Protocol for Channel Assignment in Single-radio Mesh Networks

Filipe Teixeira, Tânia Calçada, and Manuel Ricardo
INESC Porto, Faculdade de Engenharia, Universidade do Porto, Portugal
{fbt, tcalcada, mricardo}@inescporto.pt

Abstract—Wireless mesh networks (WMNs) are a simple and effective solution to provide Internet connectivity to large areas by multi-hop connecting a set of Mesh Access Points (MAPs). However, WMNs would be more efficient if multiple radio channels could be used.

This paper proposes a centralized protocol for channel assignment in single-radio WMNs based in IEEE 802.11. The protocol has the capability to discover and report all the link connections available between MAPs, independently of the channel they are operating. With this information, the Network manager can take decisions about the network topology and assign the right channel to each MAP, helping maximizing the network throughput. The protocol extends WiFIX [1] which is a low overhead solution for IEEE 802.11-based WMN. An implementation of the protocol and a testbed at the FEUP campus were created in order to validate and demonstrate the proposed protocol.

Index Terms—Channel Assignment; Mesh Testbed; Wireless Mesh Networks.

I. INTRODUCTION

To be connected to Internet has become extremely important. The increasing demand of wireless LAN connectivity beyond the infrastructured network pushes the use of IEEE 802.11 Wireless Mesh Networks (WMNs) as a simple and effective solution to extend Internet connectivity. WMNs consist in a set of multihop connected Mesh Access Points (MAPs) responsible for delivering WLAN connectivity. Despite client mobility is probable, MAP mobility may not be required in some scenarios. This scenario of static mesh topologies is particularly interesting for telecommunication operators if, in addition, full control of the network is given to them.

A simple solution for infrastructure extension using 802.11-based WMNs is the Wi-Fi network Infrastructure eXtension (WiFIX) [1]. Designed to deliver Internet access, this single-message low overhead solution is proven to be efficient in static scenarios and it offers high throughputs and low delays.

With the increase of offered load, mesh nodes interference and hidden nodes start to be relevant problems for WMNs. In this scenario, WMN can be more efficient if multiple radio channels are used [2]. However, it is also shown in [2] that using more than two wireless Network Interface Cards (NICs) can lead to severe interferences, even if they operate in orthogonal channels. To avoid the use of a split router, where each radio is a separate piece of hardware [3], a possible solution is to have one radio to serve the clients and another radio to form the mesh network. Different approaches in literature are found to solve this problem, however not all of them are compatible with the IEEE 802.11 standard [4].

As full control over the network behavior must be achieved, there is the need to gather information about all the link connections available to every MAP, independently of the channel they are operating, and report them to a central Network Manager console. Moreover, topology changes should be performed in order to minimize interferences and increase throughput by assigning the right channel to each MAP. The decisions on topology can be made manually by a person, the Network Manager, or by an Artificial Intelligence System. The protocol we propose is a centralized protocol for channel assignment in single-radio WMN.

An implementation of this protocol was performed in C language, and it was embedded in the previous WiFIX implementation. A dual-band testbed at FEUP campus was also created to study wireless mesh solutions and validate the protocol design in terms of convergence time, effective channel changes and the response to central unit decisions on topology changes. The results obtained showed that the protocol can get information about all link layer connections on different channels without introducing new signaling messages or active scans. The protocol is fast enough to report topology changes and change a MAP’s channel instantly as demanded.

The rest of the paper is organized as follows. Section II describes the mesh network architecture used. Section III refers to channel assignment protocols for single-radio WMNs. Section IV presents the proposed protocol in detail. Section V describes the testbed used to validate the protocol. Section VI presents the main conclusions and envisions future work.

II. MESH NETWORKS ARCHITECTURE

This work extends the Wi-Fi Network Infrastructure eXtension (WiFIX) [1] architecture. WiFIX is a simple and efficient solution for extending IEEE 802.11 infrastructures, using a wireless mesh network. It is based on standard IEEE 802.1D bridges [5] and a single-message protocol which is responsible for network self-organization. WiFIX defines a WMN as a set of static MAPs performing multi-hop bidirectional forwarding between the actual infrastructure and the clients, as shown in Figure 1. MAPs are equipped with two Wireless NICs, one dedicated to the mesh network and another to communicate with wireless clients. A single tree rooted at the MAP connected to the wired interface (Master MAP) is automatically created, represented in Figure 1 by the dotted lines between MAPs.
network operation on a single channel, it is possible to form a saving concepts and MAC enhancements to improve mesh wireless hops, a decentralized security framework, power functions for path selection, frame forwarding over multiple supported by WiFIX.

The solution proposed for Load-Balancing Routing in Multichannel Hybrid Wireless Networks With Single Network Interface [8] is a receiver-fixed protocol. It works in multichannel mode, using a single radio interface. This protocol is able to find multiple routes, to avoid bottlenecks, and to balance load among channels, while maintaining connectivity. The metric used to choose the best route is the downstream traffic load information of the tree. The traffic is estimated using AP-measured weighted load, which considers the distance of a node to the AP, the node’s traffic and node’s children traffic. When channel utilization is high, a node switches to another route in a different channel. An example of the routing table used is shown in Figure 3.

The primary route in Figure 3 is created using SCAN, REPLY and ASSOCIATION messages. The SCAN message is broadcasted in every channel by the connecting node, once
at time. The REPLY message sent by the potential parent is stored and the connecting node chooses the best parent according to the load of the tree. An ASSOCIATION message is then sent by the connecting node to the parent. After this phase, the load of the tree is measured at the AP and sent down the tree using an HELLO message broadcasted in every channel once at time, which allows a node tuned at its channel to receive also the load information of trees working in other channels. Having load information of all channels, a node can switch channel by sending a SWITCH message to all the child nodes. This protocol uses 5 different messages, introducing some overhead in the network.

IV. PROPOSED PROTOCOL

The proposed protocol aims to provide WiFIX with a multichannel single-radio mechanism. The infrastructure is extended by interconnecting a set of Mesh Access Points (MAP), that are responsible to establish the wireless link between clients associated with them and the wired network. Figure 4 presents the reference scenario of our solution.

Figure 4. Reference Scenario based on WiFIX architecture. Multichannel operation enables the existence of multiple trees, represented in dashed lines, each operating on a different channel. Mesh Network operates in 5 GHz band (802.11a), while clients connected to a MAP using 2.4 GHz band (802.11b/g).

In order to enable the multichannel operation, we propose a centralized approach for channel assignment in single-radio WMN, consisting of 3 phases: 1) Deliver to the Network Manager Console information about all link layer connections available between MAPs, independently of the channel they are operating; 2) with this view about all link possibilities in different channels, assign the operation channel of each MAP controlling, this way, the network topology. Those decisions can be made manually by a person, the Network Manager, or by an artificial intelligence system with the purpose of minimizing interferences and maximizing throughput; 3) use the protocol to transport those decisions in short time and correctly. Our proposed protocol implements the steps (1) and (3), while the decision (phase 2) is out of scope of this work. Our protocol also aims to not introduce additional messages, so the TR messages of WiFIX should be reused to transport channel information. As a TR message uses only 53 out of 2346 octets from a 802.11 frame, the remaining space can be used to transport information about network topology and deliver it to the Network Manager Console, keeping this a single-message solution.

Convergence time should be low, in line with the previous WiFIX solution. Besides, the protocol must be robust in order to avoid a MAP loose the connectivity to its parent MAP in case of wrong decisions on channel assignment.

A. Operation Modes

Two operation modes are associated to this protocol: the Topology Discovery and the Channel Change. The Topology Discovery mode is responsible to gather information about link connections between MAPs, whether they are on the same channel or not, and deliver that information to the Network Manager Console. The Change Channel mode is responsible to apply channel change requests from the Console to one or more MAPs, ordering them to switch channel.

B. TR message structure

In order to carry the information required in both operation modes, a new structure of TR messages is proposed, as shown in Figure 5. Besides the fields in common with WiFIX, the new Protocol Data field is introduced whose length is variable. This field includes the leading subfield Type used to describe the operation mode; the Length subfield gives information about the number of octets used in the next fields; Current Node Channel is used to broadcast the operation channel of the MAP that sent the message; the Topology Data contents depend on the operation mode; in the Topology Discovery mode it carries topology information; in the Change Channel mode, it carries the MAC address and the new channel of the MAPs notified by the manager to switch channel.

Figure 5. Topology Refresh message for Topology Discover and Change Channel modes. Protocol data is an extension to the original TR message of WiFIX.
C. Message Exchange in Topology Discovery mode

The Topology Discovery mode has two phases: 1) enable each MAP to gather the MAC address of each of its neighbors; 2) report that information to the Network Manager Console. The first phase enables the creation in every MAP of a table with all neighbors in its radio range. The second phase consists in passing the table of each MAP to the Network Manager Console; this can be done by letting every MAP to include the neighbor’s table in the retransmitted TR message. This message will be eventually received by its parent MAP, who will collect and store that information and rebroadcast it in its next transmitted TR message. This hop-by-hop mechanism from the leaf to the root allows that, after some TR messages, all the topology information reaches the Master MAP, who is then responsible to report that information to the Network Manager Console.

Figure 6 shows a message sequence diagram of the topology discovery process in a multihop tree with 3 MAPs. MAP C connects to Master MAP A via MAP B. Note that no direct radio link exists between MAP C and MAP A. Node C may also have a neighbor MAP X. The two phases of topology discovery are represented. The exchanged messages are presented followed by a table that contains an updated list of each MAC address of each MAP and their neighbors tables. The known topology after receiving a message is also shown whenever it has been updated. The first phase of discovering neighbors starts when the master MAP A sends a TR with sequence number seq1. Upon receiving the TR message with seq1, MAP B adds MAP A to its neighbor list. Then, MAP B changes the required fields in the received TR and rebroadcasts it. MAP A and MAP C receive the rebroadcast of TR message with seq1 message which allow them to add MAP B as their neighbor. Then, MAP C changes the required fields in the received TR and rebroadcasts it. MAP B receives the rebroadcast of TR message with seq1 message which allow it to add MAP C as its neighbor. If MAP C would have any neighbor, that neighbor would rebroadcast the TR message, allowing Node C to add it as a neighbor. We can conclude that after the TR message with seq1, all MAPs in this example know about their on-hop neighbors.

The second phase consists in delivering all the topology information to master MAP A. In TR message with seq2, MAP A sends an empty topology information, since it is the master. MAP B updates the received message with the information about its neighbors and rebroadcasts it. MAP A receives the TR message with seq2 sent by MAP B containing the information about all Node B neighbors. MAP C updates the received message with the information about its neighbors and rebroadcasts it. MAP B receives the TR message with seq2 sent by MAP C containing the information about all MAP C neighbors which is stored by MAP B. MAP A broadcasts a TR message with seq3 containing an empty topology information. When MAP B rebroadcasts this message, it includes the information about neighbors of MAP B and MAP C; in case a MAP has more than 1 child MAP, it includes the neighbor tables of all child MAPs. When MAP A receives the TR message with seq3 sent by MAP B, the MAP A has full knowledge of the topology.

In this message exchange model we can conclude that informing a Master MAP at 2 hops distance takes 3 cycles of TR messages (seq1 to seq3). If 3 hops were considered, the number of TR cycles needed is increased by one, and so on. Therefore, we can conclude that informing a Master MAP at n hops of distance takes n + 1 cycles of TR messages, that is n × T_{TR}, where T_{TR} is the time interval between TR messages.

D. Multichannel Operation

Figure 6 describes the topology discovery mechanism when all the MAPs are on the same channel. If two MAPs are in different channels, it is not clear how can they announce their presence to each other. A possible solution would be to do active scans which are time and energy consuming and should be avoided. Our proposed solution broadcasts each TR message in all active channels. The list of active channels is available when the network manager sets up the network. When a MAP decides to rebroadcast a TR message, it first sends the message in its main channel and then switches to other channel, sends the message and turns back to the main channel. As the receiver is tuned in its main channel, it will receive messages from the sending MAP, allowing the creation of a neighbor table with MAPs operating in multiple channels. This is the same approach used in [8], what classifies our...
protocol as a receiver-fixed, according to [4]. The deafness and multichannel hidden terminal problems described as affecting the performance of receiver-fixed protocols [4], is not problem for our solution, since the period of time that MAPs are on other channels is residual.

Consider that the neighbor of MAP C, MAP X, is on a different channel. MAP C rebroadcasts the TR message with seq1 on both channels. MAP X receives the TR message with seq1 sent by MAP C containing the information about all MAP C neighbors, which is stored by MAP X. Instead of rebroadcast the TR message with seq1 received from MAP C, MAP X stores the information about the neighbors of MAP C. When MAP X receives a TR message created by the master MAP on its main channel, MAP X rebroadcasts it on both channels. MAP C receives the TR message sent by MAP X containing the information about all MAP X and includes this information on its neighbor table. When MAP C receives the TR message with seq2, it rebroadcasts the message including the information about MAP X neighbors.

E. Message Exchange in Change Channel mode

In Change Channel mode the Master MAP receives a message from the Console. This message contains the MAC addresses of the MAPs that will change and their new channels. This topology change request is transmitted in the Topology Data field of TR message, and reaches all the MAPs in the tree. In this way, every MAP knows about the requested change and, if it is one of the MAPs affected, it changes channel and associates to a parent MAP in the new channel as fast as possible.

When a child MAP detects the parent address in the TR received, it knows that the parent is about to change and choses another parent to associate with as fast as possible. This reduces the time that child MAPs are without parent association which means loss of Internet connectivity.

F. Robustness

In order to be robust, the protocol must be improved with two mechanisms to avoid transmission errors. The first mechanism used is to Change NIC channel if no TR messages are received in a period of time (SL_TIMEOUT). This mechanism eliminates a possible mistake in the NewChannel field on the Channel Change mode and does not isolate a MAP, as it can choose a parent in another channel. This backup mechanism is very important as local reconfiguration should be avoided.

The second mechanism that adds robustness to the system is to collect information about CurrentNodeNeighbors and ChildNodeTopologyData during the SL_TIMEOUT period. That information collected during this period is passed to the parent MAP via a TR message. This mechanism avoids precipitated topology changes that would occur if one or more packets did not reach the destination due to collisions or interferences. In fact, MAPs would report a topology that was constantly changing, which is not true. If, for instance, SL_TIMEOUT is 5s and \( T_{TR} \) 1 s, 4 out of 5 messages can be lost and even so the system goes on without problems. This introduces tolerance up to 80% of packet loss in this case.

Using the mechanisms presented in this section, convergence time in a network with \( n \) hops can be described as \( SL_{-}\text{TIMEOUT} \times (n + 1) \), if \( T_{TR} \) is lower than \( SL_{-}\text{TIMEOUT} \).

V. TESTBED

In order to validate the proposed architecture we developed a prototype of the protocol and tested it in an outdoor testbed. The prototype implementation is a modified version of WiFIX [1] daemon, written in C language and designed to run in Linux Operating System. As shown in Figure 7, this new daemon, supporting the multichannel scenario proposed, runs in the stack of every MAP between 802.11 NIC driver and Linux bridge, being responsible for performing Eo11 encapsulation and to create and delete virtual interfaces, one for each tunnel created in ATCM.

Figure 7. Testbed showing the wireless links between every MAP and a control network connected to the Testbed Control. WiFIX daemon will run on every MAP, operating in master or slave modes and in different channels.

A. Band Selection

According to ROMA [9], a tested implementation with dual-radio nodes, using both 802.11b/g and 802.11a bands can be an advantage and can coexist without problems. After performing a scan at the campus of Faculdade de Engenharia (FEUP), in Porto, with an 802.11 network scanner, inSSIDer, more than 100 different networks were scanned in different channels of the 2.4 GHz band. Because of that, in our testbed 802.11a was chosen to create the mesh network while 802.11b/g was the band chosen to communicate with clients.

B. Deployment Scenario

The protocol was tested and validated using the testbed of Figure 7. The testbed was built on the roof of FEUP buildings using 4 regular computers operating as MAPs. Besides the wireless NICs, each MAP is also equipped with an Ethernet NIC to connect it to a control network, giving the testbed control console Secure Shell (SSH) access to each MAP. This control network allows to perform all the configurations needed, setup the software and capture the exchanged packets remotely. The MAC addresses and new channels of channel
change requests are stored in Change.txt file, present in every Master MAP and read periodically by the WiFIX daemon.

The buildings B, C, D form almost a perfect triangle, while building A has line-of-sight only to B building. This topology is illustrated in Figure 8, where the lines show all possible links between MAPs. Multihop can be performed between A and C or D through B building. The distances between buildings vary from 55 to 140 meters.

Figure 8. Implementation of the testbed. Lines show all available links between MAPs.

C. Impact of Frequency Switching Delay

Broadcasting a message in many channels using a single radio requires that the frequency is changed several times. This can be a problem if the time needed to switch from one channel to another is high as it leads to packet loss. In order to study the switching delay, two tests were performed. In the first test, shown in Figure 9(a), one computer was alternating between two channels and sending a PING packet after changing. Packets captured by Wireshark showed an average channel switching delay of 4 ms. Therefore, changing the channel to transmit a TR message and turn back to the main channel takes 8 ms plus the time required to send the packet.

The second test, presented in Figure 9(b) was done using three computers, two on the same channel and one on an orthogonal channel. The first computer is pinging the second every 1ms, with a packet size of 64 Byte. The second computer changes to the third computer’s channel, sends an ICMP echo request, waits for the echo reply and turns back to the previous channel. The average number of packets lost in the first MAP was 9. As each PING was done every 1ms, it means that 9 ms is the total delay to switch channel, send a packet, receive an answer, and switch back. As broadcasting a TR message in other channel does not involve waiting for the answer, the effective delay is in fact less than 9 ms, a low value considering the TR message broadcast interval \(T_{TR}\) that can be in the order of seconds.

The three computers were configured on the same BSSID address. If the BSSID was different, the PING request would not be answered. To configure a new ad-hoc network each time a channel change was performed would force scanning of other ad-hoc networks with the same SSID in the same channel. As it takes about 2-3 seconds to perform these actions, we conclude that all MAPs, even in different channels, should be in the same BSSID in order to enable fast channel switching.

In order to obtain the packet loss introduced by the channel switching delay, we need to divide the total channel switching delay by \(T_{TR}\) and multiply this result by the number of hops in the network \(h\), in percentage, as shown in Eq. 1. Using 9 ms as total delay and 2 s as \(T_{TR}\), we obtain 0.45% of packet loss per hop in the network.

\[
\frac{\text{TotalDelay}}{T_{TR}} \times h \times 100\% = h \times 0.45\%
\]  

D. Hardware used

Regular PCs running Debian 5.0.4 Linux Operating System with at least two PCI slots and one Ethernet connection were used. The PCI slots would be used to install Wireless NICs, one operating at 5 GHz for the mesh network and other one at 2.4 GHz to deal with clients. The wireless NIC used was the 3COM 3CRA675B abg PCI adapter. Madwifi driver was chosen as it showed the most consistent results in the additional tests performed [11]. Omni-directional antennas were used in order to receive signal from all neighbors.

Considering that the maximum link distance between each MAP in Figure 8 is 140 m measured in Google Earth and operating at 5.2 GHz with a transmitting power of 16 dBm, the antennas should have a gain greater than 4 dBi in order to establish the link at 54 Mbit/s. In order to have a link margin greater than 0 to operate in non-ideal conditions and compensate for unpredicted losses or interferences, the antenna used was a TP-Link TL-ANT2408C omni-directional antenna with a gain of 8 dBi. Although it is only specified for 2.4 GHz frequency band, the reflection coefficient measured in lab was practically the same for 5.2 GHz band (channel 40).

In order to avoid transmission losses, an elliptical zone between the two antennas (Fresnel Zone) must be unobstructed [10]. Therefore, the height (in meters) of the antennas was properly calculated, being in the worst case 1.4 meters above the roof level [11].

The lower height of C and D buildings is an advantage when considering the multihop scenario. In fact, the connection
between A and C or D buildings is blocked by the building next to B, forcing all traffic to go through MAP B. However, this could also be a problem if the vertical angle of the antennas is not enough to establish a connection between C or D buildings and B building. The antennas chosen have a vertical angle of 15 degrees. Considering that the shortest distance between these buildings is 55 meters, the vertical angle of the antennas allows the difference in buildings height of 14.7 meters, which does not require an adjustment of antenna position.

E. Testbed performance

In order to validate the testbed design and deployment, individual tests were carried out between every pair of buildings using iperf. Working on channel 40 (5.2 GHz), with no other networks using the same frequency band, we could achieve single-hop TCP bandwidths from 11.3 up to 24.7 Mbit/s between the different buildings. A small test using the channel 1 (2.4 GHz) was also performed in order to compare possible differences between using 802.11a and 802.11b/g. The average TCP bandwidth obtained was 6.3 Mbit/s, against 21.7 Mbit/s using 802.11a. This shows the advantage of using 802.11a to form the mesh network, as the 2.4 GHz was saturated due to 104 different networks found on that frequency band. UDP performed slightly better in all MAPs, with bandwidths between 13.1 and 27.4 Mbit/s. In all tests, the packet loss found is less than 1%.

VI. PROTOCOL PERFORMANCE EVALUATION

In order to evaluate our channel assignment protocol, a set of tests was performed on the testbed described in Section V. These tests evaluate the system in terms of convergence delay, robustness of the protocol, and effectiveness of channel change. The maximum number of MAPs inside the mesh network was also studied.

A. Convergence Delay - Node Turned On and Off

The convergence delay is the time elapsed from the change of a MAP state (turn on, turn off, or change channel) to the report of the change to the Network Manager Console. This time should be adequate for static mesh networks with occasional topology changes. SL_TIMEOUT has a great impact on the convergence time: as we shorten this value, the neighbor list is updated more often and the information sent in the next TR message is more recent.

1) Node turned on: Figure 10(a) shows the case where a MAP C will be turned on and will be connected to the master MAP A via MAP B. Using SL_TIMEOUT 5 s, and a TR broadcast interval (T_TR) of 2 s, the expected convergence delay is SL_TIMEOUT × (n + 1) = 5 × (3 + 1) = 20 s. To perform this test, Wireshark was used to listen to the exchanged packets on master MAP A. After performing the test 5 times in different periods of the day, we achieve a mean value of 22.3 s for the master MAP to have knowledge about MAP C and its neighbors, with a standard deviation of 0.7 s. The value obtained is close to the expected value and is acceptably low regarding the low signaling - one message every 2 seconds and big SL_TIMEOUT. The SL_TIMEOUT is passed as an argument when WiFix daemon is called, allowing the Network Manager to decide its value.

2) Node turned off: Using the same topology of Figure 10(a), we will now turn off the MAP C and measure the time elapsed until MAP A realises that the topology has changed. It will happen when the TR message sent from MAP B does not report C MAC address as it neighbor and does not forward information about C neighbors. Keeping the SL_TIMEOUT 5 s and the T_TR 2 s, the expected delay is SL_TIMEOUT × n = 5 × 3 = 15 s. Wireshark was listening the exchanged packets on MAP A and the protocol took a mean delay of 18.3 s to report that MAP C was not in the mesh network any longer. The standard deviation was 0.9 s. Again, the value does not differ much from the expected delay and it is acceptably low regarding the settings used and static scenario considered. As shown in Figure 11, this time can be reduced to 9.1 s if SL_TIMEOUT is reduced to 3.

3) Resistance to packet loss: Reducing too much the SL_TIMEOUT can lead to wrong topology data sent to the upper level as the neighbor list is incomplete in case a TR message is lost. SL_TIMEOUT should be greater than 2 × T_TR in order to allow at least one out of 2 messages to be lost without interference in the protocol behavior. We tested that using a T_TR of 2 s, SL_TIMEOUT equal to 5 s and discarding 50% of the received TR messages; the protocol worked as expected. A tradeoff between the convergence of the network and the resistance to packet loss should be considered by the Network Manager when setting up the network.

B. Channel Change Delay

The channel change delay is the time elapsed between a change channel request from the network manager (Master MAP) and the effective change in the selected MAP. This time should be as low as possible to allow fast channel switches. When a MAP switches, it will loose its parent. The period without connectivity is the time elapsed to find and associate with a new parent MAP. Two different tests were performed to analyse these two parameters. In the first one, as presented in Figure 10(b), MAP C will change to Ch2 and associate with
Master MAP D. In the second test, MAP B, parent of MAP C, will change to Ch2 and associate with Master MAP D.

1) Node Change: In case MAP C receives a request to change from Ch1 to Ch2, it should perform change in 4 ms plus the time to transmit the packet in each MAP. Using wireshark in MAP C to measure the time elapsed between the channel change request and the effective change, we always obtained 5 ms for the channel change delay, which is in line with the expected results: almost instant channel changes. The period without connectivity is calculated by \( SL\_TIMEOUT / 2 \), as the MAP needs to search a new parent in Ch2 as fast as possible. Using \( SL\_TIMEOUT = 5 \) s, this period is 2.5 s. After running the tests 5 times, the average time elapsed between the channel change request and the association with the new parent was 2.6 s, with a standard deviation of 0.3 s. This value is a little higher than the TR broadcast interval, which is 2 s. If we decrease \( T_{TR} \), this value can decrease. In practice, the MAP chooses the first parent available, in order to reduce the period without connectivity.

2) Parent Node Change: If MAP B is the MAP requested to change channel (parent MAP), the MAP C will be lost and without connection. This period without connectivity for child MAP should be as high as \( SL\_TIMEOUT \) because the child MAP must listen for any TR messages and, as there was no parent available, the child MAP changes itself to Ch2. After running 5 tests with \( SL\_TIMEOUT = 5 \) s, we concluded through wireshark logs in MAP C, that the period without connectivity was 5.1 s in average, with a standard deviation of 0.4 s. This shows that backup method avoids deadends from wrong decisions on channel changes.

C. Maximum number of MAPs

With the increase in number of MAPs in the mesh network, more topology data needs to be exchanged through TR messages. Using Eo11 encapsulation, only 2293 bytes are free in a 2348 802.11 frame (Figure 5), limiting the maximum number of MAPs in the mesh network. Considering Topology Discovery messages and a full mesh topology, where all \( n \) MAPs are able to see each other, each MAP have \( n - 1 \) neighbors. The space left in each message should carry all the topology data of \( n - 1 \) MAPs, which limits the number of MAPs in the mesh network to 17 [11]. However, this number can be higher, if sparser topology is considered or fractioning topology information and send them in two messages.

D. Discussion

Observing Figure 11, we can conclude that this protocol performs in a real-usage scenario with minimal differences to the expected values. The convergence delay and the period without connectivity benefit with the decrease of \( SL\_TIMEOUT \), meaning that the tradeoff between these parameters and the overhead must be carefully addressed.

VII. CONCLUSION AND FUTURE WORK

We proposed a centralized protocol for channel assignment in single-radio WMN. Embedded in WiFIX solution, this protocol gathers and reports all link layer connections on different channels with low delay and without introducing new signaling messages to WiFIX. Supporting more than 17 mesh access points in a full mesh topology, our protocol is fast enough to track topology changes and provide instantaneous channel changes commanded by a Network Manager. The proposed protocol prevents large periods without connectivity to child nodes and is highly resistant to packet loss. A testbed was designed and created to validate the protocol. The testbed works on both 2.4 and 5 GHz bands, one for clients and another for the mesh network, with links exceeding 20 Mbit/s and allowing multihop scenarios at high data rates.

Future work coming out from this work includes an automatic adjustment of the delay to choose/switch to a better parent depending on the packet loss and the automatic adjustment of the number of TR messages according to topology change rate. The support for multiple gateways on each channel and the design of a Network Manager are also being studied.

REFERENCES


