Protocol for Centralized Channel Assignment in WiFIX 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

Summary. A Wireless Mesh Network (WMN) is an effective solution to provide Internet connectivity to large areas and its efficiency may increase if multiple radio channels are used in the mesh backbone.
This paper proposes a protocol for centralized channel assignment in single-radio WMNs. This protocol has the capability to discover all the links available between Mesh Access Points (MAPs), independently of the channel they operate. With this information, a network manager can assign the right channel to each MAP in order to, for instance, maximize the network throughput. The proposed protocol extends WiFIX [1] which is a low overhead solution for implementing IEEE 802.11-based WMNs.

Key words: channel assignment, mesh testbed, wireless mesh networks

1 Introduction

The increasing demand of wireless LAN connectivity is pushing the use of IEEE 802.11 Wireless Mesh Networks (WMNs). A WMN consists of a set of Mesh Access Points (MAPs) interconnected by wireless links. The static mesh topology has low deployment costs and it is of particular interest for telecommunications operators which, in addition, will see advantages in controlling the mesh nodes from their premises.

Interference and hidden nodes affect seriously the performance of a WMN, but this performance degradation can be reduced if multiple radio channels are used to interconnect MAPs; however, the usage of multiple wireless Network Interface Cards (NICs) can lead to severe interferences [2] if the cards operate in the same band, even when they operate in orthogonal channels. A possible solution for this problem is to use a single network interface in each MAP to form the mesh network. A second NIC, operating on a different band, can be used to serve the stations associated to MAPs. Therefore, in order to take advantage of the multichannel operation and use a single radio interface to form the mesh network, a signaling protocol for channel assignment within the mesh is needed.

From the telecommunications operator point of view, full control over the network behavior is desirable. For that purpose, there is a need to collect information about the wireless links of every MAP, independently of the channel the MAP operates, and to store this information centrally, at the operator premises. The operator will also see benefits in having control over the network topology by

assigning the right channel to each MAP, so that interference can be minimized and throughput can be increased.

IEEE 802.11s [3] adds to IEEE 802.11 [4] the required functions for path selection, frame forwarding over multiple wireless hops, decentralized security, and power saving. While the main scope of 802.11s is the mesh network operation on a single channel, it is possible to form a mesh network over multiple channels. However, this multi-channel operation demands the use of multiple radios tuned on different channels. Moreover, this solution requires two different mechanisms to create the routing protocol, which leads to high overheads. A simpler solution for infrastructure extension using 802.11-based WMNs is the Wi-Fi network Infrastructure eXtension (WiFIX) [1]. Designed to provide Internet access, this single-message low overhead solution is proven to be efficient in static scenarios and to offer high throughputs and low delays. As multichannel operation was not considered solution in the WiFIX, the protocol we propose in this paper becomes in fact a multichannel operation solution for WiFIX mesh networks.

An implementation of this protocol was developed and embedded in the previous WiFIX implementation. The protocol was validated in a testbed deployed at FEUP campus. The results obtained showed that the proposed protocol can get information about all links operating on different channels without introducing new signaling messages to the WiFIX base protocol nor active scans. The proposed protocol is fast enough to report topology changes to the operator and to change a MAP's mesh channel instantly, as demanded.

The rest of the paper is organized as follows. Section 2 describes the mesh network architecture used. Section 3 surveys the channel assignment protocols used in single-radio WMNs. Section 4 presents the proposed protocol in detail. Section 5 describes the testbed used to validate the protocol. Section 6 presents the main conclusions and envisions future work.

2 WiFIX 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.

The metric used for choosing next MAP is the minimum number of hops; this metric is simple and effective when compared with the radio aware routing metrics used in [3] which are proved to have problems related with network

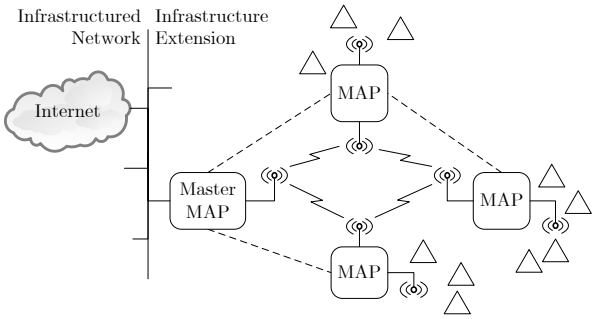


Fig. 1. WiFIX Reference Scenario. Each MAP is equipped with two NICs, one for the mesh network and the other for clients.

instability [6]. The concept of 802.1D bridges and their simple learning mechanism is used, allowing multi-hop frame forwarding among the mesh network. Ethernet-over-802.11 (Eo11) is the tunneling mechanism used to encapsulate an Ethernet frame inside an 802.11 frame, storing the original source and destination addresses of the original frame. This mechanism is used because the original 802.11 frame only supports 2 MAC addresses for forwarding purposes, which are already used by the intermediate source and destination at intermediate links.

Inside the mesh network, the active tree topology, rooted at the Master MAP, is created using the Active Topology Creation and Maintenance (ATCM) mechanism. The Topology Refresh (TR) message is sent by the Master MAP periodically and forwarded by other MAPs. Every time a MAP forwards the message, it updates parent address, TTL, sequence number and distance fields. This mechanism allows each MAP to select the best parent, which is the parent that offers the path to the Master MAP with the least number of hops. As the forwarded TR message includes the parent address, the same message is used with three purposes: (1) inform the parent MAP about a new child and create a new Eo11 tunnel to it; (2) inform other MAPs about the MAP existence; (3) announce the Master MAP, which is the path to Internet. This mechanism helps reducing the number of messages exchanged to form the active tree.

3 Channel Assignment for Single-Radio WMN

In [7], different protocols and architectures for channel assignment in WMN are presented. When it comes to single-radio networks, these protocols can be classified into 4 types: (1) Dedicated Control Channel, (2) Hopping, (3) Split Phase and (4) Receiver-fixed. In ***Dedicated Control Channel***, one channel is reserved for control packets; in the case of IEEE 802.11b/g, which is limited to three orthogonal channels, 33% of the resources become allocated to the control channel. In ***Hopping Protocols***, the nodes hop between multiple narrow-band channels, in the same or different patterns; although this solution does not need

a control channel, it demands synchronization mechanisms and it is not adequate to IEEE 802.11. The ***Split Phase*** protocols consider the division of time into cycles composed of two phases - the control phase and data phase. The multi-channel hidden terminal problem is less severe; however, fine synchronization between nodes and a proper computation of the duration of both phases is required. Moreover, carrier sense must be done on all channels simultaneously, which may also be a problem. In ***Receiver-fixed*** protocols [8], a fixed quiescent channel is assigned to each node. When a node needs to send data, the sender changes its frequency and sends data on the quiescent channel of the destination node. If the receiver is idle, it is tuned on its quiescent channel and then the data is received. When all data is sent, the sender node changes back to its quiescent channel, being free to receive data from other nodes. Receiver-fixed approach is easy to implement and it is compatible with the IEEE 802.11 standard. However, broadcast must be done in every channel, consuming extra resources.

The Load-Balancing solution proposed in [8] is a receiver-fixed protocol. It works on 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, which is broadcasted by every node in more than one channel. 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. After having load information of all channels, a node can switch channel when the channel utilization is higher than in other channels. This protocol uses 5 different messages and introduces relevant overhead in the network.

4 Proposed Protocol

The proposed protocol aims to enable multichannel operation in WiFIX mesh networks considering that only one radio is available to form the mesh network. Figure 2 presents the reference scenario of our solution. Multichannel operation enables the existence of multiple trees, represented in dashed lines, each operating on a different channel.

The centralized approach, proposed in this paper consists of 3 phases: 1) discover the network topology and deliver it to the network manager; 2) decide in which channel each MAP should operate; 3) configure network interface cards of MAPs to use the selected channel. Our proposed protocol implements phase 1) and phase 3), while the decision of phase 2) is out of scope of this paper.

Our protocol also aims to avoid additional messages. So, the TR messages of WiFIX should be reused to transport channel information. As a TR message uses only 53 out of 2348 octets from an 802.11 frame, the remaining space can be used to transport information about network topology and channel assignment decisions, keeping this a single-message solution. Convergence time should be low, in line with the previous WiFIX solution. Besides, the protocol must be

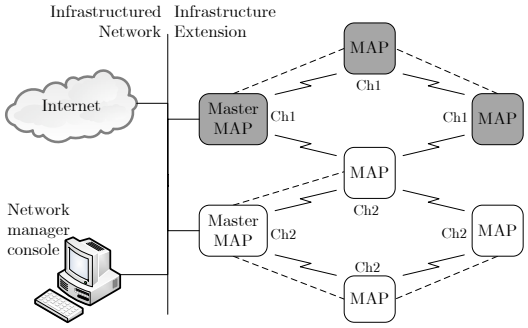


Fig. 2. Reference Scenario based on WiFIX architecture, showing multiple trees, operating on different channels.

robust in order to avoid losing the connectivity to a MAP in case of wrong decisions on channel assignment.

4.1 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 links between MAPs, whether they are on the same channel or not, and deliver that information to the network manager. The ***Change Channel*** mode is responsible to apply channel assignment decisions from the network manager to one or more MAPs.

4.2 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 3. Besides the fields in common with WiFIX, represented on the top of Figure 3, the new variable length *Protocol Data* field is introduced. 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 network manager to switch channel.

4.3 Message Exchange in Topology Discovery mode

The Topology Discovery mode has two phases: 1) gather the MAC address and the operation channel of each of its neighbors; 2) report that information to

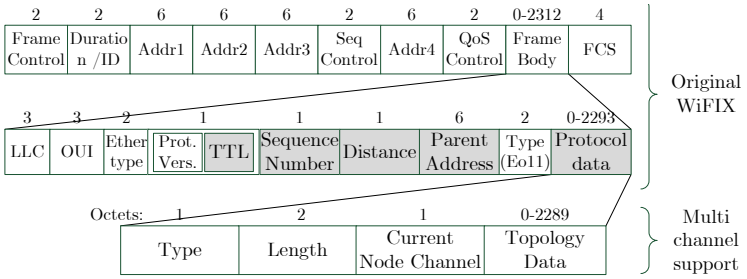


Fig. 3. Topology Refresh message for Topology Discover and Change Channel modes. Protocol data is an extension to the original TR message of WiFIX.

the network manager. The first phase enables the creation of a table in every MAP with all neighbors in its radio range by listening and storing the $CurrentNodeChannel$ from the received TR messages. The second phase consists in passing the table of each MAP to the network manager; this can be done by letting every MAP include the neighbor's table in the retransmitted TR message ($TopologyData$). This message will be eventually received by its parent MAP, who will collect and store that information and retransmit it in its next transmitted TR message. This hop-by-hop mechanism from the leaf to the root enables 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.

Figure 4 shows a message sequence diagram of the topology discovery process in a multi-hop 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. MAP 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, for each MAP, an updated list of known MAPs in the tree and their neighbors; in case no updates exist, this table is omitted. 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 rebroadcasted 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 allows it to add MAP C as its neighbor. If MAP C had 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 one 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

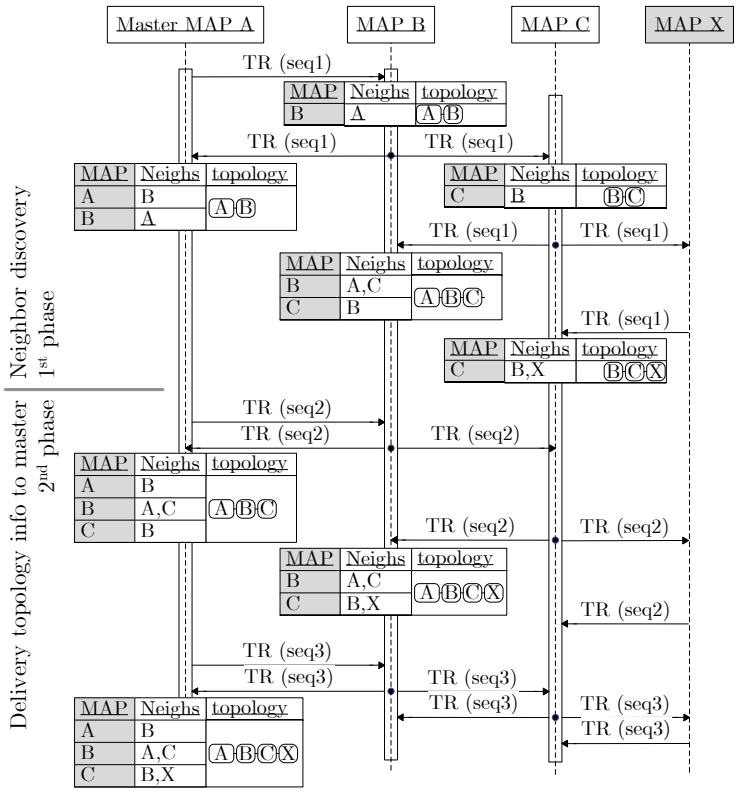


Fig. 4. Message sequence diagram in multi-hop scenario with 3 MAPs, showing the network topology known in every MAP as well as the neighbor's tables during the message exchange. Only after the TR with *seq*3, Master knows the full network topology.

information about its neighbors and rebroadcasts it. MAP A receives the TR message with *seq*2 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 *seq*2 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 *seq*3 containing an empty topology information. When MAP B rebroadcasts this message, it includes the information about neighbors of MAP B and neighbors of 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 *seq*3 sent by MAP B the, MAP A has full knowledge of the topology.

In this message exchange model we can conclude that informing a MAP 2 hops distance takes 3 cycles of TR messages (*seq*1 to *seq*3). 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 MAP at h hops of distance takes

$h + 1$ cycles of TR messages, that is $h \times T_{TR}$ seconds, where T_{TR} is the time interval between TR messages. As each MAP retransmits each TR message, the number of TR messages needed for a network of size n MAPs to converge is $(h + 1) \times n$.

4.4 Multichannel Operation

Figure 4 described 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 they can 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 its 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], which classifies our protocol as a receiver-fixed, according to [7]. To avoid that the parent MAP is tuned on another channel when its child MAP rebroadcast the TR message, each MAP should wait a small random period of time, lower than T_{TR}, before switching to another channel. The deafness and multichannel hidden terminal problems, which affect the performance of receiver-fixed protocols [7], is not a 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, as in Figure 4. MAP C rebroadcast the TR message with *seq*1 on both channels. MAP X receives the TR message with *seq*1 sent by MAP C and adds it as a neighbor. As MAP C is on a different channel (different tree), its neighbor information will not be stored by MAP X. When MAP X receives a TR message created by its Master MAP, on MAP X main channel, it rebroadcasts the message on both channels, allowing MAP C to add MAP X as a neighbor. Same as MAP X, MAP C will not store any neighbor information of MAP X, as they operate on different trees.

4.5 Message Exchange in Change Channel mode

In Change Channel mode the Master MAP receives a message from the network manager console. This message contains the MAC addresses of the MAPs selected to change and the new channels to be assigned. This topology change request is transmitted in the *Topology Data* field of TR message, and reaches all MAPs in the tree allowing every MAP to know about the requested changes. If a MAP finds itself in the list of MAPs selected to change, it first rebroadcasts the TR message, then it changes channel and finally associates to a parent MAP in the new channel as fast as possible. When a child MAP detects its parent

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

4.6 Robustness

To improve the robustness of the protocol, two mechanisms were implemented to avoid transmission errors. The first mechanism avoids MAPs isolation caused by a possible mistake in the *Topology Data* field of a Channel Change mode TR message; after a T_{reconf} time without receiving TR messages, a MAP selects and reconfigures itself on a different channel and can choose a parent in that channel. This backup mechanism is very important for avoiding manual reconfigurations.

In the second mechanism, each MAP collects information about its neighbors and its child's neighbors for a T_{Upd} period. T_{Upd} is defined as the interval between updates of the *Topology Data* field of the TR messages. After this interval, the collected information is rebroadcasted in the subsequent TR messages. This mechanism avoids premature topology changes that would occur if one or more packets did not reach the destination due to collisions, interferences or packet drop in case of high loads. If, for instance, T_{Upd} is 5 s and T_{TR} (interval between TR messages) is 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% to packet loss, in this case.

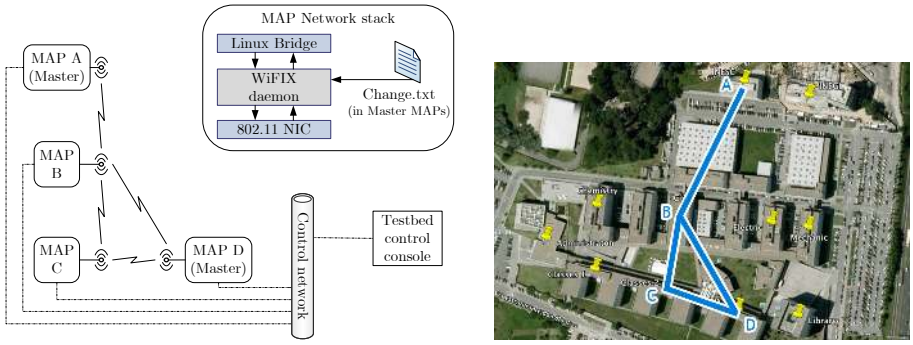
Using the mechanisms presented in this section, convergence time in a network with h hops can be described as $T_{Upd} \times (h + 1)$, if T_{TR} is lower than T_{Upd}.

5 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 5(a), this new daemon, adapted to work on the multichannel scenario proposed, runs in the stack of every MAP between 802.11 NIC driver and Linux bridge. It is responsible for performing Eo11 encapsulation and creating and deleting virtual interfaces, one for each tunnel created in ATCM.

5.1 Deployment Scenario and Hardware Used

The protocol was tested and validated using the testbed of Figure 5(a). The testbed was built on the roof of FEUP buildings using 4 regular computers, operating as MAPs, with Debian 5.0.4 Linux Operating System and at least two PCI slots. The PCI slots were used to install Wireless NICs, one for the



(a) (b)

Fig. 5. (a) Testbed showing the wireless links available and a Ethernet control network. WiFIX daemon will run on every MAP, operating in master or slave modes and in different channels. (b) Implementation of the testbed on the roof of FEUP Campus.

mesh network, operating on the less used 5 GHz band, and other one at 2.4 GHz to deal with clients. The wireless NIC used for the mesh network was the 3COM 3CRDAG675B abg PCI adapter. Madwifi driver was chosen. Besides the wireless NICs, each MAP was also equipped with an Ethernet NIC to connect it to a control network, giving the Testbed Control Secure Shell (SSH) access to each MAP. The MAC addresses of the MAPs requested to change and their new channel were stored in Change.txt file, present in the Master MAP and read periodically by the WiFIX daemon.

The buildings B, C, D form a triangle, while building A has line-of-sight only to B building. This topology is illustrated in Figure 5(b), where the lines show all possible links between MAPs. Multi-hop can be performed between A and C or D through B building. Considering that the maximum link distance between each MAP in Figure 5(b) is 140 m and operating at 5.2 GHz with a transmitting power of 16 dBm, we used 8 dBi omni-directional antennas. This allowed us to have a link margin of 4 dBi to operate in non-ideal conditions and compensate unpredicted losses or interferences. The Fresnel zone was also considered, being the antennas placed 1.4 meters above the roof level [9].

In order to validate the testbed design and its deployment, tests using *iperf* were carried between the buildings. We could achieve single-hop TCP bandwidths from 11.3 up to 24.7 Mbit/s and UDP bandwidths between 13.1 and 27.4 Mbit/s using channel 40 (5.2 GHz). Using channel 1 (2.4 GHz), the average TCP bandwidth obtained was 6.3 Mbit/s, against 21.7 Mbit/s using 802.11a, clearly showing the advantage of using 802.11a to form the mesh network, as the 2.4 GHz was saturated. Multihop performance was also measured, with TCP bandwidths of 8 Mbit/s at two hops distance. In all the tests, the packet loss was found less than 1%.

5.2 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 delay introduced by switching from one channel to another is high, as it leads to packet loss. According to the tests carried out in [9], changing to another channel to transmit a TR message and change back to the first channel takes about 9 ms, a low value considering the T_{TR} (interval between TR messages), that can be in the order of seconds. The switching delay can still be reduced to 200 μs using low switching delay hardware [10], causing minimal packet loss due to channel switching.

6 Protocol Performance Evaluation

In order to evaluate the proposed protocol, a set of tests was performed on the testbed described in Section 5. These tests evaluate the system in terms of convergence delay, robustness of the protocol, and effectiveness of channel change. The maximum number of nodes inside the mesh network was also studied.

6.1 Convergence Delay - MAP 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. This time should be adequate for static mesh networks with occasional topology changes. T_{Upd} has a great impact on the convergence time: as we shorten this value, the neighbor lists are updated more often and the information sent in the next TR message is more recent.

MAP turned on Figure 6(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 T_{Upd} 5 s, and a T_{TR} of 2 s, the expected convergence delay is $T_{Upd} \times (h+1) = 5 \times (3+1) = 20$ 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 s and big T_{Upd}.

MAP turned off Using the same topology of Figure 6(a), we will now turn off the MAP C and measure the time elapsed until MAP A realizes 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 T_{Upd} 5 s and the T_{TR} 2 s, the expected delay is $T_{Upd} \times h = 5 \times 3 = 15$ 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,

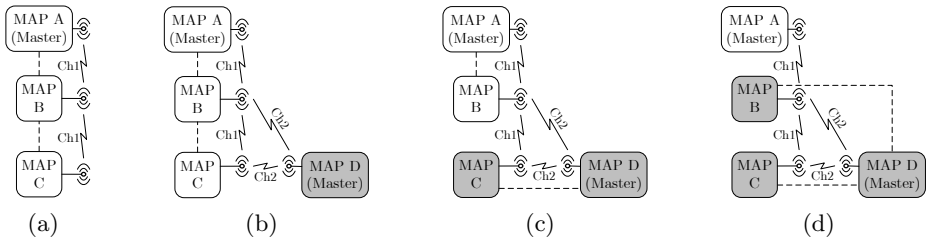


Fig. 6. (a) Multihop topology of MAP Turned On/Off test (b) Multihop topology of Channel Change test. MAP B and C listen to TR messages from Master MAP in ch 2. (c) MAP C changes to Ch2. (d) MAP B changes to Ch2 and forces MAP C to change.

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 7, this time can be reduced to 9.1 s if T_{Upd} is reduced to 3 s.

Resistance to packet loss Reducing too much the T_{Upd} can lead to wrong *topology data* sent to the upper level as the neighbor list is incomplete in case a TR message is lost. T_{Upd} 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, T_{Upd} equal to 5 s and discarding 50% of the received TR messages; the protocol worked as expected. A trade-off between the convergence of the network and the resistance to packet loss should be considered by the network manager when setting up the network.

6.2 Channel Change Delay

The channel change delay is the time elapsed between a change channel request from the network manager 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 to a new parent MAP. Two different tests were performed to analyze these two parameters. To do it, a new Master MAP D was introduced, as shown in Figure 6(b), operating in a different channel (Ch2). In the first test, 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.

MAP Change In case MAP C receives a request to change from Ch1 to Ch2, (Figure 6(c)), it is expected to perform that 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 $T_{Upd} \div 2$, as the MAP needs to search a new parent in Ch2 as

fast as possible. Using $T_{Upd} = 5$ 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 T_{TR}, 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.

Parent MAP Change If MAP B (parent MAP) changes channel, as shown in Figure 6(d), the MAP C looses its parent connectivity to the infrastructure network. This period without connectivity for child MAP should be at least T_{Upd} because the child MAP must listen for TR messages before conclude that there is no parent available and change to Ch2. After running 5 tests with $T_{Upd} = 5$ 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 dead-ends from wrong decisions on channel changes.

6.3 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 3), 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 has $n-1$ 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 [9]. However, this number can be higher, if sparser topology is considered or fractioning topology information and send them in two messages.

6.4 Discussion

By observing Figure 7, we can conclude that this protocol performs with minimal differences from the expected values in a real-usage scenario. The convergence delay and the period without connectivity benefit with the decrease of T_{Upd}, meaning that the trade-off between these parameters and the overhead must be carefully addressed.

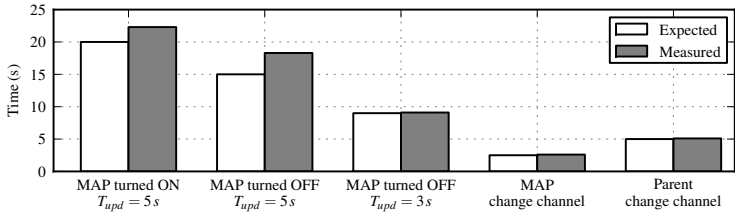


Fig. 7. Protocol behavior show little difference between expected and measured values.

7 Conclusions and Future Work

We proposed a protocol for centralized channel assignment in single-radio WMN. Embedded in the WiFIX solution, this protocol gathers and reports all link layer connections on different channels with low delay and without introducing new signaling messages. Supporting up to 17 Mesh Access Points (MAPs) 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 MAPs and is 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, with links exceeding 20 Mbit/s and allowing multi-hop 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 algorithm are also being studied.

References

1. R. Campos, R. Duarte, F. Sousa, M. Ricardo, and J. Ruela, "Network infrastructure extension using 802.1D-based wireless mesh networks," *Wireless Communications and Mobile Computing*, vol. 11, pp. 67–89, Jan 2011.
2. J. Robinson, K. Papagiannaki, C. Diot, X. Guo, and L. Krishnamurthy, "Experimenting with a multi-radio mesh networking testbed," *Proc. of WiNMee'05*, Apr 2005.
3. IEEE 802.11s/D12.0 draft amendment to standard IEEE 802.11, "Mesh networking," Tech. Rep., Jun 2011, work in progress.
4. IEEE 802.11 Work Group Part 11, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," Tech. Rep., Jun 2007.
5. IEEE 802.1D Work Group Part 11, "IEEE standard for local and metropolitan area networks: Media access control (MAC) bridges," Tech. Rep., Jun 2004.
6. R. G. Garroppo, S. Giordano, and L. Tavanti, "Implementation frameworks for IEEE 802.11s systems," *Computer Communications*, vol. 33, pp. 336–349, Feb 2010.
7. J. Crichigno, M.-Y. Wu, and W. Shu, "Protocols and architectures for channel assignment in wireless mesh networks," *Ad Hoc Networks*, vol. 6, pp. 1051–1077, Oct 2007.
8. J. So and N. H. Vaidya, "Load-balancing routing in multichannel hybrid wireless networks with single network interface," *IEEE Trans. on Vehicular Technology*, vol. 55, pp. 806–812, May 2006.
9. F. Teixeira, *Protocol for Channel Assignment in Single-radio Mesh Networks.* MSc Thesis. Faculdade de Engenharia da Universidade do Porto, Jul 2010.
10. A. Mishra, V. Shrivastava, D. Agrawal, S. Benerjee, and S. Ganguly, "Distributed channel management in uncoordinated wireless environments," *Proceedings of MobiCom'06*, Sep 2006.