Channel Assignment Protocols
for
Multi-Radio Multi-Channel Wireless Mesh Networks

MARYAM AMIRI NEZHAD

DEPARTMENT OF COMPUTER ARCHITECTURE
UNIVERSITAT POLITÈCNICA DE CATALUNYA

SUPERVISOR: DR. LLORENÇ CERDÀ ALABERN

CO-SUPERVISOR: DR. MANEL GUERRERO ZAPATA

A THESIS SUBMITTED FOR THE DEGREE OF
Doctor of Philosophy

Abstract

The increasing demand for large and low cost wireless coverage, ranging from campus to city wide areas, has motivated a high interest in multi-hop communications with Wireless Mesh Networks (WMN) based on IEEE 802.11s as the most recent and significant standard.

Channel Assignment (CA) is a mechanism which selects the best channels for an individual wireless node or the entire network aiming to increase the capacity of the network. Channel assignment has been extensively researched for multi-radio WMNs, but it is still very challenging when it comes to its implementation. Although IEEE 802.11s introduces new inter-working, routing and wireless frame forwarding at the link layer, the multi channel architecture receives less attention due to many unsolved challenges that arise while mesh service set works over multiple frequencies.

This research work tries to give a solution to the needs of designing an efficient channel assignment mechanism.

As a result we have proposed a new static channel assignment based on the fact that not all wireless links are practically useful. Our mechanism prunes the network topology by removing weak wireless links and improves the network performance by reaching a more diverse channel-radio assignation solution.

Toward designing a distributed channel assignment we propose a new game theory based formulation of channel assignment which is applicable to a realistic scenario with imperfect information at each router. We have proposed a distributed and hybrid channel assignment protocol based on the game formulation. The proposed channel assignment makes wireless routers to be able to follow the unpredictable changes in the wireless environment.

We also investigated the types of channel assignment protocols which can be adapted to the IEEE 802.11s based mesh network and improve the network good-put in terms of data delivery ratio and end-to-end delay.

Acknowledgments

I would like to express my gratitude to my PhD supervisor Dr.Llorencç Cerdà Alabern for teaching me and helping me through this thesis with his support an immense knowledge. My special thanks to my co-supervisor Dr. Manel Guerrero Zapata for his inspiration, his patient and his great effort to make this thesis possible. He provides me encouragement, good advice and good ideas throughout writing this thesis.

I like to thank Dr. Boris Bellalta for his thoughtful comments which, most of the time, shows me the right direction to accomplish my research. I always get surprised for the amount of time he has dedicated to review my papers and give me the constructive feedback. I also appreciate Dr. Enric Monte Moreno for being available whenever I had doubts in maths and try to explain everything simple.

I wish to thank Sirus Dehghan Madiseh, the head of organizational affairs at National Iranian Oil Refining and Distribution Company (NIORDC), without his support I wasn't able to pursue PhD. I also thank my colleagues at NIORDC, Hossein Danesh Alagheh Band, Hossein Sajadifar, Roghieh Jangi and Puzhan Pasdar for their encouragement and great help to do all paper works which I must have done in the company.

I also appreciate my dear friend Razieh Heydari (Raii) for trying hard to keep the contact with me and provides me emotional support.

My PhD at UPC can not be fulfilling without Victor Rodriguez Doncel, Abhishek Deb, Ruben Tous and Pere Toràn Vilarrubias for providing fun environment during coffee breaks.

I like to thank Bettina Rausche for her kindness and inspiring conversation. I wish to thank my friends Hélio Doyle and Safa Abedi for sharing good times with me and providing me warmth environment at their home. I also thank Hojat Madadi and Mahzad Shiravand for making me smile all the time.

I had amazing time with UPC feminine rugby team playing rugby and chatting every now and then. Thanks to all Catalan girls in the team for sharing with me memorable times.

I indebted to Sílvia Moncayo who inspired me to believe myself and never let me down. I appreciate her for her words and emotional support throughout the difficult times.

I wish to thank my best friend, Pooneh Irani, for all caring and emotional support she provides. She always comes up with amazing music and movie recommendations when I need to get rid of stress.

Lastly and most importantly I wish to thank my family; my parents, and my husband, Amir, for their love and their support. They are the most important and inspiring people in my life and I owe them my happiness and achievements. To them this thesis is dedicated.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Wireless mesh networks (WMN), based on commodity 802.11 radios, are emerging as a promising solutions to extend the wireless coverage in a flexible and cost-effective way without relaying on any wired infrastructure. They have broad applications in Internet access, emergency networks, public safety, and so forth [2, 47].

Technical solutions for multihop wireless networks are being specified in IEEE 802.11s [27]. IEEE 802.11s is developed as an extension of the successful IEEE 802.11 standard for WLANs (Wireless Local Area Networks) [25]. IEEE 802.11s based mesh networks are composed of mesh stations (Mesh STAs) that operate as routers. Within a mesh, packets are transmitted over multiple wireless hops providing Internet access for last mile users.

However due to the broadcast nature of the wireless media, wireless links interfere each other if there are simultaneous transmissions in them. In multi hop networks, the interference of the next hop link over the previous hop reduces the end-to-end performance drastically [24, 8]. Moreover the increasing number of devices sharing the same spectrum band, can degrade the performance of the mesh networks [24].

Interferences can be grouped in two types: external and internal. The external interference appears when two or more coexisting wireless networks work in the same frequency channel (Fig. 1.1). Although it can be eliminated by using different non-overlapping channels offered in IEEE 802.11 [25], it requires that the different networks agree on the channel distribution. Additionally, as the number of non-overlapping channels is reduced, the scenario where the external interference can be completely avoided is unrealistic. By internal interference we refer to the time overlapping transmissions by nodes of the same network which can result in collisions or transmissions errors. Both external and internal interferences limit the system performance [51, 43].

In WMNs, mesh routers can have more than one radio interface, which can

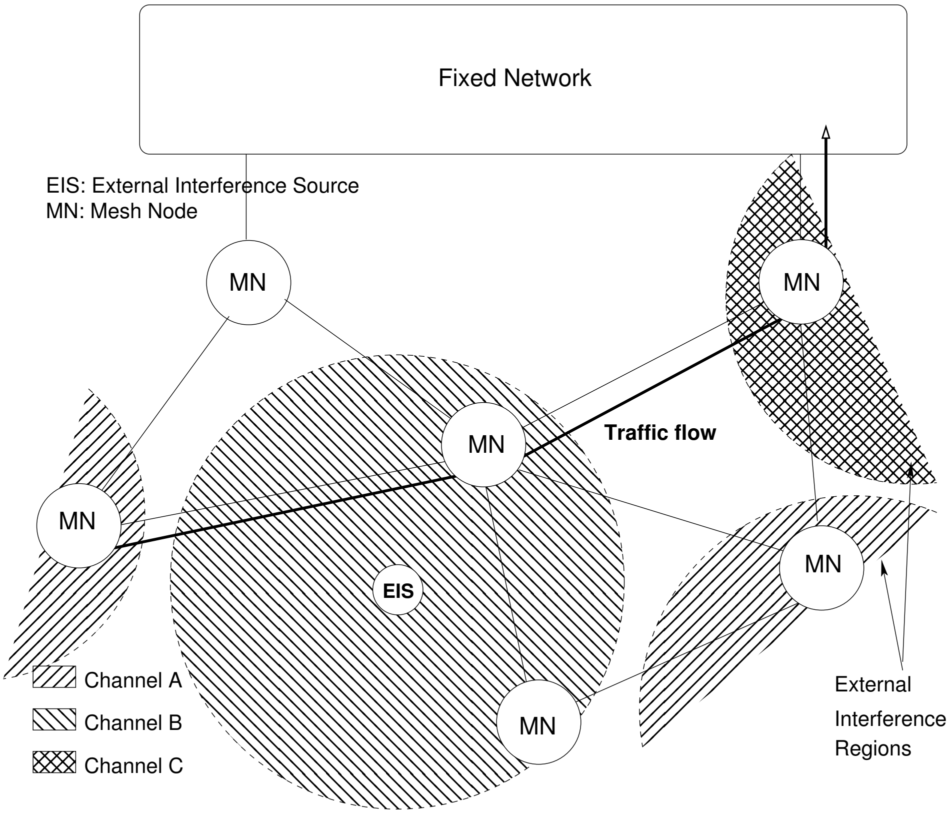


Figure 1.1: Wireless Mesh Network suffering from interference caused by external sources. If there are only three available channels (A, B and C), to avoid the external interference the mesh network has to be able to find the most suitable channel at each hop.

be tuned to different channels, forming a multi-channel WMN as shown in Fig. 1.2. Multi radio WMNs are able to offer higher network capacity by segregating the collision domains into multiple non-overlapping channels provided by IEEE 802.11 standards in the unlicensed bands.

Although IEEE 802.11s defines the mesh operation in a single channel, multi-radio mesh routers can form different meshes. The connection between different meshes is provided via bridging.

The 802.11a standard occupies a section of spectrum known as unlicensed national infrastructure (U-NII) band. The band takes up 300 MHz of spectrum and is divided into three sections of 100 MHz. The first two are next to each other and the third is 375 MHz up from the top of the second band. Due to the separation among channels, 12 of them can be considered as non-overlapping channels.

Although there are several non-overlapping channels available, the number

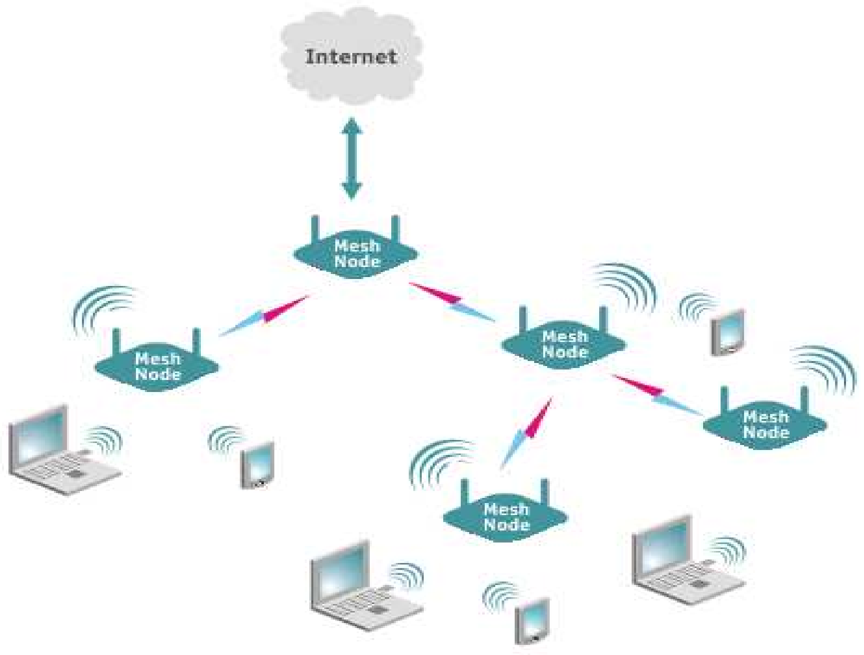


Figure 1.2: Multi-Radio Wireless Mesh Network

of channels that can be used simultaneously by a single node is limited by the number of radio interfaces installed on the node. Therefore a mechanism which selects the best channel, in terms of interference, among all available channels, is needed in order to achieve the maximum possible network performance.

1.1 Channel Assignment Mechanism

In multi-radio mesh networks, the Channel assignment (CA) [11, 51] is a mechanism which tries to find a feasible mapping between wireless channels and radio interfaces at each node with the aim of maximizing the capacity of the network.

A channel assignment solution must satisfy the following conditions to be feasible:

- The number of channels assigned to a node must be equal or less than the number of radio interfaces it has.
- The neighboring nodes must have at least a radio at a common channel to be able to communicate with each other.

This is a special challenging task in distributed wireless networks, as each channel may be shared by many wireless devices from the same or different networks, which makes it difficult to predict the amount of available bandwidth in each channel. In addition, nodes may not be aware of the criteria followed by the other nodes of the same network in the channel selection, thus increasing the uncertainty of any CA decision.

Based on the time duration between consecutive runs of the channel assignment protocol, CAs are categorized as: static; dynamic; semi-dynamic; and hybrid [11].

1.1.1 Static Channel Assignment

Most of CA proposed in the literature fall into static category [63, 60, 15, 34, 54, 6, 52, 50, 45, 4], where nodes tune their radios to certain channels permanently. Static CAs are easy to deploy but unsuccessful to cope with the changes in the wireless environment [63].

1.1.2 Dynamic Channel Assignment

Dynamic channel assignments [23, 7], on the other hand, enforce nodes to switch their interface dynamically from one channel to another between successive data transmissions. Therefore they require tight synchronizations among nodes. Dynamic approaches are only used for single radio nodes working over multiple frequencies, since they can not exploit the advantages of multi-radio networks [11].

1.1.3 Semi-Dynamic Channel Assignment

Static CAs could be easily extended to be semi-dynamic [43, 59, 3, 36, 44, 32, 57, 30, 17, 40] if the node refreshes the channels assigned to the radios on a regular time period. Semi-dynamic CAs adapt fast to the changes in the traffic pattern and the interference on the wireless medium from both internal and external sources. However to maintain the network connectivity, neighboring nodes are supposed to share a common channel [43, 14].

1.1.4 Hybrid Channel Assignment

Hybrid CAs [43, 59, 40] apply a semi-dynamic channel assignment to the fixed radio interface of each node while the other interface is controlled dynamically. Wireless nodes which use hybrid CAs, do not share common channel with their neighbors, since the dynamic radio switches to the channel of the neighboring nodes to make the connection.

CHAPTER 2

Motivations

Wireless Mesh Networks (WMNs) are supposed to be the next generation of Internet back-haul due to their easy deployment, self-configuring and self-healing properties. However the capacity of the WMNs is limited due to wireless interference.

To employ the benefits of WMNs, channel assignment problem must be solved to provide better performance by making the neighboring routers to be able to transmit over non-overlapping channels.

Using non-overlapping channels improves the performance of the network by increasing the number of wireless links which can transmit simultaneously without interfering each other. Moreover due to the increasing number of wireless devices, which operate on the same channel, a mechanism is needed to move the network or a transmission to another channel to avoid performance degradation due to appearance of an interfering device.

It is proven that by means of an efficient channel assignment the network throughput can be enhanced by the factor of 6 or 7 compared to the single channel network [45, 34]. However the channel assignment is a challenging task and although many proposals have been presented recently, It has been shown that, the proposed mechanisms are unable to make use of the available channels efficiently [58].

Many challenges make channel assignment algorithms to be inefficient.

Wireless antenna Current commodity of 802.11 antennas suffer significant switching delay from 3 *ms* to 20 *ms* [37]. The switching delay increases the end to end delay if the wireless routers, on the path between the source and the destination, switch their radio frequently. Moreover due to the imperfect filtering at wireless antenna the two antennas which installed on a single node may interfere each other if the antenna separation is less than a specific length [5, 48]. This limits the number of radio interfaces that can be installed on a node.

Wireless media The unpredictable wireless media is an important challenge toward selecting less busy channels. The increasing number of wireless devices sharing the same medium, introduces wireless interference over channels which changes randomly over time. The frequent changes in the status of the channels affects the decision of the channel selection mechanism and reduces the performance of the system by frequent switching.

Wireless link The distance between wireless routers determines the quality of the links between them [46]. Many other factors like the mobility of routers, environmental obstacles, external noise and noise figure of the antenna, determine the quality of the wireless links over time. The channel assignment must consider the temporal variation in wireless links, since the links with low delivery probability are ignored by routing protocols. Moving the low quality links to better channels has not any impact on the throughput of the network.

Routing metric The channel assignment affects the routing protocols which uses a dynamic metric, that is, changing the channel of the links changes the routing metric which considers the channel diversity of the path. On the other hand changes in the transmission path affect the channel assignment decision which considers the traffic to select a channel for the link. This is why there are many articles which consider the joint problem of channel assignment and the routing. However, those proposed channel assignment mechanisms are not flexible to be applicable with any preexisting routing protocol.

Connectivity Maintaining the connectivity between nodes which have radios on different channels is challenging since wireless nodes need to share the same channel to have a common link, that is, if the neighboring nodes do not have any radio interface over the same channel there can not be any wireless link between them.

NP-hard Assigning channels to the wireless links with the limited number of radio interface at each node is proved to be NP-hard [54, 34].

Interference pattern Adaptive channel assignments do not keep the record of interference over channels. The channel selection is done by considering the current situation of the network and channel occupancy [57, 43]. However the channel that has been busy before may be appear busy again in the near future.

Synchronization Dynamic channel assignment protocols need tight synchronization between nodes, that is, the nodes which want to initiate data transmission over a channel, which differs from their current channel, should communicate to move to the destination channel and notifying their neighbors about the time that they will switch back to the common channel.

Central coordinator Most of channel assignment protocols depend on a central node or a common channel to ease the connectivity and synchronization.

Many multi-radio mesh networks are implemented for scientific or commercial use [38]. The two best known multi-radio examples are are MobiMESH [5] and QuRiNet [58].

MobiMESH is a mesh network with mobility support and designed using the IEEE 802.11 off-the-shelf technology. In MobiMESH, the interference has proven to be a major issue, both because of external devices and the design problems of commercial IEEE 802.11 network adapters, which experience radiation leakages and interference. Moreover, it has also been reported that the distance between the two interfaces on a router severely affects the performance, even though the two interfaces are working on different orthogonal channels. MobiMESH does not benefit from any adaptive channel assignment mechanism and each radio on a router is tuned to a specific channel permanently [5].

The Quail Ridge Wireless Mesh Network (QuRiNet) is an experimental wireless mesh and environmental sensor network. QuRiNet consists heterogenous nodes which utilize dual-radios and multiple orthogonal IEEE802.11g channels. The evaluation of different channel assignment on QuRiNet shows that there is no single channel assignment algorithm that does well overall. The channel assignments are suboptimal when applied to a live mesh network because temporal variations in the link quality metrics are not taken into account [58].

This research work tries to find solutions for the need of designing an efficient channel assignment mechanism. We have proposed a static channel assignment which considers the quality of the wireless links into account and uses the orthogonal channels to prone the network topology (Chapter 5). We have also proposed an adaptive channel assignment which minimizes the internal and external interference over wireless links by means of a real time learning algorithm. The real time learner modifies the channel selection according to the changes in the environment while considering the previous situation of each channel. The synchronization between nodes is achieved using hello messages without any reliance on a central node nor the common channel (Chapter 7).

CHAPTER 3

Background and Related Work

Channel assignment has been studied extensively during the last years. Although many solutions have been considered for channel assignment [11, 51, 43, 59, 3, 36, 44, 32, 57, 30, 17, 63, 60, 15, 34, 54, 6, 52, 50, 45, 4], few proposals are adaptive to the changes in the wireless environment such as the interference induced from other wireless networks [43, 57, 40].

3.1 Static Channel Assignment

The static channel assignment problem is well studied in recent years and has been addressed in several proposals [14, 52, 45, 34, 15, 54, 3, 36, 50, 6]. The detail classifications of channel assignment methods is presented in [11, 51]. The core idea of all proposed algorithms is to use the available channels to eliminate the interference of neighboring transmissions.

A simple approach for utilizing two channels in a dual-radio network is presented in [15], while the main focus of authors is on modifying the routing parameters to benefit from multi-channel structure. Further investigations in [34] and [45] show that it is possible to increase the performance of the multi-channel network more than two factors by applying smart channel assignment algorithms.

Raniwala et al. [45] presented a traffic aware channel assignment. Given the network topology and the traffic profile, the channel assignment binds each radio to a channel such that the available bandwidth on each link is proportional to its expected load. If the traffic loads change over time the algorithm must perform channel reassignment. Skalli et al. [52] also formulated the channel assignment problem considering the traffic load of each node. Their priority based scheme, uses a common channel on all nodes to exchange control messages and to maintain the network connectivity. It gives higher priority to nodes close to the gateway to occupy better channels. But using traffic dependent schemes in presence of dynamic traffic profile is very challenging since the channel assignment output affects the routing protocol decision and in return changes the traffic load over links.

Marina et al. [34] and Subramanian et al. [54], formulated the static and traffic independent channel assignment as a topology control problem, and develop their approaches subject to minimize the link conflict weight. The channel assignment mechanisms proposed in [34, 54] assign channel to all link to preserve the network topology and do not consider the different quality offered by wireless links. Avallone et al. [6] formulated the problem to reduce the size of collision domains by assigning links to non-overlapping channels. However they also do not consider the delivery probability of the wireless links.

Recently Dhananjay et al. [14] proposed a distributed channel assignment and routing which takes the links delivery ratio into accounts to find the shortest path to the gateway. In the proposed algorithm each node follows the channel assignment pattern which is propagated by the gateway i.e. the algorithm optimizes the paths to the gateway, thus the big size of collision domains between links far from the gateway, especially in a dens network leads to a lower performance in an arbitrary traffic profile.

3.2 Semi-Dynamic and Hybrid Channel Assignment

Many semi-dynamic and hybrid CAs have been addressed in previous proposals [43, 59, 3, 36, 44, 32, 57, 30, 17], but few proposals consider the effect of the external interference [43, 57].

A simple semi-dynamic approach is proposed in [32] for a mesh network with two radio interfaces per node. Although the authors introduce a new path metric which takes into account the interface switching cost in addition to the expected transmission time [15], the proposed mechanism considers only the internal interference.

Breath first search channel assignment (BFSCA) [43] is the first interference aware CA mechanism. In BFSCA each node estimates the external interference through monitoring the wireless media and coordinates with a central node through a common channel. The central coordinator then assigns channels to links considering the distance of each link to the coordinator and the quality of the link in terms of transmission delay. The main drawbacks of BFSCA are: it needs tight synchronization between nodes and, the channel assignment algorithm is very slow and time consuming since it does an exhaustive search over all interfering links to find the best channel for each link.

Urban-X is another adaptive and hybrid channel assignment [57]. Urban-X is proposed for a network where each node must have at least three radios. One radio of all nodes is tuned to a common channel and is used for control traffic. The channel assignment considers the external interference in addition to the number of flows at each node, to make decisions. The best channel is occupied by the receiving radio of a node which has more traffic to send, although it may not receive any traffic.

3.3 Game Theory Based Formulation of Channel Assignment

Channel assignment algorithms using game theory models have been studied recently in some works [17, 21, 9, 49]. None of the proposed algorithms considers the effect of co-channel interference. All approaches consider that nodes or players have information about all strategies and payoffs, it means that all nodes make decisions based on a global payoff table. However, in a scenario having external interference, it is difficult to have a perfect knowledge of the channel use, before making decisions.

Felegyhazi et al. [17], formulate the channel assignment problem as a game where traffic flows compete for shared channels in a conflict situation. Although the algorithm converges to a stable Nash Equilibrium, their work is limited to a single hop single collision domain network, where each node participate in only one traffic flow.

Further extensions of this work for multi hop networks but limited to one collision domain are presented in [21, 9, 29], where nodes are limited to communicate with devices in their transmission range. However, although it may be an unrealistic assumption, the authors assume that each node knows about the existence of all other nodes in the network and the channel they use.

Kim et al. [29] did not put any constraints for the number of radios per node. The proposed game should be played sequentially and channels should be reallocated for any changes in the traffic profile. The approach proposed in [21] formulates the channel assignment as a cooperative game where nodes are cooperating with each other to improve the network throughput. The channel reallocation is necessary for any changes in traffic pattern. Shah et al. [49] formulate the game for multiple collision domains, but they use a static game which is limited to find a Nash Equilibrium for competing flows.

3.4 Validation and Verification of Channel Assignment Simulation

Simulating any CA in the current network simulators needs some general components which are missing in all simulators. The necessary modifications for ns-2 simulator [16], for evaluating multi radio wireless networks, are presented in [1]. The manual is restricted to the static channel assignment, which assigns a channel to the radio interface of a node before the simulation starts and keeps the configuration until the end of the simulation.

Ns-3 simulator [41] provides the basic features to simulate static or semi-dynamic CAs but they are not enough for simulating hybrid or dynamic CA mechanisms.

This research is the first work which provides the essential steps toward simulating channel assignment protocols in ns-3 simulator.

Furthermore, to the best of the our knowledge, there is no previous work in validating the channel assignment simulation.

CHAPTER 4

Contributions

4.1 Publications

4.1.1 Conferences

1. Maryam Amiri Nezhad, Llorenç Cerdà-Alabern. **Utility Based Channel Assignment Mechanism for Multi Radio Mesh Networks**. *In Proceedings of the 8th ACM international workshop on Mobility management and wireless access (MobiWac '10). Bodrum, Turkey October 17-18, 2010. ACM 2010, ISBN 978-1-4503-0277-7 PP. 68-74.*
2. Maryam Amiri Nezhad, Llorenç Cerdà-Alabern. **Adaptive Channel Assignment for Wireless Mesh Networks Using Game Theory**, *in MeshTech'11 (Fifth IEEE International Workshop on Enabling Technologies and Standards for Wireless Mesh Networking), Valencia, Spain, October 17, 2011. PP. 746-751 ISSN: 2155-6806.*
3. Maryam Amiri Nezhad, Boris Bellalta, Manel Guerrero-Zapata, Llorenç Cerdà-Alabern. **Should Next Generation Wireless Mesh Networks Consider Dynamic Channel Access?**. *In New Trends in Next Generation WLANs (Special Session at BCFIC 2012, The Baltic Congress on Future Internet Communications) April 25th, 2012.*

4.1.2 Journals

1. Maryam Amiri Nezhad, Llorenç Cerdà-Alabern, Manel Guerrero Zapata. **UBCA: A Centralized Utility Based Channel Assignment Mechanism for Multi Radio Mesh Networks**. *Accepted for publication in Scientific Research and Essays, 2011, [Impact Factor 0.445 Multidisciplinary Sciences 59/33 Q3]*

2. Maryam Amiri Nezhad, Llorenç Cerdà-Alabern, Boris Bellalta, Manel Guerrero Zapata. **A Semi-Dynamic, Game Based and Interference Aware Channel Assignment for Multi-Radio Multi-Channel Wireless Mesh Networks** . *Accepted for publication in the International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC), 2012, [Impact Factor 0.435 Telecommunications 80/58 Q3]*
3. Maryam Amiri Nezhad, Manel Guerrero Zapata, Boris Bellalta, Llorenç Cerdà-Alabern. **Simulation of Multi-Radio Multi-Channel 802.11 Mesh Networks in Ns-3**. *Submitted to The ACM Transactions on Modeling and Computer Simulation (TOMACS)", 2012 [Impact Factor 1.114 Computer Science, Interdisciplinary Applications 99/54 Q3, Mathematics, Applied 245/61 Q1]*
4. Maryam Amiri Nezhad, Boris Bellalta, Manel Guerrero Zapata, Llorenç Cerdà-Alabern, Enric Monte Moreno. **Game Theory Formulation of Channel Assignment for Multi-Radio Multi-Channel Wireless Networks with Unbalanced Resources**. *Submitted to IEEE Communications Letters 2012.*

4.2 Summary of Contributions

The paper 4.1.1-1 presents a static channel assignment which is traffic independent and considers the delivery probability and the usefulness of the wireless links. Based on estimating the wireless links usefulness, the channel assignment makes an efficient decision for assigning good channels to good links. The paper shows that the topology preserving constraint (assigning channel to all available links), as it has been done in most of static channel assignments, leads to a suboptimal solution, and relaxing this constraint improves the results considerably. The paper presents the simulation and numerical results which shows that our protocol outperforms other static channel assignment algorithms.

The paper 4.1.1-2 presents SICA, an interference aware channel assignment protocol. SICA estimates the amount of interference over channels, induced by any external wireless device, based on IEEE 802.11k standard. The paper then presents a game theory model to formulate the semi-dynamic channel assignation problem. Unlike previous game formulation in the literature, we assume a more realistic scenario by explicitly considering the presence of external interferences from other networks and, we assume that nodes do not have perfect information about others' strategies. Then we apply a real-time learning method to design a distributed algorithm which assigns channels to radios while avoiding the ripple and channel oscillation effects. The nodes continuously refine their decision accounting the changes in the wireless environment. The paper provides simulations using ns-3 and compares SICA with other channel assignment mechanisms that have been proposed in the literature. Results demonstrate the effectiveness of SICA in exploiting channel diversity, hence reducing the interference over wireless links and improving the system performance in terms of capacity and supported nodes and networks.

The paper 4.1.1-3 provides an overview of how 802.11s based mesh networks works. It tries to find out which kinds of channel assignment can be implemented in a 802.11s based mesh network. It also presents some of the recent proposals for adaptive channel selection which can be adapted to the standard. And, it shows the performance gains of dynamic channel assignment in multi-channel mesh networks compared to the single channel one.

The paper 4.1.2-1 is a revised and extended version of the paper 4.1.1-1, which uses queuing theory to show that the maximum throughput and the bottleneck delay is highly affected by the quality of the path to the gateway. The paper also presents more detailed results which compare the performance of our protocol in two more additional modes: pruning the network topology before applying the channel assignment and using our protocol while preserving the network topology. The performance of the protocol is also investigated in a network with different number of radio interfaces for each node and also for different number of available channels.

The paper 4.1.2-2 is the revised and improved version of the paper 4.1.1-2. The paper provides a modified game theory model that handles better the internal and external interference. The new game theory model is based on a set of control parameters that improve the system adaptability to the changing environment conditions. Using the new game theory model, SICA achieves a gain equal to 11% compared with the former protocol. Finally, a new random topology is introduced to evaluate SICA, as well as we extend the protocols to which it is compared, showing that SICA outperforms all of them.

The paper 4.1.2-3 shows in detail how to simulate a hybrid channel assignment protocol using ns-3 simulator without any need to modify the simulator's source code. The paper contains the definition of new classes and the presentation of class diagrams. As a specific example, we present the simulation of SICA using the presented classes. As another contribution, the paper presents the simulation verification and the validation of the proposed CA model using Markov model and the State Space Checker algorithm. The paper also describes how other existing CA protocols could be adopted to implement the presented approach.

The paper 4.1.2-4 presents a new game theory model for distributed channel assignment. The channel switching for each radio interface is modeled as a Markov model. The Markov process helps nodes to estimate the future gain of changing channels. The result is used as the prize to the game. The paper shows that by tuning the game parameter the result obtained by the learner working based on the defined model has an acceptable distance from the best response.

CHAPTER 5

Centralized and Static Channel Assignment

5.1 Introduction

Static Channel Assignment has been researched in many papers [45, 34, 6, 15, 44, 43, 54, 3, 36, 50]. However, most of the schemes disregard the delivery probability of wireless links, i.e. they suppose that all wireless links offer the same performance for data transmission. Normally, the delivery probability of a link strongly depends on its length, because the received power decreases drastically with increasing the distance. In addition, in a mesh network with a gateway most of the data traffic is directed to/from the gateway, not all wireless links are useful. Therefore links close to the gateway should be selected with higher probability by the routing protocol.

In this chapter, we propose a new channel assignment that takes these features into consideration and demonstrates its benefits by a performance comparison with other relevant channel assignments algorithms that have been proposed in the literature.

Our contributions that set our work apart from the existing approaches for channel assignment problem are as follows:

- We show that the topology preserving constraint (assigning channel to all available links) leads to a suboptimal solution, i.e. relaxing this constraint improves the results considerably.
- We propose a new centralized channel assignment algorithm which is traffic independent.
- We consider the delivery probability and the usefulness of the wireless links to make an efficient decision for assigning good channels to good links. Simulation results show the effectiveness of our approach.

- We show that the maximum throughput and the bottleneck delay is highly affected by the quality of the path to the gateway.

The rest of the chapter is organized as follows: Section 5.2 contains the description of the network model and formulation of the problem. The channel assignment mechanism is proposed in Section 5.3. We report the simulation results in Section 5.4. The discussion on the efficiency of the method is presented in Section 5.5. The work is concluded in Section 5.6.

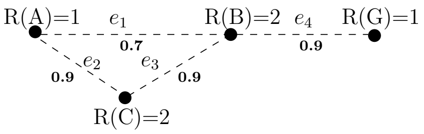
5.2 Network Model

We consider a multi-radio wireless mesh network (WMN) consisting of a set of mesh routers (nodes) where some nodes serve as gateways between the WMN and the wired network. We assume that each node has at least $R \geq 1$ radio interfaces (radios) and can tune each radio to one of the frequencies selected from C non overlapping channels. For simplicity, we assume that all radios have the same characteristic.

We model the connectivity between nodes by an undirected graph $G_t = (V, E)$; henceforth referred to as the topology graph. Here V denotes the set of nodes, whereas E denotes the set of links. A pair of nodes have a link in E if they are connected in the network. We associate to every link e a weight equal to its packet delivery probability ($p_d(e)$). Since the wireless links may interfere with each other while transmitting simultaneously, the topology graph is not sufficient to fully characterize the wireless network. To account the impact of interference on a transmission we use the interference protocol model defined in [24]. In this model, two transmissions links will interfere if they occur within the interference range of each other. The interference range of a link is usually supposed to be two times the transmission range.

To represent the interference among all possible transmissions in a network, the conflict graph is used [28]. The conflict graph $G_f = (V_f, E_f)$, contains a set of vertices corresponding to all links in the network topology. There is an edge between two vertices in the conflict graph if the corresponding links interfere with each other. We define the interference weight for a link e ($I(e)$), as the number of links that potentially interfere with e, consequently the interference weight of a link is equal to the degree of the corresponding vertex in the conflict graph.

Throughout this chapter, we use the topology graph to model the network topology, and the conflict graph based on the protocol model for the wireless interference.



Channel assigments:

(a)	(b)
$e_1 : c_1$	e_1 : none
$e_2 : c_1$	$e_2 : c_1$
$e_3 : c_2$	$e_3 : c_2$
$e_4 : c_2$	$e_4 : c_3$

Figure 5.1: Channel assignment mechanism

5.2.1 Problem Formulation

The static channel assignment problem in a multi-radio wireless mesh network is to find a feasible mapping between radios and channels (see 1.1 for feasibility conditions).

The aim of our channel assignment algorithm is to utilize the available channels effectively, to reduce the interference of all links as much as possible. In most of the cases, due to the limited number of radios per node and the big interference weight of wireless links, it is not possible to eliminate the interference over all links completely. Moreover in a channel assignment strategy each decision will limit the flexibility of the next decision, as we show in the following example.

Fig. 5.1 shows a simple network with four nodes where $R(v)$ shows the number of radios of node v. The numbers under the links show the packet delivery probability offered by each link. Consider a channel assignment algorithm which starts from node A and assigns channel c_1 to its single radio. To preserve the topology, nodes B and C must tune one of their radios to channel c_1. Therefore they lose their flexibility in making the decision for one of their radios. The algorithm finished its work by assigning channel c_2 to the other links incident on nodes B and C (Fig. 5.1(a)). In order to calculate the capacity offered by this topology, capacity can be defined as the maximum number of possible concurrent transmissions in the network [8]. Since in this network all wireless links interfere with each other and are assigned to channels c_1 or c_2, at most two transmissions can occur concurrently (one on each channel). Therefore the network capacity would be $2 * 0.9 = 1.8$.

From another standpoint, since the link between B and A is lossy (compared to other links), if the channel assignment omits this link, node B can choose two different channels for its radios, rather than c_1, and obviously we achieve a better channel utilization (Fig. 5.1(b)). The maximum capacity in the second topology is $3 * 0.9 = 2.7$, since all remaining links can transmit concurrently over different channels.

This example shows that omitting some lossy links may allow the channel assignment to reach a more optimal solution.

Note that, removing a wireless link from the network topology is possible if endpoint nodes do not share any common channel, therefore it is possible to use channel assignment to prune the network topology. On the other hand, if the lossy links are removed before channel assignment, CA may add those links to the topology by putting the nodes on a common channel, if it is not aware of removed links. Moreover the process of removing links should be done in a controlled way without affecting the network performance.

We study the problem to find a channel assignment that reduces the interference over good links with a slack restriction on preserving the network topology. In order to measure the amount of interference in a channel , c, we define the average of the interference weight of the links in c as:

$$F_c = \frac{1}{|E_c|} \sum_{e \in E_c} I_c(e) \tag{5.1}$$

where E_c is a set of links assigned to a channel, c, and $I_c(e)$ is the interference weight of a link, e, in a channel, c. So, the aim of our algorithm is keeping F_c as low as possible over all channels.

Our channel assignment assigns a priority to each link based on its performance and then visits each link in order. It then finds the best channel for the link by comparing the value of F_c for all channels and selecting the channel which has the minimum interference weight (F_c). We explain the channel assignment in more detail in Section 5.3.

5.2.2 Link Quality Estimation

To assess the delivery probability ($p_d(e)$) of a link we use the shadowing propagation model (equation (5.3)) [46]. Measurement based propagation models for radio communication systems indicate that, the average received signal power decreases logarithmically with distance. The average path loss for an arbitrary transmission-receiver separation is expressed as a function of distance.

$$\overline{P_L(d)}_{dB} = P_L(d_0) + 10\beta \log(\frac{d}{d_0}) \tag{5.2}$$

where β is the path loss exponent and is usually empirically determined by field measurements. Large β, indicates more obstructions and hence, faster decrease in average received power as distance becomes longer. The value of β depends on the specific propagation environment, here we consider an urban area, and use $\beta = 2.7$ [46]. d_0 is the close-in reference distance which is determined from measurements close to the transmitter and d is the transmitter-receiver separation distance.

In reality, the received power at certain distance may be vastly different at two different location due to the surrounding environmental clutter. Measurements have shown that at any value of d the path loss at a particular location is a random variable, thus the communication range of a wireless radio is not an ideal circle. Equation (5.3) predicts the mean received power at distance d based on the Shadowing propagation model.

$$\left[\frac{\overline{P_r(d)}}{P_r(d_0)}\right]_{dB} = -10\beta\log(\frac{d}{d_0}) + X_{dB} \tag{5.3}$$

where (X_{dB}) is a Gaussian random variable (with zero mean and standard deviation σ) and reflects the variation of received power at certain distance. We use $\sigma_{dB} = 6$ throughout this work.

Since $P_r(d)$ is a random variable with normal distribution the Q-function (equation (5.4)) can be used to determine the probability that the received signal level will exceed a particular level:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp(-\frac{u^2}{2})\mathrm{du} \tag{5.4}$$

The probability that the received signal level will exceed a certain value ζ can be calculated from cumulative density function as:

$$Pr[\overline{P_r(d)}|_{dB} > \zeta] = Q\left(\frac{\zeta - \overline{P_r(d)}}{\sigma}\right) \tag{5.5}$$

Packets are delivered correctly if the received power is greater or equal to a threshold (e.g. *RXThresh* in network simulator ns-2) [16]. Therefore, the delivery probability at distance d is given by equation (5.6)

$$p_d(e) = Pr[\overline{P_r(d)}|_{dB} \geq 10log_{10}(RXThresh)] \tag{5.6}$$

In our model we consider that, links exist only when $p_d(e) \geq 0.5$, this delivery probability is achievable if d is not longer than 131.53_m. For analytical sections of this work we use equation (5.6) to calculate the delivery probability of links. Note that *RXThresh*, antenna gain and height are set based on default values in [16].

5.3 Utility Based Channel Assignment (UBCA)

In this section we describe an algorithm (UBCA) for assigning channels to radios, which is developed based on our model. Note that UBCA is a centralized algorithm which considers a network with at least one gateway. The wireless links are assigned a priority for assigning channels. To assign the priority to the links we consider the gateway placement in the network.

We start by defining the following terms:

Free radio: Whenever the number of channels assigned to a node, is less than the number of radios it has, the node is supposed to have some free radios.

Potential link: In a multi-channel network, the availability of a link depends on the physical distance between end point nodes and the existence of common channel between them. Therefore, we call a link as potential, if the endpoint nodes are physically neighbors but they have no common channel. Note that if a link remains potential at the end of the channel assignment, it is actually removed.

Our channel assignment algorithm (UBCA) has two phases (see Alg. 5.1). In the first phase, UBCA chooses the most diverse channel set for links without having tight restriction for network connectivity. In this phase, if the algorithm can not assign any common channel to the end point nodes of a link, it marks the link as potential. In the second phase, UBCA makes the final decision for potential links: It tries to make one common channel for them through merging channels over endpoint nodes, or removes them from the topology.

At the beginning of the algorithm, each link is given a priority based on its delivery probability and utility. We describe the exact criteria for determining the priority of each link in the next section (Section. 5.3.1). UBCA visits each link based on the priority order (line 1 in Alg. 5.1). For each visiting link, the algorithm first determines a possible set of channels, and then selects the best channel for the link among that set. To select the channel for a link, UBCA investigates the effect of adding the visiting link to all possible channels and chooses the channel with lower interference average (See Alg. 5.3 and equation (5.1)).

If the possible channel set for a link is empty, then UBCA marks the link as potential for the next phase. The size of the possible channel set for a link depends on the situation of its endpoint nodes (See Alg. 5.2). If both endpoint nodes have free radios, then it is possible to assign a new channel to the visiting link, for this case the possible channel set contains all available channels. In the case that one endpoint node has no free radios, selecting a different channel from the current channel set of that node is against the first condition of the feasibility (Section 5.2.1), therefore the possible channel set for the visiting link is equal to the current channel set of the node which has no free radio (lines 3-5 in Alg. 5.2). If both end point nodes do not have any free radio, then the possible channel set must be equal to the common channel between the two nodes. Finally in the case that nodes have no free radios neither common channel, the possible channel set for the visiting link would be empty, and the link remains potential (line 9 in Alg. 5.2).

In the second phase, UBCA visits the remaining potential links. It must decide to remove a potential link from the topology or recheck the channel assignment to make a common channel between end point nodes. A possible way to create a common channel between two nodes is to select one channel from each node's current channel set and merge them to one, i.e. assigns all links in one channel to another (Alg. 5.4). For selecting two appropriate channels to merge, the algorithm must consider the interference weight of the links after merging channels. However UBCA is mostly oriented to remove unnecessary links. For this reason, in this phase, it visits the potential links in an increasing order of priority i.e. the link with the lowest priority is visited first. For each potential link, it checks whether there is any other path between the two end point nodes. If so, the link is removed, otherwise the channel merging process is applied to make the link available.

Theorem 1. The proposed channel assignment algorithm satisfies the feasibility conditions.

Proof. The feasibility conditions are mentioned in Section 5.2.1. Here we express the proof for each condition separately.

Radio-constraint: For each link, the best channel is selected among the possible channel set which is determined by Alg. 5.2. To determine the possible channel set for a link, Alg. 5.2 checks the radio-constraint condition for end point nodes through lines 2-8. Therefore, the possible channel set for a link is selected based on the current channel set of end point nodes and their free radios. Hence the radio constraint condition is preserved.

Connectivity-constraint: The second phase of Alg. 5.1, decides to remove a link under two conditions: first, there must be an alternative path between two endpoint nodes; and second, the path must be independent of the current link (see lines 3-4 of the second phase of Alg. 5.1). If these two conditions are held, then the algorithm removes the link. Therefore, even after deletion of a link, there is a path between two nodes, and thus the network remains connected.

```
  Input:
    $G_t = (V, E)$: The topology graph
    $G_f$: The conflict graph of $G_t$
    $C$: The available frequency channels
Require: $C > 0$
  Channel Assignment Phase 1:
Order potential edges of $G_t$ in non-increasing order of their priority
for Unvisited and Potential edge $e =< v, u >$ in order do
    $PCh \leftarrow Get.Possible.Channel(e, C)$
    if $PCh$ is $\varnothing$ then
        Mark $e$ as "Visited and Potential"
    else
        $c \leftarrow Find.Best.Channel(e, PCh, G_f)$
        Assign $c$ to one radio of each endpoint nodes ($v$,$u$)
    end if
end for
  Channel Assignment Phase 2:
Order all "Visited and Potential" links in $G_t$ in an increasing order of their
priority
for Potential link $e =< v, u >$ in order do
    if Any path $P$ between $u$ and $v$ and $< u, v >\notin P$ then
        Remove $e$ from $G_t$
    else
        $C_v \leftarrow$ Channels that have already been assigned to node $v$
        $C_u \leftarrow$ Channels that have already been assigned to node $u$
        $c_v \leftarrow \{c \in C_v | F_c < F_i \forall i \in C_v\}$
        $c_u \leftarrow \{c \in C_u | F_c < F_i \forall i \in C_u\}$
        $Merge.Channels(c_v, c_u)$
    end if
end for
```

Algorithm 5.1: Utility Based Channel Assignment(G_t,G_f,C)

5.3.1 Computing Links Priority

Recall that the objective of our channel assignment is to reduce the average interference weight of the high performance links by removing unnecessary links. Therefore it is necessary to visit the preferred links first. To this aim considering the delivery probability is not sufficient since links removal must be carried in such away that have the minimum impact on the paths to the gateway. To assess the role of each link in constructing the paths to the gateway we define the Utility metric for each link ($U(e)$). $U(e)$ will help us to estimate the probability of using a specific link by routing protocol. Without any traffic profile, it is pretty hard to estimate the Utility precisely, but we can have a good estimation by considering a balanced traffic over the network.

```
Input:
  $e =< u, v >$: The link between nodes $v$ and $u$
Output:
  $PCh$: A set of possible channels for the given link
Require: $C > 0$
if $v$ and $u$ have free radio then
    $PCh \leftarrow C$
else if at least one node has free radio then
    $Lim.Node \leftarrow$ The node which has no free radio
    $PCh \leftarrow$ Current channels of the $Lim.Node$
else if $v$ and $u$ have no free radio but common channels then
    $PCh \leftarrow$ Common channels between $u$ and $v$
else
    $PCh \leftarrow \varnothing$
end if
return $PCh$
```

Algorithm 5.2: Get.Possible.Channel($e =< u, v >$,C)

To compute $U(e)$, independent of specific traffic pattern, We use the shortest path first (SPF) and consider all shortest paths between the gateway and other nodes in the network. The shortest path, between two nodes, is a path with the lowest cost from one node to another. The cost of a path is the total cost of its participant links while the cost of a link is equal to the inverse of its delivery probability ($1/p_d(e)$) i.e. the expected transmission count over the link, assuming Bernoulli trials[13]. We define $U(e)$ equal to the number of times that link e participates in constructing the shortest paths between the gateway and other nodes (Alg. 5.5).

In a network with $|V|$ nodes consisting one node as a gateway, we have $|V|-1$ paths from all nodes to the gateway. Thus, we estimate the probability of using a link for a transmission to the gateway as the utility of that link ($U(e)$) over the total number of paths ($\frac{U(e)}{|V|-1}$).

Since our channel assignment algorithm tries to prune the topology by deleting some links, it is important to start channel assignment from links with the higher utility. Through our simulation study, we found that in all topologies many links have the utility equal to zero or one. Therefore, after sorting all links based on their utility, links with the same utility must be ordered based on their delivery probability. This sorting can be done by defining the priority $P(e)$ of each link e as:

$$P(e) = \gamma * \frac{U(e)}{|V|-1} + (1-\gamma) * p_d(e) \quad \forall\, e \,\in\, E \tag{5.7}$$

```
Input:
  $e =< u, v >$: The link between nodes $v$ and $u$
  $PCh$: A set of possible channels for the given link
  $G_f$: The conflict graph
Output:
  $C$: The best channel for the given link
if $|PCh| > 1$ then
    $MinF \leftarrow I(e)$
    for $c \in PCh$ do
        $E_c \leftarrow E_c \cup e$
        Compute $F_c$ from equation (5.1)
        if $F_c \leq MinF$ then
            $MinF \leftarrow F_c$
            $C \leftarrow c$
        end if
    end for
else
    $C \leftarrow PCh[1]$
end if
return $C$
```

Algorithm 5.3: Find.Best.Channel($e =< u, v >$,PCh,G_f)

```
Input:
  $c_{src}$, $c_{dst}$: The channels that must be merged

Require: $c_{src}, c_{dst} > 0$
for $v \in$ The set of nodes which have one radio tuned to channel $c_{src}$ do
    Assign $c_{dst}$ to the radio of node $v$ instead of $c_{src}$
end for
```

Algorithm 5.4: Merge.Channels(c_{src},c_{dst})

where γ is a tuning parameter subject to $0 < \gamma \leq 1$. Big γ prefer links with higher utility while small γ is preferable for networks without any gateway. We use $\gamma = 0.9$ as a default value for the algorithm. Note that $\gamma = 1$ means classifying links only considering their utility which may be equal for many links and thus results in an ambiguous classification.

5.4 Performance Evaluation

In this section, we study the performance of the proposed channel assignment algorithm using the R numerical tool [42] and the ns-2 [16] simulator. We use R to compare the capacity and interference properties of different multichannel algorithms [34, 54, 15, 14]. Detailed ns-2 simulations are used to evaluate the performance of the channel assignment algorithms in 802.11-

```
Input:
  $G_t = (V, E)$: The topology graph
  $S_g$ : Set of gateway nodes
Compute utility, considering all paths to the gateway
if $S_g \neq \varnothing$ then
    for $v \in V$ do
        $g_v \leftarrow \arg\min_{g \in S_g} \mathrm{SPF}(g, v)$
        $S_p[v] \leftarrow \mathrm{SPF}(g_v, v, \mathrm{Cost} = 1/p_d)$
    end for
end if
for $e \in E$ do
    $U(e) \leftarrow$ Number of repetition of e in $S_p$
end for
return $U$
```

Algorithm 5.5: Computing the Utility of Links (G_t,S_g)

based multi-radio mesh networks. We have add the multi-radio functionality to the physical and MAC layer of 802.11 in ns-2 simulator based on the work done in [1]. The routing tables in the ns-2 simulations are obtained using SPF, while considering $1/p_d(e)$ of any link e as its weight.

To assess the delivery probability ($p_d(e)$) of link e we use the shadowing propagation model [46] with a path loss $\beta = 2.7$ and standard deviation $\sigma_{dB} = 6$ dB. We have used the default ns-2 values for the other propagation model parameters (see [16]). In our model we consider that, links exist only when $p_d(e) \geq 0.5$. With the selected values, this delivery probability is achievable if the distance of the link is not longer than 131.5 m.

We consider a network with different number of nodes ($10 \leq N \leq 30$) which are randomly placed in a square field of 300×300 m^2.

We assume the protocol model for interference between wireless links with an interference range equal to 263.06 m (two times the communication-range). Throughout we assume that all nodes are equipped with two radios that can be tuned over 12 non-overlapping channels. We also compare the performance of channel assignment protocols with a network in which nodes are equipped with more radio interfaces and in the case that different number of channels are available.

We compare UBCA with three relevant channel assignment algorithms that have been proposed in the literature: the Common Channel Assignment (CCA) [15]; the Connected Low interference Channel Assignment algorithm (CLICA) [34]; the distributed channel assignment (ROMA) [14]; and Greedy Channel Assignment (GCA) [54].

CCA applies the same channel assignment pattern for all nodes, i.e. the first radio of all nodes is tuned to the first channel, the second radio is tuned to the second channel and so on. Therefore if each node has r radio interfaces, regardless of the number of available channels, the network created by CCA always uses r channels.

CLICA is a centralized channel assignment which tries to reduce the interference weight of the links while preserving the network topology. CLICA visits the nodes based on their priority which is defined by their distance to a reference node and the number of free radios they have. Here the reference node is the gateway. While assigning a channel to a link, each end-point node will lose one of its free radios and thus during the channel assignment the priority of the nodes will change dynamically. CLICA selects a channel for a link in such a way that leads to the lowest amount of interference weight over that link and all other links which are interfering with it [34].

ROMA is a distributed algorithm proposed for a network with at least one gateway. At the beginning of the channel assignment each gateway produces a channel sequence $(c_1, c_2, \cdots, c_n)$, and broadcasts it. The node which is i hops far from the gateway will select the c_{i-1} and c_i elements of the channel sequence, and tune its radios to the selected channels. Therefore, at the end of the procedure each node will have a common channel with its previous node on the path to the gateway, and a common channel with its neighbors at the same and lower level.

GCA is a centralized algorithm which tries to minimize the interference on wireless links by assigning interfering links to different channels. The algorithm does not consider any gateway and gives the same priority to all links. The greedy algorithm runs in several rounds and in each round it checks all channel-link pairs to find the channel which must be assigned to a link and minimizes the total interference weight in the network. The algorithm ends if it can not decrease the total interference weight in the network anymore.

We also investigate the performance of our protocol UBCA with some changes; first without removing any links, which means that UBCA acts as a topology preserving algorithm (UBCA-TopologyPreserve); and second remove the links with zero Utility from the network topology before running the channel assignment and then apply UBCA (UBCA-RemoveFirst).

UBCA-TopologyPreserve is chosen to investigate the importance of link removal in channel assignment. On the other hand UBCA-RemoveFirst is chosen to proof that the link removal should be done in a controlled way to avoid decreasing the network performance.

Single channel network is also used as a base to compare the performance of multi channel networks.

5.4.1 Topology Properties

Capacity-Gain

We use the maximum number of concurrent transmissions as an estimation for network one-hop capacity [8]. We calculate this metric in two steps: First, computing all independent sets in the conflict graph, and then selecting the set which gives us the maximum capacity factor, as we explain in the following.

The simplest way to determine the capacity factor ($C_f(C)$) of a multi-channel network with C orthogonal channels, is by considering the cardinality of the largest independent set, S, of the conflict graph of each channel $G_f(c)$ [8]. To take into account the delivery probability in the capacity metric, we calculate the summation of the delivery probability of the links in each independent set (equation (5.8)).

$$C_f(c) = \max_{\forall S \subset G_f(c)} \sum_{\forall e \in S} p_d(e) \tag{5.8}$$

Links over non-overlapping channels are able to transmit simultaneously, therefore the capacity factor of the network is the total capacity acquired over all channels (equation (5.9)).

$$C_f(C) = \sum_{\forall c \subset C} C_f(c) \tag{5.9}$$

We define the capacity gain of the multi-channel network in relation to the single channel network in equation (5.10), where $C_f(C)$ and $C_f(1)$ represent the capacity factor of a multi-channel network with C orthogonal channels, and a single channel network respectively.

$$\text{Capacity Gain} = \frac{C_f(C)}{C_f(1)} \tag{5.10}$$

Network Interference

We use two metrics to show the interference characteristic in a multi-channel network: the collision domain size; and the link conflict weight.

Size of collision domains: A collision-domain is a subset of links in which all links collide each other if they transmit simultaneously. A collision-domain in the conflict graph is a complete subgraph or clique of vertices. All vertices in a clique are connected pairwise, therefore the set of their corresponding links in the topology graph make a potential collision domain.

Maximum average interference weight: The interference weight for a link is the number of links in its interference range (Section 5.2.1). We calculate this metric taking the maximum of equation (5.1) over C. This metric is important since a link with the maximum interference weight could be a potential bottleneck for the network. In CLICA [34] the authors use a similar metric as the objective function.

5.4.2 802.11 Based Multi-Radio Performance

We use ns-2 simulator to evaluate multi-radio networks created by different static channel assignment algorithms in terms of aggregate throughput, packet delivery ratio, and average delay. In each network we produce random traffic flows. We use two type of sources: CBR traffic with fixed rate at 100 kbps and packet size equal to 1 kB; and TCP traffic with packet size equal to 1.4 kB. We consider two traffic profiles: *gateway profile* consisting of flows from the gateway to randomly selected nodes; and *random profile* consisting of flows between random pairs of nodes. The simulation time was set to 100 s. RTS/CTS mechanism is enabled. For each topology with different number of nodes, we repeated the simulation for 50 different random placement of the nodes, and report the average with the confidence intervals.

Aggregate Throughput: For TCP traffic we calculate the aggregate throughput (Mbps), dividing the total received traffic by the duration time of TCP flows.

Packet Delivery Ratio (PDR): We consider the number of correctly received packets with respect to the amount of sent packets.

Average Delay: For all received CBR packets, we calculate the average delay to verify the ability of the network to use non interfering channels to deliver data with less contention.

5.4.3 Results

As explained before, Fig. 5.2-5.4 have been computed with the R numerical tool [42], analyzing the properties of the topology graphs obtained by the channels assignment algorithms under study. Note that these properties do not reflect the fact of having a gateway. Fig. 5.5-5.17, on the other hand, show the results of analyzing the traces obtained using the ns-2 simulator [16].

Fig. 5.2 shows the capacity gain of multi-channel networks (equation (5.10)). The results are produced by different channel assignment algorithms varying the number of nodes. Using CCA the capacity gain is bounded to two, since CCA only uses two channels throughout the network [15]. The figure shows that GCA, outperforms the other mechanisms, but it performs close to UBCA as the number of nodes increases because UBCA removes some links and reaches a more diverse solution. The good performance of UBCA is because two main reasons: first, considering the delivery probability of the links during the channel assignment; and second, removing some useless links from the original network. As expected, removing useless links will reduce the collision domain size, thus resulting a considerable increase in capacity gain, but it must be considered that removing many links from the topology as it has been done in UBCA-RemoveFirst does not necessarily improves the capacity gain since it decreases the number of possible concurrent transmissions. As expected UBCA-TopologyPreserve performance is close to the other topology preserving algorithm, CLICA.

Fig. 5.3 shows the average size of the collision domains for different number of nodes. CLICA and GCA are successful in reducing the size of collision domains compared to CCA, but the reduction is much higher with UBCA and ROMA. Note that, even after increasing the network density, the size of collision domains do not change much for the topologies created by UBCA and ROMA.

Fig. 5.4 depicts the maximum interference weight of the links for different network densities. The figure shows that by increasing the network density, the maximum interference weight do not change significantly for UBCA, while for the other mechanisms increases rapidly. Although CLICA and GCA are designed to minimize this metric, they couldn't achieve a reasonable result compared to UBCA since they tries to preserve the topology and as explained in section 5.2.1, this approach couldn't achieve a diverse-channel assignment hence unsuccessful in eliminating the interference over links. The same reason justifies the higher value of maximum interference weight for UBCA-TopologyPreserve compared to the original protocol (UBCA) and UBCA-RemoveFirst.

To investigate the performance of the proposed channel assignment in a general situation, we first run the simulation for the gateway profile. Fig. 5.5 depicts the aggregate TCP throughput with different number of nodes and 5 TCP flows. The figure shows that due to a significant increase in the network capacity (see Fig. 5.2), UBCA and ROMA outperform the other mechanisms especially in dense topologies. Note that UBCA- RemoveFirst performs the same as UBCA, since it removes all links that never used to reach the gateway and the resulted CA solution will be same as UBCA for gateway traffic profile.

Fig. 5.6 shows that the packet delivery ratio in network topologies created by UBCA, CLICA and ROMA are also much better than CCA or GCA.

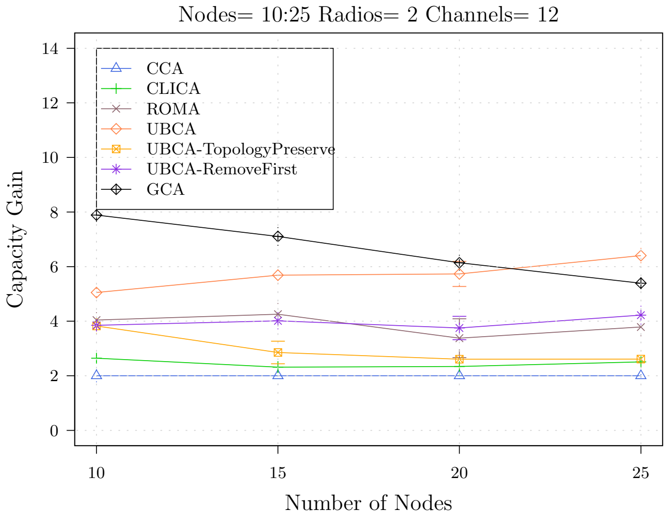


Figure 5.2: Capacity Gain.

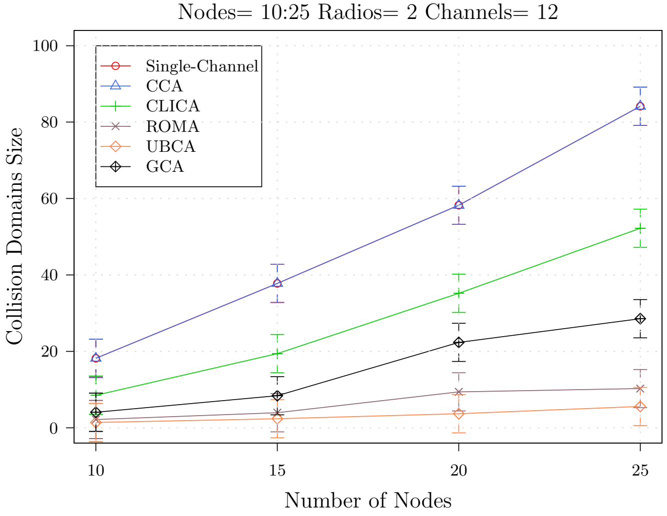


Figure 5.3: Size of the collision domains.

Moreover, in contrast to other mechanisms, delivery ratio in UBCA, CLICA and ROMA is rather insensitive to the network density. GCA as expected does not perform well for gateway traffic profile since it does not consider the gateway placement for channel assignment.

Fig. 5.7 shows that by using UBCA, CLICA and ROMA the average delay for CBR traffic is significantly lower than with the other algorithms. This confirms its smaller size of collision domains, and lower interference weight obtained in Fig. 5.3 and Fig. 5.4 respectively.

We re-run the simulation by considering different number of traffic flows. The network consists of 15 nodes randomly distributed in a square field of

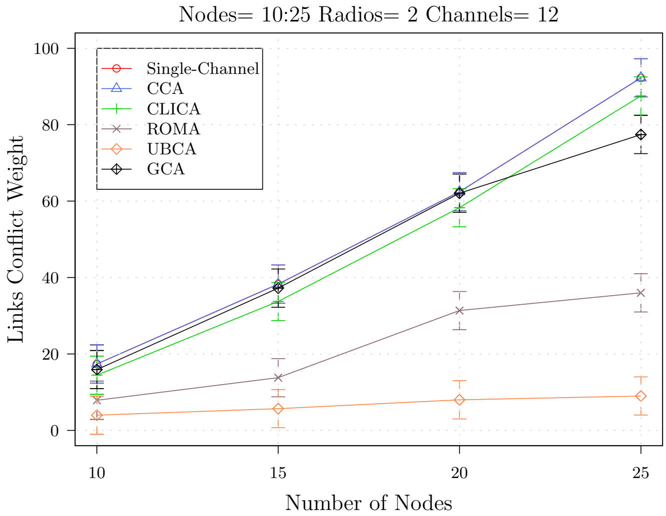


Figure 5.4: Maximum interference weight.

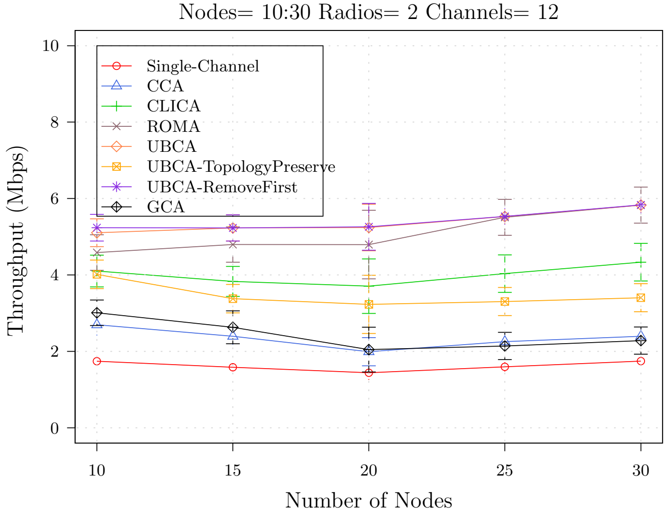


Figure 5.5: Aggregate throughput for TCP traffic (gateway profile).

$300 \times 300\ m^2$. the gateway profile in Figs. 5.8-5.9. Fig. 5.8 shows the aggregate TCP throughput with different number of nodes. As expected, due to the proper channel assignment in CLICA, ROMA and UBCA compared to CCA, the aggregate throughput improves significantly. UBCA get almost the same result as ROMA. Recall that ROMA is designed to optimize the gateway paths.

Fig. 5.9 shows that the packet delivery ratio in network topologies created by UBCA and ROMA is much higher than the others. Additionally, in UBCA and ROMA the rate of decrease in the packet delivery ratio with respect to the increase of the traffic flows is much lower than the others. Finally, Fig. 5.10 shows that in UBCA and ROMA, the average delay for CBR traffic

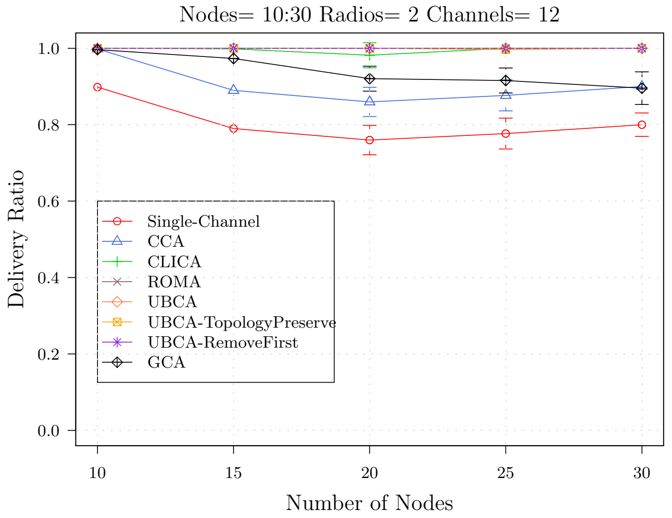


Figure 5.6: Average packet delivery ratio for CBR traffic (gateway profile).

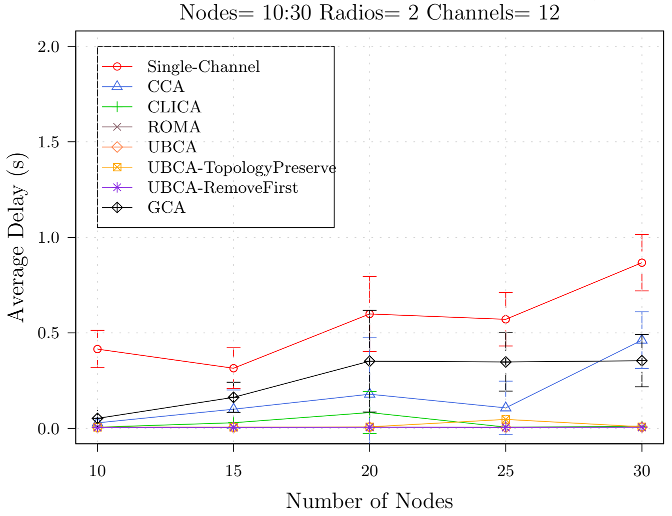


Figure 5.7: Average delay for CBR traffic (gateway profile).

is lower than the other algorithms. These results are justifiable by taking into account the fact that ROMA optimizes the paths to the gateway, and UBCA estimates the utility of the links based on their frequency to access the gateway.

We repeat the simulation for random profile of traffic flows. The aggregate throughput is shown in Fig. 5.11. The result shows that UBCA performs better for random traffic compared to others. Note that unlike the results obtained for gateway traffic profile, the network aggregate throughput using UBCA-RemoveFirst for random traffic profile is worst than the original protocol (UBCA) which shows that removing all links with the Utility equal to zero from the network topology may decrease the network throughput.

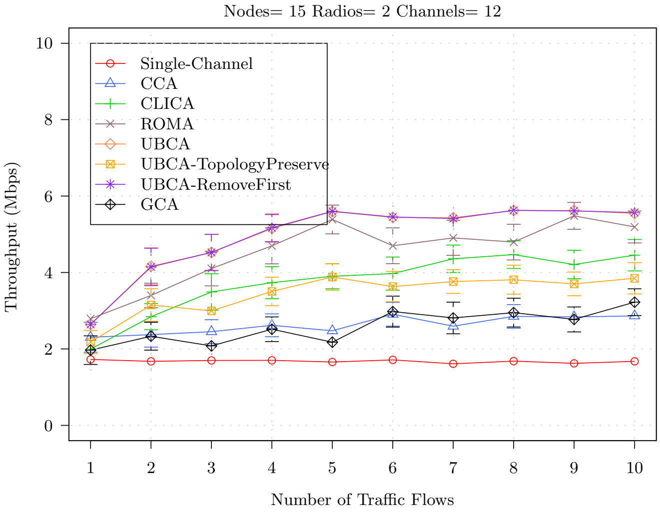


Figure 5.8: Aggregate throughput for TCP traffic (gateway profile).

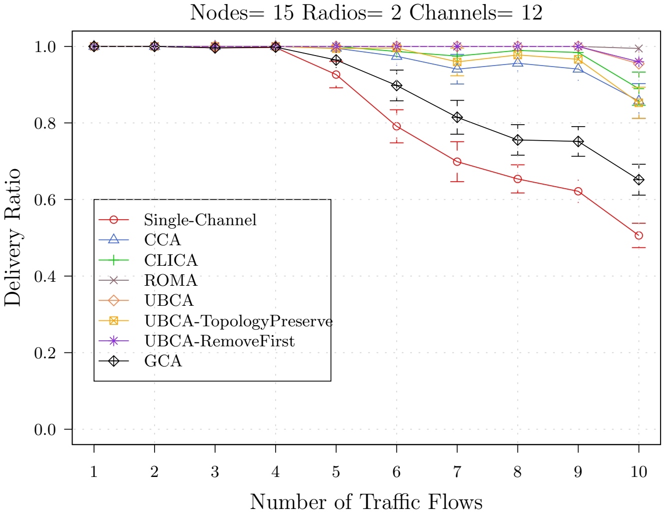


Figure 5.9: Average packet delivery ratio for CBR traffic (gateway profile).

We investigate the effect of increasing the number of radio interfaces on each node on the performance of the channel assignment protocols (except ROMA). Note that ROMA is proposed for a dual-radio network. Fig. 5.12 shows the aggregate network throughput for 5 TCP flows for different number of radio interfaces at each node. The figure shows that, increasing the number of radio interfaces at each node improves the performance of GCA significantly. It also shows that UBCA outperforms other CA mechanisms, but the performance of the network does not change a lot while using more radio interfaces for each node.

Fig. 5.13 is obtained for a network with two radio interfaces at each node while different number of channels are available. The figure shows that as

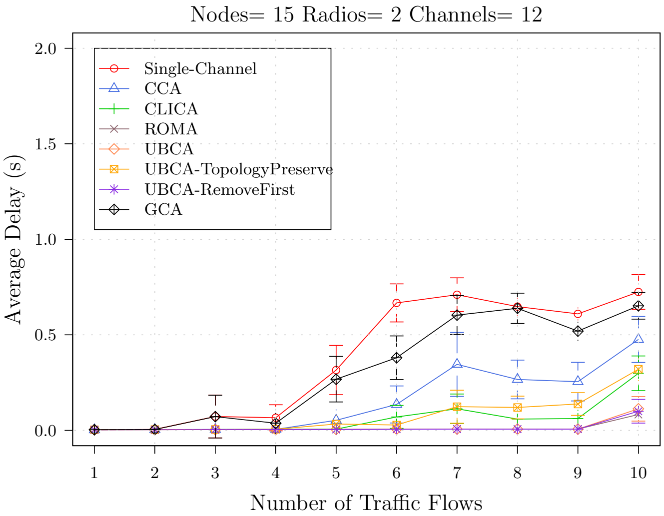


Figure 5.10: Average delay for CBR traffic (gateway profile).

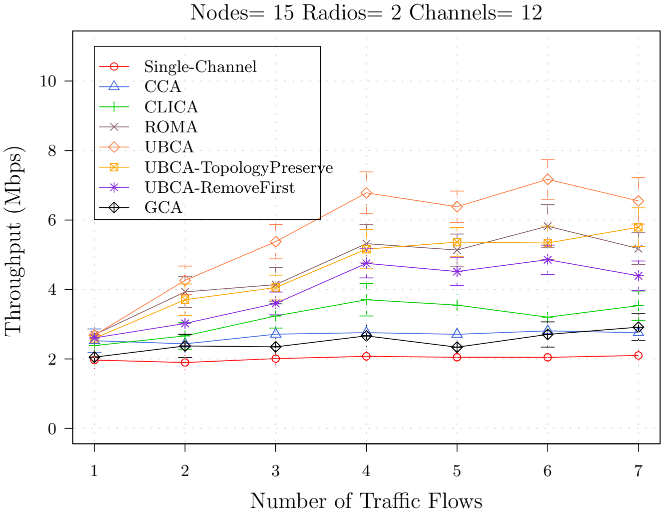


Figure 5.11: Aggregate throughput for TCP traffic (random profile).

the number of channels increase the performance of multi radio networks improves but it does not improve as the number of channels increases to more than 6. ROMA and UBCA performs better than others.

We conclude that based on the simulation results, UBCA builds a network topology with low interference with a small number of radios per node. We showed that removing some links from the network topology will lead to a better performance, even if it may cause an increase in the length of some paths between nodes. The link removal should be done in controlled way to not reduce the network performance.

The results confirm that performing channel assignment while disregard-

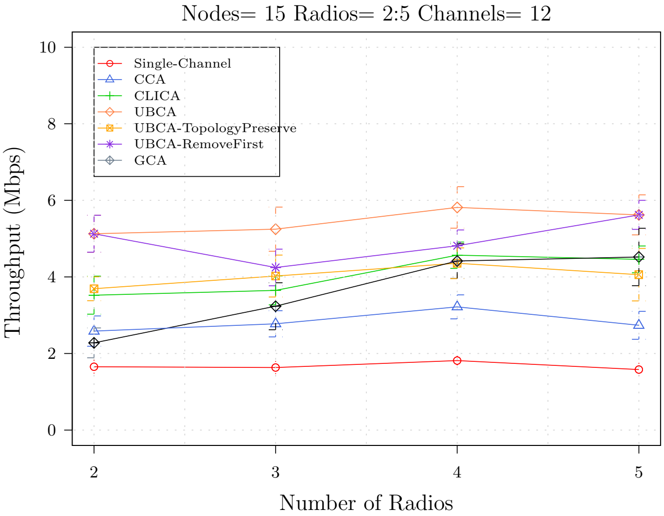


Figure 5.12: Aggregate throughput for TCP traffic (gateway profile) vs. number of radio interfaces per node.

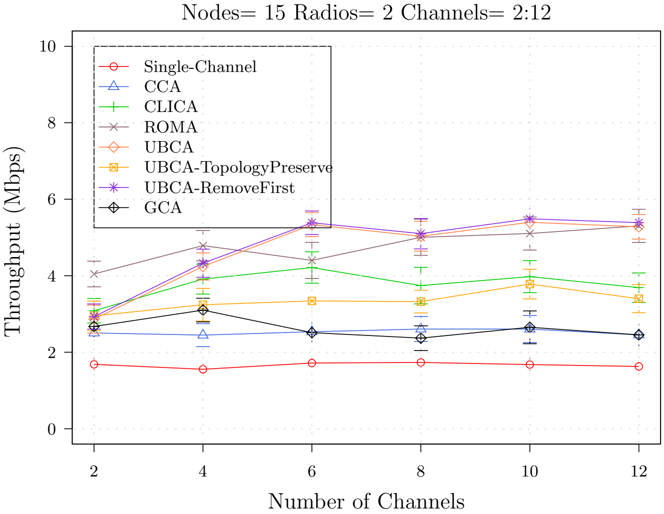


Figure 5.13: Aggregate throughput for TCP traffic (gateway profile) vs. number of available channels.

ing the gateway placement does not improve the network performance in a scenario where the traffic is oriented toward the gateway.

It is also important to note that removing the wireless links from the topology is possible if the end point nodes do not share any common channel. Therefore it must be done during the channel assignment. In other words if the links are removed from the topology before channel assignment, the CA mechanisms might make those removed links to be available later by putting the neighboring nodes on a common channel. Fig. 5.14 shows the number of removed links which are added to the topology after the channel assignment. The figure shows that if CA is not aware of the links which must be removed the links removal can not be performed correctly.

The Impact of γ

Recall that UBCA uses a weighted metric for assigning the priority to wireless links before channel assignment (Section 5.3.1). As we have explained previously, the priority metric is a weighted sum of the utility and the delivery probability of the wireless links. The weight given to the utility (U) is denoted by γ, while the weight given to the delivery probability (p_d) is $1 - \gamma$.

To investigate the impact of γ on the performance of UBCA, we repeated the simulation with $0.1 \leq \gamma \leq 1$ for three different network densities ($N = 10, 25, 50$).

Fig. 5.15 shows that by increasing γ, UBCA achieves better network throughput for TCP traffic generated in the gateway profile. Obviously the impact of different value of γ is more significant for dense networks ($N = 25, 50$), because big γ leads to decrease the interference over the links which are close to the gateway.

Fig. 5.16- 5.17 show that a larger value of γ leads to better packet delivery ratio with less delay for CBR traffic.

These simulation confirms that, since a large γ forces the channel assignment to favor the links with higher utility, for the gateway profile UBCA achieves better performance with increasing γ. For the random profile we have not obtained significant differences for different values of γ.

5.5 Discussion

In this section, we discuss the importance of considering the quality of wireless links during the channel assignment process with more details.

Recall that we associate two properties with each wireless link e: the delivery probability($p_d(e)$), defined in section 5.2.2; and the Utility ($U(e)$), defined and calculated in section 5.3.1.

We first study the relation between the quality of the path to the gateway with the throughput and bottleneck delay using queuing model.

We then evaluate the distribution of $U(e)$ over several random network topologies. Results confirms that links close to the gateway have much higher utility than other links. Therefore giving higher priority to links with higher utility to occupy better channels leads to better resource allocation.

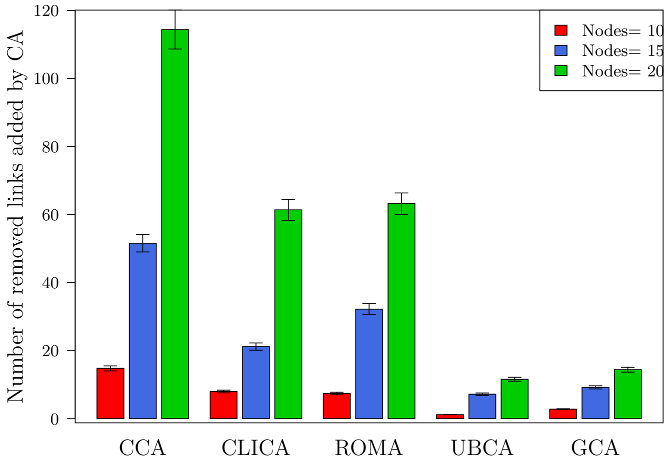


Figure 5.14: Number of zero utility links which are added by CA to the topology.

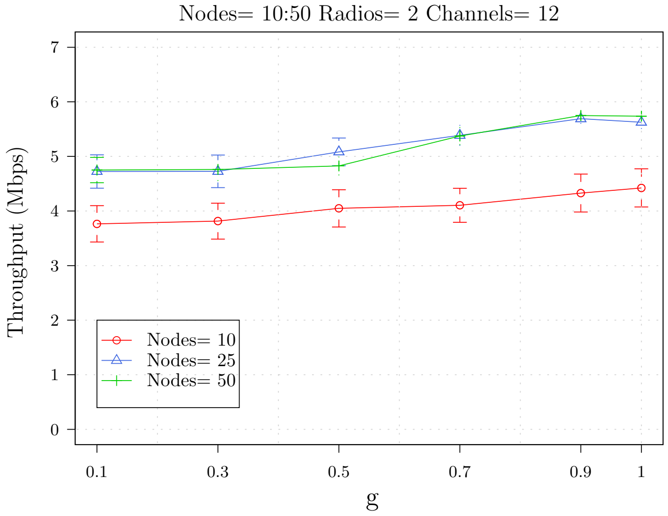


Figure 5.15: Aggregate throughput for TCP traffic using UBCA varying γ (gateway profile).

5.5.1 Bottleneck Delay and Throughput

Consider a mesh network with one gateway, we assume that all data requests from mesh routers aim to the gateway, therefore the gateway is treated as a service station in queuing system. We further assume that the gateway has an infinite buffer.

We define bottleneck delay, D_s, as the average delivery delay of data requests at the gateway while the service time of each request is identical and equal to T_s. We also define the throughput, λ, as the maximum data requesting frequency from all mesh nodes to the gateway.

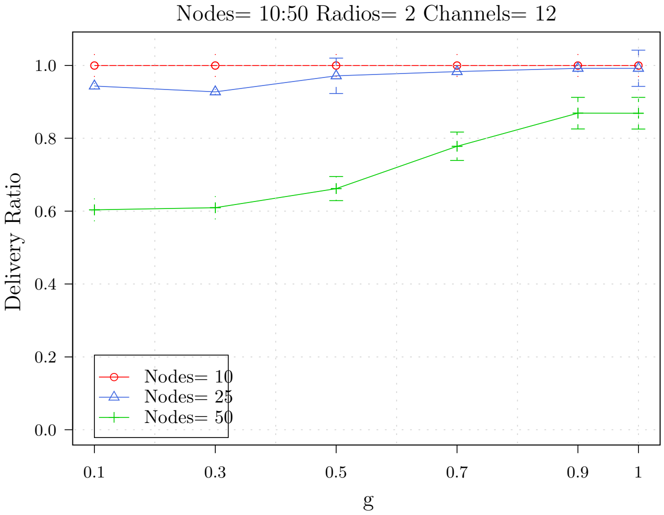


Figure 5.16: Average packet delivery ratio for CBR traffic using UBCA varying γ (gateway profile).

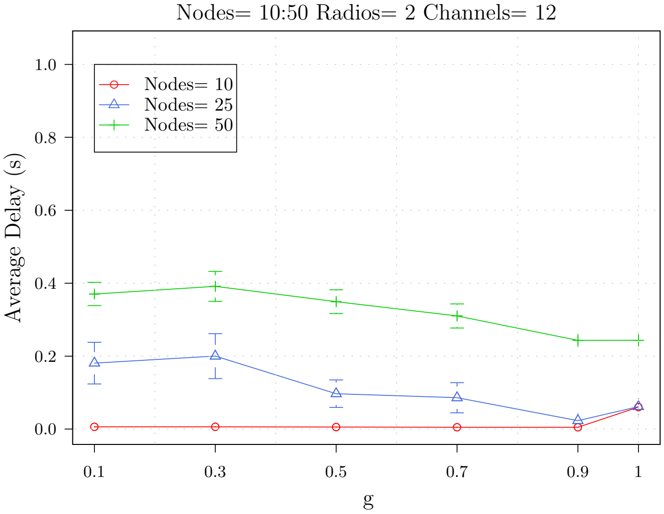


Figure 5.17: Average delay for CBR traffic using UBCA varying γ (gateway profile).

We assume that all mesh nodes (excluded from gateway) generate data request at the constant rate of τ. Considering the delivery probability of a wireless link ($p_d(e)$), if τ is the rate of data produced at one endpoint node then the data arrival at next node could be considered with the average rate of $p_d(e)\tau$. Since the delivery probability is sufficiently smaller than τ we can model the data arrival at the next node as a Poisson distribution with mean $\lambda_e = p_d(e)\tau$.

Suppose that all nodes are placed over one path to the gateway, and as a result of the channel assignment mechanism, all links are active over non-interfering channels. Since all transmissions can be done simultaneously, according to the additivity property of Poisson distribution, data requests arrival at gateway will be of Poisson distribution with mean value of λ:

$$\lambda = \tau \sum_{e \in S_e} p_d(e) \tag{5.11}$$

where S_e is the set of links which construct the path to the gateway. Equation (5.11) indicates that the quality of the shortest path to the gateway determines the upperbound for the throughput.

For a gateway with an infinite buffer; the constant service time of a data request; and data requests arrival at Poisson distribution, we can model the data incoming and outgoing as in an M/D/1 queue. Considering constant service time T_s, the expected value of T_s, is identical ($E[T_s] = T_s$) and the variation of T_s is zero. According to [59], the bottleneck delay for a linear topology can be estimated as:

$$D_s = T_s + \frac{T_s^2 \lambda}{2(1 - T_s \lambda)} \tag{5.12}$$

The data request arrival rate shows the bottleneck throughput and since the value of the *utilization factor* ($\rho = T_s \lambda$) should be smaller than one (in order to arrive at a steady state), the data arrival rate should be smaller than $\frac{1}{T_s}$. Therefore the maximum data rate for each mesh router is bounded by:

$$\tau = \frac{1}{T_s \sum_{e \in S_e} p_d(e)} \tag{5.13}$$

We consider one path to the gateway which consists of 10 mesh routers uniformly placed in a chain. We assume that after channel assignment each node can establish transmission only with its successive nodes on the path, moreover all links are established over non-overlapping channels. We change the distance between nodes to have different values for data delivery probability of wireless links, while the number of nodes is kept intact. As the distance between nodes gets larger the cost of the path to the gateway increases. Note that, all nodes (except the gateway) produces data requests at the constant rate of τ. We run the experiment varying the value of τ in the range of 20 to 100. The service time, T_s, is assumed to be 1 ms for all requests.

Fig. 5.18 shows the maximum data arrival rate, λ, at gateway, for different value of the data rate at each node, τ. As expected, the throughput drops as the distance between nodes gets larger. Fig. 5.18 depicts that, the higher the data rate, the more sensitive to the distance.

Fig. 5.19, on the other hand, shows the bottleneck delay for different data rates of mesh nodes. As expected the higher data arrival rate the bigger delay it suffers at the gateway to get serviced. Note that for $\tau = 100$ in a scenario where the distance between nodes is 50 m, the bottleneck delay goes to infinity. This phenomenon happens because for this topology, according to equation (5.13), the bottleneck data rate value is about 100, which prevent the system from reaching the steady state.

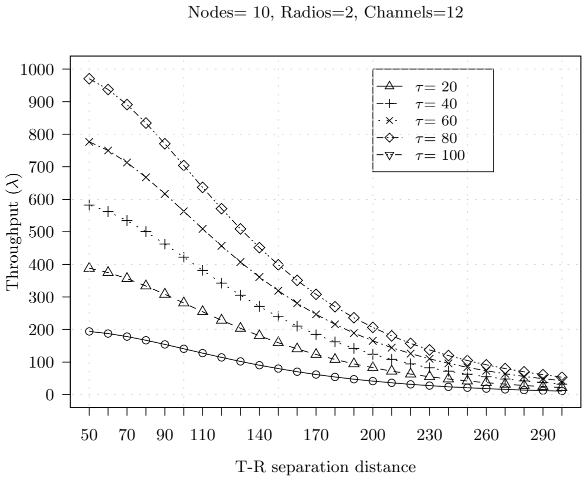


Figure 5.18: Maximum data request arrival at gateway for different data arrival rate at mesh nodes

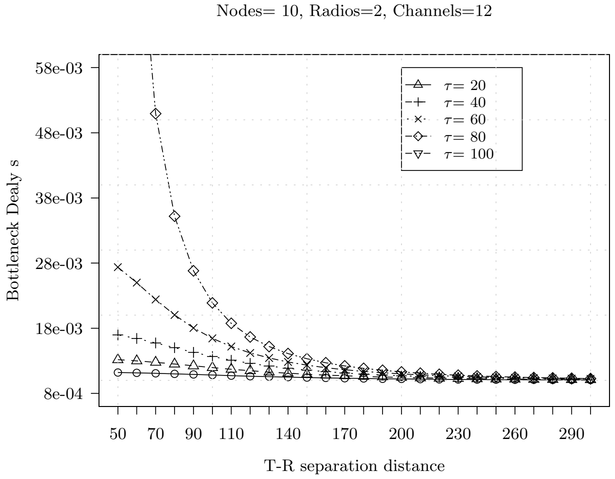


Figure 5.19: The bottleneck delay for different data arrival rate

5.5.2 Utility Distribution

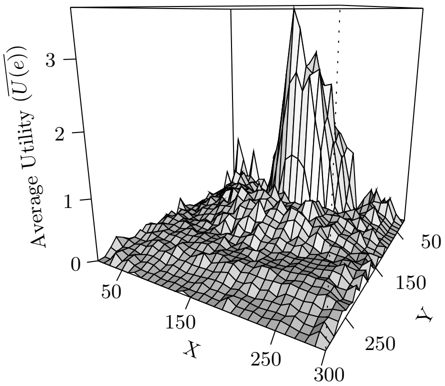


Figure 5.20: The distribution of links utility in a mesh network

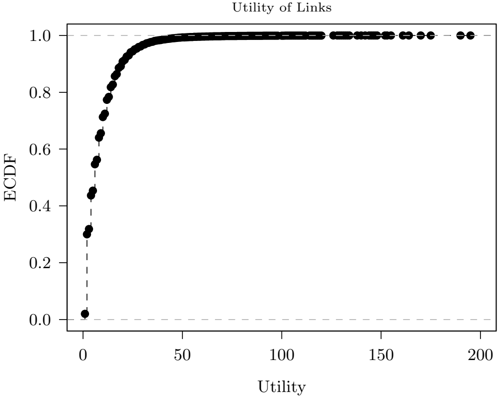


Figure 5.21: The ECDF of links utility

Category	Links Delivery Probability	Links Utility	Considering the paths to the gateway	Considering all paths
1	$\geq \overline{p_d}$	$\geq \overline{U}$	8.56%	14.39%
2	$\geq \overline{p_d}$	$< \overline{U}$	43.01%	37.18%
3	$< \overline{p_d}$	$\geq \overline{U}$	11.89%	20.03%
4	$< \overline{p_d}$	$< \overline{U}$	36.54%	28.40%

Table 5.1: The percentage of links, distributed in four categories based on their delivery probability and utility in a network

In a WMN with a gateway, the traffic distribution is usually skewed, since most of the user traffic is directed to/from the gateway. Therefore links close to the gateway should be selected with higher probability by the routing protocol. We define utility ($U(e)$) to estimate the probability of using a specific link, e, for a random traffic request (see section 5.3.1).

Note that, since a weak link with $p_d(e) < 0.5$ is not favorable for transmission, in our model we consider links with the delivery probability equal or greater than 0.5 (Section 5.2.2). In practice, routing protocols favor links which offer higher performance. Therefore most of the links with lower quality are likely useless.

We perform a simple experiment in our simulation area to verify the different types of links considering the delivery probability and the utility. We assume a network with one gateway and 24 nodes, which are distributed randomly in a $300 \times 300\ m^2$ square field. The gateway is fixed and placed on the right edge of the field. We calculate the average utility of links in different area of the network over 1000 different network topologies. Fig. 5.20 shows a perspective view of average links' utility, for all networks. As expected, in the neighborhood areas of the gateway, the average utility of the links has a considerable bigger value compared to other regions of the network. According to our experiments, 79.54% of links have the utility equal to 0, it means that, none of the network nodes use them to access the gateway, while 0.2% of links participate in more than 12 paths ($U(e) \geq 12$), i.e. the probability of using those links, for a transmission, is more than 0.5. Note that, the average of links utility ($\overline{U}$), in this network, is 0.42. To generalize our result, we calculate the utility of links regardless of the gateway, by considering not only the gateway-paths, but also, every other shortest paths between all pairs of nodes. Fig. 5.21 shows the empirical distribution function (ECDF) of links utility in this scenario. The figure shows that after considering all possible paths, a considerable amount of links have the utility close to zero.

We use the average of delivery probability ($\overline{p_d}$) and the average of utility ($\overline{U}$), to categorize the links based on their attributes: first the links with big delivery probability and utility; second the links with big delivery probability but small utility, third the links with small delivery probability but big utility; and forth the links with small delivery probability and small utility. Table 5.1 shows the percentage of links in four categories and for two mentioned scenarios: considering only paths to the gateway; and considering all paths between all pairs of nodes. As shown in table 5.1, the forth category contains at least 28% of the network links, which have the lowest amount of p_d , and are less important in the network (small U). Note that, this result is almost the same for a dense network.

From the result in this experiment, it is obvious that assigning high priority to the links in categories with higher U and p_d, to occupy better channels, will lead to better channel assignment, since these links are more probable to participate in traffic transmissions. On the other hand, it is important to disregard links in category four during the channel assignment. Removing weak links from the topology may lead to even better performance due to the following reasons:

- It reduces the potential interference over good links,
- It relaxes the restriction over channel assignment to make a common channel for weak links by considering finite number of radios per node,
- It doesn't have a big effect on the communication between nodes.

This simple experiment shows that; it is extremely unfair and illusive to threat all links identically.

5.6 Conclusions

In this chapter we have studied the static channel assignment problem in a multi-radio multi-channel mesh network. We have showed that considering the utility of the links makes it possible to estimate the usefulness of each link regardless of the traffic profile. We have presented a new algorithm, called UBCA, which assigns channel to links considering their utility without a tight constraint on preserving the topology. We have done a performance evaluation comparing UBCA with other relevant static channels assignment algorithms proposed in the literature. In our study, we have used a numerical tool to analyze the properties of the topology graphs, and detailed ns-2 simulations. Our numerical results demonstrate the effectiveness of UBCA in exploiting channel diversity for reducing the interference over wireless links with a small number of radios per node. The simulation results show that our approach increases the performance of the multi-radio mesh network significantly. Additionally, UBCA provides a considerable decrease in the size of collision domain and thus significant increase in the network capacity. The ns-2 simulations proved that pruning the network from some useless links leads to a better channel utilization and thus reduces the average delay and increases in the packet delivery ratio and throughput.

CHAPTER 6

Game Theory Modeling of Distributed Channel Assignment

6.1 Introduction

Game theory is considered to solve the non-cooperative and distributed channel assignment problem where nodes have conflict of interests for the wireless bandwidth. Many game theory based channel assignment solutions have been proposed recently [17, 55, 30, 21, 49]. However they do not consider the dynamic nature of wireless environment, which results in a different amount of available bandwidth in each channel due to presence of external devices. Moreover, they assume perfect information at each player, and in some cases there is no constraint for the number of radios installed on nodes [30].

In a scenario with external interference, it is not possible to have the perfect knowledge of the channel use before moving to that channel. The other shortage of the previous proposals is assuming all nodes to seek for a maximum identical amount of bandwidth while in a wireless network, nodes may reduce their data rate to avoid congestion or to save energy. The proposed games are solved disregarding the effect of past events on future decisions.

In this chapter we present a novel Game Theory based formulation for channel assignment problem which is general enough to be applicable in heterogeneous networks consisting of nodes equipped with different number of radio interfaces. We solve the game by means of an adaptive learning algorithm based on multiplicative weight updates. The numerical results show that, the learner is capable to cope with the random changes in the environment and reach a satisfactory stable situation in a limited amount of time.

The rest of the chapter is organized as follows: Section 6.2 contains the description of the problem and the game model. The best response to the game with one collision domain is discussed in Section 6.3. The general multiplicative weight learner algorithm is explained in Section 6.4 to solve the game for multi collision domain network. Numerical results is presented in Section 6.5 and the chapter is concluded in Section 6.6.

6.2 Problem Formulation

6.2.1 Scenario

We consider a multi radio wireless network that consists of N wireless nodes which are equipped with several radio interfaces. We assume that there are $|C|$ non-overlapping channels available.

We define $I_i < |C|$ as the number of radio interfaces installed on node i. Each wireless channel suffers external interference caused from external devices which belong to other networks. The goal of a node is to use the channels which have more available bandwidth in order to satisfy its required data rate.

We define the normalized available bandwidth of a channel c, $B_{free}(c) = \frac{BW(free,c)}{BW(total)}$, as the amount of bandwidth which is free over channel c related to the total bandwidth of the channel, where $BW(total)$ is the same for all channels. We assume that nodes can estimate the occupancy of channels using any of the bandwidth estimation methods proposed in the literature [51, 40, 43].

We define $r_{i,c}$ as the number of radios that node i has placed on channel c. Since the node does not get any benefit from placing more than one radio in the same channel, $r_{i,c}$ is limited to 1 ($r_{i,c} \leq 1$). Therefore, the bandwidth share available for node i over channel c is equal to:

$$B_{i,quota}(c) = \frac{B_{free}(c) r_{i,c}}{max(\sum_{n \in N} r_{n,c}, 1)} \tag{6.1}$$

We define ϑ_i as the bandwidth that node i wants to acquire. This minimum bandwidth can be satisfied using one or more channels at the same time.

6.2.2 Channel Assignment Game

We consider the problem of finding the best channel assignment as a game played between each node and the environment. The environment consists of all other nodes of the same network, as well as the devices from other networks that also operate in the same set of channels, that is, all wireless nodes which work in the interference range of node i. From now on we use the terms node and player interchangeably. We assume that, players are rational and selfish. However they do not know exactly the payoff matrix of the game, but they are able to measure the current strategy's payoff at the time. The game is played over several rounds and players can modify their decision in order to obtain a higher payoff.

We define s_i as the strategy of player i, which is the channel allocation vector of the player, given by:

$$s_i = (r_{i,1}, r_{i,2}, ..., r_{i,C}), \quad i = 1, \ldots, N \tag{6.2}$$

A node strategy, s_i, describes whether it has a radio over a specific channel or not which can be either a pure strategy or a mixed one.

Note that the total number of radios employed by player i is given by:

$$\sum_{k=1}^{C} r_{i,k} \leq I_i \tag{6.3}$$

We define the strategy matrix (strategy profile), S, as the strategy vector of all players at a given time:

$$S = \begin{pmatrix} s_1 \\ s_2 \\ ... \\ s_N \end{pmatrix} \tag{6.4}$$

By S_{-i} we shall refer to the strategy matrix which consists all nodes' strategies except player i.

We define the payoff of playing the strategy vector s_i equal to the bandwidth share that node i gets from all channels, on which it has radios, to the total available bandwidth of all channels in its neighborhood.

$$F(s_i, S_{-i}) = \frac{\sum\limits_{c \in C} B_{i,quota}(c)}{\sum\limits_{c \in C} B_{free}(c)} \tag{6.5}$$

We define φ_i as the minimum payoff that node i expects to gain considering its required data rate (ϑ_i).

$$\varphi_i = \frac{\vartheta_i}{\sum\limits_{c \in C} B_{free}(c)} \tag{6.6}$$

Nodes are not able to know the payoff of all possible strategies but they can choose a channel by measuring the payoff in the current strategy profile. However they must avoid *channel oscillation.* Channel oscillation happens when nodes detect that a given channel is occupied by many nodes and they decide to switch to another channel at the same time, eventually they will find the target channel occupied again. To avoid channel oscillation we use a Markov model to compute the expected payoff of selecting any channel discounting the expectation of the neighboring nodes (Section 6.2.3).

6.2.3 Markov Model

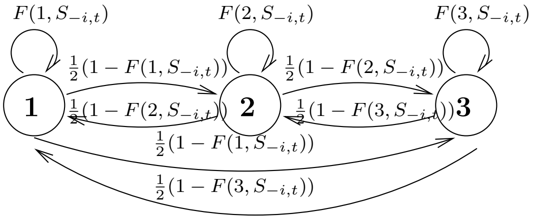


Figure 6.1: Markov model of transition from one strategy to another

The behavior of a node can be modeled as a Markov chain, when the node keeps the current strategy with the probability proportional to the payoff it gets and changes to other strategies with the probability proportional to the complementary of staying in the current strategy. Fig. 6.1 shows a chain with 3 strategies or states available for a node with one radio interface while $S_{-i,t}$ is the current strategy of the environment. The probability of staying in strategy s_i for a node is self transition over s_i and is equal to $p_{s_i s_i} = F(s_i, S_{-i,t})$. We assume that, the node has the same tendency to leave its strategy and occupy any other strategies. Therefore the probability of departure from strategy s to any other strategy $\acute{s}$ is considered as $p_{s_i \acute{s}_i} = \frac{1-F(s_i,S_{-i,t})}{|S_i|-1}$. Note that for a node with I_i radios, $|S_i| = \binom{|C|}{I_i}$ is the number of possible channel selections (strategies) while $|C|$ channels are available. The transition matrix of the Markovian process is:

$$P_i(S_{-i,t}) = \begin{bmatrix} F(1,S_{-i,t}) & \frac{1-F(1,S_{-i,t})}{|S_i|-1} & \frac{1-F(1,S_{-i,t})}{|S_i|-1} & \dots \\ \frac{1-F(2,S_{-i,t})}{|S_i|-1} & F(2,S_{-i,t}) & \frac{1-F(2,S_{-i,t})}{|S_i|-1} & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \tag{6.7}$$

We define the predicted payoff of selecting any strategy as the stationary vector, that is, the probability of selecting a strategy after a hight number of steps while $S_{-i,t}$ remains unchanged ($\rho_{i,t} = \rho_{i,t} P_i(S_{-i,t})$).

6.3 One Collision Domain Solution

We first investigate the scenario where nodes have complete information and the game is played simultaneously at each stage. Later, we will provide a new algorithm to solve the game in a general scenario. This section is a benchmark to justify the results of the general algorithm.

We define the goal of the channel assignment as maximizing the total bandwidth that all nodes acquire in the network. We further assume that there is only one collision domain.

The best response is the strategy profile that leads to the maximum achievable bandwidth considering both the number of radios (I_i) and the bandwidth required by each node (φ_i).

We define best response as the strategy that minimizes the difference between the bandwidth that a node gains and the bandwidth it requires. Therefore, given that S is the game strategy profile (Equation (6.4)), the best response is the answer of the following optimization problem:

Minimize:

$$G(S) = \sum_{i \in N} \left[\frac{\vartheta_i}{\sum\limits_{c \in C} B_{i,quota}(c)} \right] \tag{6.8}$$

subject to:

$$\sum_{k \in C} r_{i,k} \leq I_i \quad \forall i \in N$$

$$\sum_{i \in N} \xi_i r_{i,c} \leq B_{free}(c) \quad \forall c \in C$$

Note that, ξ_i shows the bandwidth demand of a node for each channel (equation (6.9)), since we assume that, a node which has radio interfaces over more than one channel will use all channels proportionally.

$$\xi_i = \frac{\vartheta_i}{\sum_{c \in C} r_{i,c}} \tag{6.9}$$

To obtain the best response, the minimization problem is solved using Lagrangian multipliers:

$$
\begin{aligned}
&L(G(S), \lambda_1, \ldots, \lambda_{N+|C|}) = \\
&\sum \left[\frac{\vartheta_i}{\sum_{c \in C} B_{i,quota}(c)} \right] - \qquad \lambda_1 \left[\sum_{k=1}^{C} r_{1,k} - I_1 \right] - \ \ldots \\
&-\lambda_N \left[\sum_{k=1}^{C} r_{N,k} - I_N \right] - \qquad \lambda_{N+1} \left[\sum_{i \in N} r_{i,1} \xi_i - B_{free}(1) \right] - \ \ldots \\
&-\lambda_{N+|C|} \left[\sum_{i \in N} r_{i,|C|} \xi_i - B_{free}(|C|) \right]
\end{aligned}
$$

In detail, the equations that are solved are:

$$
\frac{\partial L}{\partial r_{i,c}} = 0 \quad \forall i \in N; \quad \forall c \in C
$$

together with the complementary slackness conditions:

$$
\lambda_i \left[\sum_{k=1}^{C} r_{i,k} - I_i \right] = 0 \quad \forall i \in N
$$

$$
\lambda_{N+c} \left[\sum_{i \in N} r_{i,c} \xi_i - B_{free}(c) \right] = 0 \quad \forall c \in C
$$

To Solve the optimization problem we must assume that a node knows the payoff of all possible strategies, which is only possible if all nodes are in one collision domain. In the next section we provide an algorithm which solves the game without any need to deal with the optimization problem.

6.4 General Multi Collision Domain Solution

We propose a real-time learner algorithm which uses the local information at each node and is based on a multiplicative weight update scheme [18]. The main idea of the real-time learner is as follows: 1) each player assigns a weight to each possible strategy considering its payoff, 2) if the current strategy does not provide the desired payoff, the player selects a random strategy considering the weights, 3) the player updates the weights based on the local observation of the channel occupancy.

Note that the properties that make this algorithm interesting are:

- The algorithm minimizes the loss that players suffered without knowing the complete cost matrix of all strategy combinations.
- If other players are adversarial, it is proven that the algorithm converges to the *Minimax* solution [18].
- If other players are not adversarial, the solution obtained by the player is better than the *Minimax* solution. Note that the external interference which occupies part of the spectrum, can be considered as another player, which does not have a minimization strategy.

Let $E[F(S_{-i,t})]$ be the *expected payoff vector* of node i. Therefore considering the strategy profile $S_{-i,t}$, the expected payoff of selecting strategy s_i is $E[F(s_i, S_{-i,t})]$:

$$E[F(s_i, S_{-i,t})] = \delta E[F(s_i, S_{-i,t-1})] + (1-\delta)F(s_i, S_{-i,t}) \tag{6.10}$$

where δ is the memory tuning parameter that takes values in the range $[0, 1)$. Equation (6.10) works as a low pass filter with an exponential impulse response, that approximates the mean value of $F(s_i, S_{-i,t})$.

A node assigns a non-negative weight ($w(s_i)$) to each element of $E[F(s_i, S_{-i,t})]$. Initially $E[F(s_i, S_{-i,t})]$ is unknown to player i, but as the game is played repeatedly in a sequence of game rounds $(1, \ldots, T)$, it is updated at each round. In each round $t \in [1, \ldots, T]$, the player plays a mixed strategy based on the weights assigned to the elements of $E[F(s_i, S_{-i,t})]$. The probability of selecting the strategy s_i, is calculated as the related magnitude of its weight to the total weights of available strategies:

$$p_t(s_i) = \frac{w_t(s_i)}{\sum\limits_{s \in S_i} w_t(s)} \tag{6.11}$$

Initially all weights are set to 1, thus, the probability of selecting any strategy is identical following a uniform distribution. Once one strategy has been selected, the node observes the network state, that is, the external interference and the channels occupied by its neighbors. Based on that, it measures its obtained payoff and updates the weight assigned to strategy s_i using the following formula:

$$w_t(s_i) = \begin{cases} 1, & F(s_i, S_{-i,t}) \geq \varphi_i \text{ OR} \\ & F(s_i, S_{-i,t}) \geq max(\rho_{i,t}) \\ w_{t-1}(s_i)\, \beta^{(1-E[F(s_i,S_{-i,t})])}, & \text{otherwise} \end{cases} \tag{6.12}$$

Here β is the game parameter in the range of $(0, 1)$ which yields a multiplicative update [18]. The player selects a new strategy only if it does not obtain the expected payoff (6.6) and the current payoff is lower than the maximum payoff computed by the Markov model (Section 6.2.3). This implies that, a node stops oscillating between channels when the bandwidth it gets with the current strategy is adequate. Note that, reaching a stable solution is a key point since oscillating between channels reduces the network performance due to the switching delay of the current wireless antennas.

The performance of the decision made by the learner using the multiplicative weights scheme, depends on the value of β. The main theorem [18, 19] concerning this algorithm is:

Theorem 1 *For any matrix M with n rows and entries in $[0, 1]$, and for any sequence of mixed strategies $Q_1, \ldots, Q_T$ played by the environment, the sequence of mixed strategies $P_1, \ldots, P_T$ produced by the algorithm satisfies*

$$\sum_{t=1}^{T} M(P_t, Q_t) \leq \frac{ln(\frac{1}{\beta})}{1-\beta} \min_{P} \sum_{t=1}^{T} M(P, Q_t) + \frac{1}{1-\beta} ln\ n \tag{6.13}$$

The proof can be found in [18, 19]. The theorem simply implies that, the excellency of the decisions made by the learner using the multiplicative weights scheme, depends on the value of β. A high β value introduces minor changes to the weights, and the learner follows the environment more accurately but slowly. Therefore it is applicable to a scenario where the environment changes less frequently. On the contrary, a low β value imposes big changes in the weights, which introduces a higher error to the decision but, it is adequate to a scenario with frequent changes.

Note that the best solution reached by the learner is not necessarily the Nash Equilibrium, since it has been shown that multiplicative weights update learning algorithm cannot work for Nash Equilibrium in general bi-matrix games [12].

6.5 Performance Evaluation

The performance of the real-time learner channel assignment (RTLCA) algorithm (Section 6.4) is evaluated and compared with the best response solution (Section 6.3) and two other game models proposed in the literature.

We first consider a scenario with $N = 5$ nodes, where each node has $I_i = 2$ radio interfaces. The number of available channels is $|C| = 5$. All channels have the same total bandwidth and it is normalized to 1. The amount of external interference in each channel (i.e. environment strategy) follows a uniform random variable which picks values in the range between 0.2 and 0.8. The values of external interference are kept unchanged during the simulation.

The learner algorithm is examined for different values of β, δ and φ. Remember that δ is the memory tuning parameter for computing the payoff (Equation (6.10)), β is the game parameter (Equation (6.12)), and φ is the minimum payoff that a node expects to gain (Equation (6.6)). Each point in the charts shows the result of running the algorithm for 50 rounds and is averaged over 50 runs with different random seeds. The error bars shows 95% of confidence interval.

Fig. 6.2 shows the average distance between the payoff obtained by the real-time learner algorithm and the maximum achievable payoff from the best response for different values of δ. We have run the simulation for $\beta \in \{0.1, 0.9\}$ and $\varphi \in \{0.2, 0.8\}$. The figure shows that changing φ has a big impact on the results. With small φ the difference between the best possible payoff and the payoff obtained by the algorithm is less than 0.1, which implies that the algorithm can reach the expected results easily. For a network where nodes have high expectation for payoff (big value of φ), reaching a good result depends on tuning β and δ. By selecting a small value for β the best result is achieved when δ is big (and vice versa). This is because of the fact that a small β imposes big changes to the weights assigned to the strategies and nodes are prone to select different strategies in each round, i.e. in each round the solution is less dependant to the past which is get adjusted by having a big value for δ as a memory tuning parameter.

Fig. 6.3 shows the values of $G(S)$ (Equation (6.8)) for the strategy profile that the game reaches after 50 rounds. Fig. 6.3 confirms that in each scenario the minimum $G(S)$ is acquired when the difference between the payoff obtained by the system and the best payoff reaches 0.

Fig 6.4 shows the average number of radio interfaces over each channel (left y-axis) for the final strategy profile after running the game for 50 rounds.

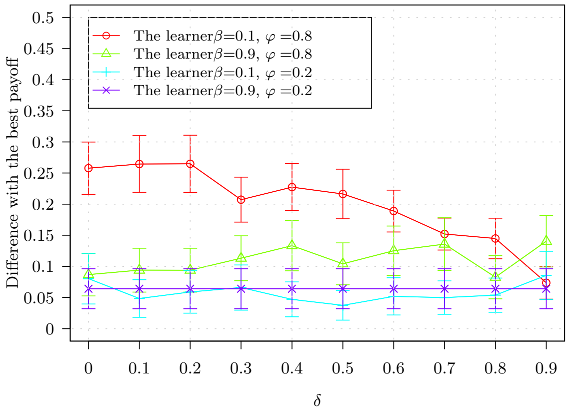


Figure 6.2: The performance of the learner related to the best response vs. different values of δ

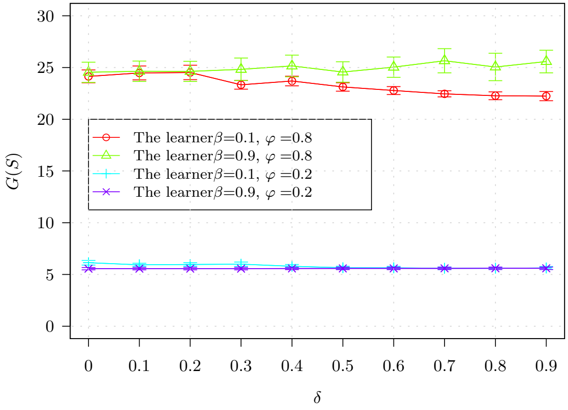


Figure 6.3: The performance of the learner vs. different values of δ

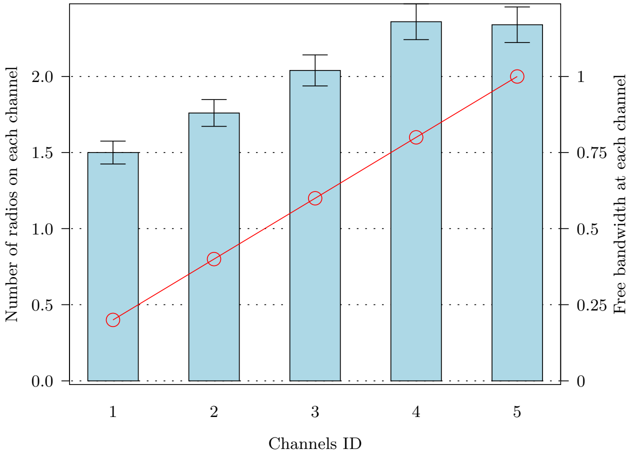


Figure 6.4: The average number of radio interfaces on each channel

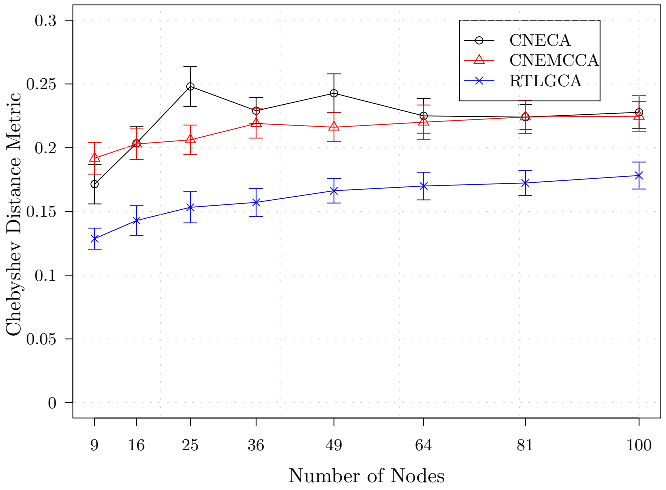


Figure 6.5: Chebyshev distance between the obtained and the demand payoff vs. different number of nodes

The results are obtained for $\varphi = 0.8$, $\beta = 0.9$ and $\delta = 0.8$. The portion of free bandwidth over each channel (right y-axis) is shown by a red line. The figure shows that for the stable strategy the number of radio interfaces on each channel is relative to the available bandwidth on the channel.

We evaluate the performance of RTLCA with two other game based channel assignments proposed in the literature: Centralized Nash Equilibrium Channel Assignment (CNECA) [35]; and Centralized Nash Equilibrium Multi Collision domain Channel Assignment (CNEMCCA) [55]. CNECA considers all nodes interfere each others while CNEMCCA is designed for multi collision domain networks. Both algorithms distribute the interfering radios on different channels fairly. However none of the proposed algorithms considers the different available bandwidth for different channels.

Fig. 6.5 is obtained for grid networks with different number of nodes ($9 \leq N \leq 100$). The number of available channels is $|C| = 12$. Nodes' requested payoff follows a uniform random variable which picks values in the range between 0.1 and 1 (Equation (6.6)). We obtain results for RTLCA with $\beta = 0.2$ and $\gamma = 0.8$. The other parameters are the same as for Fig 6.2.

Fig. 6.5 shows the Chebyshev distance metric between the obtained payoff and the payoff which is required by the players for different game based channel assignments. The results show that, RTLCA is more successful in providing the required bandwidth, for players, in unpredictable wireless environments.

6.6 Conclusion

In this chapter we have presented a Game Theory based formulation for the channel assignment problem in multi-radio wireless networks. The considered problem is formulated using a repeated game with incomplete information and it is general enough to be applicable in any wireless network. We solve the game using a multiplicative weights learning algorithm which adapts to the changes in the environment and reaches a desired solution in a limited amount of time.

CHAPTER 7

Distributed and Hybrid Channel Assignment

7.1 Introduction

In this chapter we propose SICA, a Semi-dynamic and Interference aware Channel Assignment protocol for IEEE 802.11 based WMNs. SICA estimates the amount of interference over channels, induced by any external wireless device, based on IEEE 802.11k standard [26]. We then use game theory to formulate the channel assignation problem. The model we use in this chapter is the simplified version of the game we have proposed in Chapter 6. The main contributions that sets our work apart from others are the following ones:

- A novel game theory formulation of the channel assignment problem, considering external and internal interferences.
- A decision making strategy assuming imperfect information at each mesh router that allows a fast network adaption to the changing wireless environment.
- A fully distributed channel assignment algorithm which preserves the network connectivity and supports any routing protocol.
- A new protocol which applies channel load estimation, interface switching, control message exchange and data delivery mechanisms in addition to channel assignment.

We evaluate SICA through simulations using ns-3 [41] and compare it with other channel assignment mechanisms that have been proposed in the literature [57, 43, 21]. Results demonstrate the effectiveness of SICA in exploiting channel diversity, hence reducing the interference over wireless links and improving the system performance in terms of capacity and supported nodes and networks.

The chapter is organized as follows. In Sections 7.2–7.5 we explain SICA architecture, channel assignment algorithm and simulation details. The performance evaluation is done in Section 7.6, and Section 7.7 ends the chapter with the concluding remarks.

7.2 SICA Architecture

We present a multi-radio multi-channel mechanism which mitigates the impact of the interference, and improves the performance of the wireless mesh networks by driving the benefits of non-overlapping channels. The distributed multi-channel architecture considers the channel selection mechanism, describes the switching process of the antennas and controls data buffering and transmitting. Nodes use a distributed algorithm to occupy the best channel based on the information gathered during the channel sensing periods. Channel assignment is viewed as a lower layer mechanism which does not consider the traffic load and, therefore any routing protocol can be applied to the network.

We describe SICA in the specific case where nodes are equipped with two radio interfaces, each one being able to use a set C (with cardinality $|C| > 1$) of non overlapping channels. However, SICA could be easily extended to a network where nodes are equipped with a number of radios larger than two. The radios will be referred to as the *receiving radio* and the *transmitting radio*, and denoted by R and T, respectively.

The distributed channel assignment mechanism selects and assigns the best channel to the R radio of each node. Then, nodes switch the T radio accordingly. For example, if a generic node A tunes its R radio over channel $c \in C$, each neighboring node, which aims to send traffic to A, will switch its T radio to channel c before start transmission. The T radio remains on channel c until all packets, which are addressed to node A, have been sent, or until a maximum period of time, called T_{max}.

In the following sections we explain the details of SICA. We explain the channel sensing mechanism and CA algorithm in sections 7.2.1, 7.2.2 and 7.2.4, respectively. The synchronization and switching of R and T radios are explained in sections 7.3, 7.4 and 7.5.

7.2.1 External interference estimation

To estimate the amount of external interference, mesh nodes use the Clear Channel Assessment (CCA) mechanism for spectrum sensing [31]. CCA is based on energy detection during a specific period of time. At a given time all nodes on the same channel stop transmission and start sensing the channel. The required time synchronization for the common sensing period is achieved

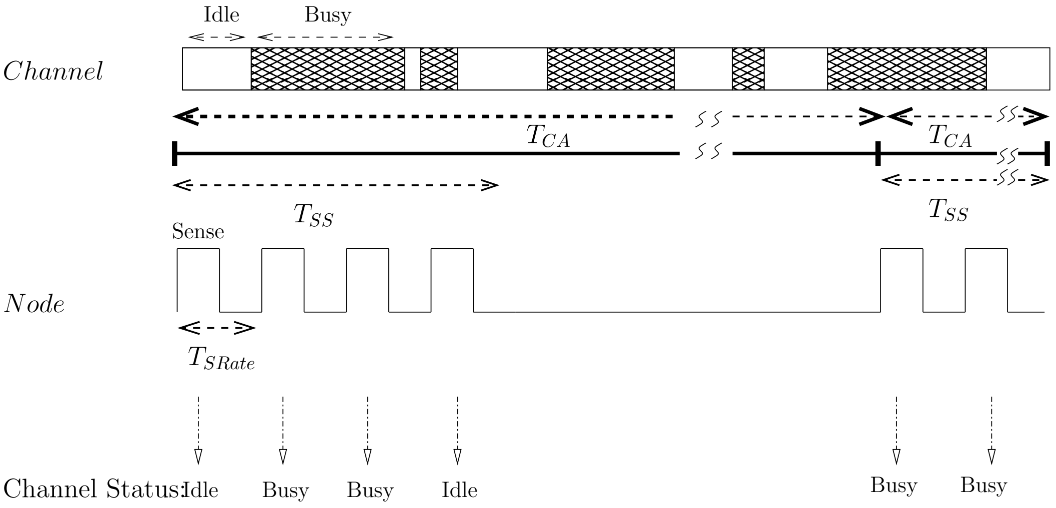


Figure 7.1: Sensing a channel by gathering samples

through sending messages (see Section 7.3). Since all nodes working on the same channel must remain silent during channel state assessment, a big sensing period will degrade the network throughput. However, a long enough sensing period is necessary to have a precise estimation [57, 26]. In SICA, each node senses only one channel at each sensing period (the channel of its receiving radio). Then, at the end of each sensing period, each node exchanges the channel state information with its neighbors.

As shown in Fig 7.1, during the sensing period (T_{SS}) every node monitors the channel by taking samples at the sense rate (T_{SRate}). T_{CA} is the period of time between consecutive runs of channel assignment algorithm.

The channel status would be monitored as either *idle* or *busy*. Define $T_{i,busy}(c)$ as the time that channel is sensed busy during the sensing period and $T_{i,idle}(c)$ as the amount of time that the channel is sensed idle.

IEEE 802.11k standard for radio resources measurement [26] proposes a simple formulation to compute the channel load as the percentage of time that the node senses the medium as busy. At the end of the sensing period, node i estimates the *normalized bandwidth* (or duty cycle) consumed by external networks over a channel c, as:

$$B_{i,neig}(c) = \frac{T_{i,busy}(c)}{T_{i,busy}(c) + T_{i,idle}(c)} \tag{7.1}$$

The mesh node then uses the channel load to make decision in channel assignment algorithm (see Section 7.2.2).

7.2.2 Game Theory Based Channel Assignment Model

We use a game theory model for the distributed channel assignment in SICA, which is adaptive to the external interference.

In our model each node is a rational player which tries to occupy the best channel for its R radio. The best channel is a channel which suffers less external interference and it is not shared by many neighboring nodes in the same network. From this point forward, we use the terms node and player interchangeably.

Let N be the number of nodes of the network, and $f_{i,c}$ the number of R radios of player i using channel c ($f_{i,c} \in \{0,1\}$). Define the strategy of player i, s_i, as its channel allocation vector, given by:

$$s_i = (f_{i,1}, f_{i,2}, \ldots, f_{i,|C|}), \qquad i = 1, \ldots, N \tag{7.2}$$

A player strategy describes whether it has a radio over a specific channel or not. Note that the total number of R radios employed by player i is given by

$$f_i = \sum_{k=1}^{|C|} f_{i,k} \tag{7.3}$$

In a dual radio network with one R radio for each node, $f_i = 1$.

We define the strategy matrix (strategy profile), S, as the strategy vector of all players at a given time:

$$S = \begin{pmatrix} s_1 \\ s_2 \\ \cdots \\ s_N \end{pmatrix} \tag{7.4}$$

By S_{-i} we refer to the strategy matrix consisting of all nodes' strategies except player i. Note that, node i may not know S_{-i} completely.

We formulate a game theory model where each player i chooses a channel c trying to minimize a *loss function*. Each mesh router uses two separate costs for selecting a channel. The first cost is according to the channel load estimated in Section 7.2.1 (Equation (7.1)). The second cost is according to internal interference induced from neighboring nodes. To estimate the internal interference over a channel, mesh routers compute how congested is the channel in the neighborhood. Let N_i be the number of nodes in the interference range of node i (two-hops neighbors based on the interference protocol model presented by [24]). We represent by $R_i(c)$ the number of nodes in the set N_i that have tuned their R radio to channel c at a given time:

$$R_i(c) = \sum_{k \in N_i} f_{k,c} \tag{7.5}$$

We define the density of interfering nodes over channel c by

$$\frac{R_i(c)}{N_i} \tag{7.6}$$

The mesh router then merges the costs by taking the weighted average of the individual cost as a *bandwidth loss function*:

$$M_{i,B}(c, S_{-i}) = \alpha \cdot B_{i,neig}(c) + (1 - \alpha)\frac{R_i(c)}{N_i} \tag{7.7}$$

where $\alpha \in [0, 1]$ is the control parameter.

However, the cost of one node's decision depends not only on the available bandwidth of the selected channel, but also on the switching delay penalty if a node's radio switches frequently. According to [57, 37] current 802.11 commodities suffer a considerable switching delay (D_s), that ranges from 80 μs to 22 ms. We consider the magnitude of the switching delay related to the *Hello* interval, T_H (explained in Section 7.4).

If the hello interval is large enough the effect of the switching delay is negligible and nodes are allowed to switch frequently to other channels. On the other hand a considerable switching delay should result in a higher channel switching cost, making nodes to switch between channels less frequently.

Let c_i the channel being used by node i for the R radio, we assume that a *switching delay loss function*, for any channel, is given by:

$$M_{i,D}(c, S_{-i}) = \begin{cases} \frac{D_s}{T_H}, & c \neq c_i \\ 0, & \text{otherwise} \end{cases} \tag{7.8}$$

Finally, we combine bandwidth and switching delay costs in the loss function given by:

$$M_i(c, S_{-i}) = \gamma \, M_{i,B}(c, S_{-i}) + (1 - \gamma) \, M_{i,D}(c, S_{-i}) \tag{7.9}$$

where $\gamma \in [0, 1]$ is a tuning parameter. Note that the loss function codomain is $[0, 1]$.

It is not feasible nor necessary for a player to compute $M_i(c, S_{-i})$ for all possible values of S_{-i}. Each player computes the *loss* value for one strategy profile at a time. In Section 7.2.3 we explain how this method solves the game effectively.

To sum up, we have defined a game with the following properties:

- Nodes are rational players and try to occupy the most vacant frequency channels.
- Nodes do not have knowledge about their neighbors criteria of making decision, beforehand.
- Each channel decision imposes a cost (in the range of 0 to 1) to a node, as a function of switching delay and available bandwidth on the selected channel.
- The game is played in several rounds, as the external parameters introduced by the environment may differ in each round, the environment is unpredictable and can remain permanently in the transient state.

7.2.3 Solving the channel assignment game

Due to the changes in the co-channel interference, the game outlined in the previous section has no deterministic loss matrix, therefore using common approaches to solve the game is impossible. Our solution is based on real-time learning approach proposed by [18, 19].

As define before, $M_i(c, S_{-i})$ is the loss matrix of node i, i.e. the rows of $M_i(c, S_{-i})$ are the strategies of node i (the channels, $c \in C$, it can choose), and the columns are all possible strategies of the other players, S_{-i}.

Each node assigns non-negative weights ($w_i(c)$) to the rows of $M_i(c, S_{-i})$. We assume that, the number of rows in $M_i(c, S_{-i})$ is the same for all nodes and equal to the number of orthogonal channels ($|C|$).

Initially $M_i(c, S_{-i})$ is unknown to player i, but this game can be played repeatedly in a sequence of *game rounds* $(1, \ldots, T)$. To avoid the channel oscillation in each round t ($t \in 1, \ldots, T$), the player plays a mixed strategy based on the weights ($w_{i,t}(c)$) assigned to the rows of $M_i(c, S_{-i})$. The probability of selecting the channel c, is calculated as:

$$P_{i,t}(c) = \frac{w_{i,t}(c)}{\sum_{c \in C} w_{i,t}(c)} \tag{7.10}$$

Initially, all weights are set to 1, which means that the probability of selecting any channel is identical. After selecting a channel, the node gathers information from its neighbors and updates the loss that is suffered (equation (7.9)). Then, the weights are updated as:

$$w_{i,t}(c) = w_{i,t-1}(c)\, \beta^{M_i(c,S_{-i})} \tag{7.11}$$

where $\beta \in [0,1]$ is the game parameter [19].

As explained in Chapter. 6 with β near 1, the algorithm introduces minor changes to the weights and the learner, follows the environment more accurate but slowly. On the contrary, a β close to zero, imposes big changes in the weights, and introduces a higher error to the decision. Therefore, it is adequate to a scenario with frequent changes.

In our simulations we found that $\beta = 0.2$ leads to better results (see Section 7.6). We use the same β for all players.

7.2.4 Channel Assignment Mechanism

Alg. 7.1 summarizes the channel assignment mechanism previously described. Recall that the main idea of SICA is to use all the available information at each node, which is gathered from its neighbors and sensing the channels, and selects the best channel by playing a game with mixed strategies. As explained in Section 7.2.3, the game is played in rounds that we refer to as *channel assignment periods*, and represent its duration by T_{CA}. Each node i runs Alg. 7.1 at every T_{CA}.

```
Input:
  $N_i$: set of one and two-hops neighbors of node $i$.
if this is the first assignment then
    Set $w_{i,t}(c) = 1 \; \forall c \in C$
    Assign a random channel $c_i$ to the $R$ radio
else
    Compute $P_{i,t}(c) \; \forall c \in C$ (Eq. (7.10))
    Compute the Cumulative Distribution Function (CDF) out of $P_{i,t}(c)$
    $\forall c \in C$.
    Select channel $c_i$ using Inverse Transform Sampling.
    Assign channel $c_i$ to the $R$ radio.
end if
Switch the $R$ radio to channel $c_i$ (Alg. 7.2)
Use $CCA$ to estimate $B_{i,neig}(c)$ (Eq. (7.1))
Inform other neighbors about $B_{i,neig}(c)$
for $c \in$ {channels used by $R$ radio of $N_i$ nodes} do
    Calculate $M_i(c, S_{-i})$ (Eq. (7.9))
    Update $w_{i,t}(c)$ (Eq. (7.11))
end for
```

Algorithm 7.1: SICA(N_i)

Parameter Name	Description	Possible Value	Default
T_H	Hello interval	10 ms - 100 ms	20 ms
T_{CA}	Channel assignment interval	$T_H \ll T_{CA}$	10 s
T_{SS}	Channel sense interval	$T_H < T_{SS} < T_{CA}$	5 s
T_{SRate}	Channel sense rate	$T_{SRate} \ll T_{SS}$	1 ms
α	Bandwidth loss function tuning parameter	$0 \leq \alpha \leq 1$	1
β	Channel weight parameter	$0 < \beta < 1$	0.2
γ	Loss function tuning parameter	$0 \leq \gamma \leq 1$	0.8
$\|C\|$	Number of available orthogonal channels	$1 < \|C\|$	8
D_s	Switching delay of the radio	80 μs - 20 ms	300 μs

Table 7.1: Channel Assignment Parameters

Four main reasons make SICA to be an efficient channel assignment algorithm:

- Nodes are not required to have the perfect information about other players' strategies and loss functions.
- Nodes are supposed to be selfish players trying to occupy the best channels.
- It is not necessary for a node to estimate the external interference over all channels. Each node only estimates the interference over the channel of its R radio.
- The proposed channel assignment eliminates the *Channel Oscillation* problem. This problem happens when some nodes find a channel empty and try to occupy it simultaneously. In such situation, the nodes will switch back as they will find it busy by the others that have switched to that channel too. Playing a mixed strategy, as previously described, avoids channel oscillation since each node selects the destination channel randomly with a predefined probability.

Table 7.1 summarizes the parameters of the channel assignment mechanism and the default value for each one of them.

7.3 Mesh nodes synchronization mechanism

Unlike most CAs proposed in the literature, in SICA there is not a common control channel shared by all nodes. In SICA, the synchronization is achieved through exchanging messages over the data channels. Since each node can assign a different channel to its receiving (R) radio, nodes must be aware of the channels used by their neighbors' R radios. In SICA, a node broadcasts *Hello* messages to report the channel of its R radio to its neighbors.

It is not necessary to send *Hello* messages over all available channels, except when a new node joins the network or when a node stops receiving *Hello* messages from any neighbor. Once a node knows the channels used by the R radio of its one-hop neighbors, the node switches the T radio to those channels and sends *Hello* messages every T_H seconds.

We refer to the period between *Hello* messages as T_H. This period must be long enough to minimize the overhead caused by the switching time of the T radio, and small enough to keep the information updated for all nodes.

A *Hello* packet contains the following information, besides a sequence number:

1. Channel and MAC address of the R radio,
2. Channels of the R radios of the node's one-hop neighbors,
3. Spectrum sensing information,
4. Channel switching attempt information.

Additionally, each node has to broadcast the the channel used by the R radios of its one-hop neighbors in the *Hello* messages. Therefore the neighbors are able to keep the information of two-hops nodes.

Spectrum sensing information contains: *i*) the estimated consumed bandwidth by external interferences over the receiving channel; and *ii*) the time units remaining before the start of the next sensing period. Neighboring nodes need information about the upcoming sensing period to avoid initiating any transmission over the channel that is going to be sensed.

Channel switching attempt information is the expected time units before the moment that a node will switch its R radio to a new channel (See Section 7.5). Note that there is no need to have a tight synchronizations between nodes since nodes are aware of changes through *Hello* messages.

7.4 Data delivery mechanism

After gathering information from the neighbors, a node may start transmitting data. One node may have packets to deliver to different neighbors which have their R radio on different channels. In our model each channel is associated with a sequential first-in first-out (FIFO) queue. Packets are added to the corresponding queue according to the receiving channel of the neighbors. When the T radio switches to each channel, it sends all or some of the packets in the associated queue. We use a different queue for *Hello* messages, which has higher priority than data packets' queue.

A mesh node uses a Round Robin approach to visit all the channels for which it has data to sent, defined as the subset $C_i \subset C$. To avoid starvation, after switching, the T radio will stay in one channel for at most a specific period of time (T_{max}). We assume that the switching delay of the T radio is constant and equal to D_s. Therefore if a node has to send data over $C_i \subset C$ channels, it will take $|C_i|\, T_{max} + (|C_i| - 1)\, D_s$ to visit them. In order to have the same opportunities to transmit over C_i channels, the node computes T_{max} as:

$$T_{max} = \frac{T_H - (|C_i| - 1)\, D_s}{|C_i|} - D_t \tag{7.12}$$

where T_H is the period between *Hello* messages and D_t is the delay before the T radio starts the transmission. Note that C_i is given by the number of different channels used by node's neighbors.

Fig. 7.2 shows an example describing the whole process. In the example the node has $|C_i| = 2$ channels over which it has data to send. Every T_H a *Hello* packet is pushed at the front of the C_i queues. We show *Hello* and Data packets with black and shaded boxes, respectively. T_{max} shows the maximum duration of time that a node may remain in each channel (see Equation (7.12)). In detail, Fig. 7.2-a shows the node status before T_H starts. The mechanism starts from channel one (Fig. 7.2-b) and, after T_{max} it switches to channel two, where it is able to transmit only three packets before to switch again to channel one.

After switching the T radio to a channel, to avoid collision with any ongoing transmission on the channel, a node must wait at least for D_t units of time before it starts transmitting, (see Section 7.5.2). After D_t, the node sends packets during at most T_{max}, then it switches the T radio to channel two and the transmission process is repeated for this channel (Fig. 7.2-b). While there are packets in the queues, the node will round robin among them until the end of T_H (Fig. 7.2-c).

Note that the T radio can not switch to a channel and initiate any transmission when the channel is being sensed by any of the neighboring nodes. Moreover nodes must consider the switching attempt of the R radio which is announced to the neighbors before it starts. The switching mechanisms of T and R radios are explained with more details in the following section.

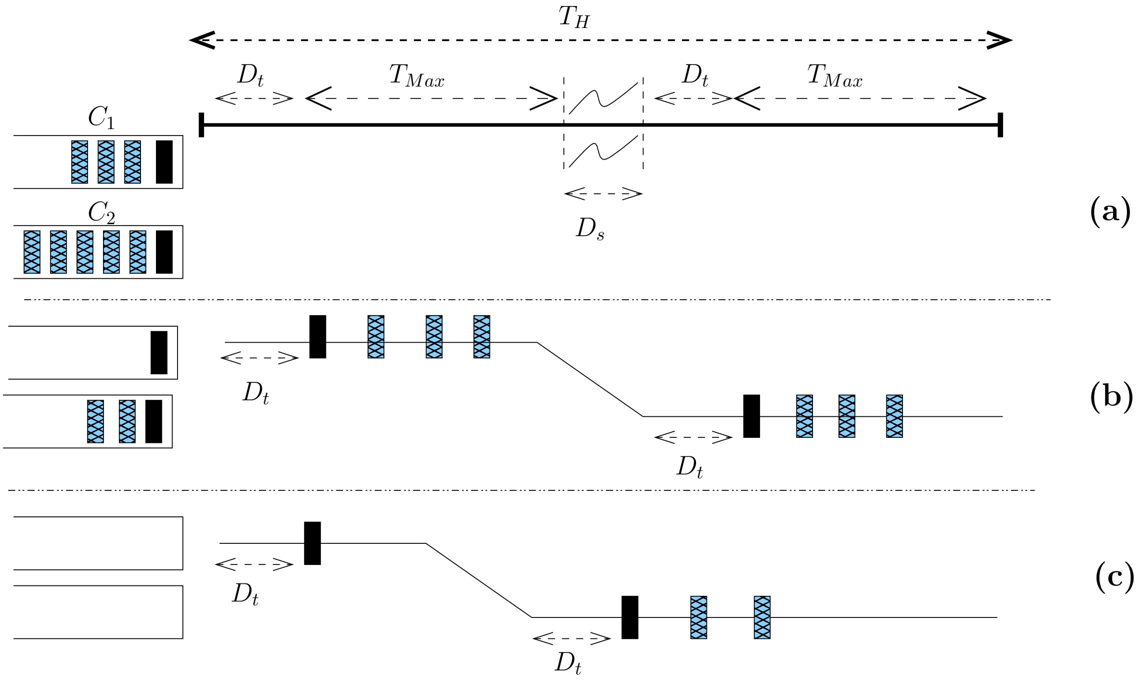


Figure 7.2: Channel queues and data delivery mechanism

7.5 Channel Switching Mechanism

The switching mechanism consists of two different protocols for switching T and R radios, respectively.

7.5.1 Switching the R radio

When a node decides to switch the R radio to any channel, it must announce the switching attempt through *Hello* messages. The switching attempt information in a *Hello* message, consists the following fields:

1. The destination channel to which the R radio will switch,
2. The time to switch (T_S).

The switching time (T_S) contains the remaining time in the current channel until the R radio switches to the new channel. This time must be longer than the *Hello* interval to make sure that all neighbors are informed. Therefore the neighbors will consider the new channel for upcoming transmissions. A node follows the steps in Alg. 7.2 to tune the R radio to the defined channel.

If a node misses any information about a switching attempt of a neighbor, the node would fail to send packets to it. The algorithm tries to prevent this

```
Input:
  c: The target channel for switching
if there is any transmission on R radio then
    Wait until the end of transmission
else
    $T_S \leftarrow Random(2T_H : 3T_H)$
    Set the timer delay to $T_S$
    while the timer is running do
        Update the switching attempt information in Hello messages
    end while
    Switch the R radio to channel c
end if
```

Algorithm 7.2: Switch The R Radio (c)

by selecting a sufficiently large T_S (see Table7.1). Moreover the node always gets information about a lost neighbor from other common neighbors, thus, updating its information.

7.5.2 Switching The *T* Radio

The T radio switches channels more often than R. Alg. 7.3 describes the switching mechanism for the T radio. Here each node checks all queues sequentially (Round Robin) and if there is any data waiting for transmission, it switches the T radio to the corresponding channel and starts sending data after D_t units of time.

When a node switches to a new channel it may fail to hear an on-going transmission between any other nodes on the same channel, so it could be a hidden terminal for the ongoing transmission and must avoid transmitting immediately to prevent the collision. Consequently after switching, a node may wait for D_t before starting any transmission.

The node remains on the target channel until the end of the transmission or at most for T_{max}. Then it proceeds to check the other queues.

7.6 Performance Evaluation

In this section, we study the performance of the proposed channel assignment algorithm using ns-3 simulator [41] for 802.11-based multi-radio mesh networks. We use a network where the mesh routers initialize their routing tables using Shortest Path First (SPF), minimizing the number of hops. We assume a two ray ground propagation model with a radio range of 250 m. Wireless nodes can tune their radio to any channel among 8 non-overlapping channels (according to IEEE 802.11a standard). RTS/CTS mechanism is

```
Input:
  $Q_c$: The packet queue associated with the channel $c$
  $C_i$: Channels of $R$ radios of one-hop neighbors
for $c \in C_i$ do
    Set the timer delay to $T_{max}$
    if $|Q_c| > 0$ then
        Switch the $T$ radio to channel $c$
        Wait for $D_t$ period of time
        while the timer is running AND $|Q_c| > 0$ do
            $P \leftarrow$ Pop one packet from $Q_c$
            Send $P$
        end while
    end if
end for
```

Algorithm 7.3: Switch The T Radio (Q_{C_i}, C_i)

disabled.

We compare SICA with three relevant channel assignment mechanisms proposed in the literature: the Centralized Breath First Channel Assignment (BFSCA) [43]; the semi-dynamic interference aware channel assignment (Urban-X) [57]; and the Nash Equilibrium Channel Assignment (NEMCA) [29]. A baseline single radio network working over one channel is also introduced for completeness.

7.6.1 Description of BFSCA, Urban-X and NEMCA

Breath First Search Channel Assignment (BFSCA) [43] is a centralized mechanism that considers one node in the WMN as a coordinator, which is responsible for assigning channels to all nodes' radios in the network. It also considers that one radio at each node is tuned to a common channel for control messages. Each node estimates the external interference through monitoring the wireless media, sending the results of such action to the central node through the common channel. The coordinator then assigns channels to links and informs nodes. Nodes redirect the traffic to the common channel before switching to a new channel and, therefore, they need to be tightly synchronized as, otherwise the channel switching mechanism would interrupt the traffic transmission. Channel assignment in BFSCA uses a graph theory based interference model to find the interfering links [24]. It sorts all links based on their distance to the central node and then tries to assign different channels to the interfering links but if such a channel is not found it assigns a random channel to the link.

Urban-X [57] uses three radios for each node: one R and one T radio, as in SICA, and a third radio which is tuned to a common channel for all nodes.

The common channel stays unchanged through the life time of the network. Channel assignment in Urban-X is distributed and takes into account the amount of flows a node has to send, and the estimated external interference over the channels. Nodes need to have information about the number of flows their neighbors have. Then Urban-X assigns a priority to each node based on the number of active flows it has. Nodes having higher priority have more chances to occupy the best channels. Nodes broadcast control messages over the common channel up to two-hops neighbors. After switching to a channel, the T radio remains there for a predefined period of time (40 ms).

Nash Equilibrium Channel Assignment (NEMCA) [29] is a game theory based channel assignment which only considers the internal interference. NEMCA is distributed and played by rational nodes. Nodes consider the internal interference over channels without keeping any memory. Each node selects a list of channels which are better than the current channel it has, from where it selects the next channel to use using a uniform random probability.

7.6.2 Simulation Parameters and Performance Metrics

We have evaluated the performance of the protocols for different number of nodes for two different node placement topologies: grid and random, and with different number of traffic flows. The traffic is generated by 100 kbps CBR flows with packet size of $L = 8000$ bits sent over vertical and horizontal directions in the grid topology and between random nodes in the random topology. Data queues are simulated in a way that they drop the old packets automatically to avoid saturation. We assume that the maximum time duration that a packet can remain in a queue is 1 s for data and 20 ms (T_H) for *Hello* messages.

The channels receiving interference from external networks, are chosen randomly. In the simulations, we consider that, at each time 50% of channels are busy due to the external interference. A channel with external interference is modeled as an on-off process, such that the channel is sensed busy and idle during the on and off states, respectively. Note that as the channel is detected busy due the external interference, nodes are not allowed to transmit during that state. The duration of the *busy* state has been fixed to a constant value, while the duration of the *idle* state is chosen exponentially distributed. The duration of the busy and idle periods have been varied to produce different interference loads.

The SICA parameters have been set using the values of Table 7.1. The specific parameters of other protocols are set according to the values given in [43, 57, 29].

We consider three network performance measures:

- *Data delivery ratio:* ratio of the total amount of data which is correctly received by the destinations, to the total amount of data packets transmitted by the sources.
- *Average end to end delay:* mean delay of the packets to reach the destination.
- *Control overhead:* ratio of the total number of control messages sent between nodes, to the total number of correctly received packets.

7.6.3 Results

Grid Topology

Figures 7.3-7.5 show the network performance for different number of nodes and two CBR traffic flows of 100 kbps. Every 50 s, external interference is introduced over 4 channels chosen randomly.

In these simulations the duration of the busy state of the external interference is fixed to 10 ms, while the mean duration of the idle state is 8 ms. The results have been obtained averaging over 10 runs of 1000 s simulation time with different seeds. The error bars in the figures show 95% confidence intervals.

Fig. 7.3 shows that the delivery ratio is 20% higher in SICA and BFSCA than in Urban-X or NEMCA. This is a significant improvement, since Urban-X and BFSCA use 3 radios and SICA uses only 2. This result shows that the game theory approach used in the channel assignment of SICA outperforms the optimized centralized algorithm used in BFSCA for large networks. The bigger error bars in NEMCA and the single-channel network shows that they suffer more unexpected external interference which is avoided by adaptive schemes. Moreover, NEMCA, as a non adaptive channel assignment, performs worse than a single channel network because of the radio switching penalty and channel assignment overheads for making inefficient decisions without considering the external interference.

In Fig. 7.4 we can see that the average end to end delay is lower in SICA and BFSCA than the other protocols. SICA leads to a lower delay thanks to the fast switching mechanism of the T radio over all channels (see Section 7.4) and because it avoids the channels which suffer external interference. The average delay in Urban-X is much higher than others because it keeps the T radio in each channel for a predefined period of time, regardless of having data to send, therefore forcing a considerable delay for data waiting in the queues of the other channels.

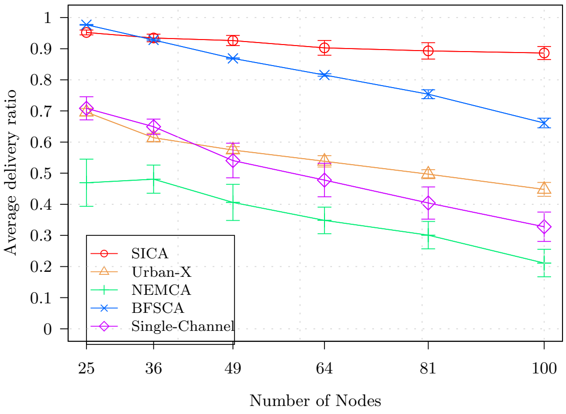


Figure 7.3: Data delivery ratio vs. number of nodes (Grid topology)

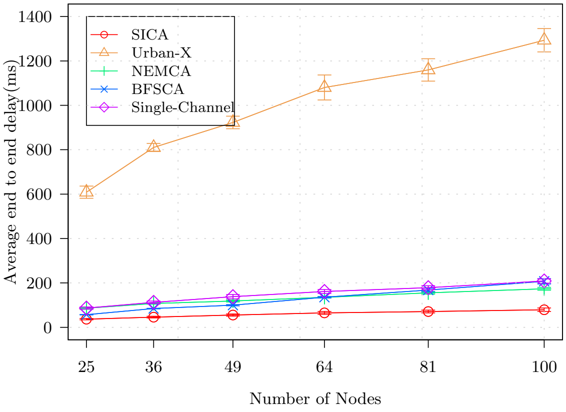


Figure 7.4: Average end to end delay vs. number of nodes (Grid topology)

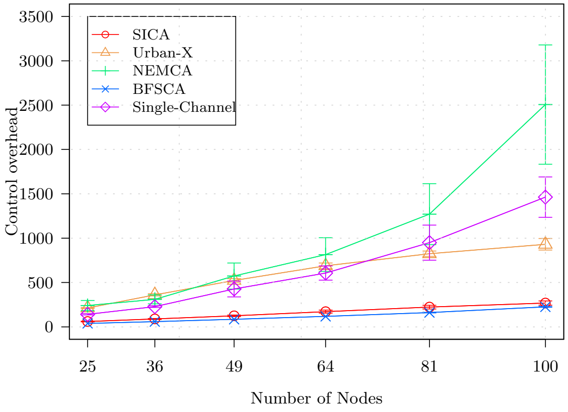


Figure 7.5: Control overhead vs. number of nodes (Grid topology)

Fig. 7.5 shows that SICA and BFSCA have similar control overhead. BFSCA uses a common channel for control packets which should be forwarded until the coordinator node, in SICA on the other hand a node broadcasts control messages only over all channels which are used by its neighbors. The control overhead in NEMCA is much higher than others since it broadcasts many control messages for channel assignment but has a low data delivery ratio. The control messages of the single channel network consist of *Hello* messages that nodes send every T_H seconds to their neighbors to announce their presence.

Random Topology

Fig. 7.6- 7.8, show the packet delivery ratio, average end to end delay and control overhead for different number of nodes which are randomly distributed over a 1000×1000 m^2 area. Each point in the figures shows the average over 50 runs. The error bars in the figures show 95% confidence intervals. Fig. 7.6 shows that the delivery ratio in SICA is higher than other protocols and it does not drop a lot as the number of nodes increases. For BFSCA and Urban-x on the other hand, the delivery ratio drops fast as the number of nodes increases. NEMCA performs the same as a single channel network following the reasons explained for Fig. 7.3.

Fig. 7.7 shows that the average end to end delay of all protocols is much lower compared to Urban-X due to the same reason explained for Fig. 7.4.

Fig. 7.8 confirms the reason explained for Fig. 7.5, showing that control overhead for NEMCA is higher than other protocols, while SICA and BFSCA lead to lower control overhead.

Increasing the number of traffic sources

Fig. 7.9-7.11 are obtained using a 8×8 grid network while other parameters are the same as the parameters considered for the grid scenario.

Fig. 7.9 shows that the delivery ratio of SICA is higher than others in presence of high traffic load. The delivery ratio of BFSCA drops fast increasing the number of traffic flows, since any channel switching interrupts the data transmission and nodes are forced to deliver data packets through common channel which offers a high load over the common channel (Section 7.6.1).

Urban-X performs better in presence of high load traffic compared to BFSCA since it considers the traffic load for making decision, but it results in a considerable high end to end delay (Fig. 7.10).

Fig. 7.11 shows that the control overhead of SICA and BFSCA is almost better than others.

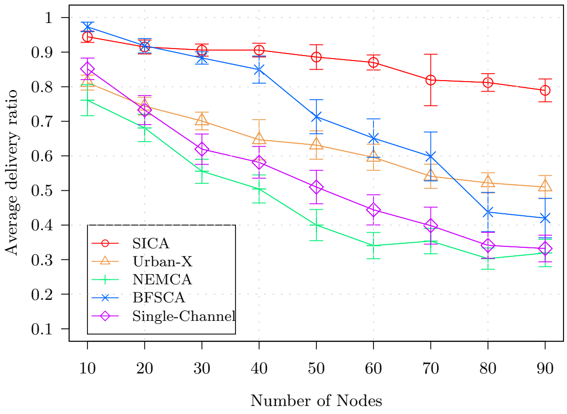


Figure 7.6: Data delivery ratio vs. number of nodes (Random topology)

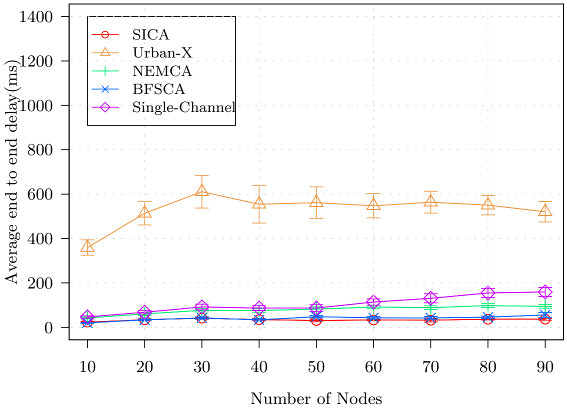


Figure 7.7: Average end to end delay vs. number of nodes (Random topology)

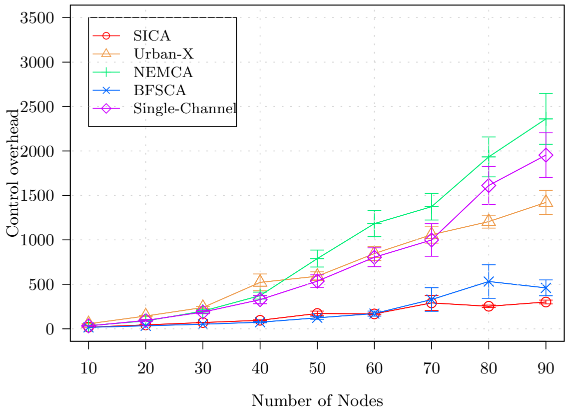


Figure 7.8: Control overhead vs. number of nodes (Random topology)

Changing the interference load

Fig. 7.12 is obtained using a 8×8 grid network while other parameters are the same as the parameters considered for the grid scenario.

Fig. 7.12 compares the delivery probability obtained with all protocols, varying the load of the interference. The x-axis of these figures shows the amount of the external interference, which is varied by changing the duration of the busy state of the interference process between 5 ms and 20 ms, and maintaining the mean duration of the idle state equal to 8 ms. We introduced interference over 4 channels. Fig. 7.12 shows that, even with a high interference load, the delivery ratio in SICA and BFSCA does not change, when the interference is increased. On the other hand, delivery ratio in Urban-X and NEMCA drops fast. The result confirms that, SICA and BFSCA are much more robust and less sensitive to the external interference than Urban-X.

Data delivery ratio: temporal evolution

In order to have a more detailed view of the protocol's behavior, Fig. 7.13 shows the time evolution of the delivery ratio obtained with different channel assignment protocols using a 8×8 grid network. Other parameters are the same as the parameters considered for the grid scenario. The values shown in the figure have been obtained repeating the simulation for 20 different random seeds and averaging the delivery ratio over 5 s periods.

Figure shows that in SICA the delivery ratio is kept more stable than in others. BFSCA is also able to offer a high packet delivery ratio with a few variations. Urban-x shows high variations in delivery probability and NEMCA is incapable to avoid the interference over channels.

SICA performance as function of α, γ and β

Finally, we investigate the sensitivity of SICA to the α, γ and β tuning parameters that characterize the response of the game theory model (see Section 7.2.2). Fig. 7.14-7.16 are obtained using a 8×8 grid network while other parameters are the same as the parameters considered for the grid scenario.

Recall that α is the weight which controls the related magnitude of external and internal interference and γ is the weight between the *bandwidth* and *switching delay* loss functions, while β is used to update the mixed probabilities of the game. Fig. 7.14- 7.16 show the delivery ratio obtained with SICA varying α, γ and β, respectively. The figures show the values obtained for a grid network with different number of nodes (depicted with different line types), and using other parameters the same as the parameters considered for the grid scenario.

Fig. 7.14 shows that SICA performs better for the high values of α. This comes from the fact that in our simulation scenario, nodes are prevented from communicating during the time that a channel is occupied by external interference, which has a higher impact than the internal interference on the system performance.

Fig. 7.15 shows that the performance of SICA is not sensitive to γ but values of γ in the range of $[0.4, 1]$ give better results, which implies that it is more important to give a higher weight to the bandwidth loss functions rather than switching delay loss function. This comes from the fact that we have chosen a *Hello* interval ($T_H = 20$ ms) significantly larger than the switching delay ($D_s = 300$ μs).

Regarding β, Fig. 7.16 shows that the best results are obtained with $\beta < 0.4$. Recall that choosing a lower value for β makes the algorithm to adapt faster to the changes of external interference.

7.7 Conclusions

In this chapter we have investigated the channel assignment problem in multi-radio wireless mesh networks. We have proposed a new semi-dynamic channel assignment protocol called SICA. We presented a novel formulation for channel assignment problem using game theory and have solved the game using a real time learning mechanism. SICA is a distributed channel assignment and assumes that nodes do not have perfect knowledge about other's nodes strategies. We have done a performance evaluation comparing SICA with other channel assignments mechanisms proposed in the literature. Simulation results show the efficiency of SICA in assigning proper channels to radios by avoiding external interference.

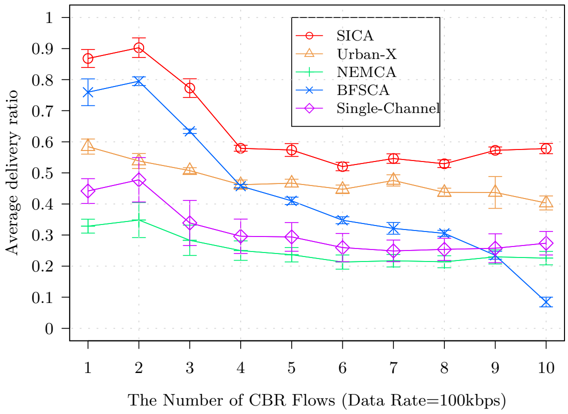


Figure 7.9: Data delivery ratio vs. number of CBR traffics

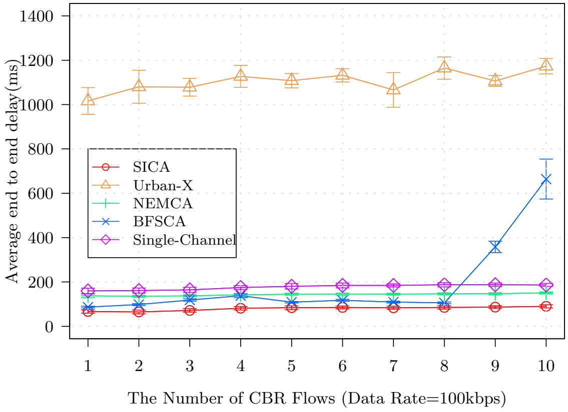


Figure 7.10: Average end to end delay vs. number of CBR traffics

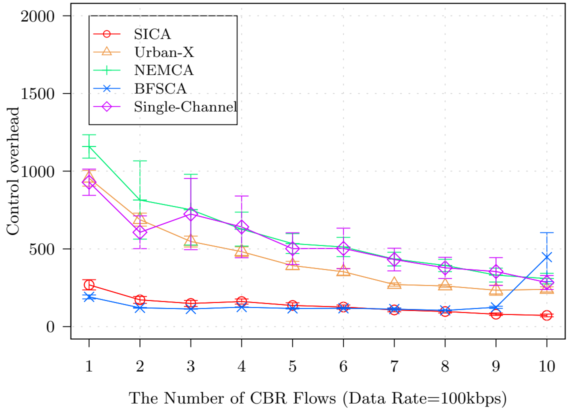


Figure 7.11: Control overhead vs. number of CBR traffics

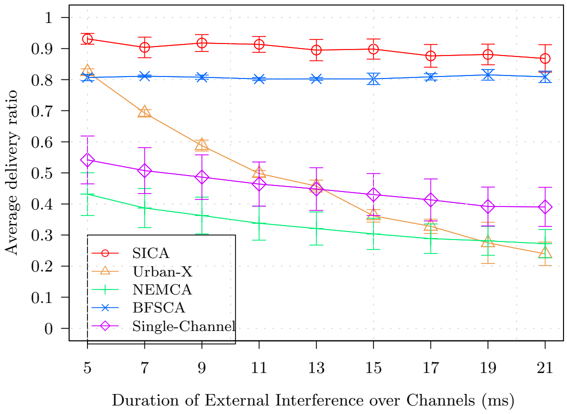


Figure 7.12: Data delivery ratio vs. Busy duration of external interference

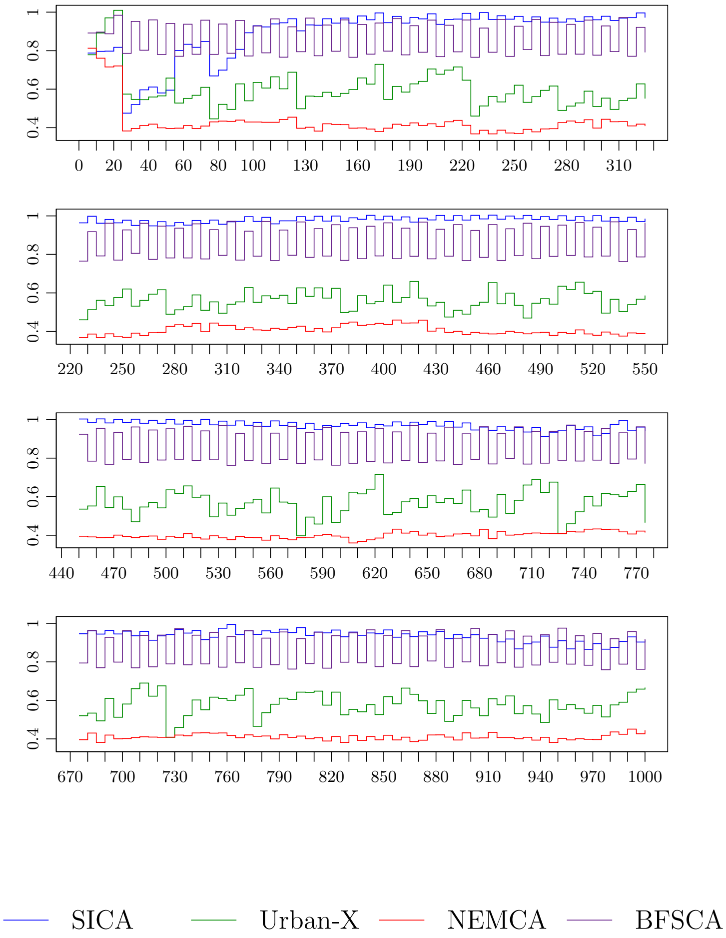


Figure 7.13: Data delivery ratio over time

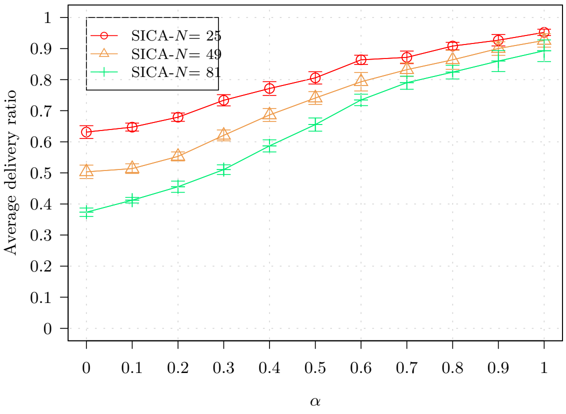


Figure 7.14: Data delivery ratio vs. α

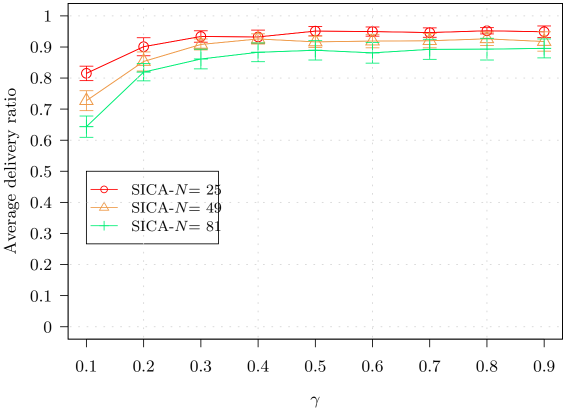


Figure 7.15: Data delivery ratio vs. γ

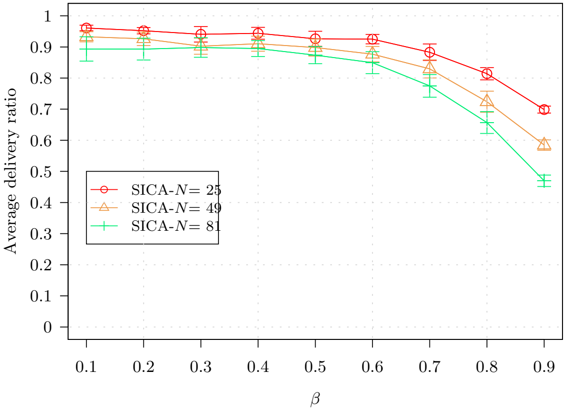


Figure 7.16: Data delivery ratio vs. β

CHAPTER 8

Validation and Verification of Multi-Radio Multi-Channel Mesh Networks Simulation

8.1 Introduction

Implementing a multi-radio multi-channel mesh network is very challenging due to the following reasons:

- The transmitting and receiving antennas installed on a single mesh node should be separated enough to reduce the noise of the transmitter on the own receiver [48, 14].
- The current 802.11 commodity devices suffer interference caused from non-overlapping frequencies over each other due to lack of prefect filtering at the antenna [48].
- The MAC layer of IEEE 802.11 standard should be modified to support channel assignment mechanisms [11].
- The routing protocol must be modified to take the advantage of the underlying channel assignment [14, 15, 43, 59].

Channel assignment mechanisms are usually evaluated via simulations due to the multiple considerations and complex scenarios required to evaluate them [34, 54, 6, 52, 4, 17, 40, 57, 30]. However most of current network simulators do not support multi-radio routers nor dynamic re-configuration of the radio interface, requiring modifications in to their core-level to allow the evaluation of CA mechanisms [1].

Ns-3 [41] is a young simulator which allows the wireless mesh nodes to be equipped with more than one radio interface, and makes it possible to change the frequency of the radio during the simulation runtime. Although those features are necessary in order to simulate a multi-radio multi-channel network, they are not enough for simulating hybrid or dynamic CA mechanisms. Multi radio mesh networks simulated in ns-3 based on the recently published standard, IEEE 802.11s [27], are assumed to share common channels. The ns-3 modules for IEEE 802.11s can be used to simulate static or semi-dynamic CAs but they are insufficient for dynamic or hybrid CA. As in IEEE 802.11s, the peer links are formed over common channels, the Peer Management Protocol must be changed to be able to send beacons for neighboring mesh points which do not share any common channel. This modification is incompatible with IEEE 802.11s standard which assumes that mesh points operate in a single channel [27]. According to IEEE 802.11s standard, multi radio stations form different meshes on different channels, which are unified in a single LAN using the layer two bridging. We consider the problem of simulating hybrid CAs without restrict our work to IEEE 802.11s mesh networks.

In this chapter we show in detail how to simulate a hybrid channel assignment protocol using ns-3 simulator [41] without any need to modify the simulator's source code (Section 8.2). We use the simulator version 3.9 released on August 2011. As a specific example, we present the simulation of the channel assignment proposed in [40] (Section 8.3), followed by the simulation verification and the validation of the proposed CA model using a Markov chain (Section 8.4- 8.5). The conclusion remarks are presented in Section 8.6.

8.2 Simulation Components For Multi Radio Mesh Networks

In this section, we introduce the required extensions to ns-3 simulator [41] for simulating hybrid CA mechanisms. These extensions include the basic functions on top of which the CA mechanism can be implemented. In addition, a CBR traffic generator and a specific routing strategy are presented as required components, to evaluate the performance of the CA mechanism.

8.2.1 Multi Radio Wireless Nodes

Multi radio wireless mesh nodes are the required basic building blocks of the simulation scenarios. Fig. 8.1 shows a wireless mesh node equipped with several radio interfaces.

The figure depicts a general cross-layer channel assignment protocol that interacts with the new components which must be added to the simulation: traffic generators; routing protocol; and the channel sensing mechanism.

In Program 1 it can be seen how to make a multi radio node (*MRNode*)

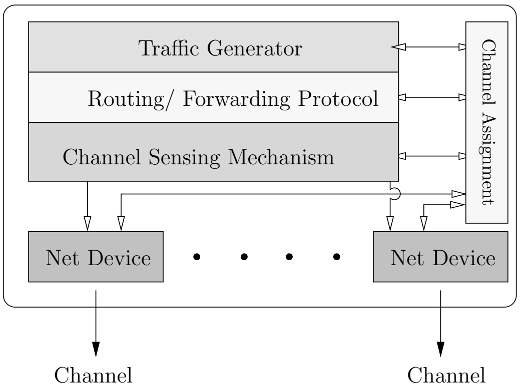


Figure 8.1: Node object

in ns-3. Where I is the number of radio interfaces which must be installed on a node. Note that in ns-3 a unique ID starting from 0 is automatically assigned to each radio interface.

We configure the MAC and physical layers according to Table 8.1 which makes IEEE 802.11a based radio interfaces. The radio propagation model is set to the fixed range propagation model with a maximum range equal to 250 m. We also define a single transmission rate equal to $6Mbps$ for data and control packets transmission. Note that this configuration is not fixed and it could be changed to any of the propagation and channel loss models that ns-3 supports [41].

Program 1 Creating a multi radio wireless node

```
1  Ptr<Node> MRNode = CreateObject<Node> (); ///< Create a node
2  Ptr<WifiNetDevice> radio;
3  for (int i=1; i< I ; i++ ){
4    radio=CreateObject<WifiNetDevice> ();///< Create one radio interface
5    radio->SetMac (macConfiguration);///< Configure the MAC layer of the
         interface
6    radio->SetPhy(phyConfiguration);///< Configure the physical
        attributes of the interface
7    MRNode->AddDevice(radio);}///< Add the device to the node
8   Ptr<NewClass> NewObject = CreateObject<NewClass>
9   ();///< Create another object , i.e. CA
10   MRNode ->AggregateObject(NewObject);///< Attach the object to the
       node
```

Lines 8- 10 of Program 1 shows how any other object (CA mechanism, routing protocol, etc.) is created and aggregated to the wireless mesh node. Fig. 8.2 shows the class diagram of a node in ns-3.

Sections 8.2.2- 8.2.3 provide a brief explanation of the components which

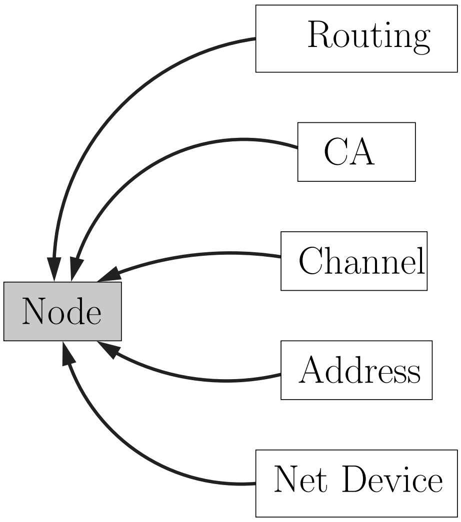


Figure 8.2: Class Diagram of the Node

must be added to the multi radio node to make it possible to simulate the channel assignment mechanism.

8.2.2 Network Devices

Configuring The Radio Interface

Ns-3 indicates each channel with a unique *ID*, which starts from 0. By default the channel assigned to all radios in a node is set to 0. To reconfigure the channel assigned to a radio interface, a node has to change the ID of the default channel in the physical layer of the radio interface. All channels is ns-3 are supposed to be non-overlapping.

Program 2 shows the process of assigning channel c to the first radio interface (*netDevice*) installed on *MRNode*.

Note that, if the interface is busy due to sending or receiving, it is not possible to switch to another frequency. Program 2 checks the status of the device (Line 5) before setting the new channel.

It is also possible to acquire the remaining time until the device gets idle (Line 9) and set the channel afterward (Line 11).

Data Service Components

To transmit packets in a multi radio multi channel architecture, where wireless nodes use a dynamic or hybrid CA, some extra tasks must be done

Parameter Name	**Description**	**Value**
Standard	MAC and Physical layer standard	ns3::WIFI_PHY_STANDARD_80211a
PropagationDelay	The propagation delay between the specified source and destination	ns3::ConstantSpeedPropagationDelayModel
PropagationLoss	Modalize the propagation loss through a transmission medium	ns3::RangePropagationLossModel
MaxRange	Radio range	250 m
RemoteStationManager	Data and control packets transmission rate	ns3::ConstantRateWifiManager

Table 8.1: Physical and MAC layer configuration

to acquire the channel on which the packets must be sent, in addition to computing the time for switching a radio interface to the desired channel. Fig. 8.3 shows the main process that a node follows to deliver packets on the presented multi-radio multi-channel architecture. It comprises three steps: Destination Channel Query, Time Stamp Assignment and Packet Queuing

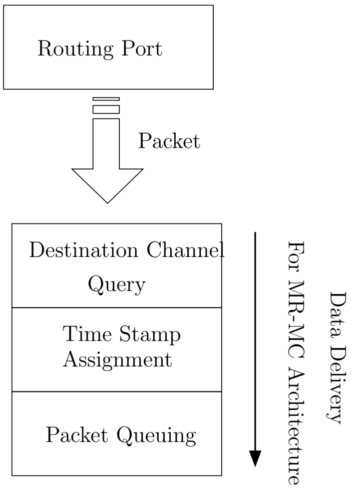


Figure 8.3: Data Services For Channel Assignment Mechanism

Unlike single channel networks, a node in a multi channel architecture needs to consider the possibility of having next hop neighbors over different channels. Therefore, to forward a packet, a wireless mesh node needs to know the channel over which it can send the packet, in addition to the the address of the next hop mesh node.

Program 2 Assign channel c to a radio interface (netDevice)

```
Ptr <NetDevice> netDevice= MRNode->GetDevice(0); ///< Get the address of the radio interface
Ptr <WifiNetDevice> wifiNetDevice=
netDevice->GetObject<WifiNetDevice>();
Ptr<WifiPhy> netDevicePhyicalLayer = wifiNetDevice->GetPhy();///< Get the access to the physical layer
   if (!netDevicePhyicalLayer->IsStateBusy())
   netDevicePhyicalLayer->SetChannelNumber(c); ///< Change the channel of the radio interface
   else
   {
     Time delayToIdle =
    netDevicePhyicalLayer->GetDelayUntilIdle(); ///< Get the time left until the device get free
     Simulator::Schedule(delayToIdle,&SwitchInterface, this, c); ///< Schedule the timer to change the channel later
   }
```

The *Destination Channel Query* function finds the channel on which the packet must be sent. A *Packet Queuing* mechanism buffers packets and keeps them until the sender switches one radio interface to the corresponding destination channel. To avoid saturation in a given queue, the packets that have been waiting for a period of time exceeding a certain threshold, are dropped from the queue. For such purpose, a time stamp is assigned to each new arriving packet and used as a reference to drop the packets.

a. Destination Channel Query: Multi radio nodes keep the following information about their neighbors in addition to their address:

- The number of radio interfaces a node has,
- The channel assigned to each radio interface,
- At which time each node will switch from one channel to another,
- At which time each node can receive packets on each channel.

The mentioned information is kept in the *Neighbor Table.* To forward a packet, a node gets the receiving channel of the next hop neighbor. With this information the node can tune a radio to that channel and send the pending packets.

Nodes use control messages to update the information about their neighbors in the *Neighbor Table.* This is necessary for the adaptive channel allocation mechanisms, where a node changes the channel of its radio interface frequently.

Every time that a node informs its neighbors about switching the channel of its radio interface, the neighbors will update the entry in the *Neighbor Table*.

In a multi channel architecture, nodes may need to monitor the available channels to acquire the list of busy channels. When forwarding a packet, the node should make sure that the receiver is not in the monitoring mode. Therefore, the node needs to keep the next monitoring period for all of its neighbors to avoid initializing any transmission.

Program 3 shows an entry of the *Neighbor Table*.

Program 3 An entry of the Neighbor Table

```
struct NeighborTableEntry
{
  uint32_t m_id; ///< The ID of the neighbor
  uint32_t m_neighborRadioNo;///< The number of radio interfaces that
       the neighbor has
  uint32_t m_neighborChannel; ///< The channel used by neighbor's
      radio interface
  Address m_Addr;///< MAC address of the neighbor's radio interface
  Time m_switchTime;///< Shows the remaining time that a neighbor
      stay on the current channel
  uint32_t m_neighborNewChannel;///< New channel for the receiving
      radio
  Time m_monitorTime; ///< Shows the remaining time until the
      neighbor starts monitoring a channel
  Time m_updateTime;///< Time stamp, which shows the moment that the
      information is updated
  bool close; ///< If the entry expired or not
}
```

b. Timestamp Assignment: Nodes must time stamp the packets waiting to be sent over a certain channel, in the case the destination channel is busy, the older packets must be discarded to avoid saturation. Therefore, it is possible to control the length of the buffers.

The timestamp assigned to each packet is equal to the time at which the packet was received from the routing agent ($T_{enqueu}(p)$). We define $T_{wait}(p)$ as the maximum amount of time that a packet is allowed to stay in the buffer before it is transmitted over the wireless media. To remove the old packets, a node checks the current time of the system ($T_{current}$), and deletes those packets that $T_{current} > T_{enqueu}(p) + T_{wait}(p)$.

To avoid the queues from getting saturated in high traffic rate, nodes may select a smaller value for $T_{wait}(p)$.

c. Data Queuing: In multi-radio multi-channel networks, the number of

Element	Description
Packet Type	The type of the packet: Data or Control
Packet	A pointer to the packet
Expire Time	The time at which the packet expires and should be dropped

Table 8.2: Packet buffers elements

available frequencies is bigger than the number of radio interfaces at each router ($|C| > I$). Therefore neighboring nodes may set their antennas over different channels. A wireless mesh node which has traffic for more than one neighbor, might need to switch to different channels to be able to deliver its packets to the next hop. During the time that a node transmits data over a channel, packets destined for other channels must be buffered to be sent later.

We create sequential First-in, First-out queues for buffering packets in ns-3 for different channels. Each queue corresponds to one of the available channels. The defined queues have the ability to eliminate the old packets automatically as described in Section 8.2.2.

Table. 8.2 shows the fields of each element of the queue with a brief description.

8.2.3 External Interference and Channel Sensing

Simulating two separate wireless networks which interfere each other is not possible in ns-3. Separated wireless networks simulated in the same scenario do not have any effect over each other since ns-3 treated them separately. Note that, Signal to Noise plus Interference (SNIR) is defined as part of the physical layer of the WiFi device [33]. Therefore setting the noise figure weight for a radio interface affects all packets which have been received over different channels. For an adaptive channel assignment we have a scenario where each channel is occupied by different amounts of interference. Nodes estimate the amount of interference through sensing or monitoring all channels at the same time [43, 57]. Simulating this scenario is possible only if the interference threshold is set separately for each channel.

Therefore, to simulate the external interference on wireless channels, we have created an *Interference Emulator* for each channel (Program 4). The *interference emulator* is based on a semi Markov model of the possible channel status (Fig. 8.4), where the status of a channel is considered as either *Busy* or *Idle*.

The emulator keeps the channel *Busy* for predefined period of time (*Busy-*

Duration). The duration of *Idle* state is determined using an exponential random variable with mean equal to *MeanIdleTime*.

In the constructor (Line 8 of Program 4), a timer is initialized to call the *ChangeStatus* function when it expires. Initially the delay of the timer is set to an exponentially distributed random variable with mean equal to 8 *ms*.

ChangeStatus changes the status of the channel from *Idle_State* to *Busy_State* or vice versa. It sets the timer to *BusyDuration* after setting the channel status to *Busy_State*. Whenever it changes the channel status to *Idle_State* it sets the timer to a randomly selected duration.

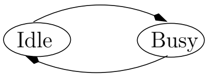


Figure 8.4: Channel Status

Program 4 External Interference Emulator

```
1      ChannelEmu::ChannelEmu():
2
3      busyDuration(<MilliSeconds> BusyDuration), ///< The duration of
           Busy status
4      nextTime(ExponentialVariable(MeanIdleTime)), ///< The random
           variable which determines the duration of Idle status
5      Status(Idle_State),///< Initial status of the emulator
6      statusTimer(Timer::CANCEL_ON_DESTROY)
7
8      { statusTimer.SetDelay(nextTime.GetValue());
9        statusTimer.SetFunction(&ChannelEmu::ChangeStatus, this); ///<
             Set the function which is called when the timer is expired
10       statusTimer.Schedule();}
```

To monitor a channel, a node checks the *Interference Emulator Status* attached to the channel during the sensing period.

The amount of interference over a channel, induced by an external network, can be varied by setting different values for *MeanIdleTime* and *BusyDuration*.

8.2.4 Routing Modifications

When simulating multi radio multi channel WMNs, considering a proper routing protocol is challenging. The routing have to be considered as a joint problem with the channel assignment [3, 36, 43, 44], since any change in the channel radio mapping affects the quality of the links between nodes and may trigger the routing protocol to re-route the traffic. On the other hand, the changes in the traffic pattern have an impact on the next decision of

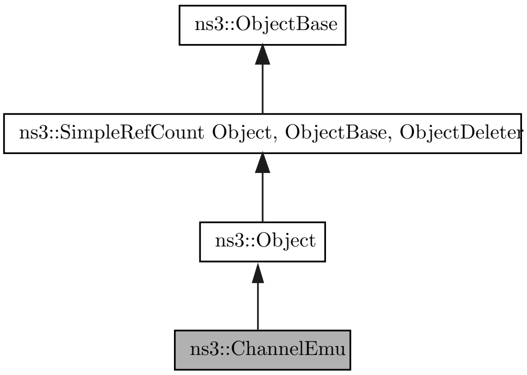


Figure 8.5: Class diagram of Interference Emulator

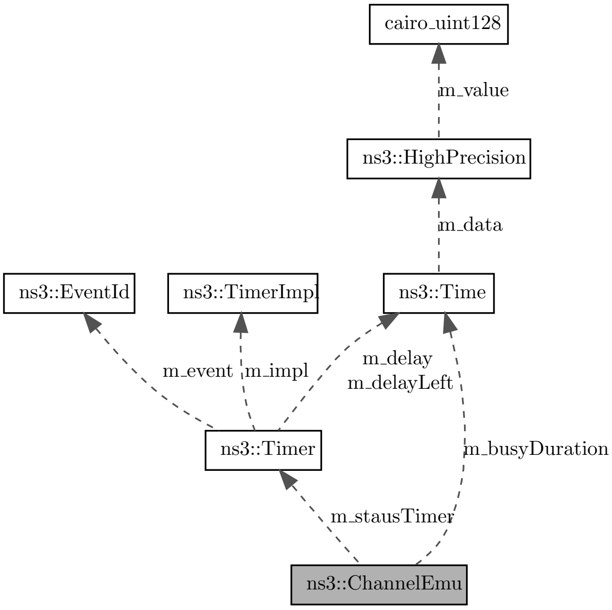


Figure 8.6: Collaboration class diagram of Interference Emulator

the channel assignment protocol. Therefore, the routing protocol and the routing metric must be modified to adapt to the channel assignment [15].

In ns-3 the available routing protocols are not applicable for multi radio wireless networks when using dynamic or hybrid channel assignment. Nevertheless, as we are only focusing on fixed WMNs, where nodes are fixed or have limited mobility, the static routing is a simple way to avoid complexity related to the dynamic routing protocols. The Global Routing simulated in ns-3, provides static routing only for wired networks by filling the routing tables at the beginning of the simulation. In the case of wireless networks there is no static routing since the topology of a wireless network is determined by the propagation model and other parameters at run time [41].

Therefore, we have attached a static routing table to each node and initialized it using Shortest Path First (SPF) algorithm, minimizing the number of hops between each node and any other destination. Each node knows about the next hop node on the paths to all other nodes in the network. Program 5 shows an entry of the *Routing Table*.

Program 5 A routing table entry

```
 struct RoutingEntry
  {
    uint32_t m_dst;///<The Id of the destination node
    uint32_t m_nextHop; ///<The Id of the next hop node toward the
        destination
    double  m_metric; ///<The metric value of the path
}
```

The *Routing Table* is filled using a file containing all shortest paths between all nodes. We have created this file using R numerical tool [42] feeding the position of nodes in the network.

The routing header which is attached to each packet has the following elements:

- Sequence number,
- Node ID of the source,
- Node ID of the destination,
- Node ID of the next hop device to the destination,
- The time of originating the packet.

The relay nodes on the path from the source to the destination, update the ID of the next hop node in the header and froward the packet.

8.2.5 Traffic Generator

Channel assignment problem can be formulated in a way to reduce the time overlapping of low rate traffics and high rate traffics by using wireless mesh nodes to communicate over different channels based on their respective data rates [62, 30]. This approach increases the throughput of high rate traffics.

The traffic generated in ns-3, must be provided by an intermediate agent to interact with the dynamic CA in the case that, there is the need to control the traffic rate. Moreover for dynamic or hybrid CA, the packets can not be sent immediately through the media after the next hop has been selected, since the radio interface should first switch to the destination channel. For simplicity, we have created a simple constant bit rate (CBR) packet source that generates traffic with a specific packet size (1000 *b*) at the constant rate (i.e., 100 *kbps*). The packet generator is attached to the source node. Program 6 shows our CBR traffic generator.

Program 6 Constant Bit Rate (CBR) Packet Generator

```
1 GenerateCBRTraffic( uint32_t pSize, uint32_t pRate)
2 {
3   Ptr<Packet> p= Create<Packet>(pSize); ///< Create a packet with the
        defined size
4   RoutingPort(p);///< Deliver packet to the routing protocol
5   Time packetInterval= MilliSeconds(pSize/pRate); ///< Compute the
        time duration between creating each packet
6   Simulator::Schedule (packetInterval, &GenerateCBRTraffic,this
7   ,pSize,pRate);///< Call the function to create the next packet after
         the interval
8 }
```

The packet is created (Line 3) and delivered to the routing agent (Line 4) which adds the routing header to it and sends it to the data service (Fig 8.3). Then, the packet is enqueued to the corresponding queue (Section 8.2.2).

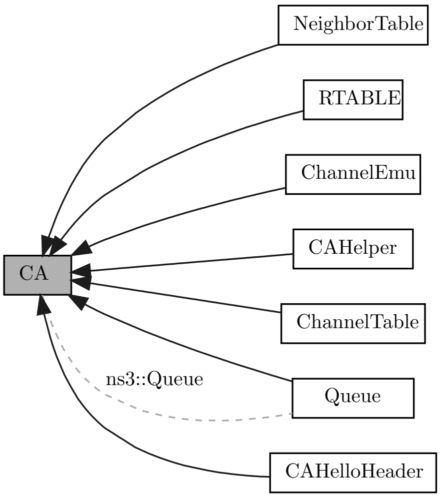


Figure 8.7: Class Diagram of the CA Class

8.3 An Example of CA Implementation: SICA

Fig. 8.7-8.8 show the class diagram and the collaboration diagram of using the defined classes to model a channel assignment protocol. In the following

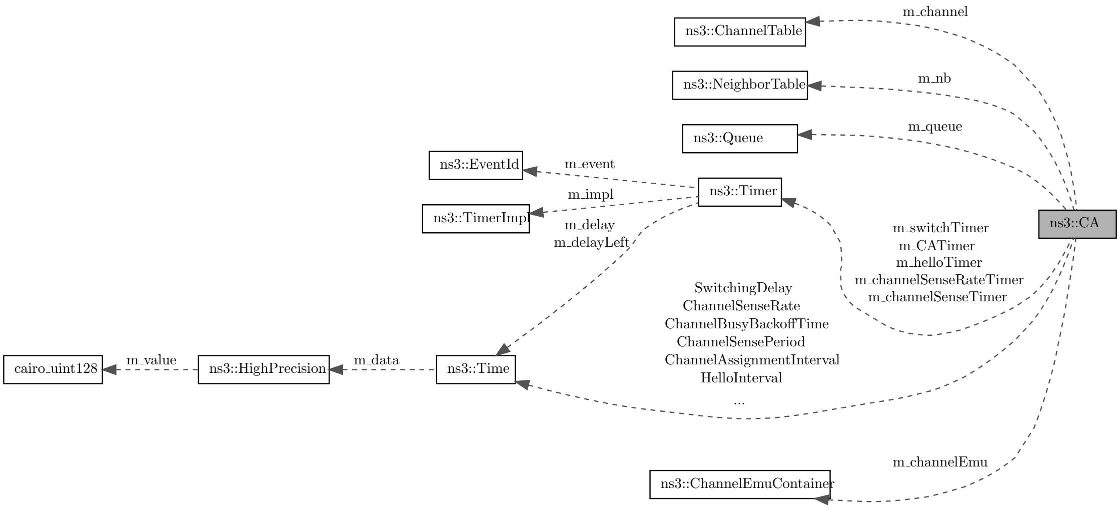


Figure 8.8: Collaboration Class Diagram of the CA Class

section we show with more detail the simulation of a channel assignment mechanism, called SICA, in ns-3 using the components introduced before. Moreover we provide the details of some important process regarding the *Data Forwarding* and *Channel Selection.*

Semi dynamic Interference aware Channel Assignment mechanism (SICA), is a protocol proposed and simulated in ns-3 [41], for wireless mesh networks [40]. The source code of SICA is available and can be accessed at [39].

SICA is implemented in a network where nodes are equipped with two radio interfaces, each one is able to use a set C (with cardinality $|C| > 1$) of non overlapping channels. The radios will be referred to as the *receiving radio* and the *transmitting radio*, and denoted by R and T, respectively.

The aim of the channel assignment mechanism is to select the channels which suffer less interferences in terms of both internal and external interference. The channel assignment mechanism selects and assigns a channel to the R radio of each node. Then, the node switches the T radio according to the receiving channel of its neighbors to start transmission. After a channel switch, the T radio remains on the same channel until all packets, which are addressed to the same destination node, have been sent, or until a maximum period of time has expired [40].

Nodes estimate the amount of external interference on a channel via sensing the channel. Then, they use control packets called *Hello* to exchange channel sensing information and inform their neighbors about upcoming interface switching for the receiving radio (R).

8.3.1 External Interference Estimation

Using the *Channel Emulators* (Section 8.2.3), the status of a channel can be monitored as *Busy* or *Idle*. Each node monitors the channel's status for its receiving radio (R) once in each sensing period of length T_{SS} seconds. For monitoring a channel c, the node checks the status of the corresponding *Channel Emulator* (c_e) at a pre-defined rate (T_{SRate}) (Program 7).

Program 7 Monitoring Channel

```
1 void SenseChannel(uint32_t c)
2 {
3   Ptr<ChannelEmu> emu=GetChannelEmu(c);///< Get the interference
        emulator attached to the channel
4   if (emu->IsBusy())
5   T_busy+=ChannelSenseRate; ///< Increase the channel busy time if the
        channel is busy
6   else
7   T_idle+=ChannelSenseRate; ///< Increase the channel idle time if the
        channel is idle
8   m_channelSenseTimer.Schedule();///< Reschedule the sense timer for
        the next sensing period
9 }
```

T_{busy} (Line 5) and T_{idle} (Line 7), are global variables which save the duration of the *Busy* and the *Idle* state respectively.

At the end of the monitoring period the node estimates the amount of external interference over a channel (B_{ext}) using Equation (8.1).

$$B_{ext}(c) = \frac{T_{busy}(c)}{T_{busy}(c) + T_{idle}(c)} \tag{8.1}$$

Note that, each node senses only one channel during the monitoring period. It then sends this information to its neighbors and as its neighbors do the same, it gathers information about other channels via the control packets received from its neighbors (see Section 8.3.3).

The internal interference is estimated based on the *Interference Protocol Model* proposed in [24]. Each node is informed by its neighbors about the current channel of their receiving radio. Therefore a node i can calculate the number of neighboring nodes over channel c ($R_i(c)$). Then the node estimates the density of interfering nodes on channel c by $\frac{R_i(c)}{N_i}$ where N_i represents the total number of neighbors of node i.

In SICA, nodes use the average of the estimated external and internal interference over a channel as a metric to select a channel that has more available capacity.

8.3.2 Channel Selection Mechanism

The channel selection mechanism is developed as a repeated game which uses the interference estimation information and selects the best possible channel for the receiving radio of a node.

Initially the channel selection mechanism calculates and assigns a weight to all available channels based on the amount of internal and external interference estimated over a channel. Then, the weights are updated using the multiplicative weight update technique proposed in [18, 19].

We define $w_t(c)$ as the weight assigned to channel c at time t. The channel assignment computes the probability of selecting a channel as :

$$P_t(c) = \frac{w_t(c)}{\sum_{c \in C} w_t(c)} \tag{8.2}$$

Initially all weights are set to 1, thus, the probability of selecting any channel is identical. After selecting a channel, the node gathers information from its neighbors and updates the loss that it is suffered. Then, the weights are updated as follows:

$$w_{i,t}(c) = w_{i,t-1}(c)\, \beta^{M_i(c)} \tag{8.3}$$

where $M_i(c)$ is the loss suffered by node i considering the external and internal interference over channel c [40], and β is the game parameter in the range of $(0, 1)$ [19].

Then SICA selects a radom channel considering the probability of each channel as shown in Program 8.

Program 8 computes and sorts the probability of selecting a channel according to Equation. (8.2) (Line 11- 18 and Line 20). Then it sorts the list of available channels based on the weights (probabilities) assigned to each channel (Line 19). Line 21 computes the cumulative probability vector by adding up each probability value with all probabilities lower than it. Then the probabilities are fed to the empirical random variable generator (Line 22-23) which generates a random number as an index for the available channels' list (Line 24 of Program 8).

Program 8 Channel Selection Mechanism

```
uint32_t SelectRandomChannel(std::vector<uint32_t> channels)
{
  /// channels contains the list of available channels
  std::vector<double> prob; ///< A vector to keep the probability of
      selecting any channel
  std::vector<double> weight;///< A vector to keep the weights
      assigned to  each channel
  std::vector<double> cProb;///< A vector to keep the cumulative
      probability  of channels
  EmpiricalVariable  emRnd;///< An empirical random variable to select
       a random number based on cumulative probabilities
  double totalWeight=0;
  double tempW;
  /// read the weights of channels
  for (std::vector<uint32_t>::iterator i=channels.begin(); i!=channels
      .end(); ++i)
    {
      tempW=m_channel.GetChannelWeight(*i);
      weight.push_back(tempW);
      totalWeight+=tempW;
    }
  for (uint32_t i=0; i<weight.size(); i++)
    prob.push_back(weight[i]/totalWeight);///< Compute the
      probabilities using the weights
  sort(channels.begin(),channels.end(),CompareChannelsWeight(*this));
      ///<  Sort channels considering their weights
  sort(prob.begin(),prob.end());///< Sort the probabilities
  cProb=ComputeCumulativeProbability(prob);///< Adds up the
      probabilities to have the cumulative probability vector
  for (uint32_t i=0; i< cProb.size(); i++)
   emRnd.CDF(i,cProb[i]);///< Feeds the probabilities to the random
      variable
  return  channels[emRnd.GetInteger()];///< Select a  random channel
      considering the assigned probabilities
}
```

8.3.3 Control Packet Elements

Unlike most CAs proposed in the literature [43, 44, 57], in SICA there is no common control channel shared by all nodes. In SICA, the synchronization is achieved through exchanging packets over the data channels. Since each node can assign a different channel to its receiving (R) radio, nodes must be aware of the channels used by their neighbors' R radios. In SICA, a node broadcasts *Hello* packets to report the channel of its R radio to its neighbors. Fig 8.9 shows the content of a *Hello* packet.

In addition to the receiving channel announcement, *Hello* packets are used to inform about the channel sensing information and the receiving channel of the neighboring nodes, so a node can compute the external and internal interference for each channel. In order to do that, there is a field in each packet which contains the amount of external interference over the receiving

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Hello Sequence Number                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Originator Node ID                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Hello Originator Time                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Numboer Of Radios| Receiving Channel (C) Busy Duration Over|CNext Receiving Channel|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Receiving Radio MAC Address (Part 1)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Receiving Radio MAC Address (Part 2)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Remaining Time Before Switching To New Channel           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Remaining Time Before Sensing Current Channel            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Number Of Neighbors Receiving Channel -1  Receiving Channel -2  Receiving Channel -3|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Neighbor 1 Node ID                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Neighbor 2 Node ID (if exist)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Neighbor 3 Node ID (if exist)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8.9: Hello Packet Elements

channel estimated by the node. The number of neighbors and the channels of their receiving radio is also added at the end of each *Hello* packet.

The remaining time before a node switches its R radio to another channel and the remaining time before the node starts sensing the current channel is also announced via *Hello* packets. The announcements inform the neighboring nodes about upcoming channel switching or sensing events.

8.3.4 Data And Control Transmission

For each channel $c \in C$, SICA maintains two sequential First In First Out queues for buffering packets (Section 8.2.2): control packets queue ($Q_{ctrl}(c)$); and data packets queue ($Q_d(c)$). $Q_{ctrl}(c)$ has higher priority than $Q_d(c)$. The source and each node in the middle of the path to the destination, push each packet to the corresponding queue attached to each channel. Each packet is time stamped when it is pushed to the queue. Using the timestamp as a reference, packets that remain in the queue for a longer time than a certain threshold are discarded to avoid saturation (see Section 8.2.2).

When the transmitter radio switches to a channel, it fetches the stored packets from the queues and sends them until there are no packets left in the queues or until a maximum amount of time has elapsed [40]. Program 9 shows how packets are sent over a channel c.

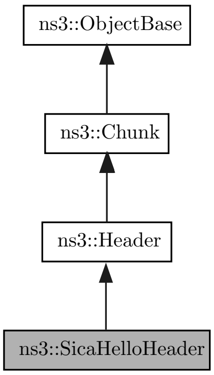


Figure 8.10: Hello Header Class Diagram

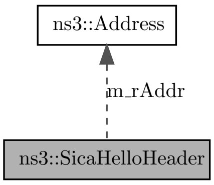


Figure 8.11: Hello Header Collaboration Class Diagram

TInterfaceReadyToSend (Line 4) is a function which gets the ID of the channel and the estimation of the required time to send a packet by the T radio, and checks whether it is possible or not to send a packet over the channel.

In the following cases the *TInterfaceReadyToSend* returns *False*, which prevents the node from sending packets over channel c:

- The channel is being sensed by any of the neighbors.
- The channel is busy due to the external interference.
- The T interface did not switch to the channel successfully.
- The estimated transmission time is less than the remaining time over the current channel.

8.4 Validating the game model

The channel assignment in SICA is modeled as a repeated game, where nodes compete to occupy the most vacant channel (as explained in detail in Section 8.3.2). The game is played repeatedly in a sequence of *game rounds*. In each round, the player plays a mixed strategy based on the weights assigned to each channel.

Program 9 Send packets over channel c

```
uint32_t helloQueueSize=$Q_{ctrl}$.GetSize(c);///< Get the size of the queue which keeps control messages
uint32_t dataQueueSize=$Q_d$.GetSize(c);///< Get the size of the queue which keeps data messages
Time txEstimation=EstimateTxDuration(maxPacketSize,wifiphy);
while ((helloQueueSize>0 || dataQueueSize>0 )&& TInterfaceReadyToSend(c,txEstimation))
{
  if (helloQueueSize>0)
  {
  protocolNumber=SICA_HELLO_PORT;///< Set the port number to control port number
  qEntry=$Q_{ctrl}$.Dequeue(c);///< Fetch a queue entry of a control message
  }
  else
  {
    protocolNumber=SICA_DATA_PORT;///< Set the port number to data port number
    qEntry=$Q_d$.Dequeue(c);///< Fetch a queue entry of a data message
  }
  if ( qEntry)
  {
    SendPacket(qEntry->GetPacket()->Copy(),m_tInterface, protocolNumber);///< Send data or control messages using different port number
  }
  helloQueueSize=$Q_{ctrl}$.GetSize(c);
  dataQueueSize=$Q_d$.GetSize(c);
}
```

In the following section we explain the Markov model we use to validate the game theory model used in SICA. The Markov model is used to proof that, the game theory based channel assignment algorithm converges to a steady state if the algorithm is played in sufficiently enough rounds in the case that the environment does not change. We simulate the game model using R numerical tool [42] and compare the result with the results obtained using the Markov model to compute the probability of selecting channels. We define the process of selecting a channel for a radio interface at each node as a Markov chain.

Fig. 8.12 shows the Markov chain that models how a node selects the channels. The channels set (strategy set) was mapped to the state space of the Markov model. The payoff $P_i = 1 - M_i$, is considered as the reward for the transition between channel c and $\acute{c}$ by node i.

We define the state transition matrix (Q) by using the Boltzmann distribution (equation (8.4)) [56]. Boltzmann distribution is used to solve the Markov model, by defining the transition from state c to $\acute{c}$ related to the

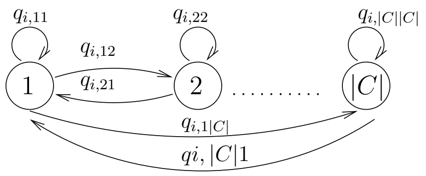


Figure 8.12: Markovian process of changing channel at node i

reward obtained on state $\acute{c}$.

We assume that the probability of transmission from channel c to $\acute{c}$ is related to the reward of the destination channel normalized by the total reward of all channels:

$$Q_{i,c\acute{c}} = \frac{e^{\frac{P_i[\acute{c}]}{\lambda}}}{\sum_{k \in C} e^{\frac{P_i[k]}{\lambda}}} \tag{8.4}$$

Here λ is the learning parameter. For our model we define $\lambda = 1 - \beta$, where β is the learning parameter used for the Game model of SICA (see Section 8.3). Note that small values of λ enable the player to choose the optimal strategy more accurately.

The stationary transition matrix (i.e the eigenvalues of Q) is computed using the global balance equation for ergodic Markov chains:

$$\begin{cases} \rho(Q - I) = 0 \\ \rho E = e \end{cases} \tag{8.5}$$

Where I is the identity matrix, E is the square unit matrix and e is the unit row matrix.

By solving equation (8.5) we get the stationary channel transition vector for node i as $\rho = e(Q + E - I)^{-1}$. Using the R tool we've simulated the Game based learning algorithm of SICA for a node. We define $\acute{\rho}$ as the final probability vector of selecting each channel equal to the mixed strategy vector obtained for the node. We compare $\acute{\rho}$ obtained for different rounds of SICA algorithm, with ρ obtained from the Markov model (Fig. 8.13-8.14). Moreover we run SICA in ns-3 simulator and report the probability distribution of selecting channels over 200 s of simulation run. (Fig. 8.15-8.16).

Fig. 8.13 shows the probability of selecting a channel at each node. For clarifying, we show the available bandwidth of each channel with a red line

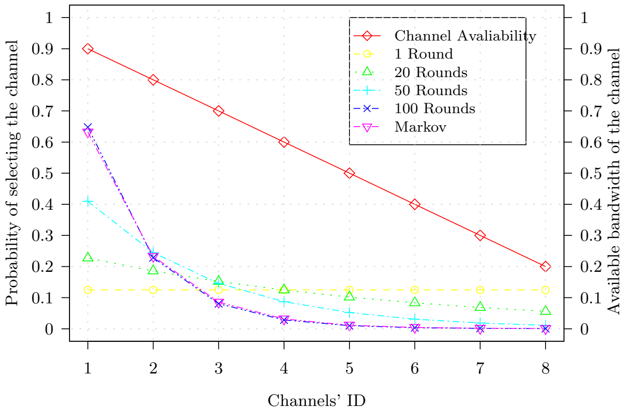


Figure 8.13: The probability of selecting a channel for a node regarding the channel occupancy for different rounds of the game

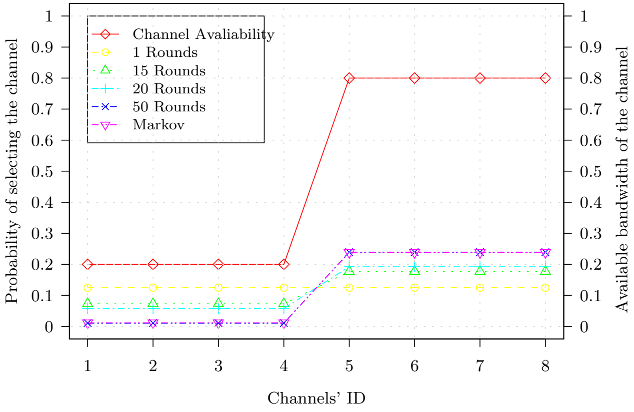


Figure 8.14: The probability of selecting a channel for a node regarding the channel occupancy for different rounds of the game

at the same figure with the probability distribution of selecting a channel. As shown, the available bandwidth in each channel decreases linearly from channel 1 to 8. The right y-axis shows the available bandwidth related to the total bandwidth of a channel while the left y-axis shows the probability of selecting the channel.

Fig. 8.13 shows that the weighted learning algorithm matches quite closely the result from solving the Markov model after 100 rounds for $\beta = 0.9$.

We examine SICA and the Markov model for the case that 50% of the channels (channels with the IDs from 5 to 8) are occupied by the external interference. The interference occupies 80% of the bandwidth of those channels, while channels with the IDs from 1 to 4 are almost free.

Fig. 8.14 shows that, in this specific scenario, when the decision about selecting the best channels, is more clear, the learning algorithm of SICA converges to the Markov solution much faster, in this case only after 15 rounds.

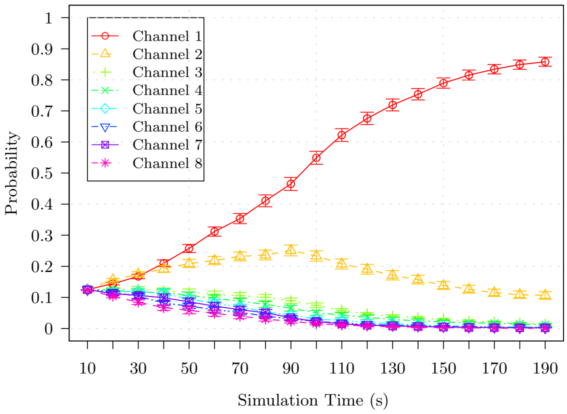


Figure 8.15: The probability of selecting a channel for a node over time

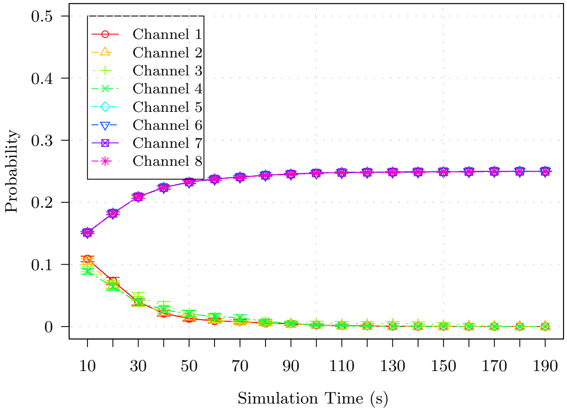


Figure 8.16: The probability of selecting a channel for a node over time

To study the behavior of SICA over time, we simulate SICA using ns-3 simulator for a grid network of size 5 × 5 nodes and 2 CBR traffic flows of 100 kbps. The results are obtained averaging over 10 simulation runs and the error bars show the 90% confidence interval.

Fig 8.15 shows how the probability of selecting a target channel changes over time, when a different amount of external interference is introduced on each channel. Specifically, the channel with the higher ID suffers more external interference. The figure shows that, the nodes tend to select the channel which offers more bandwidth, which is the expected behavior of any CA.

Fig 8.16 shows the average probability of selecting each channel over time when the external interference is introduced on the first four channels (channels with the IDs from 1 to 4) while other channels are free. The figure shows that the probability of selecting any of the busy channels tends to 0 over time, while the probability of selecting any of the free channels (channels with the IDs $5 - 8$) increases to 0.25 which confirms the results shown in Fig 8.14. Moreover, it can be observed how SICA achieves a proportional utilization of the four free channels.

8.5 Verifying the Simulation Model

To verify the simulation in ns-3, we design a state space explorer (SSE) which is invoked by the major events that happen in the simulations and performs state checking to verify the correctness of the simulation [53].

We assume that the simulation, as a system, consists of a group of entities joined together to accomplish the packet delivery goal and hence virtually, represents the operation of the real system.

We consider each radio interface as an *entity* with the following attributes:

- The frequency channel it is using.
- The number of packets it has received.
- The number of packets that has sent.
- The number of packets that has dropped.

The *State* of the system (s_t) is the complete description of the system at any time which includes all entities and values of their attributes. We consider those attributes which are relevant to the objective of interest. s_t represents a function of all entities which assigns a value to all attributes, we consider that, ξ contains all possible states of the system, and call it *state space.*

The *Event* (e_t) is any occurrence which changes the state of the system. In our model we consider the following events:

- Sending a packet (*Send*, unconditional event).
- Receiving a packet (*Receive*, conditional, $e_t = Receive|\ e_{t_0} = Send\ \ t_0 < t$).
- Dropping a packet (*Drop*, conditional , $e_t = Drop|\ e_{t_0} = Send\ \ t_0 < t$).
- Changing the channel of the radio interface (*Channel-Change*, unconditional event).

We define $\mathcal{E}$ as the set containing all above events. The first three events occurred between two entities: the sender and the receiver.

An ***Assertion*** is a property which must be hold for all states of the system to be able to proof the correctness of the simulation.

For the simulation of the channel assignment protocol we define the following assertions:

- Only one channel must be assigned to a radio,
- There must be a common channel between the sender radio and the receiver radio at any given time.

We've designed SSE in ns-3, which starts from the initial state of the system and reaches the successor state following the simulation events, and checks for any violation from the assertions.

s_0 is defined as the initial state of the simulation. We define $\acute{s}$ is explored by s_0 if $\acute{s}$ is the successor state of the system after the triggering event e $(e \in \mathcal{E})$ and we notate it as: $s_0 \overset{e}{\rightarrow} \acute{s}$. The event handler is triggered by any of the events in $\mathcal{E}$ and it calls the *state control* procedure to check the assertion of the successor event.

$f_{i,c}$ is defined as the number of receiving radios of node i which are tuned to channel c. For any node i the assertion condition is:

$$\sum_{k=1}^{|C|} f_{i,k} = I_i \ , \ \ \forall i \in N \tag{8.6}$$

where I_i represents the number of radio interfaces of node i.

For any transmission pairs (i, j) the assertion condition is:

$$\exists c \in C \ , \ \ f_{i,c} . f_{j,c} = 1 \tag{8.7}$$

Alg. 8.1, shows the state check algorithm which is triggered by sending, receiving or dropping packet events and checks if any violations happen from the assertions.

Alg 8.2 shows the process of checking *Channel-Change* event. The state checker gets the node which selects the new channels and checks whether the number of channels selected by the node is equal or less than the number of radio interfaces it has. In SICA each node selects only one channel at a time and all nodes are equipped with two radio interfaces, thus the violation from the assertions never happens for the *Channel-Change* events.

```
Input:
  $S_t$: The current state of the system at time $t$.
  $e_{t-1}$: The last event which moved the system to state $S_t$.
  $P$: The pointer to the packet in the case of sending and receiving events.
if $S_t == S_0$ then
    exit(0)
else
    $src = e_{t-1} -> Sender()$
    $dst = e_{t-1} -> Receiver()$
    for $c \in C$ do
        if $f_{src,c} \cdot f_{dst,c}$ then
            Write('Assertion holds')
            exit(0)
        end if
    end for
    Write('Assertion violated')
    exit(1)
end if
```

Algorithm 8.1: StateCheck1(S_t,e_{t-1},P)

Fig. 8.17-8.18 are obtained running *State Space Check* algorithm (Alg. 8.1) for checking the simulation events of SICA. The simulation parameters are the same as the parameters considered for Fig. 8.16.

Fig. 8.17, shows the state space control for *Send*, *Receive* and *Drop* events. The figure depicts the number of events that happened during 200 s of simulation run, and the number of events for which the assertion holds. The figure shows that, SICA does not violate any of the assertions.

Fig. 8.18 shows the number of packets which have been sent, received or dropped on each channel when the external interference is introduced over channels with the IDs from 1 to 4. The figure confirms that SICA makes the network to use the capacity of the free channels efficiently.

Fig. 8.19 shows the average number of radio interfaces over each channel during the simulation time. Note that the simulation parameters are the same as Fig. 8.18. The figure shows that the number of radio interfaces over the set of channels that are occupied by external interference decreases over time. It proves that, the channel assignment successfully detects and avoids the channels which are occupied by external interference as it is expected.

Fig. 8.20 shows the average number of packets (with a size of 1kb) in each data queue over the time. The figure shows that the proposed data delivery mechanism ensures that the number of waiting packets in each channel is lower than the default maximum queue size in ns-3 (100 packets) [41]. In addition the maximum number of packets waiting in a queue at the begin-

```
Input:
  $S_t$: The current state of the system at time $t$.
  $e_{t-1}$: The last event which moved the system to state $S_t$.
if $S_t == S_0$ then
    exit(0)
else
    $N = e_{t-1}- > Node()$
    $C = N- > Channels()$
    $I = N- > Radios()$
    if $C <= I$ then
        Write('Assertion holds')
        exit(0)
    end if
    Write('Assertion violated')
    exit(1)
end if
```

Algorithm 8.2: StateCheck2(S_t,e_{t-1})

ning of the simulation (marked by red), happened over those channels that are kept busy by external interference.

8.6 Conclusions

In this chapter we have presented the extensions which must be done in ns-3 simulator to simulate channel assignment mechanisms for multi radio wireless networks. We provided the simulation details of the Semi-dynamic Interference aware Channel Assignment (SICA) mechanism which is proposed in Chapter 7 [40] as an example. In addition the source code of SICA is published and available at [39]. We verified the simulation of SICA using a state space checker which checks the relevant simulation events for any violation from the feasibility conditions of channel assignment solution. The results prove the correctness of the simulation and show that SICA is capable of utilizing the channels in a fair and efficient way. Moreover we justified the results obtained by the Game theory based model of SICA, using a Markov chain model.

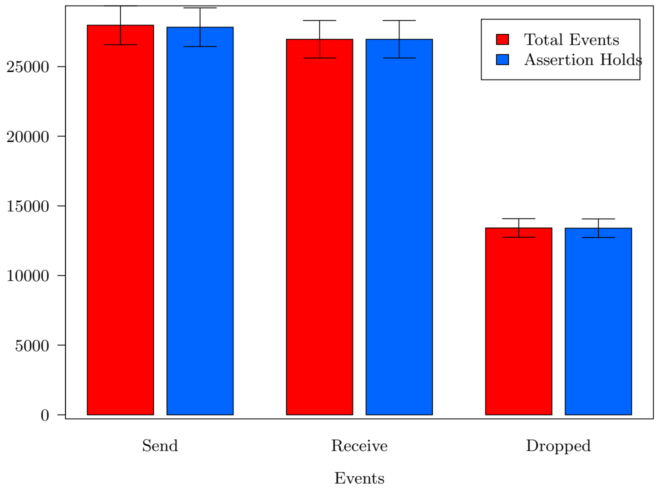


Figure 8.17: Protocols events and the assertions

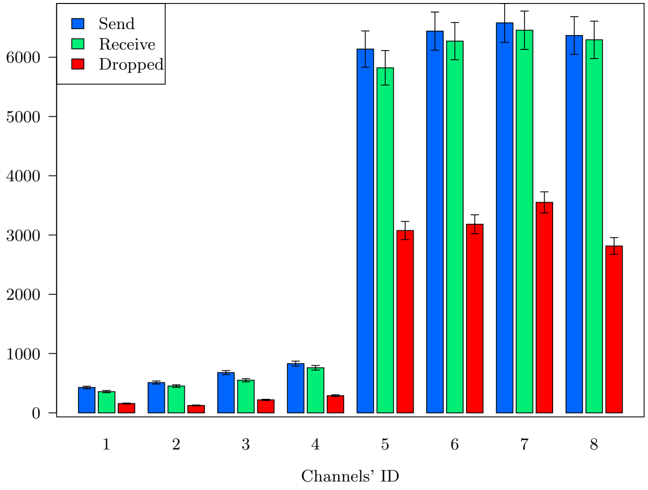


Figure 8.18: Number of events over each channel

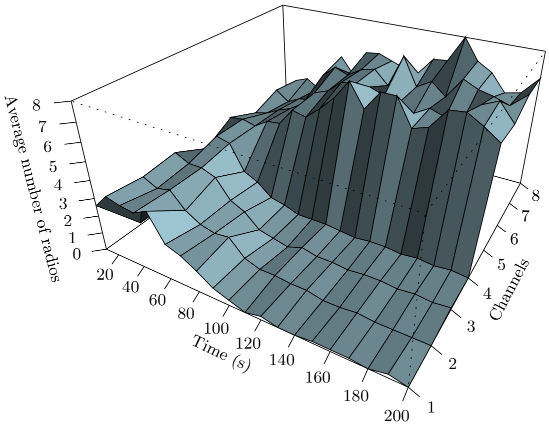


Figure 8.19: The average number of radio interfaces over each channel during the simulation time

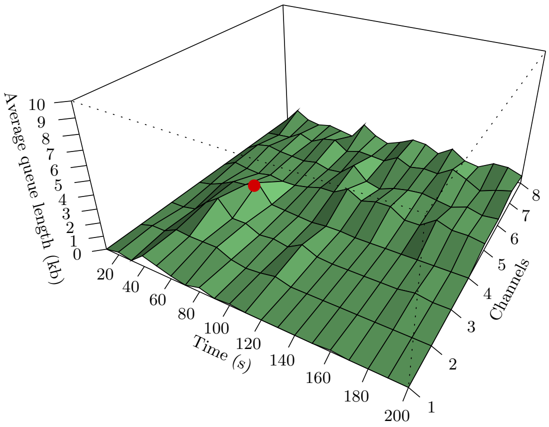


Figure 8.20: The average length of queues during the simulation time

CHAPTER 9

Next Generation Wireless Mesh Networks and Dynamic Channel Access

9.1 Introduction

IEEE 802.11s [27] defines the mesh operation in a single channel although multi-radio mesh stations (mesh STAs) can form different meshes. The connection between different meshes is provided via bridging. Mesh STAs can initiate the channel switching mechanism which moves the mesh or part of it to another channel. The STAs which do not want to follow the channel switch request may join another mesh. Although it is possible to have dynamic multi-channel networks, the network performance may be compromised drastically if mesh STAs join and leave their mesh too frequently due to channel switching overheads. In addition, the default path selection metric of the standard does not consider the dynamic channel selection of mesh STAs. And, it offers low performance in multi-channel multi-radio networks compared to other path metrics which consider the channel diversity for selecting the path [22].

This chapter, provides -in Section 9.2- an overview of how 802.11s based mesh networks work. It also presents some of the recent proposals for adaptive channel selection -in Section 9.3. And, it shows the performance gains of dynamic channel assignment in multi-channel mesh networks compared to the single channel one -in Section 9.4. The chapter is concluded in Section 9.5.

9.2 Current 802.11s Standard for Wireless Mesh Networks

Since 2004, the Task Group S has been developing an amendment for 802.11 standard to create multi hop wireless networks based on WLAN technology. The new standard called IEEE 802.11s that was finally released in 2011 [27],

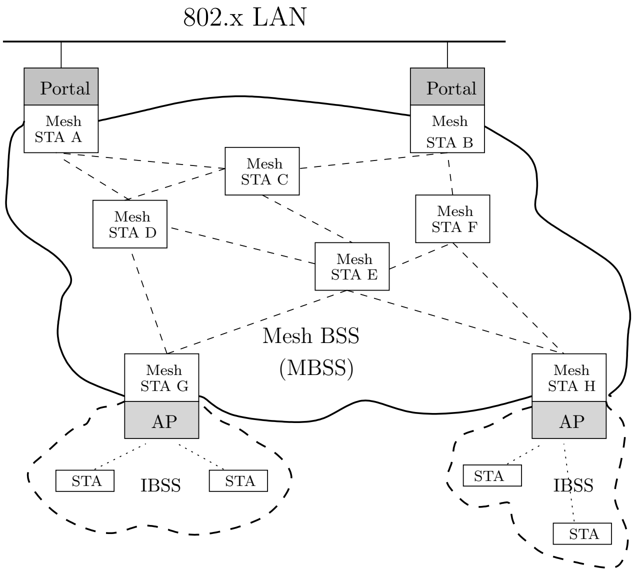


Figure 9.1: 802.11s Wireless Mesh Network

introduces wireless frame forwarding, routing capabilities (Path Selection) at MAC layer, interworking and security.

Interworking makes the 802.11s mesh network to look like a single Ethernet object to the outside (Fig. 9.1). A mesh BSS (MBSS) is an IEEE 802.11 LAN consisting of autonomous STAs [27]. Mesh STAs, which form the MBSS, forward packets wirelessly inside the mesh but they do not communicate with non mesh STAs (Fig. 9.1).

IEEE 802.11s added some fields -extension address fields, mesh sequence and time-to-live fields- to the data and management frames compared to conventional 802.11 frames. The address extentions allow for a total of six addresses in mesh data frames. This is useful when a mesh station acts as a proxy for some stations which do not belong to the mesh network, e.g. mesh STAs G and H in Fig. 9.1. The extended source and destination addresses carry the address of the source and destination which use the mesh STA as a proxy.

9.2.1 Mesh Formation

Active scanning (Probe frame transmission) or Passive scanning (observation of mesh beacons) are used by mesh STAs to detect each other. Through beacons, mesh STAs transfer the following information:

- Mesh ID: The ID of the mesh network.

- Mesh configuration: The path selection and path metric identifier, the congestion control mode, the synchronization method identifier, etc.
- Mesh parameters: Parameters that are supported by the transmitter mesh STA.
- Mesh Channel Switch Parameters.

Mesh STAs are considered to have only one radio interface and hence, the default operation for IEEE 802.11s mesh networks is in single channel. However, as Fig. 9.2-a shows multi-radio stations can form different meshes on different channels where the layer two bridging may be used to unify the different meshes in a single LAN. Fig. 9.2-b shows the three meshes formed on different channels.

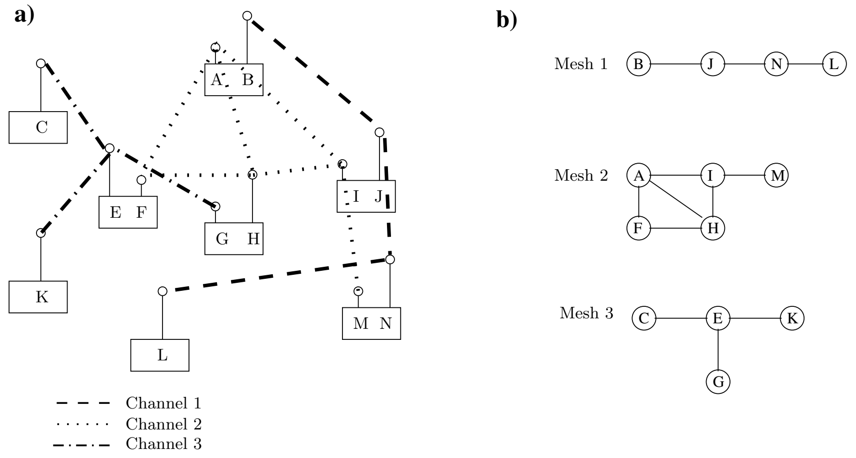


Figure 9.2: multi-radio mesh STAs

MBSS Channel Switching

A mesh STA can initiate a channel switching announcement to move the mesh to another channel due to any reason, such as interference or radar appearance. The mesh STA shall inform each of the peer mesh STAs that the mesh is moving to a new channel. It maintains the mesh peerings by advertising the switching event using Channel Switch Announcement elements together with Mesh Channel Switch Parameters element in Beacon frames, Probe Response frames, and Channel Switch Announcement frames until the intended channel switch time.

The channel switch has to be scheduled so that all mesh STAs in the MBSS, including mesh STAs in power save mode, have the opportunity to receive at least one Channel Switch Announcement element before the switch.

A mesh STA that receives a Channel Switch Announcement element may choose not to perform the specified switch, but to take an alternative action. For example, it may choose to move to a different MBSS.

It is important to remark that, the standard does not provide any default mechanism for selecting the target channel, hence opening the door for each manufacturer to implement their own solutions.

9.2.2 Medium Access Control

Mesh STAs use Mesh Coordination Function (MCF) to access the medium. MCF consists of a mandatory contention based and an optional contention free channel access mechanisms.

The contention based MCF relies on Enhanced Distributed Channel Access (EDCA), which provides limited support for quality of service (QoS) by defining four different traffic categories at the MAC layer [25]. A mesh STA using EDCA is able to transmit multiple frames whose total transmission duration does not exceed the transmission opportunity limit (TXOP). The receiver acknowledges any successful frame reception.

MCF Controlled Channel Access (MCCA) is an optional access method that allows mesh STAs to access the medium at selected times with lower contention than would otherwise be possible. Not all mesh STAs are required to use MCCA. MCCA may be used by a subset of mesh STAs within the mesh.

MCCA enabled mesh STAs, use management frames to make reservations for transmissions. The mesh STA has to transmit an MCCA Setup Request frame to initiate a reservation, the reserved TXOP is called MCCA Opportunity (MCCAOP). MCCAOP has a precise starting time and a pre-defined maximum duration, which requires a tight synchronization for all mesh STAs using MCCA. Note that the standard does not provide any default mechanism for scheduling the MCCAOPs, which may affect the performance of the MCCA [10].

The mesh STA advertises the MCCAOP via beacon frames. To avoid conflict with mesh STAs outside the beacon reception range, the mesh STA also includes its neighbors' MCCAOP reservation in the advertising frame. This implies that MCCA assumes the interference from nodes laying outside the two hops neighborhood of a link is negligible, which may degrade the performance of the channel access in realistic conditions [10].

The mesh STA which has reserved the MCCAOP will access the medium using EDCA and does not have any priority over other mesh STAs which does not support MCCA.

9.2.3 Congestion Control

Mesh networks suffer from interference due to the broadcast nature of the wireless medium. The mesh STAs, in the middle of the mesh, face more interference due to the high number of neighbors they have, compared to mesh STAs at the edge of the network. Since the access to the media relies on the carrier sense mechanism of 802.11 [25], core mesh STAs have less opportunities to access the media and hence they are prone to suffer congestion and start dropping frames. Dropping data frames is costly in a mesh network because a frame may travel several hops before reaching the congestion point.

IEEE 802.11s provides a congestion signaling protocol which allows the congested mesh STAs to advertise the expected duration of the congestion to the neighboring mesh STAs and inform them to slow down or to stop transmitting. The reduction of the frame arrival rate to a congested mesh STA avoids wasting the mesh resources for transmission of packets that with high probability will not be handled/forwarded by the congested mesh STA [20].

9.2.4 Synchronization and Beaconing

Beaconing procedure for mesh STAs is introduced in 802.11s. Since beacons are not acknowledged, a Mesh beacon collision avoidance (MBCA) mechanism is introduced to avoid collision between beacons transmitted by hidden nodes. Using MBCA, a mesh STA advertises its beacon interval and target beacon transmission times (TBTTs) in addition to its neighbors beacon interval. Upon receiving the beacon timing from its neighbors, the mesh STA uses these information to select its TBTT and beacon interval so that its beacons do not collide with the neighbors' beacons within the two hops transmission range.

In 802.11s the synchronization is similar to the timing synchronization function (TSF) of the original 802.11 standard [25]. In order to enable minimal synchronization capabilities between mesh STAs using MCCA, MBCA and power saving mode, an extensible synchronization framework is also introduced to support different synchronization mechanisms in the mesh networks. Within the framework, the neighbor offset synchronization method is defined as the default mandatory synchronization method. Using the neighbor offset synchronization method, a mesh STA have to maintain a timing offset value between its own time synchronization function (TSF) timer and the TSF timer of the mesh STA with which the mesh STA is synchronized. A mesh STA can initiate its TSF independently and can update the TSF timer offset based on the time-stamp received in the beacons from other mesh STAs.

9.2.5 Path Selection

IEEE 802.11s provides a mandatory default path selection and path metric, but any other approaches could be also used.

The path metric is called Airtime and reflects the amount of channel resources consumed by transmitting a 1 kB frame over a particular link, considering the data rate, overhead and transmission errors.

The default path selection, Hybrid Wireless Mesh Protocol (HWMP) combines the operations of a proactive tree oriented approach with an on demand path selection inspired from Ad-Hoc On Demand Distance Vector (AODV) protocol.

HWMP elements are the path request (PREQ), path reply (PREP), path error (PERR), and root announcement (RANN). The path metric is announced through PREQ, PREP, and RANN elements. An independent sequence number is propagated by mesh STAs to others through HWMP elements. The sequence number is used to discover stale paths and maintain loop free connections.

HWMP supports two modes of operation depending on how the mesh network is configured:

- On-demand mode: The functionality of this mode is always available, independent of whether a root mesh STA is configured in the MBSS or not. It allows mesh STAs to communicate using peer-to- peer paths.
- Proactive tree building mode: In this mode, an additional proactive tree building functionality is added to the on-demand mode. This can be performed by configuring a mesh STA as root mesh STA using either the proactive PREQ or RANN mechanism. The proactive PREQ mechanism creates paths from all the mesh STAs to the root, using only group-addressed communication. The RANN mechanism creates paths between the root and each mesh STA using acknowledged communication.

9.3 Dynamic Channel Access

IEEE 802.11s standard proposed a channel switching mechanism which moves the mesh to another frequency to avoid performance degradation due to interference or presence of radar. As all mesh STAs are supposed to work in the same channel, the access to the channel becomes competitive specially in the core of the network. Many research findings show that the capacity per node in such scenario drops significantly when the network size increases [24, 51]. In a multi-hop single channel network with all links running, the end-to-end performance suffers low throughput and unfairness [61]. Multi-channel mesh networks are able to provide significant capacity gain compared to single channel networks by placing neighboring links over different non overlapping channels [51, 11].

Assuming that mesh STAs have two or more radios, hybrid channel assignment protocols can be used to enhance IEEE 802.11s mesh networks, where at least one radio interface at each mesh STA is controlled dynamically to communicate with neighboring mesh STAs on different channels [40, 57].

We have provided an overview of some channel assignment algorithms proposed recently in Chapter 3.2 and Chapter 7. Here we compare their performance against the single channel network via simulation.

9.4 Performance Evaluation of Multi-Channel and Single Channel Mesh

We have evaluated the performance of a single channel and a multi-channel mesh network for different number of nodes placed in a grid topology and with different number of traffic flows. The traffic is generated by 150 kbps CBR flows with packet size of $L = 8000$ bits sent over vertical and horizontal directions in the grid topology. Each flow is between one node in an edge toward the other node placed at the opposite edge. For a grid of $xSize \times ySize$, the vertical flows are between (x,y) and (x+Size-1, y) while the horizontal flows are between (x,y) and (x,y+ySize-1). The simulation duration is 1000 s.

The channels receiving interference from external networks, are chosen randomly. A channel with external interference is modeled as an on-off process, such that the channel is sensed busy and idle during the on and off states, respectively. Note that when the channel is detected busy due to the external interference, nodes are not allowed to transmit during that state. The duration of the *busy* state has been fixed to a constant value (15 ms), while the duration of the *idle* state is exponentially distributed with mean equal to 8 ms. In the simulations, external interference is introduced over 4 channels chosen randomly.

Different multi-channel mesh networks are evaluated using the channel assignment protocols explained before (Section 9.3). The specific parameters of the protocols are set according to the values given in [40, 43, 57].

We consider three network performance measures:

- *Data delivery ratio:* ratio of the total amount of data which is correctly received by the destinations, to the total amount of data packets transmitted by the sources.

- *Average end-to-end delay:* mean delay of the packets to reach the destination.

- *Control overhead:* ratio of the total number of control messages sent between nodes, to the total number of correctly received packets.

9.4.1 Increasing the number of nodes

Fig. 9.3- 9.5, show the packet delivery ratio, average end to end delay and control overhead of 3 CBR (150 kbps) traffic flows for different number of nodes in grid topology. The traffic sources and destinations are the same in all simulations. Each point in the figures shows the average over 10 runs. The error bars in the figures show 95% confidence intervals.

Fig. 9.3 shows that the delivery ratio of multi-channel networks is higher than the single channel network as the number of nodes increases. Moreover, as expected, the specific channel assignment protocol has a big impact on the efficiency of the multi-channel network.

The delivery ratio in SICA is higher than in other protocols and it does not decrease as the number of nodes increases. For BFSCA, on the other hand, the delivery ratio drops fast as the number of nodes increases since the centralized protocol is not scalable enough. The single channel network performs as a multi channel network using Urban-X in terms of delivery ratio, but it results in a higher delay due to the higher congestion in the single channel.

Fig. 9.4 shows that the average end to end delay of dynamic channel assignment protocols is much lower compared to single channel. Urban-X results in a higher delay compared to other channel assignment protocols, since it keeps the transmitting radio on each channel for a predefined period of time, after switching, regardless of the amount of traffic waiting to be sent.

Fig. 9.5 shows that SICA and BFSCA lead to lower control overhead. The control overheads of the single channel network are the *Hello* messages that nodes send to their neighbors to inform them about themselves.

9.4.2 Increasing the number of traffic sources

Fig. 9.6-9.8 are obtained using a 6 × 6 grid network while the number of CBR flows of 150 kbps increases from 2 to 7. The results have been obtained averaging over 10 runs of 1000 s with different seeds. The error bars in the figures show 95% confidence intervals.

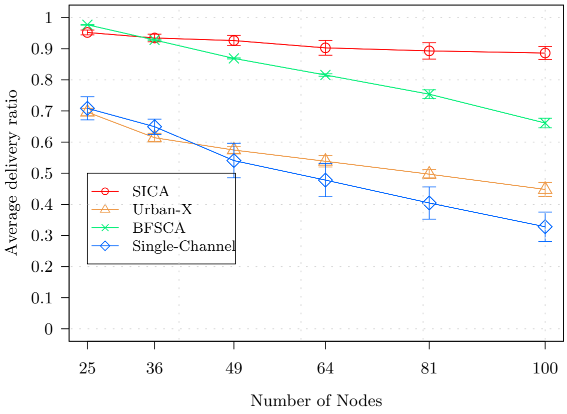


Figure 9.3: Data delivery ratio vs. number of nodes (Grid topology)

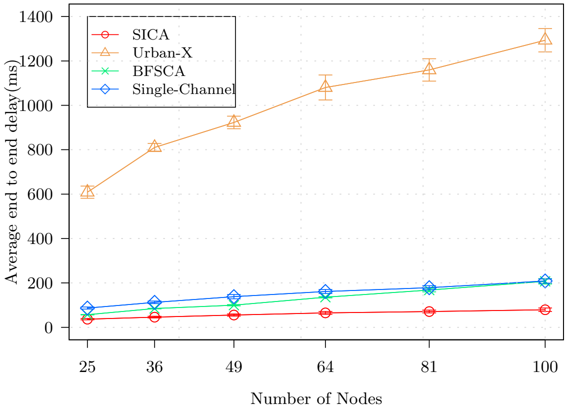


Figure 9.4: Average end to end delay vs. number of nodes (Grid topology)

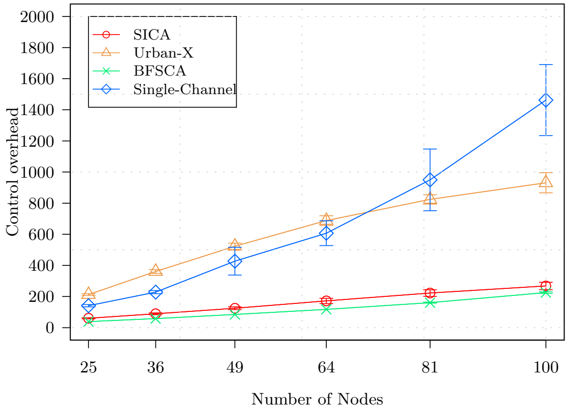


Figure 9.5: Control overhead vs. number of nodes (Grid topology)

Fig. 9.6 shows that the delivery ratio of SICA is higher than others in presence of high traffic load. The delivery ratio of BFSCA drops fast when the number of traffic flows increases, since any channel switching interrupts the data transmission and nodes are forced to deliver data packets through the common channel, which saturates it (Chapter 3.2). The delivery ratio obtained by using a single channel network is lower than multi-channel networks due to the lower capacity of a single channel.

Urban-X performs better and more robust in presence of high traffic load compared to BFSCA since it considers the traffic load for making decisions, but it results in a considerable high end-to-end delay (Fig. 9.7). Moreover, multi-channel approaches lead to a lower end to end delay compared to the single channel network, although the dynamic protocols suffer from the radio switching delay.

Fig. 9.8 shows that the control overhead of SICA and BFSCA is better than for Urban-X.

9.5 Conclusion

In this chapter we overview the main features of IEEE 802.11s -the recent standard for wireless mesh networks. In addition, we analyze some of the most recent channel assignment protocols which are potential candidates to enhance the operation of actual IEEE 802.11s-based mesh networks. Simulation results show the benefits of using Dynamic Channel Allocation mechanisms in multi-channel mesh networks compared against single channel or static channel selection approaches. The presented results show that, more attention should be directed at designing a smart channel selection mechanism and a channel aware path selection metric for the current standard.

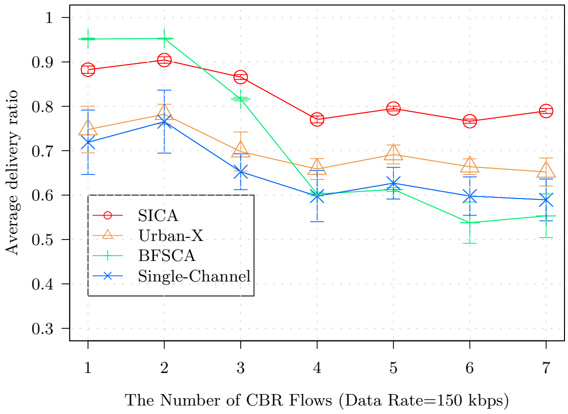


Figure 9.6: Data delivery ratio vs. number of CBR traffics

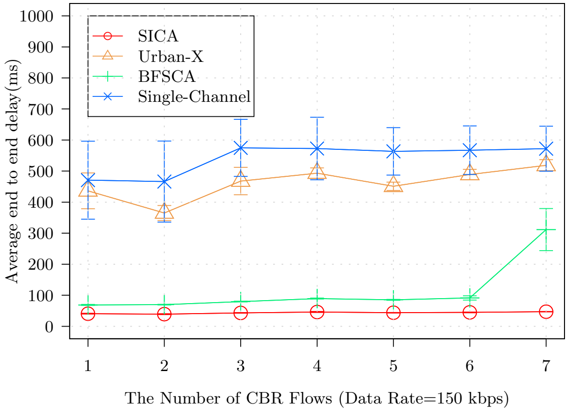


Figure 9.7: Average end to end delay vs. number of CBR traffics

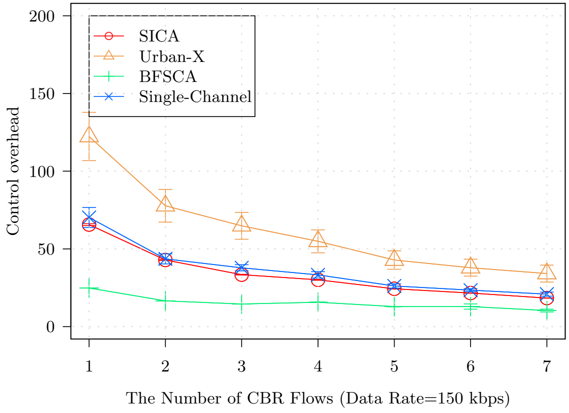


Figure 9.8: Control overhead vs. number of CBR traffics

CHAPTER 10

Conclusions and Future Research Directions

10.1 Concluding Remarks

In this research work we have studied the channel assignment problem in multi radio multi channel wireless mesh networks.

We have proposed a new centralized and static channel assignment protocol which considers the utility of the links and makes it possible to estimate the usefulness of each link regardless of the traffic profile. The channel assignment then assigns channel to links considering their utility without a tight constraint on preserving the topology. We have done a performance evaluation comparing our proposal with other relevant static channels assignment algorithms proposed in the literature. In our study, we have used a numerical tool to analyze the properties of the topology graphs, and detailed ns-2 simulations. Our numerical results demonstrate the effectiveness of our protocol in exploiting channel diversity for reducing the interference over wireless links with a small number of radios per node. Simulation results show that our approach increases the performance of the multi-radio mesh network significantly. Additionally, our protocol provides a considerable decrease in the size of collision domain and thus a significant increase in the network capacity. The ns-2 simulations proved that pruning the network from some useless links leads to a better channel utilization and thus reduces the average delay and increases in the packet delivery ratio and throughput.

To design a distributed channel assignment, we have provided a new game theory based formulation of channel assignment problem. The considered problem is formulated using a repeated game with incomplete information and it is general enough to be applicable in any wireless network. We solve the game using a multiplicative weights learning algorithm which adapts to the changes in the environment and reaches a desired solution in a limited amount of time.

We have proposed a new semi-dynamic channel assignment protocol called SICA. SICA is developed using the game theory model and the real time learning mechanism. SICA is a distributed channel assignment and assumes that nodes do not have perfect knowledge about other nodes' strategies. We have done a performance evaluation comparing SICA with other channel assignments mechanisms proposed in the literature using ns-3. Simulation results show the efficiency of SICA in assigning proper channels to radios by avoiding external interference.

To evaluate the channel assignment algorithms through simulation, we have presented the extensions which must be done in ns-3 simulator to simulate channel assignment mechanisms for multi radio wireless networks. We provided the simulation details of SICA as an example. We verified the simulation of SICA using a state space checker which checks the relevant simulation events for any violation from the feasibility conditions of channel assignment solution. The results prove the correctness of the simulation and show that SICA is capable of utilizing the channels in a fair and efficient way. Moreover we justified the results obtained by the Game theory based model of SICA, using a Markov chain model.

Finally we overview the main features of IEEE 802.11s -the recent standard for wireless mesh networks. In addition, we analyze some of the most recent channel assignment protocols which are potential candidates to enhance the operation of actual IEEE 802.11s-based mesh networks. Simulation results show the benefits of using Dynamic Channel Allocation mechanisms in multi-channel mesh networks compared against single channel or static channel selection approaches. The presented results show that, more attention should be directed at designing a smart channel selection mechanism and a channel aware path selection metric for the current standard.

10.2 Brief Future Research Plan

- To improve the channel assignment mechanism by considering the intra-channel interference between the receiving and the transmitting radio. Due to the imperfect filtering of current 802.11 based radios, receiving and transmitting antennas installed on a wireless router may interfere each other even though they are working on different non-overlapping channels. It is desirable if the channel assignment considers this fact for assigning channels to the co-located radios.

- To study the mechanism of adapting a dynamic channel allocation mechanism for IEEE 802.11s based wireless mesh networks. Although it is possible to have static and semi-dynamic CA in a mesh service set, having a hybrid CA needs more effort to be applicable in IEEE 802.11s based mesh. The mesh peering management protocol should be provided by an intermediate agent to be able to establish the connection between mesh routers which use hybrid CA.

- To study the interaction between CA and routing protocols. Because of the reactive nature of the multi channel routing metrics, the routing decision can invoke the channel assignment to re-assign channels and this will invoke the routing to change the routes. To avoid unnecessary changes in channels and routes some tuning may needs to apply to channel assignment.

Bibliography

[1] R. Aguero-Calvo and J. Perez-Campo. *Adding Multiple Interface Support in NS-2.* University of Cantabria, Cantabria, Spain, 2007. 12, 27, 83

[2] I. F. Akyildiz and Xudong Wang. A survey on wireless mesh networks. *IEEE Communications Magazine*, 43(9), 2005. 1

[3] M. Alicherry, R. Bhatia, and Li Erran Li. Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks. *IEEE Journal on Selected Areas in Communications*, 24(11):1960–1971, 2006. 4, 9, 10, 17, 91

[4] Maryam Amiri Nezhad and Llorenç Cerdà-Alabern. Utility based channel assignment mechanism for multi radio mesh networks. In *Proceedings of the 8th ACM international workshop on Mobility management and wireless access*, MobiWac '10, pages 68–74, New York, NY, USA, 2010. ACM. 4, 9, 83

[5] Alberto Pollastro Antonio Capone, Stefano Napoli. Mobimesh: An experimental platform for wireless mesh networks with mobility support. In *Proceeding of ACM QShine 2006 (WiMESHNets 2006, Workshop on "Wireless mesh: moving towards applications").* ACM, August 2006. 5, 7

[6] Stefano Avallone and Ian F. Akyildiz. A channel assignment algorithm for multi-radio wireless mesh networks. *Proceedings of 16th International Conference on Computer Communications and Networks, ICCN 2007*, 31(7):1343–1353, 2008. 4, 9, 10, 17, 83

[7] Paramvir Bahl and Ranveer Chandra. Ssch: Slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 216–230, 2004. 4

[8] H. Balakrishnan, C.L. Barrett, V.S.A. Kumar, M.V. Marathe, and S. Thite. The distance-2 matching problem and its relationship to the mac-layer capacity of ad hoc wireless networks. *IEEE Journal on Selected Areas in Communications*, 22(6):1069 – 1079, Aug. 2004. 1, 19, 29

[9] Tingting Chen and Sheng Zhong. Perfectly fair channel assignment in non-cooperative multi-radio multi-channel wireless networks. *Computer Communications*, 32(6):1058 – 1061, 2009. 11

[10] C. Cicconetti, L. Lenzini, and E. Mingozzi. Scheduling and dynamic relocation for ieee 802.11s mesh deterministic access. In *5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks. SECON '08.*, pages 19–27, 2008. Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on. 114

[11] J. Crichigno, M.Y. Wu, and W. Shu. Protocols and architectures for channel assignment in wireless mesh networks. *Ad Hoc Networks*, 6(7):1051–1077, 2008. 3, 4, 9, 83, 117

[12] Constantinos Daskalakis, Rafael Frongillo, Christos H. Papadimitriou, George Pierrakos, and Gregory Valiant. On learning algorithms for nash equilibria. In *Proceedings of the Third international conference on Algorithmic game theory. SAGT'10*, pages 114–125, 2010. 54

[13] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of the 9th ACM International Conference on Mobile Computing and Networking (MobiCom '03)*, San Diego, California, September 2003. 25

[14] Aditya Dhananjay, Hui Zhang, Jinyang Li, and Lakshminarayanan Subramanian. Practical, distributed channel assignment and routing in dual-radio mesh networks. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, pages 99–110, New York, NY, USA, 2009. ACM. 4, 9, 10, 26, 27, 83

[15] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 114–128. ACM New York, NY, USA, 2004. 4, 9, 10, 17, 26, 27, 31, 83, 92

[16] Kevin Fall and Kannan Varadhan (eds.). The ns manual (formerly ns notes and documentation), 2003. 12, 21, 26, 27, 30

[17] M. Felegyhazi, M. Cagalj, S.S. Bidokhti, and J.-P. Hubaux. Non-cooperative multi-radio channel allocation in wireless networks. In *26th IEEE International Conference on Computer Communications. IEEE INFOCOM 2007.*, pages 1442 –1450, May 2007. 4, 9, 10, 11, 47, 83

[18] Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the ninth annual conference on Com-*

putational learning theory. COLT '96, pages 325–332, New York, NY, USA, 1996. ACM. 52, 53, 54, 64, 97

[19] Yoav Freund and Robert E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, October 1999. 54, 64, 65, 97

[20] Desheng Fu, Barbara Staehle, Rastin Pries, and Dirk Staehle. On the potential of ieee 802.11s intra-mesh congestion control. In *Proceedings of the 13th ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems*, MSWIM '10, pages 299–306, New York, NY, USA, 2010. ACM. 115

[21] Lin Gao, Xinbing Wang, Youyun Xu, and Wen Chen. Distributed multi-radio channel allocation in multi-hop ad hoc networks. In *IEEE International Conference on Communications, 2008. ICC '08*, pages 3156–3160, 2008. 11, 47, 59

[22] S. Ghannay, S. M. Gammar, F. Filali, and F. Kamoun. Multi-radio multi-channel routing metrics in ieee 802.11s-based wireless mesh networks and the winner is. In *First International Conference on Communications and Networking. ComNet 2009.*, pages 1–8, 2009. Communications and Networking, 2009. ComNet 2009. First International Conference on. 111

[23] M.X. Gong, S.F. Midkiff, and S. Mao. On-demand routing and channel assignment in multi-channel mobile ad hoc networks. *Ad Hoc Networks*, 7(1):63–78, 2009. 4

[24] Piyush Gupta and P.R Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388 –404, Mar 2000. 1, 18, 63, 71, 96, 116

[25] IEEE-SA. Ieee standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications, 2007. IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999). 1, 114, 115

[26] IEEE-SA. Ieee standard for information technology telecommunications and information exchange between systems local and metropolitan area networks specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 1: Radio resource measurement of wireless lans, 2008. IEEE Std 802.11k-2008 (Amendment to IEEE Std 802.11-2007). 59, 61

[27] IEEE-SA. Ieee standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 10: Mesh networking, 2011. IEEE Std 802.11s-2011 (Amendment to IEEE Std 802.11-2007). 1, 84, 111, 112

[28] Kamal Jain, Jitendra Padhye, Venkata N. Padmanabhan, and Lili Qiu. Impact of interference on multi-hop wireless network performance. In *Proceedings of the 9th annual international conference on Mobile computing and networking. MobiCom '03.*, pages 66–80, New York, NY, USA, 2003. ACM. 18

[29] H.-K. Kim, C.-Y. Oh, and T.-J. Lee. Fair and Collision-Aware Multi-Channel Assignment Based on Game Theory for Wireless Multi-Hop Networks. *IEICE Transactions on Communications*, 92:1282–1290, 2009. 11, 71, 72

[30] Sok-Hyong Kim and Young-Joo Suh. Rate-based channel assignment algorithm for multi-channel multi-rate wireless mesh networks. In *Proceedings of the IEEE Global Telecommunications Conf. IEEE GLOBECOM 2008*, pages 1–5, 2008. 4, 9, 10, 47, 83, 93

[31] L. Kleinrock and F. Tobagi. Packet switching in radio channels: Part i–carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Transactions on Communications*, 23(12):1400–1416, 1975. 60

[32] Pradeep Kyasanur and Nitin H. Vaidya. Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 10(1):31–43, 2006. 4, 9, 10

[33] Mathieu Lacage and Thomas R. Henderson. Yet another network simulator. In *Proceeding from the 2006 workshop on ns-2: the IP network simulator*, WNS2 '06, New York, NY, USA, 2006. ACM. 90

[34] Mahesh K. Marina, Samir R. Das, and Anand Prabhu Subramanian. A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks. *Computer Networks*, 54(2):241 – 256, 2010. 4, 5, 6, 9, 10, 17, 26, 27, 28, 30, 83

[35] Jean Pierre Hubaux Mark Felegyhazi. Game theory in wireless networks: A tutorial. EPFL Technical report: LCA-REPORT-2006-2, 2 2006. 57

[36] A. H. Mohsenian-Rad and V. W. S. Wong. Joint logical topology design, interface assignment, channel allocation, and routing for multi-channel wireless mesh networks. *IEEE Transactions on Wireless Communications*, 6(12):4432–4440, 2007. 4, 9, 10, 17, 91

[37] D. Murray, M. Dixon, and T. Koziniec. Scanning delays in 802.11 networks. In *The 2007 International Conference on Next Generation Mobile Applications, Services and Technologies, 2007. NGMAST '07.*, pages 255 –260, 9 2007. 5, 63

[38] R. Musaliou-Elefteri. *Practical Wireless Mesh Networks and Their Applications.* BiblioBazaar, 2011. 7

[39] Maryam Amiri Nezhad. Semi-dynamic interference aware channel assignment (sica), February 2012. 95, 108

[40] Maryam Amiri Nezhad and Llorenç Cerdà-Alabern. Adaptive channel assignment for wireless mesh networks using game theory. In *Proceeding of the IEEE 8th Int Mobile Adhoc and Sensor Systems (MASS) Conf*, pages 746–751, 2011. 4, 9, 48, 83, 84, 95, 97, 99, 108, 117

[41] ns-3 development team. Manual: an in-depth coverage of the architecture and core of ns-3 (no. 3.9), 1 2011. 12, 59, 70, 83, 84, 85, 92, 95, 107

[42] r development core team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2008. 26, 30, 93, 101

[43] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot. Interference-aware channel assignment in multi-radio wireless mesh networks. In *Proceeding of 25th IEEE Int. Conf. Computer Communications INFOCOM 2006*, pages 1–12, 2006. 1, 4, 6, 9, 10, 17, 48, 59, 71, 72, 83, 90, 91, 98, 117

[44] Ashish Raniwala and Tzi-cker Chiueh. Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network. *Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies. INFOCOM 2005.*, 3:2223–2234, August 2005. 4, 9, 10, 17, 91, 98

[45] Ashish Raniwala, Kartik Gopalan, and Tzi-cker Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 8(2):50–65, 2004. 4, 5, 9, 17

[46] Theodore S. Rappaport. *Wireless Communications: Principles and Practice (2nd Edition)*. Prentice Hall PTR, 2 edition, January 2002. 6, 20, 27

[47] Dipankar Raychaudhuri and Mario Gerla. *Emerging Wireless Technologies and the Future Mobile Internet*. Cambridge University Press, 2011. 1

[48] Joshua Robinson, Konstantina Papagiannaki, Christophe Diot, Xingang Guo, and Lakshman Krishnamurthy. Experimenting with a multi-radio mesh networking testbed. In *In Proceedings of the First Workshop on Wireless Network Measurements. WiNMee 2005.*, 2005. 5, 83

[49] I. Shah, S. Jan, and Kok-Keong Loo. Selfish flow games in non-cooperative multi-radio multi-channel wireless mesh networks with imperfect information. In *6th International Conference on Wireless and Mobile Communications (ICWMC), 2010*, pages 219 –225, 9 2010. 11, 47

[50] Birui Shao, Jun Tao, and Fangfang Wang. Static channel assignment with the physical interference model for maximum capacity in multi-radio multi-channel wireless mesh networks. In *Proceeding of 9th Int Grid and Cooperative Computing (GCC) Conf*, pages 338–343, 2010. 4, 9, 17

[51] Weisheng Si, Selvadurai Selvakennedy, and Albert Y. Zomaya. An overview of channel assignment methods for multi-radio multi-channel wireless mesh networks. *Parallel and Distributed Computing*, 70:505–524, May 2010. 1, 3, 9, 48, 116, 117

[52] H. Skalli, S. Ghosh, S. K. Das, and L. Lenzini. Channel assignment strategies for multiradio wireless mesh networks: Issues and solutions. *IEEE Communications Magazine,*, 45(11):86–95, 2007. 4, 9, 83

[53] Ahmed Sobeih, Marcelo D'Amorim, Mahesh Viswanathan, Darko Marinov, and Jennifer C. Hou. Assertion checking in j-sim simulation models of network protocols. *Simulation*, 86:651–673, November 2010. 105

[54] A.P. Subramanian, H. Gupta, S.R. Das, and J. Cao. Minimum interference channel assignment in multiradio wireless mesh networks. *IEEE Transactions on Mobile Computing*, pages 1459–1473, 2008. 4, 6, 9, 10, 17, 26, 27, 83

[55] Rohith Dwarakanath Vallam, Arun A. Kanagasabapathy, and C. Siva Murthy. A non-cooperative game-theoretic approach to channel assignment in multi-channel multi-radio wireless networks. *Wireless Networks Journal*, 17(2):411–435, 2 2011. 47, 57

[56] Liu Weibing, Wang Xianjia, and Huang Binbin. Evolutionary markov games based on neural network. In *Proceedings of the 6th International Symposium on Neural Networks: Advances in Neural Networks - Part III*, ISNN 2009, pages 109–115, Berlin, Heidelberg, 2009. Springer-Verlag. 101

[57] Marco Di Felice Wooseong Kim, Andreas J. Kassler and Mario Gerla. Urban-x: Towards distributed channel assignment in cognitive multi-radio mesh networks. In *IFIP Wireless Days*, Venice, Italy, 2010. 4, 6, 9, 10, 59, 61, 63, 71, 72, 83, 90, 98, 117

[58] D. Wu and P. Mohapatra. From theory to practice: Evaluating static channel assignments on a wireless mesh network. In *Proceedings of the IEEE INFOCOM, 2010*, pages 1 –5, march 2010. 5, 7

[59] Xiaobing Wu, Jiangchuan Liu, and Guihai Chen. Analysis of bottleneck delay and throughput in wireless mesh networks. In *Proceedings of the IEEE Int Mobile Adhoc and Sensor Systems (MASS) Conf*, pages 765–770, 2006. 4, 9, 10, 41, 83

[60] Leiming Xu, Yong Xiang, and Meilin Shi. A novel channel assignment algorithm based on topology simplification in multi-radio wirelesss mesh networks. In *Proceedings of the 25th IEEE Int. Performance, Computing, and Communications Conf. IPCCC 2006*, 2006. 4, 9

[61] S. Xu and T. Saadawi. Does the ieee 802.11 mac protocol work well in multihop wireless ad hoc networks? *IEEE Communications Magazine*, 39(6):130–137, 2001. 117

[62] Sook-Chin Yip, Su-Wei Tan, and Teong-Chee Chuah. Data rate-aware channel assignment algorithm for multi-rate multi-channel wireless mesh networks. In *Proceedings of the 6th Int Wireless and Pervasive Computing (ISWPC) Symp*, pages 1–6, 2011. 93

[63] Li Zhang, Xinheng Wang, and Caixing Liu. Channel assignment in multi-radio multi-channel wireless mesh network by topology approach. In *Proceedings of the WRI Int. Conf. Communications and Mobile Computing CMC '09*, volume 2, pages 358–362, 2009. 4, 9

Appendices

Utility Based Channel Assignment Mechanism for Multi Radio Mesh Networks

Maryam Amiri Nezhad and Llorenç Cerdà-Alabern
Univ. Politècnica de Catalunya
Computer Architecture Dep.
Jordi Girona 1-3
E-08034, Barcelona, Spain
{maryam,llorenc}@ac.upc.edu

ABSTRACT

We address the channel assignment problem in a multi-radio mesh network that involves assigning channels to radio interfaces for eliminating the effect of wireless interference. Due to the insufficient number of frequency channels and available radios per node, interference is still present which limits the available bandwidth on wireless links and eventually decrease the achievable throughput. In this paper we investigate the effect of considering the diverse delivery probability of the wireless links on the channel assignment solutions. We show that it is possible to classify the wireless links and omit some of them to benefit from a more diverse-channel solution. We propose a new channel assignment aiming to minimize the interference over high performance links. Finally a performance study is carried to assess the effectiveness of our proposed algorithm. Evaluations show that the multi-channel network obtained from our proposed algorithm achieves significant improvement in terms of reducing the interference and increasing the network capacity.

Categories and Subject Descriptors

C.2.1 [**Computer Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Algorithms, Performance

Keywords

Channel Assignment, Mesh Network, Multi-Radio, Wireless

1. INTRODUCTION

Wireless mesh networks (WMN), based on commodity 802.11 radios, are promising solutions to provide broadband network coverage at low cost. Mesh networks however suffer from serious interference problems, limiting their capacity due to the broadcast nature of the wireless medium. A common method to improve the capacity is to use multiple orthogonal channels that are already available in 802.11 standard. The main idea is to reduce the interference by using different channels for neighboring links. A wireless router can use multiple channels if it is equipped with multiple radio interfaces (radios). Here the challenge is assigning channels to radios subject to limit the interference over the links while maintaining the network connectivity.

Many papers (and references therein) [2, 6, 10, 12, 14, 15] have been published on channel assignment problem, proposing solutions based on different criteria. However, most of the schemes disregard the delivery probability of wireless links, i.e. they suppose that all wireless links offer the same performance for data transmission. Normally, the delivery probability of a link strongly depends on its length, because the received power decreases drastically with increasing the distance. On the other hand all wireless links are not useful, in a mesh network with a gateway most of the user traffic is oriented to/from the gateway, therefore links close to the gateway are more probable to be selected by the routing protocol.

In this work, we propose a new channel assignment that takes these features into consideration and demonstrate its benefits by a performance comparison with other relevant channel assignments algorithms that have been proposed in the literature.

Our contributions that set our work apart from the existing approaches for channel assignment problem are as follows:

- We show that the topology preserving constraint (assigning channel to all available links) leads to a suboptimal solution, i.e. relaxing this constraint may improve the results considerably.
- We propose a new centralized channel assignment algorithm which is traffic independent.
- We consider both the delivery probability of the wireless links and their usefulness to make a better decision for assigning good channels to good links. Simulation results show the goodness of our approach.

The rest of the paper is organized as follows: We start by describing the network model and formulation of our problem in Section 2. In Section 3 we present our algorithm. We report the simulation results in Section 4. The related works are summarized in Section 5 and the paper is concluded in Section 6.

2. MODEL AND PROBLEM FORMULATION

2.1 Network Model

We consider a multi-radio wireless mesh network (WMN) consisting of a set of mesh routers (nodes) where some nodes serve as

MobiWac'10, October 17–18, 2010, Bodrum, Turkey.

gateways between the WMN and the wired network. We assume that each node has at least $R \geq 1$ radio interfaces (radios) and can tune each radio to one of the frequencies selected from C non overlapping channels. For simplicity, we assume that all radios have the same characteristic and transmission range.

We model the connectivity between nodes by an undirected graph $G_t = (V, E)$; henceforth referred to as the topology graph. Here V denotes the set of nodes, whereas E denotes the set of links. A pair of nodes have a link in E if they are connected in the network. We associate to every link e a weight equal to its packet delivery probability ($p_d(e)$).

The topology graph is not sufficient to fully characterize the wireless network, because the wireless links may interfere each other while transmitting simultaneously. To account the impact of interference on a transmission we use the interference protocol model defined in [8]. In this model, two transmissions links will interfere if they occur within the interference range of each other. The interference range of a link is usually supposed to be two times the transmission range.

To represent the interference among all possible transmissions in a network, the conflict graph is introduced by Jain et.all [9]. The conflict graph $G_f = (V_f, E_f)$, contains a set of vertices corresponding to all links in the network topology. There is an edge between two vertices in the conflict graph if the corresponding links interfere each other. We define the interference weight for a link e ($I(e)$), as the number of links that potentially interfere with e, consequently the interference weight of a link is equal to the degree of the corresponding vertex in the conflict graph.

Throughout this paper, we use the topology graph to model the network topology, and the conflict graph based on the protocol model for the wireless interference.

2.2 Problem Formulation

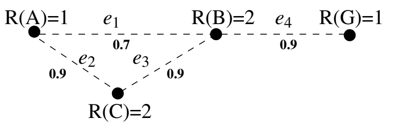
The channel assignment problem in a multi-radio wireless mesh network is to find a feasible mapping between radios and channels. A feasible mapping between radios and channels must satisfy the following conditions (feasibility conditions):

Radio constraint: The number of channels assigned to a node must be equal or less than the number of radios it has.

Connectivity constraint: The channel assignment must preserve the connectivity of the network, i.e. after assigning channels to radios, there must be at least one path between all pairs of nodes.

The aim of a channel assignment algorithm is to utilize the available channels effectively, to reduce the interference of all links as much as possible. In most of the cases, due to the limited number of radios per node and the big interference weight of wireless links, it is not possible to eliminate the interference over all links completely. Moreover in a channel assignment strategy each decision will limit the flexibility of the next decision, as we show in the following example.

Fig. 1 shows a simple network with four nodes where $R(v)$ shows the number of radios of node v. The numbers under the links show the packet delivery probability offered by each link. Consider a channel assignment algorithm which starts from node A and assigns channel c_1 to its single radio. To preserve the topology, nodes B and C must tune one of their radios to channel c_1. Therefore they lose their flexibility in making the decision for one of their radios. The algorithm finished its work by assigning channel c_2 to the other links incident on nodes B and C (Fig. 1(a)). We are interested in computing the capacity offered by this topology. We define the capacity equal to the maximum number of possible concurrent transmissions in the network [3]. Since in this network all wireless links interfere each other and are assigned to channels c_1 or c_2, at most two transmissions can occur concurrently (one on each channel). Therefore the network capacity would be $2 * 0.9 = 1.8$.



Channel assigments:

(a)	(b)
$e_1 : c_1$	e_1 : none
$e_2 : c_1$	$e_2 : c_1$
$e_3 : c_2$	$e_3 : c_2$
$e_4 : c_2$	$e_4 : c_3$

Figure 1: Channel assignment mechanism

From another standpoint, since the link between B and A is lossy (compared to other links), if the channel assignment omits this link, node B can choose two different channels for its radios, rather than c_1, and obviously we achieve a better channel utilization (Fig. 1(b)). The maximum capacity in the second topology is $3 * 0.9 = 2.7$, since all remaining links can transmit concurrently over different channels.

This example shows that omitting some lossy links may allow the channel assignment to reach a better solution.

Therefore we tackle the problem to find a channel assignment that tries to reduce the interference over good links, with a slack restriction on preserving the network topology. In order to measure the amount of interference in a channel c, we define the average of the interference weight of the links in c as:

$$F_c = \frac{1}{|E_c|} \sum_{e \in E_c} I_c(e) \tag{1}$$

where E_c is a set of links assigned to a channel c, and $I_c(e)$ is the interference weight of a link e in a channel c. So, the aim of our algorithm is keeping F_c as low as possible over all channels.

3. UTILITY BASED CHANNEL ASSIGNMENT (UBCA)

In this section we describe an algorithm (UBCA) for assigning channels to radios, which is developed based on our previous discussion. We start by defining the following terms:

Free radio: Whenever the number of channels assigned to a node, is less than the number of radios it has, the node is supposed to have some free radios.

Potential link: In a multi-channel network, the availability of a link depends on the physical distance between end point nodes and the existence of common channel between them. Therefore, we call a link as potential, if the endpoint nodes are physically neighbors but they have no common channel. Note that if a link remains potential at the end of the channel assignment, it is actually removed.

Our channel assignment algorithm (UBCA) has two phases (see Alg. 1). In the first phase, UBCA chooses the most diverse channel set for links without having tight restriction for network connectivity. In this phase, if the algorithm can not assign any common channel to the end point nodes of a link, it marks the link as potential. In the second phase, UBCA makes the final decision for potential links: It tries to make one common channel for them through merging channels over endpoint nodes, or removes them from the topology.

At the beginning of the algorithm, each link is given a priority based on its delivery probability and utility. We describe the exact criteria for determining the priority of each link at the end of this section (Section. 3.1). UBCA visits each link based on the priority order (line 1 in Alg. 1). For each visiting link, the algorithm first determines a possible set of channels, and then selects the best channel for the link among that set. To select the channel for a link, UBCA investigates the effect of adding the visiting link to all

Algorithm 1 Utility Based Channel Assignment(G_t,G_f,C)

```
  Input:
    $G_t = (V, E)$: The topology graph
    $G_f$: The conflict graph of $G_t$
    $C$: The available frequency channels
Require: $C > 0$
  Channel Assignment Phase 1:
 1: Order potential edges of $G_t$ in non-increasing order of their priority
 2: for Unvisited and Potential edge $e =< v, u >$ in order do
 3:     $PCh \leftarrow Get.Possible.Channel(e, C)$
 4:     if $PCh$ is $\varnothing$ then
 5:         Mark $e$ as "Visited and Potential"
 6:     else
 7:         $c \leftarrow Find.Best.Channel(e, PCh, G_f)$
 8:         Assign $c$ to one radio of each endpoint nodes ($v$,$u$)
 9:     end if
10: end for
  Channel Assignment Phase 2:
 1: Order all "Visited and Potential" links in $G_t$ in an increasing order of their priority
 2: for Potential link $e =< v, u >$ in order do
 3:     if Any path $P$ between $u$ and $v$ and $< u, v > \notin P$ then
 4:         Remove $e$ from $G_t$
 5:     else
 6:         Select least interfering channels from $v$ and $u$ and merge them
 7:     end if
 8: end for
```

possible channels and chooses the channel with lower interference average (See Alg. 3).

If the possible channel set for a link is empty, then UBCA marks the link as potential for the next phase. The size of the possible channel set for a link depends on the situation of its endpoint nodes (See Alg. 2). If both endpoint nodes have free radios, then it is possible to assign a new channel to the visiting link, for this case the possible channel set contains all available channels. In case that one endpoint node has no free radios, selecting a different channel from the current channel set of that node is against the first condition of the feasibility (Section 2.2), therefore the possible channel set for the visiting link is equal to the current channel set of the node which has no free radio (lines 3-5 in Alg. 2). If both end point nodes do not have any free radio, then the possible channel set must be equal to the common channel between the two nodes. Finally in the case that nodes have no free radios neither common channel, the possible channel set for the visiting link would be empty, and the link remains potential (line 9 in Alg. 2).

In the second phase UBCA visits the remaining potential links. It must decide to remove a potential link from the topology or recheck the channel assignment to make a common channel between end point nodes. A possible way to create a common channel between two nodes is to select one channel from each node's current channel set and merge them to one (assigns all links in one channel to another). For selecting two appropriate channels to merge, the algorithm must consider the interference weight of the links after merging channels. UBCA is mostly oriented to remove unnecessary links. For this reason, in this phase, it visits the potential links in an increasing order of priority, i.e the link with the lowest priority is visited first. For each potential link, it checks whether there is any other path between the two end point nodes. If so, the link is removed, otherwise the channel merging process is applied to make the link available.

Theorem 1. The proposed channel assignment algorithm satisfies the feasibility conditions.

Proof. The feasibility conditions are mentioned in Section 2.2. Here we express the proof for each condition separately.

Algorithm 2 Get.Possible.Channel($e =< u, v >$,C)

```
  Input:
    $e =< u, v >$: The link between nodes $v$ and $u$
  Output:
    $PCh$: A set of possible channels for the given link
Require: $C > 0$
 1: if $v$ and $u$ have free radio then
 2:     $PCh \leftarrow C$
 3: else if at least one node has free radio then
 4:     $Lim.Node \leftarrow$ The node which has no free radio
 5:     $PCh \leftarrow$ Current channels of the $Lim.Node$
 6: else if $v$ and $u$ have no free radio but common channels then
 7:     $PCh \leftarrow$ Common channels between $u$ and $v$
 8: else
 9:     $PCh \leftarrow \varnothing$
10: end if
11: return $PCh$
```

Algorithm 3 Find.Best.Channel($e =< u, v >$,PCh,G_f)

```
  Input:
    $e =< u, v >$: The link between nodes $v$ and $u$
    $PCh$: A set of possible channels for the given link
    $G_f$: The conflict graph
  Output:
    $C$: The best channel for the given link
 1: if $|PCh| > 1$ then
 2:     $MinF \leftarrow I(e)$
 3:     for $c \in PCh$ do
 4:         $E_c \leftarrow E_c \cup e$
 5:         Compute $F_c$ from equation (1)
 6:         if $F_c \leq MinF$ then
 7:             $MinF \leftarrow F_c$
 8:             $C \leftarrow c$
 9:         end if
10:     end for
11: else
12:     $C \leftarrow PCh[1]$
13: end if
14: return $C$
```

Radio-constraint: For each link, the best channel is selected among the possible channel set which is determined by Alg. 2. To determine the possible channel set for a link, Alg. 2 checks the radio-constraint condition for end point nodes through lines 2-8. Therefore, the possible channel set for a link is selected based on the current channel set of end point nodes and their free radios. Hence the radio constraint condition is preserved.

Connectivity-constraint: The second phase of Alg. 1, decides to remove a link under two conditions: first, there must be an alternative path between two endpoint nodes; and second, the path must be independent of the current link (see lines 3-4 of the second phase of Alg. 1). If these two conditions are held, then the algorithm removes the link. Therefore, even after deletion of a link, there is a path between two nodes, and thus the network remains connected.

3.1 Computing Links Priority

Recall that the objective of our channel assignment is to reduce the average interference weight of the links by removing unnecessary links. Thus it is necessary to visit the preferred links first. Considering only the delivery probability is not sufficient since links removal must be carried in such away that have the minimum impact on the paths to the gateway. To assess the role of each link in constructing the paths to the gateway we define the Utility metric for each link ($U(e)$). $U(e)$ will help us to estimate the probability of using a specific link. Without any traffic profile, it is pretty hard to estimate the Utility precisely, but we can have a good estimation by considering a balanced traffic over the network.

To compute $U(e)$, independent of specific traffic pattern, we consider all shortest paths between the gateway and other nodes in the network. We use the shortest path first (SPF) while considering the cost of a link equal to the inverse of its delivery probability ($1/p_d(e)$) i.e. the expected transmission count over the link, assuming Bernoulli trials. We define $U(e)$ equal to the number of times that link e participates in constructing the shortest paths between the gateway and other nodes.

In a network with $|V|$ nodes (one node is the gateway), we have $|V|-1$ paths from all nodes to the gateway. Thus, we estimate the probability of using a link for a transmission to the gateway as the utility of that link ($U(e)$) over the total number of paths ($\frac{U(e)}{|V|-1}$)

Since our channel assignment algorithm tries to prune the topology by deleting some links, it is important to start channel assignment from links with the higher utility. Through our simulation study, we found that in all topologies many links have the utility equal to zero or one. Therefore, after sorting all links based on their utility, links with the same utility must be ordered based on their delivery probability. This sorting can be done by defining the priority $P(e)$ of each link e as equation (2), where γ is a tuning parameter subject to $0 < \gamma < 1$. Big γ prefer links with higher utility while small γ is preferable for networks without any gateway. We use $\gamma = 0.9$ throughout this work. Note that $\gamma = 1$ means classifying links only considering their utility which may be equal for many links and thus results in an ambiguous classification.

$$P(e) = \gamma * \frac{U(e)}{|V|-1} + (1-\gamma) * p_d(e) \quad \forall e \in E \tag{2}$$

4. PERFORMANCE EVALUATION

In this section, we study the performance of the proposed channel assignment algorithm using the R numerical tool [16] and the NS-2 [7] simulator. We use R to compare the capacity and interference properties of different multichannel algorithms [10, 15]. Detailed NS-2 simulations are used to evaluate the performance of the channel assignment algorithms in 802.11-based multi-radio mesh networks. We have add the multi-radio functionality to the physical and MAC layer of 802.11 in NS-2 simulator based on the work done in [1]. The routing tables in the NS-2 simulations are obtained using SPF, while considering $1/p_d(e)$ of any link e as its weight.

To assess the delivery probability ($p_d(e)$) of link e we use the shadowing propagation model [13] with a path loss $\beta = 2.7$ and standard deviation $\sigma_{dB} = 6$ dB. We have used the default NS-2 values for the other propagation model parameters (see [7]). In our model we consider that links exist only when $p_d(e) \geq 0.5$. With the selected values, this delivery probability is achievable if the distance of the link is not longer than 131.5 m.

We consider a network with different number of nodes ($10 \leq N \leq 50$) which are randomly placed in a square field of 300×300 m^2.

We assume the protocol model for interference between wireless links with an interference range equal to 263.06 m (two times the communication-range). Throughout we assume that all nodes are equipped with two radios that can be tuned over 12 non-overlapping channels.

We compare UBCA with three relevant channel assignment algorithms that have been proposed in the literature: the Common Channel Assignment (CCA) [6]; the Connected Low interference Channel Assignment algorithm (CLICA) [10]; and the distributed channel assignment (ROMA) [5].

CCA applies the same channel assignment pattern for all nodes, i.e. the first radio of all nodes is tuned to the first channel, the second radio is tuned to the second channel and so on. Therefore if each node has r radio interfaces, regardless of the number of available channels, the network created by CCA always uses r channels.

CLICA is a centralized channel assignment which tries to reduce the interference weight of the links while preserving the network topology. CLICA visits the nodes based on their priority which is defined by their distance to a reference node and the number of free radios they have. Here the reference node is the gateway. While assigning a channel to a link, each end-point node will lose one of its free radios and thus during the channel assignment the priority of the nodes will change dynamically. CLICA selects a channel for a link in such a way that leads to the lowest amount of interference weight over that link and all other links which are interfering with it [10].

ROMA is a distributed algorithm proposed for a network with at least one gateway. At the beginning of the channel assignment each gateway produces a channel sequence ($c_1, c_2, \cdots, c_n$), and broadcasts it. The node which is i hops far from the gateway will select the c_{i-1} and c_i elements of the channel sequence, and tune its radios to the selected channels. Therefore, at the end of the procedure each node will have a common channel with its previous node on the path to the gateway, and a common channel with its neighbors at the same and lower level.

4.1 Topology Properties

4.1.1 Capacity-Gain

We use the maximum number of concurrent transmissions as an estimation for network one-hop capacity [3]. We calculate this metric in two steps: First, computing all independent sets in the conflict graph, and then selecting the set which gives us the maximum capacity factor, as we explain in the following.

The simplest way to determine the capacity factor ($C_f(C)$) of a multi-channel network with C orthogonal channels, is by considering the cardinality of the largest independent set S of the conflict graph $G_f(C)$ [3]. To take into account the delivery probability in the capacity metric, we calculate the summation of the delivery probability of the links in each independent set (equation (3)). We define the capacity gain of the multi-channel network in relation to the single channel network in equation (4), where $C_f(C)$ and $C_f(1)$ represent the capacity factor of a multichannel network with C orthogonal channels, and a single channel network respectively.

$$C_f(C) = \max_{\forall S \subset G_f(C)} \sum_{\forall e \in S} p_d(e) \tag{3}$$

$$\text{Capacity Gain} = \frac{C_f(C)}{C_f(1)} \tag{4}$$

4.1.2 Network Interference

We use two metrics to show the interference characteristic in a multi-channel network: the collision domain size, and the link conflict weight.

Size of collision domains: A collision-domain is a subset of links in which all links collide each other if they transmit simultaneously. A collision-domain in the conflict graph is a complete subgraph or clique of vertices. All vertices in a clique are connected pairwise, therefore the set of their corresponding links in the topology graph make a potential collision domain.

Maximum average interference weight: The interference weight for a link is the number of links in its interference range (Section 2.2). We calculate this metric taking the maximum of equation (1) over C. This metric is important since a link with the maximum interference weight could be a potential bottleneck for the network.

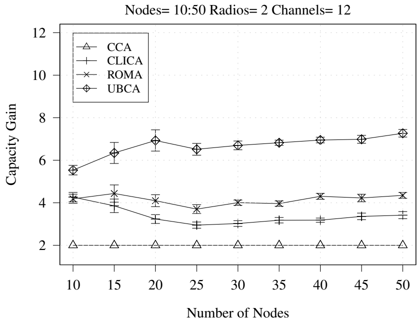


Figure 2: Capacity Gain.

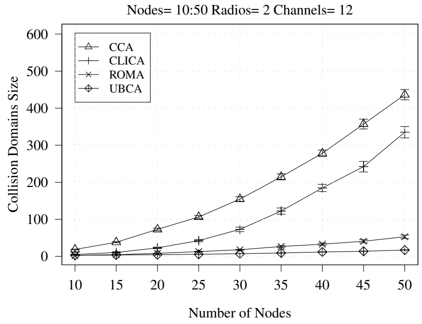


Figure 3: Size of the collision domains.

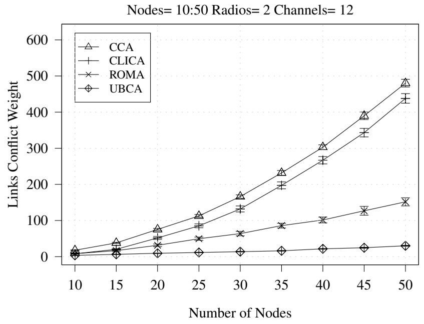


Figure 4: Maximum interference weight.

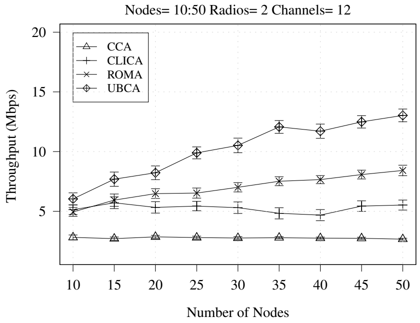


Figure 5: Aggregate throughput for TCP traffic (random profile).

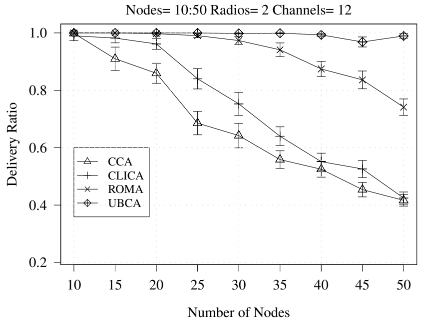


Figure 6: Average packet delivery ratio for CBR traffic (random profile).

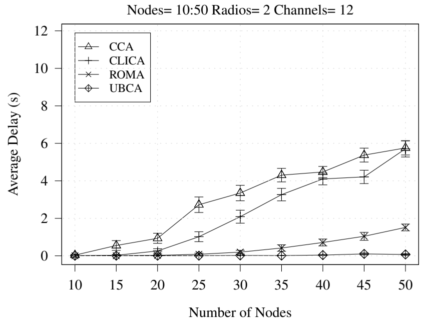


Figure 7: Average delay for CBR traffic (random profile).

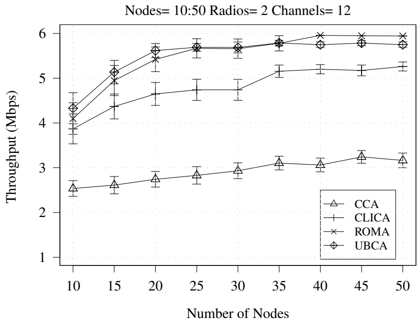


Figure 8: Aggregate throughput for TCP traffic (gateway profile).

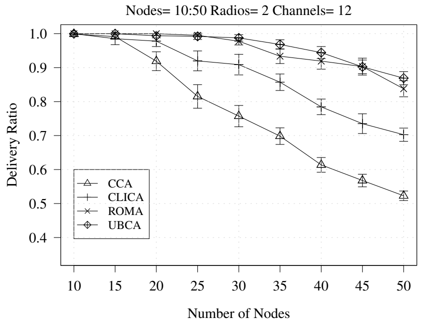


Figure 9: Average packet delivery ratio for CBR traffic (gateway profile).

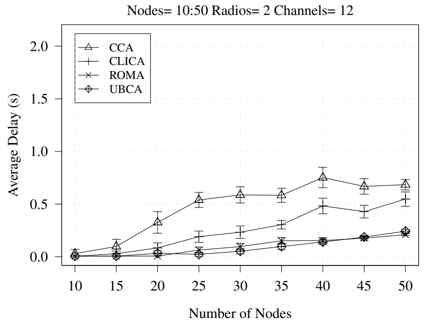


Figure 10: Average delay for CBR traffic (gateway profile).

In CLICA [10] the authors use a similar metric as the objective function.

4.2 802.11 Based Multi-Radio Performance

We use NS-2 simulator to evaluate multi-radio networks created by different channel assignment algorithms in terms of aggregate throughput, packet delivery ratio, and average delay. In each network we produce random traffic flows. The number of flows is equal to the 40% of the number of nodes in each topology. We use two type of sources: CBR traffic with fixed rate at 100 kbps and packet size equal to 1 kB; and TCP traffic with packet size equal to 1.4 kB. We consider two traffic profiles: *gateway profile* consisting of flows from the gateway to randomly selected nodes; and *random profile* consisting of flows between random pairs of nodes. The simulation time was set to 100 s. RTS/CTS mechanism is enabled. For each topology with different number of nodes, we repeated the simulation for 50 different random placement of the nodes, and report the average with the confidence intervals.

Aggregate Throughput: For TCP traffic we calculate the aggregate throughput (Mbps), dividing the total received traffic by the duration time of TCP flows.

Packet Delivery Ratio (PDR): We consider the number of correctly received packets with respect to the amount of sent packets.

Average Delay: For all received CBR packets, we calculate the average delay to verify the ability of the network to use non interfering channels to deliver data with less contention.

4.3 Results

As explained before, Fig. 2-4 have been computed with the R numerical tool [16], analyzing the properties of the topology graphs obtained by the channels assignment algorithms under study. Note that these properties do not reflect the fact of having a gateway. Fig. 8-7, on the other hand, show the results of analyzing the traces obtained using the NS-2 simulator [7].

Fig. 2 shows the capacity gain of multi-channel networks (equation (4)). The results are produced by different channel assignment algorithms varying the number of nodes. Using CCA the capacity gain is bounded to two, since CCA only uses two channels throughout the network [6]. The figure shows that our proposal, UBCA,

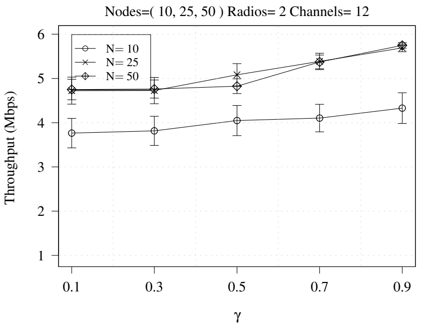


Figure 11: Aggregate throughput for TCP traffic using UBCA varying γ (gateway profile).

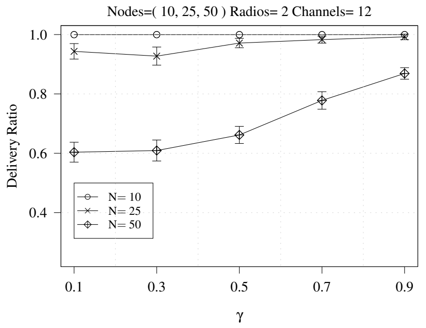


Figure 12: Average packet delivery ratio for CBR traffic using UBCA varying γ (gateway profile).

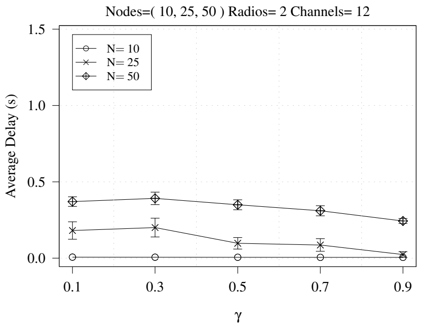


Figure 13: Average delay for CBR traffic using UBCA varying γ (gateway profile).

outperforms the other mechanisms. This is because two main reasons: First, considering the delivery probability of the links during the channel assignment; and second, removing some useless links from the original network. As expected, removing useless links will reduce the collision domain size, thus resulting a considerable increase in capacity gain.

Fig. 3 shows the average size of the collision domains for different number of nodes. CLICA is successful in reducing the size of collision domains compared to CCA, but the reduction is much higher with UBCA and ROMA. Note that, even after increasing the network density, the size of collision domains do not change much for the topologies created by UBCA and ROMA.

Fig. 4 depicts the maximum interference weight of the links for different network densities. The figure shows that by increasing the network density, the maximum interference weight do not change significantly for UBCA, while for the other mechanisms increases rapidly. Although CLICA is designed to minimize this metric, it couldn't achieve a reasonable result compared to UBCA since it tries to preserve the topology and as explained in section 2.2,this approach couldn't achieve a diverse-channel assignment hence unsuccessful in eliminating the interference over links.

To investigate the performance of the proposed channel assignment in a general situation, we first run the simulation for the random profile. Fig. 5 depicts the aggregate TCP throughput with different number of nodes. The figure shows that due to a significant increase in the network capacity (see Fig. 2), UBCA outperforms the other mechanisms specially in dense topologies.

Fig. 6 shows that the packet delivery ratio in network topologies created by UBCA is also much better than the others. Moreover, in contrast to other mechanisms, delivery ratio in UBCA is rather insensitive to the network density.

Fig. 7 shows that by using UBCA, the average delay for CBR traffic is significantly lower than with the other algorithms. The out-performance of UBCA confirms its smaller size of collision domains, and lower interference weight obtained in Fig. 3 and Fig. 4 respectively.

We re-run the simulation by considering the gateway profile in Figs. 8-9. Fig. 8 shows the aggregate TCP throughput with different number of nodes. As expected, due to the proper channel assignment in CLICA, ROMA and UBCA compared to CCA, the aggregate throughput improves significantly. UBCA get almost the same result as ROMA. Recall that ROMA is designed to optimize the gateway paths.

Fig. 9 shows that the packet delivery ratio in network topologies created by UBCA and ROMA is much higher than the others. Additionally, in UBCA and ROMA the rate of decrease in the packet delivery ratio with respect to the increase of the network density, is much lower than the others. Finally, Fig. 10 shows that in UBCA and ROMA, the average delay for CBR traffic is lower than the other algorithms. These results are justifiable by taking into account the fact that ROMA optimizes the paths to the gateway, and UBCA estimates the utility of the links based on their frequency to access the gateway.

We conclude that our simulation results show that UBCA builds a network topology with low interference with a small number of radios per node. We showed that removing some links from the network topology will lead to a better performance, even if it may cause an increase in the length of some paths between nodes.

4.3.1 The Impact of γ

Recall that UBCA uses a weighted metric for assigning the priority to wireless links before channel assignment (Section 3.1). As we have explained previously, the priority metric is a weighted sum of the utility and the delivery probability of the wireless links. The weight given to the utility (U) is denoted by γ, while the weight given to the delivery probability (p_d) is $1-\gamma$.

To investigate the impact of γ on the performance of UBCA, we repeated the simulation with $0.1 \leq \gamma \leq 0.9$ for three different network densities ($N = 10, 25, 50$).

Fig. 11 shows that by increasing γ, UBCA achieves better network throughput for TCP traffic generated in the gateway profile. Obviously the impact of different value of γ is more significant for dense networks ($N = 25, 50$), because big γ leads to decrease the interference over the links which are close to the gateway.

Fig. 12- 13 show that a larger value of γ leads to better packet delivery ratio with less delay for CBR traffic.

These simulation confirms that, since a large γ forces the channel assignment to favor the links with higher utility, for the gateway profile UBCA achieves better performance with increasing γ. For the random profile we have not obtained significant differences for different values of γ.

5. RELATED WORK

The static channel assignment problem is well studied in recent years and has been addressed in several proposals [2, 4–6, 10–12, 14, 15]. The core idea of all proposed algorithms is to use the available channels to eliminate the interference of neighboring transmissions. A simple approach for utilizing two channels in a dual-radio network is presented in [6], while the main focus of authors is on modifying the routing parameters to benefit from multi-channel structure. Further investigations in [10] and [12] show that it is possible to increase the performance of the multi-channel network more than two factors by applying smart channel assignment algorithms. Raniwala et. all [11, 12] presented a centralized (and also a distributed) traffic aware channel assignment. Given the network topology and the traffic profile, the channel assignment binds each radio to a channel such that the available bandwidth on each link is proportional to its expected load. If the traffic loads change over time the algorithm must perform channel reassignment, thus using traffic aware schemes in presence of a dynamic traffic profile is very challenging. Marina et. all [10] and Subramanian et. all [15], formulated the channel assignment problem as a topology control problem, and develop their approaches subject to minimize the link conflict weight. Avallone et. all [2] formulated the problem to acquire the smaller size of collision domains. However none of these works consider the delivery probability of the wireless links. Recently Dhananjay et.all [5] proposed a distributed channel assignment and routing which takes the links delivery ratio into accounts. In the proposed algorithm each node follows the channel assignment pattern which is propagated by the gateway i.e. the algorithm optimizes the paths to the gateway, thus the big size of collision domains specially in a dens network may lead to a lower performance in an arbitrary traffic profile.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have studied the channel assignment problem in a multi-radio multi-channel mesh network. We have showed that considering the utility of the links makes it possible to estimate the usefulness of each link regardless of the traffic profile. We have presented a new algorithm, called UBCA, which assigns channel to links considering their utility without a tight constraint on preserving the topology. We have done a performance evaluation comparing UBCA with other relevant channels assignment algorithms proposed in the literature. In our study we have used a numerical tool to analyze the properties of the topology graphs, and detailed NS-2 simulations. Our numerical results demonstrate the effectiveness of UBCA in exploiting channel diversity for reducing the interference over wireless links with a small number of radios per node. The simulation results show that our approach increases the performance of the multi-radio mesh network significantly. Additionally, UBCA provides a considerable decrease in the size of collision domain and thus significant increase in the network capacity. The NS-2 simulations proved that pruning the network from some useless links leads to a better channel utilization and thus reduces the average delay and increases in the packet delivery ratio and throughput.

Our future work will focus on formulating the proposed channel assignment problem to obtain the optimal bound for network capacity and interference. We will seek for an optimal solution while considering the delivery probability of wireless links and relaxing the constraint on topology preserving. We will also investigate the distributed version of UBCA.

7. ACKNOWLEDGMENTS

This work was supported by the Ministry of Education of Spain under grants TSI2007-66869-C02-01 and TSI2007-66869-C02-02, and by the European NoE EuroNF.

8. REFERENCES

[1] R. Aguero-Calvo and J. Perez-Campo. *Adding Multiple Interface Support in NS-2*. University of Cantabria, Cantabria, Spain, 2007.

[2] S. Avallone and I. F. Akyildiz. A channel assignment algorithm for multi-radio wireless mesh networks. *Comput. Commun.*, 31(7):1343–1353, 2008.

[3] H. Balakrishnan, C. Barrett, V. Kumar, M. Marathe, and S. Thite. The distance-2 matching problem and its relationship to the mac-layer capacity of ad hoc wireless networks. *Selected Areas in Communications, IEEE Journal on*, 22(6):1069 – 1079, Aug. 2004.

[4] J. Crichigno, M. Wu, and W. Shu. Protocols and architectures for channel assignment in wireless mesh networks. *Ad Hoc Networks*, 6(7):1051–1077, 2008.

[5] A. Dhananjay, H. Zhang, J. Li, and L. Subramanian. Practical, distributed channel assignment and routing in dual-radio mesh networks. In *SIGCOMM '09: Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, pages 99–110, New York, NY, USA, 2009. ACM.

[6] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 114–128. ACM New York, NY, USA, 2004.

[7] K. Fall and K. V. (eds.). The ns manual (formerly ns notes and documentation), 2003.

[8] P. Gupta and P. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388 –404, Mar 2000.

[9] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 66–80, New York, NY, USA, 2003. ACM.

[10] M. K. Marina, S. R. Das, and A. P. Subramanian. A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks. *Computer Networks*, 54(2):241 – 256, 2010.

[11] A. Raniwala and T.-c. Chiueh. Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network. *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, 3:2223–2234, August 2005.

[12] A. Raniwala, K. Gopalan, and T.-c. Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 8(2):50–65, 2004.

[13] T. S. Rappaport. *Wireless Communications: Principles and Practice (2nd Edition)*. Prentice Hall PTR, 2 edition, January 2002.

[14] H. Skalli, S. Ghosh, S. K. Das, and L. Lenzini. Channel assignment strategies for multiradio wireless mesh networks: Issues and solutions. *Communications Magazine, IEEE*, 45(11):86–95, 2007.

[15] A. Subramanian, H. Gupta, S. Das, and J. Cao. Minimum interference channel assignment in multiradio wireless mesh networks. *IEEE Transactions on Mobile Computing*, pages 1459–1473, 2008.

[16] R. D. C. Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008.

Adaptive Channel Assignment for Wireless Mesh Networks Using Game Theory

Maryam Amiri Nezhad and Llorenç Cerdà-Alabern
Department of Computer Architecture
Universitat Politecnica de Catalunya, Barcelona, Spain
E-mail: maryam@ac.upc.edu, llorenc@ac.upc.edu

***Abstract*—Channel assignment has been extensively researched for multi-radio wireless mesh networks, but it is still very challenging when it comes to its implementation. In this paper we propose a semi-dynamic and distributed channel assignment mechanism called SICA that uses game theory. To the best of our knowledge this is the first game formulation of channel assignment which takes the co-channel interference into account. SICA is an interference aware, distributed channel assignment which preserves the network connectivity without relying on a common channel nor central node for coordination between mesh routers. SICA applies an on-line learner algorithm which assumes that nodes do not have perfect information. We have simulated SICA and compared against another interference-aware channel assignment mechanism proposed in the literature called Urban-X. Simulation results show that SICA outperforms Urban-X, even using fewer radio interfaces per node.**

***Keywords*-Channel Assignment; Multi-Radio; Multi-Channel; Wireless Mesh Network; Game Theory; Online Learning;**

I. INTRODUCTION

Multiple-antenna technologies are well known to offer significant improvement in capacity through the use of multiple frequencies offered in IEEE 802.11 standards. The network capacity can be further enhanced if the network employs an intelligent channel assignment which seeks a proper mapping between the available channels and the radios at every node.

Many channel assignment approaches fall under the *static* category, where mesh nodes tune an antenna to a specific channel permanently (see [1], [2]). Due to the variable nature of the wireless medium, the channel assignment mechanism must be flexible enough to adapt to the erratic traffic or interference pattern. Static CAs are unable to cope with the external interference but they can easily be extended to *semi-dynamic* by refreshing the channel assignment at regular time intervals in response to the changes in the traffic pattern or co-channel interference (see e.g. [3]–[5]).

In this work we propose SICA, a semi-dynamic interference aware channel assignment algorithm for IEEE 802.11 based WMN. We estimate the amount of interference over channels, induced by any wireless enabled devices, based on IEEE 802.11k standard. We then use game theory to formulate the problem. Unlike previous game formulation in the literature (see e.g. [5]–[7]), we assume a more realistic scenario where nodes do not have perfect information about others strategies and channels suffer external interference from neighboring networks. Then we apply the online learning method to design a distributed algorithm which tries to assign the best channel to each radio. The nodes continuously refine their decision accounting the changes in the wireless environment. The main contributions that sets our work apart from others are the following ones:

- A novel game theory formulation of channel assignment, considering external and internal interference.
- A decision making strategy which assumes imperfect information at each router but adapts fast to the changes in the wireless environment.
- A fully distributed CA algorithm which preserves the network connectivity and supports any routing protocol.
- A self-contained protocol which applies channel load estimation, interface switching, control message exchange and data delivery mechanisms in addition to channel assignment.

We evaluate SICA through simulations using ns-3, and compare it with another distributed and interference aware channel assignment mechanism that has been proposed in the literature. Results demonstrate the effectiveness of SICA in exploiting channel diversity, hence reducing the interference over wireless links, even with a small number of radios per node.

II. SICA ARCHITECTURE

We introduce a multi-radio multi-channel architecture which reduces the impact of the wireless interference, and improves the performance of the network by driving the benefits of non-overlapping channels. Channel assignment is viewed as a lower layer mechanism which doesn't consider the traffic load. Our goal is to reduce the effect of the interference inside the mesh network and with any other co-located wireless networks. The distributed multi-channel architecture considers the channel selection mechanism, describes the switching process of the antennas and controls data buffering and transmitting. Nodes use a distributed algorithm to occupy the best channel based on the information gathered during the channel sensing periods.

We shall describe SICA with nodes equipped with 2 radio interfaces, each provided with a set C (with cardinality $|C| > 1$) of non overlapping channels. However, SICA could be easily extended to a network where nodes are equipped with a number of radios larger than 2. The radios will be referred to as the *receiving radio* and the *transmitting radio*, and denoted by R and T, respectively.

The distributed channel assignment selects and assigns a channel to the R radio of each node. Then, nodes switch the T radio accordingly. For example, if a generic node A tunes its R radio over channel $c \in C$, each neighboring node, which aims to send traffic to A, will switch its T radio to channel c before start transmission. The T radio remains on channel c until all the packets addressed to node A have been sent, or until a maximum period of time (T_{max}).

In the following sections we explain the details of SICA. We explain the channel sensing mechanism, CA algorithm and its implementations in sections III, IV and V, respectively. The synchronization and switching of R and T radios are explained in sectionVI.

III. EXTERNAL INTERFERENCE ESTIMATION

To estimate the amount of external interference, mesh nodes use the clear channel assessment (CCA) mechanism for spectrum sensing [8]. CCA is based on energy detection during a specific period of time. At a given time all nodes on the same channel stop transmission and start sensing the channel, the synchronization is achieved through sending messages (see section VI). Since all nodes working on the same channel must remain silent during listening to the carrier, a big sensing period will degrade the network throughput. On the other hand, a long enough sensing period is necessary to have a precise estimation [4]

During the sensing period (T_{SS}) every node monitors the channel by taking samples at the sense rate (T_{SRate}). The channel status would be monitored as either *idle* or *busy*. Define $T_{i,busy}(c)$, the time that a channel is sensed as busy during the sensing period. On the contrary $T_{i,idle}(c)$ shows the amount of time that the channel is sensed as idle. IEEE 802.11k standard for radio resources measurement proposes a simple formulation to compute the channel load as the percentage of time that the node sensed the medium as busy. At the end of sensing period node i estimates the *normalized bandwidth* (or duty cycle) consumed by external networks over a channel c, as:

$$B_{i,neig}(c) = \frac{T_{i,busy}(c)}{T_{i,busy}(c) + T_{i,idle}(c)} \tag{1}$$

The mesh nodes then use the channel load to make decision in channel assignment algorithm (see Section IV).

IV. CHANNEL ASSIGNMENT ALGORITHM

We have used a game theory model to formulate the distributed channel assignment, which is adaptive to the external interference. In our model each node is a rational player which tries to occupy the best channel for its R radio. The best channel is a channel which suffers less external interference and it is not shared by many neighboring nodes of the same network. From this point forward we use the terms node and player interchangeably.

Let N be the number of nodes of the network, and $f_{i,c}$ the number of R radios of player i using channel c ($f_{i,c} \in \{0,1\}$). Define the strategy of player i, s_i, as its channel allocation vector, given by $s_i = (f_{i,1}, f_{i,2}, ..., f_{i,|C|})$, $i = 1, \cdots N$.

A player strategy describes whether it has a radio over a specific channel or not. Note that the total number of R radios employed by player i is given by $f_i = \sum_{k=1}^{|C|} f_{i,k}$. Since only one R radio is used, $f_i = 1$. We define the strategy matrix (strategy profile), S, as the strategy vector of all players at a given time: $S = [s_1^T \; s_2^T \; ... \; s_N^T]$. By S_{-i} we shall refer to the strategy matrix consisting of all nodes' strategies except player i. Note that the node may not know S_{-i} completely.

We formulate a game theory model where each player i chooses a channel c trying to minimize a *loss function*. Each mesh router derives two separate costs for selecting a channel. The first cost is according to the channel load estimated in Section III (equation (1)). The second cost is according to internal interference induced from neighboring nodes. To estimate the internal interference over a channel, mesh routers compute how congested is the channel in the neighborhood. Let N_i be the number of nodes in the interference range of node i (two-hops neighbors based on interference protocol model [9]). We shall represent by $R_i(c)$ the number of nodes in the set N_i that have tuned their R radio to channel c at a given time:

$$R_i(c) = \sum_{k \in N_i} f_{k,c} \tag{2}$$

We define the density of interfering nodes over channel c by $\frac{R_i(c)}{N_i}$. The mesh router then merges the costs by taking the average of the individual cost as *bandwidth loss function*:

$$M_{i,B}(c, S_{-i}) = \frac{1}{2}\left(B_{i,neig}(c) + \frac{R_i(c)}{N_i}\right) \tag{3}$$

However, the cost of one node's decision depends not only on the available bandwidth of the selected channel, but also the switching delay penalty. According to [4], [10] current 802.11 commodities suffer a considerable switching delay (D_s) varying from 80 μs to 22 ms. A big switching delay affects the performance of the protocol if the radio switches frequently. We consider the magnitude of the switching delay related to the *Hello* interval, T_H (explained in Section VI). If the hello interval is big enough the effect of the switching delay is negligible. On the other hand a considerable switching delay should give a higher cost, making nodes to switch between channels less frequently.

Let c_i the channel being used by node i, we assume that a *switching delay loss function* for any channel c, is given by:

$$M_{i,D}(c, S_{-i}) = \begin{cases} \frac{D_s}{T_H}, & c \neq c_i \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Finally, we combine bandwidth and switching delay costs in the loss function given by:

$$M_i(c, S_{-i}) = \gamma M_{i,B}(c, S_{-i}) + (1-\gamma) M_{i,D}(c, S_{-i}) \tag{5}$$

Note that, γ is a tuning parameter ($\gamma \in [0,1]$), and the loss function co-domain is $[0,1]$. It is not feasible nor necessary for a player to compute $M_i(c, S_{-i})$ for all possible values of S_{-i}. Each player computes the *loss* value for one strategy profile at a time, in Section IV-A we explain how this method solves the game effectively.

To sum up, we defined a game with the following properties:

- Nodes are rational players and try to occupy the most vacant frequency channels.
- Nodes do not have knowledge about their neighbors criteria of making decision beforehand.
- Each channel decision imposes a cost (in the range of 0 to 1) to a node, as a function of the switching delay and the available bandwidth over the selected channel.
- The game is played in several rounds and the external parameters introduced by the environment may differ in each round, i.e. the environment is unpredictable.

A. Solving the game

Due to the changes in the co-channel interference, the game outlined in the previous section has no deterministic loss matrix, therefore using common approaches to solve the game is impossible. Our solution is based on the online learning approach proposed by Freund and Schapire in [11].

Let M_i be the loss matrix of node i, i.e. the rows of M_i are the strategies of node i (the channels $c \in C$ it can choose), and the columns are all possible strategies of the other players, S_{-i}. Each node assigns non-negative weights ($w_i(c)$) to the rows of M_i. We assume that, the number of rows in M_i is the same for all nodes and equal to the number of orthogonal channels ($|C|$).

Initially M_i is unknown to player i, but this game can be played repeatedly in a sequence of *game rounds* $(1, \cdots T)$. To avoid channel oscillation in each round t ($t \in 1, \cdots T$), the player plays a mixed strategy based on the weights ($w_{i,t}(c)$) assigned to the rows of M_i. The probability of selecting channel c, is calculated as:

$$P_{i,t}(c) = \frac{w_{i,t}(c)}{\sum_{c \in C} w_{i,t}(c)} \tag{6}$$

At the beginning all weights are set to 1, thus, the probability of selecting any channel is identical. After selecting a channel, the node gathers information from its neighbors and updates the loss that is suffered (equation (5)). Then, the weights are updated as:

$$w_{i,t}(c) = w_{i,t-1}(c)\,\beta^{M_i(c,S_{-i})} \tag{7}$$

where β is the game parameter in the range of $(0,1)$. A big β introduces minor changes to the weights and the learner follows the environment slowly but more accurately. Therefore it is applicable to a scenario where the environment changes less frequently. On the contrary, a small β imposes big changes in the weights, and introduces bigger error to the decision but adequate to a scenario with frequent changes. In our simulations we found that $\beta = 0.2$ leads to better results (see Section VII). We use the same β for all players. Note that the best solution reached by the learner is not necessarily the Nash Equilibrium. Since it has been shown that multiplicative weights updates learning algorithm cannot work for Nash Equilibrium in general bi-matrix games [12].

V. CHANNEL ASSIGNMENT IMPLEMENTATION

Alg. 1 summarizes the implementation of the channel assignment previously described. Recall that the main idea of SICA is using available information on each node, gathered from its neighbors, and selecting the best channel by playing a game with mixed strategies. As explained in section IV-A, the game is played in rounds that we shall refer to as *channel assignment periods*, and represent its duration by T_{CA}. Each node i runs Alg. 1 at every T_{CA}.

Algorithm 1 SICA(N_i)

```
Input:
  $N_i$: set of one and two-hops neighbors of node $i$.
if this is the first assignment then
    Set $w_{i,t}(c) = 1 \ \forall c \in C$
    Assign a random channel $c_i$ to the $R$ radio
else
    Compute $P_{i,t}(c) \ \forall c \in C$ (Eq. (6))
    Assign channel $c_i$ to the $R$ radio with probability $P_{i,t}$
end if
Switch the $R$ radio to channel $c_i$
Use $CCA$ and estimate $B_{i,neig}(c)$ (Eq. (1))
Inform other neighbors about $B_{i,neig}(c)$
for $c \in \{$channels used by $R$ radio of $N_i$ nodes$\}$ do
    Calculate $M_i(c, S_{-i})$ (Eq. (5))
    Update $w_{i,t}(c)$ (Eq. (7))
end for
```

Four main reasons call for SICA to be an efficient channel assignment algorithm:

- Nodes are not required to have the perfect information about other players' strategies and loss functions.
- Nodes are supposed to be selfish players trying to occupy the best channels.
- It is not necessary for a node to estimate the external interference over all channels. In our proposal each node senses the channel of its R radio and uses the information of its neighbors about other channels.
- The proposed channel assignment eliminates the *Channel Oscillation* problem. This problem happens when some nodes find a channel empty and try to occupy it simultaneously, finally they will switch back when they find it busy by others. Playing a mixed strategy, as previously described, avoids channel oscillation since each node selects the destination channel randomly with a predefined probability.

Param.	Possible Value	Default Value
T_H	10 ms - 100 ms	20 ms
T_{CA}	$T_H \ll T_{CA}$	5 s
T_{SS}	$T_H < T_{SS} < T_{CA}$	1 s
T_{SRate}	$T_{SRate} \ll T_{SS}$	1 ms
β	$0 \leq \beta < 1$	0.2
γ	$0 \leq \gamma \leq 1$	0.8
D_s	80 μs - 20 ms	300 μs

Table I: Channel Assignment Parameters

VI. MESH NODES SYNCHRONIZATION MECHANISM

Unlike Urban-X and many other CAs in the literature, in SICA there is no common channel between all nodes but the synchronization is achieved through exchanging messages. Since each node can assign a different channel to its receiving (*R*) radio, the network topology may appear to be partitioned. To avoid network partitioning, nodes must be aware of the channels used by their neighbors' *R* radios. A node broadcasts *Hello* messages to report the channel of its *R* radio to its neighbors.

It is not necessary to send *Hello* messages over all available channels, except when a new node joins the network or when a node stops receiving *Hello* messages from any neighbors. Once a node knows the channels used by the *R* radios of its one-hop neighbors, then the node switches the *T* radio to those channels and sends *Hello* messages every specific period of time (T_H).

After gathering information from the neighbors, a node may start transmitting data. One node may have packets to deliver to different neighbors on different channels. In our model each channel is associated with a queue. Packets are added to the corresponding queue according to the receiving channel of the neighbors. When the *T* radio switches to each channel, it sends all or some of the packets in the associated queue. We use a different queue for *Hello* messages, which has higher priority than data packets' queue.

A mesh node uses Round Robin for visiting all channels for which it has data to sent. To avoid starvation, after switching, the *T* radio will stay in one channel for at most a specific period of time (T_{max}). We assume that the switching delay of the *T* radio is constant and equal to D_s. Therefore to have the same opportunity to transmit over $|C_i|$ channels, the node computes T_{max} as $\frac{T_H-(|C_i|-1)D_s}{|C_i|}$.

Fig. 1 shows an example describing the data delivery process over two channels. For simplicity we merged the data and *Hello* packets in the same queue (shaded and black boxes, respectively), but *Hello* messages have higher priority than data. Fig. 1-a shows the channel status before T_H starts. The mechanism starts from channel one (Fig. 1-b). After switching the *T* radio to a channel, a node must wait at least for D_t units of time before it starts transmitting, to avoid collision with any ongoing transmission on the channel (see section VI-B). After D_t, the node sends packets during at most T_{max}, then it switches the *T* radio to channel two and the transmission process is repeated for this channel (Fig. 1-b). While there are packets in the queues, the node will round robin among them until the end of T_H (Fig. 1-c).

A. Switching the R radio

When a node decides to switch the *R* radio to any channel, it must announce the switching attempt through *Hello* messages. The switching attempt information in a *Hello* message, consists two fields: the destination channel to which the *R* radio will switch; and the switching timer (T_S).

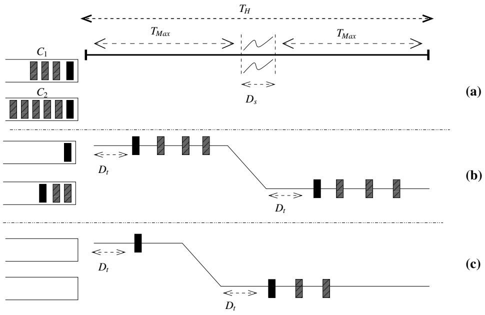


Figure 1: Data delivery mechanism

The switching timer (T_S) is the units of time before the *R* radio will switch to the new channel. This timer must be longer than the *Hello* interval ($2T_H < T_S \ll T_{CA}$) to make sure that all neighbors are informed. The timer is used to inform the neighbors about the channel switching attempt before the switching process is completed. Therefore the neighbors will consider it for upcoming transmissions.

If a node misses any information about a switching attempt of a neighbor, the node would fail to send packets to it. The algorithm tries to prevent this by selecting a sufficiently large T_S. Moreover the node always gets information about a lost neighbor from other common neighbors, thus, updating its information.

B. Switching The T Radio

The *T* radio switches more often than the *R*. Each node checks all queues sequentially and if there is any data waiting for transmission, it switches the *T* radio to the corresponding channel and starts sending data after a delay (D_t). When a node switches to a new channel it may fail to hear the previous CTS/RTS between any other nodes on the same channel, thus it must avoid transmitting immediately to prevent the collision with ongoing transmissions. Consequently after switching, a node may wait for D_t before starting any transmission.

The node remains on the target channel until the end of the transmission or at most for T_{max}. Then it proceeds on checking other queues.

VII. PERFORMANCE EVALUATION

In this section, we study the performance of the proposed channel assignment algorithm using ns-3 simulator [13] for 802.11-based multi-radio mesh networks. We use a network where the mesh routers initialize their routing tables using Shortest Path First (SPF), minimizing the number of hops. We assume a two ray ground propagation model with a radio range of 250 m. Wireless nodes can tune their radio to any channel among 8 non-overlapping channels (according to IEEE 802.11a standard). CTS/RTS mechanism is disabled.

We compared SICA with another interference-aware channel assignment proposed in the literature called Urban-X [4]. Urban-X uses three radios for each node: an *R* and *T* radios, as in SICA, and a third radio which is tuned to a common channel for all nodes. The common channel stays unchanged through the life time

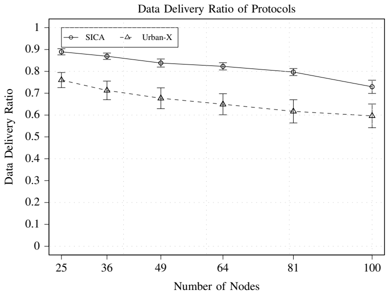


Figure 2: Data delivery ratio vs. number of nodes

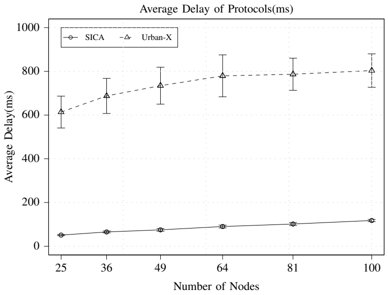


Figure 3: Average end to end delay vs. number of nodes

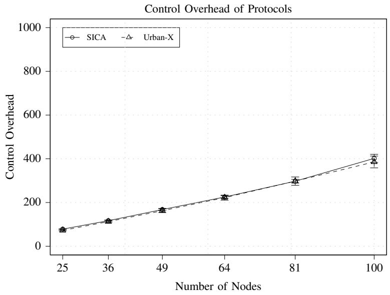


Figure 4: Control overhead vs. number of nodes

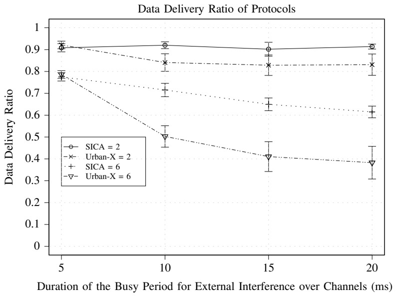


Figure 5: Data delivery ratio vs. the number of channels that suffer external interference

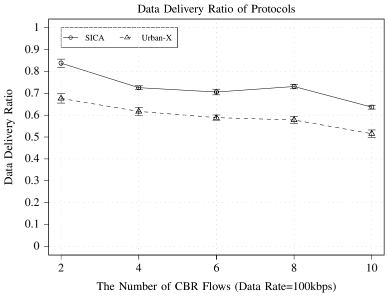


Figure 6: Data delivery ratio vs. CBR traffic loads

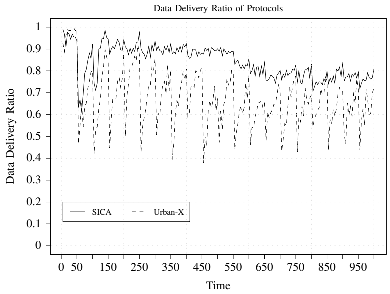


Figure 7: Data delivery ratio over time

of the network. Channel assignment in Urban-X takes into account the amount of flows a node has to send, and the estimated external interference over the channels. Nodes need to have information about the number of flows their neighbors have. Then Urban-X assigns a priority to each node based on the number of active flows it has, and nodes having higher priority have more chances to occupy the best channels (those with less traffic from external networks). Nodes broadcast control messages over the common channel up to two-hops neighbors. After switching to a channel, the *T* radio remains there for a predefined period of time (40 ms).

We've used a dual-radio network to evaluate SICA, while for Urban-X, we have added an extra radio for each node for the common channel. We have evaluated the performance of the protocols for different number of nodes which are placed in a grid topology. The traffic is generated by 100 kbps CBR flows sent over vertical and horizontal directions in the grid. The packet size is equal to 1 kB and the average hop count between a source and a destination is 6. The channels having interference from external networks are chosen randomly. We have done simulations using different number of channels. A channel with external interference is modeled as an on-off process, i.e. the channel is sensed busy and idle during the on and off states respectively. The duration of the *busy* state has been fixed to a constant value, while the duration of the *idle* state is chosen exponentially distributed. The duration of the busy and idle periods have been varied to produce different interference loads. The SICA parameters and the assigned values are shown in Table I. The Urban-X specific parameters are set according to the values given in [4].

A. 802.11 Based Multi-Radio Performance

We consider three network performance measures:

- *Data delivery ratio:* ratio of the total amount of data which is correctly received by the destinations, to the total amount of data packets transmitted by the sources.
- *Average end to end delay:* mean delay of the packets to reach the destination.
- *Control overhead:* ratio of the total number of control messages sent between nodes, to the total number of correctly received packets.

Figures 2-4 show the network performance for different number of nodes and two CBR traffic flows of 100 kbps. Every 50 s, external interference is introduced over 4 channels chosen randomly. In these simulations the duration of busy state of the external interference is fixed to 10 ms, while the mean duration of the idle state is 8 ms. The results have been obtained averaging over 10 runs of 1000 s simulation time with different seeds. The error bars in the figures show 95% confidence intervals.

Fig. 2 shows that the delivery ratio is 10% higher in SICA than in Urban-X. This is a significant improvement, since Urban-X uses 3 radios and SICA uses only 2. The result shows that the game theory approach used in the channel assignment of SICA outperforms the priority scheme used in Urban-X.

In Fig. 3 we can see that the average end to end delay is much lower in SICA than in Urban-X. SICA leads to a lower delay because of the fast switching of the *T* radio over all channels, while Urban-X keeps the *T* radio in each channel for a predefined period of time, regardless of having data to send.

Fig. 4 shows that both protocols have a similar control overhead in terms of *Hello* messages. Urban-X uses a specific common radio for exchanging control messages, and each control message is sent over two hops. SICA, on the other hand, sends control messages only over those channels where a node has neighbors.

Fig. 5-7 are obtained using a 7×7 grid network while other parameters are the same as in Fig. 2.

Fig. 5 compares the delivery probability obtained with SICA and Urban-X, varying the number of channels with external interference and the interference load. The x-axis shows the load of the external interference, which has been varied by changing the duration of the busy state of the interference between 5 ms and 20 ms, and maintaining the mean duration of the idle state equal to 8 ms. We introduced interference over 2 and 6 channels. Fig. 5 shows that, even with a high interference load, the delivery ratio in SICA changes from 90 to 60%, when the interference is increased from 2 to 6 channels. On the other hand, in Urban-X, the delivery ratio drops from 90 to 40%. The result confirms that, SICA is much more robust and less sensitive to the external interference than Urban-X.

Fig. 6 compares the delivery ratio obtained with SICA and Urban-X varying the number of CBR sources between 2 and 10. Fig. 6 shows that SICA outperforms Urban-X, confirming the conclusions drawn from Fig. 2.

In order to have a more detailed view of the protocol's behavior, Fig. 7 shows the time evolution of the delivery ratio obtained with SICA and Urban-X. The values shown in the figure have been obtained repeating the simulation for 20 different random seeds and averaging the delivery ratio over 5 s periods. The figure shows that in SICA the delivery ratio is kept more stable than in Urban-X. Recall that every 50 s external interference is introduced over 4 channels chosen randomly. Fig. 7 shows that Urban-X has a considerable drop in delivery ratio at these time instants. SICA, on the contrary, is less sensitive to the changes of external interference, demonstrating that SICA adapts faster than Urban-X to the external interference.

We have also investigated the sensitivity of SICA to γ and β tuning parameters of the game model (see section IV). We omit the details here due to space constrains. The results show that the performance of SICA is not very sensitive to γ, and the best results are obtained with $\beta \approx 0.2$.

VIII. CONCLUSIONS

In this paper we have considered the channel assignment problem in multi-radio wireless mesh networks. We have proposed a new semi-dynamic protocol called SICA. SICA uses game theory and online prediction concepts for a distributed channel selection where nodes do not have perfect knowledge about others strategies. We have done a performance evaluation comparing SICA with Urban-X, which is an adaptive channel assignment algorithm proposed in the literature. Simulation results show the efficiency of SICA in exploiting channel diversity for avoiding external interference and reducing the internal interference with only 2 radios per node. Moreover, even if Urban-X uses 3 radios per node, SICA outperforms Urban-X in terms of delivery ratio and delay.

ACKNOWLEDGMENTS

This work was supported by the Spanish government and Generalitat de Catalunya through projects TIN2010-21378-C02-01 and 2009-SGR-1167, respectively, and by the European Commission through the NoE EuroNF. We also acknowledge Enric Monte Moreno for helpful discussions during the project.

REFERENCES

[1] J. Crichigno, M. you Wu, and W. Shu, "Protocols and architectures for channel assignment in wireless mesh networks," *Ad Hoc Networks*, vol. 6, pp. 1051–1077, 2008.

[2] W. Si, S. Selvakennedy, and A. Y. Zomaya, "An overview of channel assignment methods for multi-radio multi-channel wireless mesh networks," *J. Parallel Distrib. Comput.*, vol. 70, pp. 505–524, May 2010.

[3] K. N. Ramachandran *et al.*, "Interference-aware channel assignment in multi-radio wireless mesh networks," in *INFOCOM*, 2006, pp. 1–12.

[4] W. Kim *et al.*, "Urban-x: Towards distributed channel assignment in cognitive multi-radio mesh networks," in *IFIP Wireless Days*, Venice, Italy, 2010.

[5] M. Felegyhazi, M. Cagalj, S. Bidokhti, and J.-P. Hubaux, "Non-cooperative multi-radio channel allocation in wireless networks," in *INFOCOM*, May 2007, pp. 1442 –1450.

[6] L. Gao, X. Wang, Y. Xu, and W. Chen, "Distributed multi-radio channel allocation in multi-hop ad hoc networks." in *ICC'08*, 2008, pp. 3156–3160.

[7] I. Shah, S. Jan, and K.-K. Loo, "Selfish flow games in non-cooperative multi-radio multi-channel wireless mesh networks with imperfect information," in *ICWMC*, September 2010, pp. 219 –225.

[8] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part i–carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE Trans. Commun.*, vol. 23, no. 12, pp. 1400–1416, 1975.

[9] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388 –404, Mar 2000.

[10] D. Murray, M. Dixon, and T. Koziniec, "Scanning delays in 802.11 networks," in *NGMAST*, 9 2007, pp. 255 –260.

[11] Y. Freund and R. E. Schapire, "Adaptive game playing using multiplicative weights," *Games and Economic Behavior*, vol. 29, no. 1-2, pp. 79–103, October 1999.

[12] C. Daskalakis *et al.*, "On learning algorithms for nash equilibria," in *Algorithmic Game Theory*, 2010, pp. 114–125.

[13] ns-3 development team, "ns-3 manual (release ns-3.9)," 2011.

Should Next Generation Wireless Mesh Networks consider Dynamic Channel Access?

Maryam Amiri Nezhad*, Boris Bellalta†, Manel Guerrero Zapata*, Llorenç Cerdà-Alabern*
*Dept. of Computer Architecture
Universitat Politecnica de Catalunya, Barcelona, Spain
Email: maryam, guerrero,llorenc@ac.upc.edu
†Dept. of Information and Communication Technologies
Universitat Pompeu Fabra
Barcelona, Spain,
Email:boris.bellalta@upf.edu

Abstract

The increasing demand for large and low cost wireless coverage, ranging from campus to city wide areas, has motivated a high interest in multi-hop communications with IEEE 802.11s as the most significant and successful standard for wireless mesh networks. Although IEEE 802.11s introduces new interworking, routing and wireless frame forwarding at the link layer, the multi channel architecture receives less attention. In this paper we provide insights into the IEEE 802.11s standard and explain some channel assignment (CA) protocols which can be considered for wireless mesh networks to improve their performance by limiting the negative interference effects.

1. Introduction

Multihop wireless networks are emerging as a promising architecture to extend the wireless coverage in a flexible and cost-effective way without relaying on any wired infrastructure. They have broad applications in Internet access, emergency networks, public safety, and so forth [1, 2].

Technical solutions for multihop wireless networks are being specified in IEEE 802.11s [3]. IEEE 802.11s is developed as an extension of the successful IEEE 802.11 standard for WLANs (Wireless Local Area Networks) [4]. IEEE 802.11s based mesh networks are composed of mesh stations (Mesh STAs) that operate as routers. Within a mesh, packets are transmitted over multiple wireless hops. However due to the broadcast nature of the wireless media, wireless links interfere each other if there are simultaneous transmissions in them. In multi hop networks, the interference of the next hop link over the previous hop reduces the end-to-end performance drastically [5, 6]. A possible solution to eliminate the interference of radio transmissions is that nearby links transmit over different non-overlapping frequencies using multi-radio nodes.

Multi-radio, multi-channel mesh networks (MRMC-WMN) have been vastly studied during recent years [7–9], and research results show that they achieve significant performance gains when compared to single channel networks [10, 11]. Nevertheless, channel selection mechanisms for MRMC-WMNs are very challenging to design, since many formulations of this problem turn out to be NP-hard [12–14].

IEEE 802.11s defines the mesh operation in a single channel although multi-radio mesh STAs can form different meshes. The connection between different meshes is provided via bridging. Mesh STAs can initiate the channel switching mechanism which moves the mesh or part of it to another channel. The STAs which do not want to follow the channel switch request may join another mesh. Although it is possible to have dynamic multi-channel networks, the network performance may be compromised drastically if mesh STAs join and leave their mesh too frequently due to channel switching overheads. In addition, the default path selection metric of the standard does not consider the dynamic channel selection of mesh STAs. And, it offers low performance in multi-channel multi-radio networks compared to other path metrics which consider the channel diversity for selecting the path [15].

This paper, provides -in Section 2- an overview of how 802.11s based mesh networks work. It also presents some of the recent proposals for adaptive

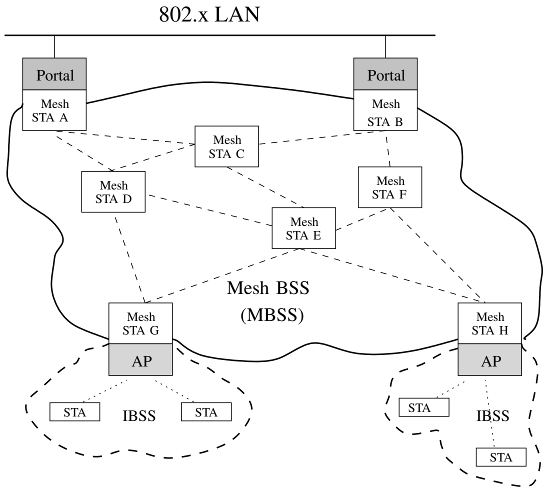


Figure 1: 802.11s Wireless Mesh Network

channel selection -in Section 3. And, it shows the performance gains of dynamic channel assignment in multi-channel mesh networks compared to the single channel one -in Section 4. The paper is concluded in Section 5.

2. Current 802.11s Standard for Wireless Mesh Networks

Since 2004, the Task Group S has been developing an amendment for 802.11 standard to create multi hop wireless networks based on WLAN technology. The new standard called IEEE 802.11s that was finally released in 2011 [3], introduces wireless frame forwarding, routing capabilities (Path Selection) at MAC layer, interworking and security.

Interworking makes the 802.11s mesh network to look like a single Ethernet object to the outside (Fig. 1). A mesh BSS (MBSS) is an IEEE 802.11 LAN consisting of autonomous STAs [3]. Mesh STAs, which form the MBSS, forward packets wirelessly inside the mesh but they do not communicate with non mesh STAs (Fig. 1).

IEEE 802.11s added some fields -extension address fields, mesh sequence and time-to-live fields- to the data and management frames compared to conventional 802.11 frames. The address extentions allow for a total of six addresses in mesh data frames. This is useful when a mesh station acts as a proxy for some stations which do not belong to the mesh network, e.g. mesh STAs G and H in Fig. 1. The extended source and destination addresses carry the address of the source and destination which use the mesh STA as a proxy.

2.1. Mesh Formation

Active scanning (Probe frame transmission) or Passive scanning (observation of mesh beacons) are used by mesh STAs to detect each other. Through beacons, mesh STAs transfer the following information:

- Mesh ID: The ID of the mesh network.
- Mesh configuration: The path selection and path metric identifier, the congestion control mode, the synchronization method identifier, etc.
- Mesh parameters: Parameters that are supported by the transmitter mesh STA.
- Mesh Channel Switch Parameters.

Mesh STAs are considered to have only one radio interface and hence, the default operation for IEEE 802.11s mesh networks is in single channel. However, as Fig. 2-a shows multi-radio stations can form different meshes on different channels where the layer two bridging may be used to unify the different meshes in a single LAN. Fig. 2-b shows the three meshes formed on different channels.

2.1.1. MBSS Channel Switching. A mesh STA can initiate a channel switching announcement to move the mesh to another channel due to any reason, such as interference or radar appearance. The mesh STA shall inform each of the peer mesh STAs that the mesh is moving to a new channel. It maintains the mesh peerings by advertising the switching event using Channel Switch Announcement elements together with Mesh Channel Switch Parameters element in Beacon frames, Probe Response frames, and Channel Switch Announcement frames until the intended channel switch time.

The channel switch has to be scheduled so that all mesh STAs in the MBSS, including mesh STAs in power save mode, have the opportunity to receive at least one Channel Switch Announcement element before the switch.

A mesh STA that receives a Channel Switch Announcement element may choose not to perform the specified switch, but to take an alternative action. For example, it may choose to move to a different MBSS.

It is important to remark that, the standard does not provide any default mechanism for selecting the target channel, hence opening the door for each manufacturer to implement their own solutions.

2.2. Medium Access Control

Mesh STAs use Mesh Coordination Function (MCF) to access the medium. MCF consists of a mandatory

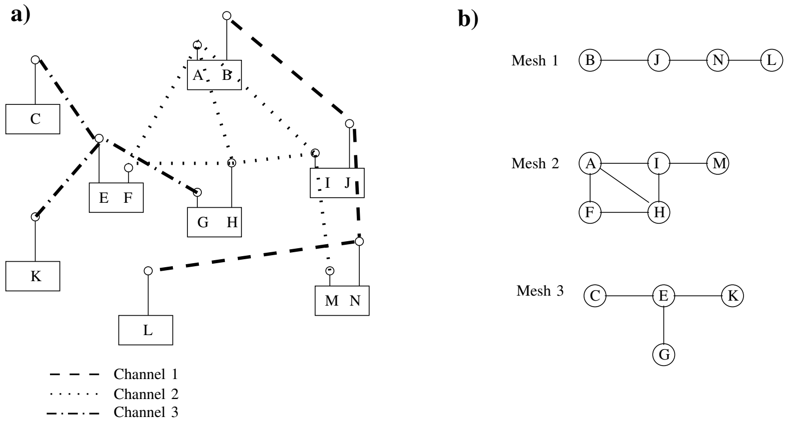


Figure 2: multi-radio mesh STAs

contention based and an optional contention free channel access mechanisms.

The contention based MCF relies on Enhanced Distributed Channel Access (EDCA), which provides limited support for quality of service (QoS) by defining four different traffic categories at the MAC layer [4]. A mesh STA using EDCA is able to transmit multiple frames whose total transmission duration does not exceed the transmission opportunity limit (TXOP). The receiver acknowledges any successful frame reception.

MCF Controlled Channel Access (MCCA) is an optional access method that allows mesh STAs to access the medium at selected times with lower contention than would otherwise be possible. Not all mesh STAs are required to use MCCA. MCCA may be used by a subset of mesh STAs within the mesh.

MCCA enabled mesh STAs, use management frames to make reservations for transmissions. The mesh STA has to transmit an MCCA Setup Request frame to initiate a reservation, the reserved TXOP is called MCCA Opportunity (MCCAOP). MCCAOP has a precise starting time and a pre-defined maximum duration, which requires a tight synchronization for all mesh STAs using MCCA. Note that the standard does not provide any default mechanism for scheduling the MCCAOPs, which may affect the performance of the MCCA [16].

The mesh STA advertises the MCCAOP via beacon frames. To avoid conflict with mesh STAs outside the beacon reception range, the mesh STA also includes its neighbors' MCCAOP reservation in the advertising frame. This implies that MCCA assumes the interference from nodes laying outside the two hops neighborhood of a link is negligible, which may degrade the performance of the channel access in realistic conditions [16].

The mesh STA which has reserved the MCCAOP will access the medium using EDCA and does not have any priority over other mesh STAs which does not support MCCA.

2.3. Congestion Control

Mesh networks suffer from interference due to the broadcast nature of the wireless medium. The mesh STAs, in the middle of the mesh, face more interference due to the high number of neighbors they have, compared to mesh STAs at the edge of the network. Since the access to the media relies on the carrier sense mechanism of 802.11 [4], core mesh STAs have less opportunities to access the media and hence they are prone to suffer congestion and start dropping frames. Dropping data frames is costly in a mesh network because a frame may travel several hops before reaching the congestion point.

IEEE 802.11s provides a congestion signaling protocol which allows the congested mesh STAs to advertise the expected duration of the congestion to the neighboring mesh STAs and inform them to slow down or to stop transmitting. The reduction of the frame arrival rate to a congested mesh STA avoids wasting the mesh resources for transmission of packets that with high probability will not be handled/forwarded by the congested mesh STA [17].

2.4. Synchronization and Beaconing

Beaconing procedure for mesh STAs is introduced in 802.11s. Since beacons are not acknowledged, a Mesh beacon collision avoidance (MBCA) mechanism is introduced to avoid collision between beacons transmitted by hidden nodes. Using MBCA, a mesh STA advertises its beacon interval and target beacon transmission times (TBTTs) in addition to its neighbors beacon interval. Upon receiving the beacon timing from its neighbors, the mesh STA uses these information to select its TBTT and beacon interval so that its beacons do not collide with the neighbors' beacons within the two hops transmission range.

In 802.11s the synchronization is similar to the timing synchronization function (TSF) of the original 802.11 standard [4]. In order to enable minimal synchronization capabilities between mesh STAs using MCCA, MBCA and power saving mode, an extensible synchronization framework is also introduced to support different synchronization mechanisms in the mesh networks. Within the framework, the neighbor offset synchronization method is defined as the default mandatory synchronization method. Using the neighbor offset synchronization method, a mesh STA have to maintain a timing offset value between its own time synchronization function (TSF) timer and the TSF timer of the mesh STA with which the mesh STA is synchronized. A mesh STA can initiate its TSF independently and can update the TSF timer offset based on the time-stamp received in the beacons from other mesh STAs.

2.5. Path Selection

IEEE 802.11s provides a mandatory default path selection and path metric, but any other approaches could be also used.

The path metric is called Airtime and reflects the amount of channel resources consumed by transmitting a 1 *kB* frame over a particular link, considering the data rate, overhead and transmission errors.

The default path selection, Hybrid Wireless Mesh Protocol (HWMP) combines the operations of a proactive tree oriented approach with an on demand path selection inspired from Ad-Hoc On Demand Distance Vector (AODV) protocol.

HWMP elements are the path request (PREQ), path reply (PREP), path error (PERR), and root announcement (RANN). The path metric is announced through PREQ, PREP, and RANN elements. An independent sequence number is propagated by mesh STAs to others through HWMP elements. The sequence number is used to discover stale paths and maintain loop free connections.

HWMP supports two modes of operation depending on how the mesh network is configured:

- On-demand mode: The functionality of this mode is always available, independent of whether a root mesh STA is configured in the MBSS or not. It allows mesh STAs to communicate using peer-to-peer paths.
- Proactive tree building mode: In this mode, an additional proactive tree building functionality is added to the on-demand mode. This can be performed by configuring a mesh STA as root mesh STA using either the proactive PREQ or RANN mechanism. The proactive PREQ mechanism creates paths from all the mesh STAs to the root, using only group-addressed communication. The RANN mechanism creates paths between the root and each mesh STA using acknowledged communication.

3. Dynamic Channel Access

IEEE 802.11s standard proposed a channel switching mechanism which moves the mesh to another frequency to avoid performance degradation due to interference or presence of radar. As all mesh STAs are supposed to work in the same channel, the access to the channel becomes competitive specially in the core of the network. Many research findings show that the capacity per node in such scenario drops significantly when the network size increases [5, 7]. In a multi-hop single channel network with all links running, the end-to-end performance suffers low throughput and unfairness [18]. Multi-channel mesh networks are able to provide significant capacity gain compared to single channel networks by placing neighboring links over different non overlapping channels [7, 8].

Assuming that mesh STAs have two or more radios, hybrid channel assignment protocols can be used to enhance IEEE 802.11s mesh networks, where at least one radio interface at each mesh STA is controlled dynamically to communicate with neighboring mesh STAs on different channels [19, 20]. Here we provide an overview of some channel assignment algorithms proposed recently, and compare their performance against the single channel network via simulation.

3.1. Breath First Search Channel Assignment (BFSCA)

Breath First Search Channel Assignment (BFSCA) [21] is a centralized mechanism that considers one node in the WMN as a coordinator, which is responsible for assigning channels to all nodes' radios in the network. One radio at each node is tuned to a common channel for control messages. BFSCA assigns the same channel to the end points of a link, and therefore to maintain the topology, neighboring nodes are advertised to change the channel simultaneously.

Each node estimates the external interference through monitoring the packets on the wireless media.

From monitoring, the node estimates the data rate and the number of external devices working over the channel. BFSCA assumes that nodes can acquire the MAC address of all other mesh STAs in their network. Therefore, they are able to calculate the amount of external interference through analyzing the packets gathered during the monitoring mode. Then the node sends the results of such action to the central node through the common channel.

The coordinator assigns channels to links and informs nodes. Nodes redirect the traffic to the common channel before switching to a new channel and, therefore, they need to be tightly synchronized. Otherwise the channel switching mechanism would interrupt the data packet transmissions.

Channel assignment in BFSCA uses a graph theory based interference model to find the interfering links [5]. It sorts all links based on the distance to the central node and their quality in terms of delay, and then tries to assign different channels to the interfering links. If such a channel is not found, BFSCA assigns random channels to the links.

Since the number of radio interfaces on each node is limited, when a channel is assigned to a link, the nodes at the end points of the link loose their flexibility to choose any channel for the other links connected to them, therefore BFSCA gives more priority to the links of those node for assigning channels to them.

The coordinator then sends messages to the nodes which should switch their radios based on the new configuration.

3.2. Urban-x: Distributed Channel Assignment

Urban-X [20] considers three radios for each node: one receiving (*R*) and one transmitting (*T*) radio, and a third radio which is tuned to a common channel for all nodes. The common channel stays unchanged through the life time of the network. The channel assignment assigns different channels to the *R* radio interface of each node. Then a node which has data to send establishes the connection with the intended receiver by switching its *T* radio to the receiving channel of the intended receiver.

The channel assignment in Urban-X is distributed and takes into account the amount of flows a node has to send, and the estimated external interference over the channels.

The amount of external interference over a channel is acquired by sensing the channel while all nodes belonging to the mesh are silent. The synchronization is achieved through sending control messages over the common channel.

Nodes need to have the information about the number of flows their neighbors have. Each node sends the amount of traffic waiting to be sent on each channel through control messages which are broadcasted over the common channel up to two-hops neighbors.

Then, Urban-X assigns a priority to each node based on the number of active flows it has among its neighbors. Nodes with a higher priority have more chances to occupy the best channels. After switching to a channel, the *T* radio remains there for a predefined period of time (40 ms).

3.3. Semi-dynamic Interference aware Channel Assignment (SICA)

SICA [19] is a distributed semi dynamic channel assignment which is designed based on Game theory formulation. SICA assumes at least two radio interfaces for each node: one receiving (*R*) and one transmitting (*T*) radio. The channel assignment assigns a channel to the *R* radio subject to minimize the interference over the receiving channel, while nodes control the *T* radio dynamically to establish the connections with their neighbors.

SICA does not relay on any common channel nor central node for synchronization. The synchronization is achieved through sending control messages over all channels where a node has a neighbor. Moreover the nodes use a similar mechanism of 802.11s (Section 2.1.1) to switch the channel of a radio by announcing the channel switch event in advance.

The channel assignment in SICA is distributed and based on Game theory where nodes are assumed to be rational and selfish trying to occupy most vacant channels. In addition, nodes are assumed to be unaware of others strategy or the type of game they are playing.

Each node estimates the amount of external interference over a channel via sensing the channel. Then, the channel assignment considers the external interference and the number of neighboring nodes over a channel in addition to the switching penalty to choose the best channel.

The nodes use a multiplicative weight update learning method [22] to find better channels over time following the changes in the wireless environment.

4. Performance Evaluation of Multi-Channel and Single Channel Mesh

We have evaluated the performance of a single channel and a multi-channel mesh network for different number of nodes placed in a grid topology and with different number of traffic flows. The traffic is generated by 150 kbps CBR flows with packet size of $L = 8000$ bits sent over vertical and horizontal directions in the grid topology. Each flow is between one node in an edge toward the other node placed at the opposite edge. For a grid of $xSize \times ySize$, the vertical flows are between (x,y) and (x+Size-1, y) while the horizontal flows are between (x,y) and (x,y+ySize-1). The simulation duration is 1000 s.

The channels receiving interference from external networks, are chosen randomly. A channel with external interference is modeled as an on-off process, such that the channel is sensed busy and idle during the on and off states, respectively. Note that when the channel is detected busy due to the external interference, nodes are not allowed to transmit during that state. The duration of the *busy* state has been fixed to a constant value (15 ms), while the duration of the *idle* state is exponentially distributed with mean equal to 8 ms. In the simulations, external interference is introduced over 4 channels chosen randomly.

Different multi-channel mesh networks are evaluated using the channel assignment protocols explained before (Section 3). The specific parameters of the protocols are set according to the values given in [19–21].

We consider three network performance measures:

- *Data delivery ratio:* ratio of the total amount of data which is correctly received by the destinations, to the total amount of data packets transmitted by the sources.
- *Average end-to-end delay:* mean delay of the packets to reach the destination.
- *Control overhead:* ratio of the total number of control messages sent between nodes, to the total number of correctly received packets.

4.1. Increasing the number of nodes

Fig. 3- 5, show the packet delivery ratio, average end to end delay and control overhead of 3 CBR (150 kbps) traffic flows for different number of nodes in grid topology. The traffic sources and destinations are the same in all simulations. Each point in the figures shows the average over 10 runs. The error bars in the figures show 95% confidence intervals.

Fig. 3 shows that the delivery ratio of multi-channel networks is higher than the single channel network as the number of nodes increases. Moreover, as expected, the specific channel assignment protocol has a big impact on the efficiency of the multi-channel network.

The delivery ratio in SICA is higher than in other protocols and it does not decrease as the number of nodes increases. For BFSCA, on the other hand, the delivery ratio drops fast as the number of nodes increases since the centralized protocol is not scalable enough. The single channel network performs as a multi channel network using Urban-X in terms of delivery ratio, but it results in a higher delay due to the higher congestion in the single channel.

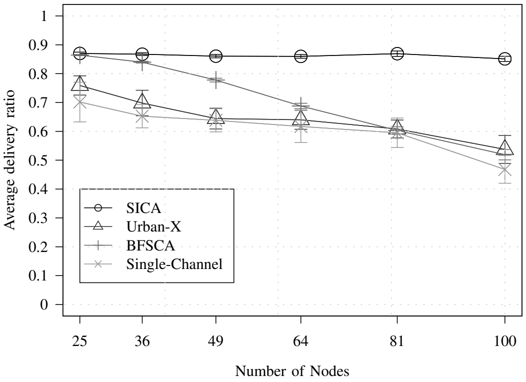


Figure 3: Data delivery ratio vs. number of nodes (Grid topology)

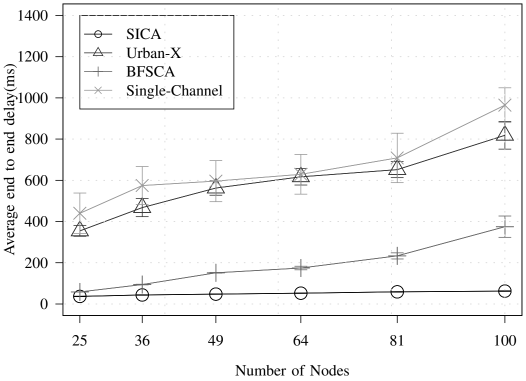


Figure 4: Average end to end delay vs. number of nodes (Grid topology)

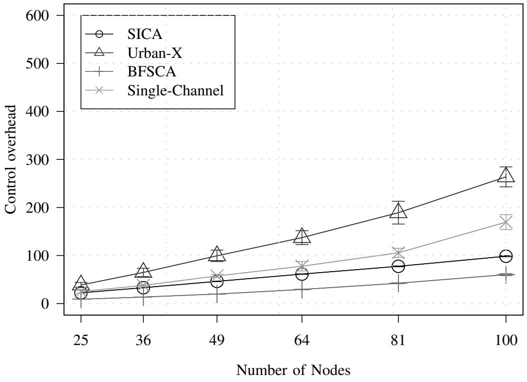


Figure 5: Control overhead vs. number of nodes (Grid topology)

Fig. 4 shows that the average end to end delay of dynamic channel assignment protocols is much lower compared to single channel. Urban-X results in a higher delay compared to other channel assignment protocols, since it keeps the transmitting radio on each channel for a predefined period of time, after switching, regardless of the amount of traffic waiting to be sent.

Fig. 5 shows that SICA and BFSCA lead to lower control overhead. The control overheads of the single channel network are the *Hello* messages that nodes send to their neighbors to inform them about themselves.

4.2. Increasing the number of traffic sources

Fig. 6-8 are obtained using a 6×6 grid network while the number of CBR flows of 150 kbps increases from 2 to 7. The results have been obtained averaging over 10 runs of 1000 s with different seeds. The error bars in the figures show 95% confidence intervals.

Fig. 6 shows that the delivery ratio of SICA is higher than others in presence of high traffic load. The delivery ratio of BFSCA drops fast when the number of traffic flows increases, since any channel switching interrupts the data transmission and nodes are forced to deliver data packets through the common channel, which saturates it (Section 3.1). The delivery ratio obtained by using a single channel network is lower than multi-channel networks due to the lower capacity of a single channel.

Urban-X performs better and more robust in presence of high traffic load compared to BFSCA since it considers the traffic load for making decisions, but it results in a considerable high end-to-end delay (Fig. 7). Moreover, multi-channel approaches lead to a lower end to end delay compared to the single channel network, although the dynamic protocols suffer from the radio switching delay.

Fig. 8 shows that the control overhead of SICA and BFSCA is better than for Urban-X.

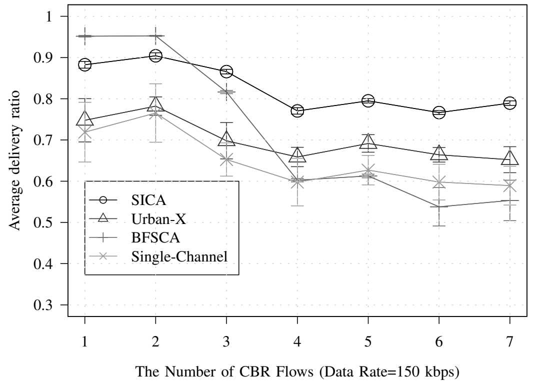


Figure 6: Data delivery ratio vs. number of CBR traffics

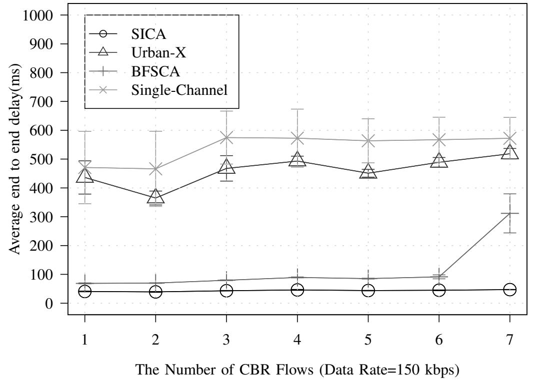


Figure 7: Average end to end delay vs. number of CBR traffics

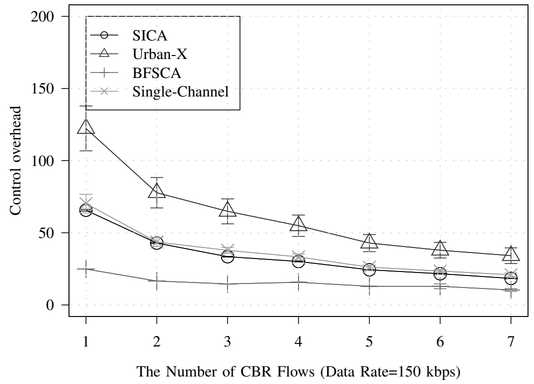


Figure 8: Control overhead vs. number of CBR traffics

5. Conclusion

In this paper we overview the main features of IEEE 802.11s -the recent standard for wireless mesh networks. In addition, we analyze some of the most recent channel assignment protocols which are potential candidates to enhance the operation of actual IEEE 802.11s-based mesh networks. Simulation results show the benefits of using Dynamic Channel Allocation mechanisms in multi-channel mesh networks compared against single channel or static channel selection approaches. The presented results show that, more attention should be directed at designing a smart channel selection mechanism and a channel aware path selection metric for the current standard.

Acknowledgments

This work was supported by the Spanish Government and Generalitat de Catalunya through projects TIN2010-21378-C02-01, TEC2008-0655, and 2009-SGR-1167, respectively, and by the European Commission through the NoE EuroNF.

References

[1] I. F. Akyildiz and X. Wang, "A survey on wireless mesh networks," *IEEE Commun. Mag.*, vol. 43, no. 9, 2005.

[2] D. Raychaudhuri and M. Gerla, *Emerging Wireless Technologies and the Future Mobile Internet*. Cambridge University Press, 2011. [Online]. Available: http://dx.doi.org/10.1017/CBO9780511921117

[3] "Ieee standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 10: Mesh networking," 2011, iEEE Std 802.11s-2011 (Amendment to IEEE Std 802.11-2007).

[4] "Ieee standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," 2007, iEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999).

[5] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388 –404, Mar 2000.

[6] H. Balakrishnan, C. Barrett, V. Kumar, M. Marathe, and S. Thite, "The distance-2 matching problem and its relationship to the mac-layer capacity of ad hoc wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 22, no. 6, pp. 1069 – 1079, Aug. 2004.

[7] W. Si, S. Selvakennedy, and A. Y. Zomaya, "An overview of channel assignment methods for multi-radio multi-channel wireless mesh networks," *J. Parallel Distrib. Comput.*, vol. 70, pp. 505–524, May 2010. [Online]. Available: http://dx.doi.org/10.1016/j.jpdc.2009.09.011

[8] J. Crichigno, M. Wu, and W. Shu, "Protocols and architectures for channel assignment in wireless mesh networks," *Ad Hoc Networks*, vol. 6, no. 7, pp. 1051–1077, 2008.

[9] V. S. Kapse and U. N. Shrawanakar, "Survey of channel assignement schemes in wireless mesh network," in *Proc. 3rd Int Electronics Computer Technology (ICECT) Conf*, vol. 3, 2011, pp. 103–107.

[10] A. Raniwala, K. Gopalan, and T.-c. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 8, no. 2, pp. 50–65, 2004.

[11] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM New York, NY, USA, 2004, pp. 114–128.

[12] A. Subramanian, H. Gupta, S. Das, and J. Cao, "Minimum interference channel assignment in multiradio wireless mesh networks," *IEEE Transactions on Mobile Computing*, pp. 1459–1473, 2008.

[13] A. H. Mohsenian-Rad and V. W. S. Wong, "Joint logical topology design, interface assignment, channel allocation, and routing for multi-channel wireless mesh networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 12, pp. 4432–4440, 2007.

[14] M. K. Marina, S. R. Das, and A. P. Subramanian, "A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks," *Computer Networks*, vol. 54, no. 2, pp. 241 – 256, 2010.

[15] S. Ghannay, S. M. Gammar, F. Filali, and F. Kamoun, "Multi-radio multi-channel routing metrics in ieee 802.11s-based wireless mesh networks and the winner is," pp. 1–8, 2009, communications and Networking, 2009. ComNet 2009. First International Conference on.

[16] C. Cicconetti, L. Lenzini, and E. Mingozzi, "Scheduling and dynamic relocation for ieee 802.11s mesh deterministic access," pp. 19–27, 2008, sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on.

[17] D. Fu, B. Staehle, R. Pries, and D. Staehle, "On the potential of ieee 802.11s intra-mesh congestion control," in *Proceedings of the 13th ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems*, ser. MSWIM '10. New York, NY, USA: ACM, 2010, pp. 299–306. [Online]. Available: http://doi.acm.org/10.1145/1868521.1868569

[18] S. Xu and T. Saadawi, "Does the ieee 802.11 mac protocol work well in multihop wireless ad hoc networks?" pp. 130–137, 2001, communications Magazine, IEEE.

[19] M. A. Nezhad and L. Cerda-Alabern, "Adaptive channel assignment for wireless mesh networks using game theory," in *Proc. IEEE 8th Int Mobile Adhoc and Sensor Systems (MASS) Conf*, 2011, pp. 746–751.

[20] M. D. F. Wooseong Kim, Andreas J. Kassler and M. Gerla, "Urban-x: Towards distributed channel assignment in cognitive multi-radio mesh networks," in *IFIP Wireless Days*, Venice, Italy, 2010.

[21] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," in *Proc. 25th IEEE Int. Conf. Computer Communications INFOCOM 2006*, 2006, pp. 1–12.

[22] Y. Freund and R. E. Schapire, "Game theory, on-line prediction and boosting," in *COLT '96: Proceedings of the ninth annual conference on Computational learning theory*. New York, NY, USA: ACM, 1996, pp. 325–332.

UBCA: A Centralized Utility Based Channel Assignment Mechanism for Multi Radio Mesh Networks

Maryam Amiri Nezhad[1] , Llorenç Cerdà-Alabern , Manel Guerrero Zapata
Univ. Politècnica de Catalunya Computer Architecture Dep.
Jordi Girona 1-3 E-08034, Barcelona, Spain

Abstract

We address the channel assignment problem in a multi-radio mesh network that involves assigning channels to radio interfaces for eliminating the effect of wireless interference. Due to the insufficient number of frequency channels and available radios per node, interference is still present which limits the available bandwidth on wireless links and eventually decreases the achievable throughput. In this paper we investigate the effect of considering the diverse delivery probability of the wireless links on the channel assignment solutions. We show that it is possible to classify the wireless links and omit some of them to benefit from a more diverse-channel solution. We propose a new channel assignment mechanism aiming to minimize the interference over high performance links. Finally a performance study is carried out to assess the effectiveness of our proposed algorithm. Evaluations show that the multi-channel network obtained from our proposed algorithm achieves significant improvement in terms of reducing the interference and increasing the network capacity.

Keywords: Wireless Mesh Network, Channel Assignment, Multi-Radio, Multi-Channel

1 Introduction

Wireless mesh networks (WMN), based on commodity 802.11 radios, are promising solutions to provide broadband network coverage at low cost. Mesh networks however suffer from serious interference problems, limiting their capacity due to the broadcast nature of the wireless medium. A common method to improve the capacity is to use multiple orthogonal channels that are already available in 802.11 standard. The main idea is to reduce the interference by using different channels for neighboring links. A wireless router can use multiple channels if it is equipped with multi-radio interfaces (radios). Here the challenge is assigning channels to radios subject to limit the interference over the links while maintaining the network connectivity.

Many papers (Crichigno et al., 2008; Avallone et al., 2008; Dhananjay et al., 2009; Marina et al., 2010) have been published on channel assignment problem, proposing solutions based on different criteria. However, most of the schemes disregard the delivery probability of wireless links, i.e. they suppose that all wireless links offer the same performance for data transmission. Normally, the delivery probability of a link strongly depends on its length, because the received power decreases drastically with increasing the distance. In addition, in a mesh network with a gateway most of the data traffic is directed to/from the gateway, not all wireless links are useful. Therefore links close to the gateway should be selected with higher probability by the routing protocol.

In this work, we propose a new channel assignment that takes these features into consideration and demonstrates its benefits by a performance comparison with other relevant channel assignments algorithms that have been proposed in the literature.

Our contributions that set our work apart from the existing approaches for channel assignment problem are as follows:

- We show that the topology preserving constraint (assigning channel to all available links) leads to a suboptimal solution, i.e. relaxing this constraint improves the results considerably.

---

[1] Corresponding author. Email:maryam@ac.upc.edu. Tel: +34 9340 54055

- We propose a new centralized channel assignment algorithm which is traffic independent.
- We consider the delivery probability and the usefulness of the wireless links to make an efficient decision for assigning good channels to good links. Simulation results show the effectiveness of our approach.
- We show that the maximum throughput and the bottleneck delay is highly affected by the quality of the path to the gateway.

The rest of the paper is organized as follows: Section 2 contains the description of the network model and formulation of the problem. The channel assignment mechanism is proposed in Section 3. We report the simulation results in Section 4. The discussion on the efficiency of the method and the related works is presented in Section 5. The paper is concluded in Section 6.

2 Network Model

We consider a multi-radio wireless mesh network (WMN) consisting of a set of mesh routers (nodes) where some nodes serve as gateways between the WMN and the wired network. We assume that each node has at least $R \geq 1$ radio interfaces (radios) and can tune each radio to one of the frequencies selected from C non overlapping channels. For simplicity, we assume that all radios have the same characteristic.

We model the connectivity between nodes by an undirected graph $G_t=(V,E)$; henceforth referred to as the topology graph. Here V denotes the set of nodes, whereas E denotes the set of links. A pair of nodes have a link in E if they are connected in the network. We associate to every link e a weight equal to its packet delivery probability ($p_d(e)$). Since the wireless links may interfere with each other while transmitting simultaneously, the topology graph is not sufficient to fully characterize the wireless network. To account the impact of interference on a transmission we use the interference protocol model defined in Gupta and Kumar (2000). In this model, two transmissions links will interfere if they occur within the interference range of each other. The interference range of a link is usually supposed to be two times the transmission range.

To represent the interference among all possible transmissions in a network, the conflict graph is used (Jain et al., 2003). The conflict graph $G_f=(V_f,E_f)$, contains a set of vertices corresponding to all links in the network topology. There is an edge between two vertices in the conflict graph if the corresponding links interfere with each other. We define the interference weight for a link e ($I(e)$), as the number of links that potentially interfere with e, consequently the interference weight of a link is equal to the degree of the corresponding vertex in the conflict graph.

Throughout this paper, we use the topology graph to model the network topology, and the conflict graph based on the protocol model for the wireless interference.

2.1 Problem Formulation

The channel assignment problem in a multi-radio wireless mesh network is to find a feasible mapping between radios and channels. A feasible mapping between radios and channels must satisfy the following conditions (feasibility conditions):

Radio constraint: The number of channels assigned to a node must be equal or less than the number of radios it has.

Connectivity constraint: The channel assignment must preserve the connectivity of the network, i.e. after assigning channels to radios, there must be at least one path between all pairs of nodes.

The aim of a channel assignment algorithm is to utilize the available channels effectively, to reduce the interference of all links as much as possible. In most of the cases, due to the limited number of radios per node and the big interference weight of wireless links, it is not possible to eliminate the interference over all links completely. Moreover in a channel assignment strategy each decision will limit the flexibility of the next decision, as we show in the following example.

Fig. 1 shows a simple network with four nodes where $R(v)$ shows the number of radios of node v. The numbers under the links show the packet delivery probability offered by each link. Consider a channel assignment algorithm which starts from node A and assigns channel c_1 to its single radio. To preserve the topology, nodes B and C must tune one of their radios to channel c_1. Therefore they lose their flexibility in making the decision for one of their radios. The algorithm finished its work by assigning channel c_2 to the other links incident on nodes B and C (Fig. 1(a)). In order to calculate the capacity offered by this topology, capacity can be defined as the maximum number of possible concurrent transmissions in the network. Since in this network all wireless links interfere with each other and are assigned to

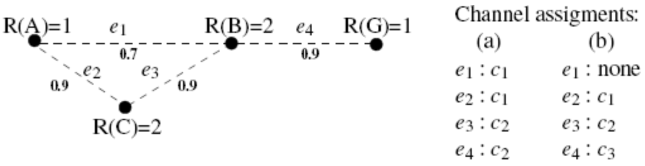


Figure 1: Channel assignment mechanism

channels c_1 or c_2, at most two transmissions can occur concurrently (one on each channel). Therefore the network capacity would be 2*0.9=1.8.

From another standpoint, since the link between *B* and *A* is lossy (compared to other links), if the channel assignment omits this link, node *B* can choose two different channels for its radios, rather than c_1, and obviously we achieve a better channel utilization (Fig. 1(b)). The maximum capacity in the second topology is 3*0.9=2.7, since all remaining links can transmit concurrently over different channels.

This example shows that omitting some lossy links may allow the channel assignment to reach a more optimal solution.

Note that, removing a wireless link from the network topology is possible if endpoint nodes do not share any common channel, therefore it is possible to use channel assignment to prune the network topology. On the other hand, if the lossy links are removed before channel assignment, CA may add those links to the topology by putting the nodes on a common channel, if it is not aware of removed links.

Moreover the process of removing links should be done in a controlled way without affecting the network performance.

We study the problem to find a channel assignment that reduces the interference over good links with a slack restriction on preserving the network topology. In order to measure the amount of interference in a channel, *c*, we define the average of the interference weight of the links in *c* as:

$$F_c = \frac{1}{|E_c|} \sum_{e \in E_c} I_c(e) \qquad (1)$$

where E_c is a set of links assigned to a channel, *c*, and $I_c(e)$ is the interference weight of a link, *e*, in a channel, *c*. The aim of our algorithm is keeping F_c as low as possible over all channels.

Our channel assignment assigns a priority to each link based on its performance and then visits each link in order. It then finds the best channel for the link by comparing the value of F_c for all channels and selecting the channel which has the minimum interference weight (F_c). We explain the channel assignment in more detail in Section 3.

2.2 Link Quality Estimation

To assess the delivery probability ($p_d(e)$) of a link we use the shadowing propagation model (equation (3)) (Rappaport, 2002). Measurement based propagation models for radio communication systems indicate that, the average received signal power decreases logarithmically with distance. The average path loss for an arbitrary transmission-receiver separation is expressed as a function of distance.

$$\overline{P_L(d)}_{dB} = P_L(d_0) + 10\beta log(\frac{d}{d_0}) \qquad (2)$$

where β is the path loss exponent and is usually empirically determined by field measurements. Large β, indicates more obstructions and hence, faster decrease in average received power as distance becomes longer. The value of β depends on the specific propagation environment, here we consider an urban area, and use β=2.7 (Rappaport, 2002). d_0 is the close-in reference distance which is determined from measurements close to the transmitter and *d* is the transmitter-receiver separation distance.

In reality, the received power at certain distance may be vastly different at two different location due to the surrounding environmental clutter. Measurements have shown that at any value of *d* the path loss at a particular location is a random variable, thus the communication range of a wireless radio is not an ideal circle. Equation (3) predicts the mean received power at distance *d* based on the Shadowing propagation model.

$$\left[\frac{\overline{P_r(d)}}{P_r(d_0)}\right]_{dB} = -10\beta log(\frac{d}{d_0}) + X_{dB} \qquad (3)$$

where (X_{dB}) is a Gaussian random variable (with zero mean and standard deviation σ) and reflects the variation of received power at certain distance. We use σ_{dB}=6 throughout this work.

Since $P_r(d)$ is a random variable with normal distribution the Q-function (equation (4)) can be used to determine the probability that the received signal level will exceed a particular level:

$$Q(x)=\frac{1}{\sqrt{2\pi}}\int_{x}^{\infty}\exp(-\frac{u^2}{2})du \qquad (4)$$

The probability that the received signal level will exceed a certain value ζ can be calculated from cumulative density function as:

$$Pr[\overline{P_r(d)}\ |_{dB}>\zeta]=Q\left(\frac{\zeta-\overline{P_r(d)}}{\sigma}\right) \qquad (5)$$

Packets are delivered correctly if the received power is greater or equal to a threshold (e.g. *RXThresh* in network simulator NS-2) (Fall and Varadhan, 2003) . Therefore, the delivery probability at distance d is given by equation (6)

$$p_d(e)=Pr[\overline{P_r(d)}\ |_{dB}\geq 10log_{10}(RXThresh)] \qquad (6)$$

In our model we consider that, links exist only when $p_d(e)\geq 0.5$, this delivery probability is achievable if d is not longer than 131.53_m. For analytical sections of this work we use equation (6) to calculate the delivery probability of links. Note that *RXThresh*, antenna gain and height are set based on default values in Fall and Varadhan (2003).

3 Utility Based Channel Assignment (UBCA)

In this section we describe an algorithm (UBCA) for assigning channels to radios, which is developed based on our model. Note that UBCA is a centralized algorithm which considers a network with at least one gateway. The wireless links are assigned a priority for assigning channels. To assign the priority to the links we consider the gateway placement in the network.

We start by defining the following terms:

Free radio: Whenever the number of channels assigned to a node, is less than the number of radios it has, the node is supposed to have some free radios.

Potential link: In a multi-channel network, the availability of a link depends on the physical distance between end point nodes and the existence of common channel between them. Therefore, we call a link as potential, if the endpoint nodes are physically neighbors but they have no common channel. Note that if a link remains potential at the end of the channel assignment, it is actually removed.

Our channel assignment algorithm (UBCA) has two phases (see Alg. 1). In the first phase, UBCA chooses the most diverse channel set for links without having tight restriction for network connectivity. In this phase, if the algorithm can not assign any common channel to the end point nodes of a link, it marks the link as potential. In the second phase, UBCA makes the final decision for potential links: It tries to make one common channel for them through merging channels over endpoint nodes, or removes them from the topology.

At the beginning of the algorithm, each link is given a priority based on its delivery probability and utility. We describe the exact criteria for determining the priority of each link in the next section (Section. 3.1). UBCA visits each link based on the priority order (line 1 in Alg. 1). For each visiting link, the algorithm first determines a possible set of channels, and then selects the best channel for the link among that set. To select the channel for a link, UBCA investigates the effect of adding the visiting link to all possible channels and chooses the channel with lower interference average (See Alg. 3 and equation (1)).

If the possible channel set for a link is empty, then UBCA marks the link as potential for the next phase. The size of the possible channel set for a link depends on the situation of its endpoint nodes (See Alg. 2). If both endpoint nodes have free radios, then it is possible to assign a new channel to the visiting link, for this case the possible channel set contains all available channels. In the case that one endpoint node has no free radios, selecting a different channel from the current channel set of that node is against the first condition of the feasibility (Section 2.1), therefore the possible channel set for the visiting link is equal to the current channel set of the node which has no free radio (lines 3-5 in Alg. 2). If both end point nodes do not have any free radio, then the possible channel set must be equal to the common channel between the two nodes. Finally in the case that nodes have no free radios neither common channel, the possible channel set for the visiting link would be empty, and the link remains potential (line 9 in Alg. 2).

In the second phase, UBCA visits the remaining potential links. It must decide to remove a potential link from the topology or recheck the channel assignment to make a common channel between end point nodes. A possible way to create a common channel between two nodes is to select one channel from each node's current channel set and merge them to one (assigns all links in one channel to another) (Alg. 4). For selecting two appropriate channels to merge, the algorithm must consider the interference weight of the links after merging channels. However UBCA is mostly oriented to remove unnecessary links. For this reason, in this phase, it visits the potential links in an increasing order of priority i.e. the link with the lowest priority is visited first. For each potential link, it checks whether there is any other path between the two end point nodes. If so, the link is removed, otherwise the channel merging process is applied to make the link available.

Theorem 1. The proposed channel assignment algorithm satisfies the feasibility conditions.

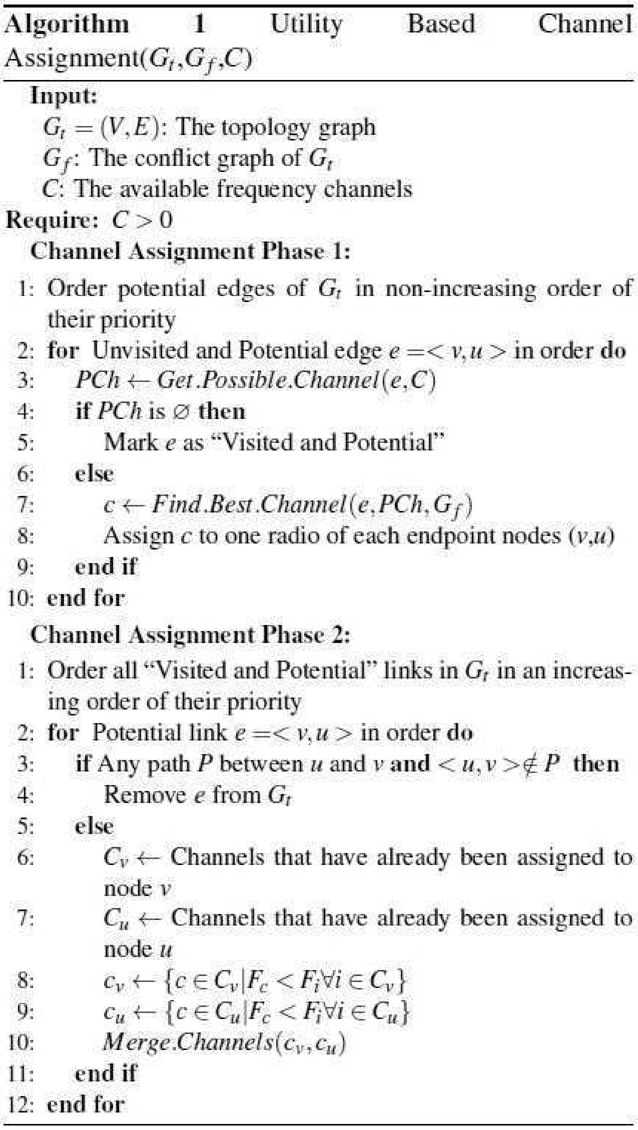
Proof. The feasibility conditions are mentioned in Section 2.1. Here we express the proof for each condition separately.

Radio-constraint: For each link, the best channel is selected among the possible channel set which is determined by Alg. 2. To determine the possible channel set for a link, Alg. 2 checks the radio-constraint condition for end point nodes through lines 2-8. Therefore, the possible channel set for a link is selected based on the current channel set of end point nodes and their free radios. Hence the radio constraint condition is preserved.

Connectivity-constraint: The second phase of Alg. 1, decides to remove a link under two conditions: first, there must be an alternative path between two endpoint nodes; and second, the path must be independent of the current link (see lines 3-4 of the second phase of Alg. 1). If these two conditions are held, then the algorithm removes the link. Therefore, even after deletion of a link, there is a path between two nodes, and thus the network remains connected.

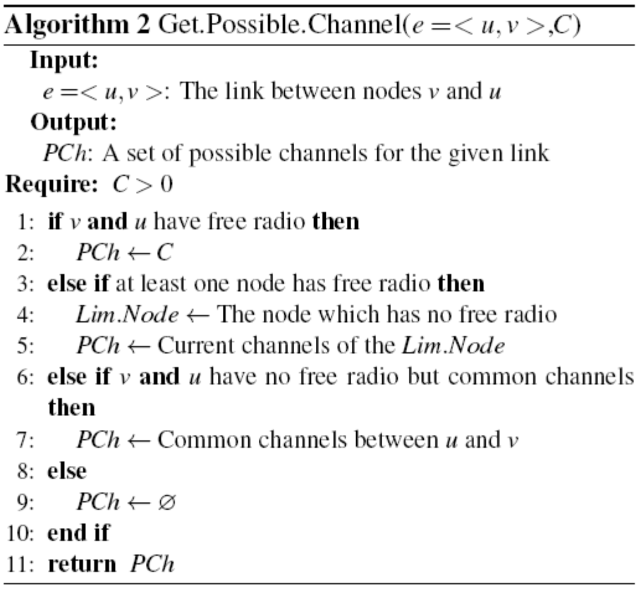
3.1 Computing Links Priority

Recall that the objective of our channel assignment is to reduce the average interference weight of the high performance links by removing unnecessary links. Therefore it is necessary to visit the preferred links first. To this aim considering the delivery probability is not sufficient since links removal must be carried in such a way that have the minimum impact on the paths to the gateway. To assess the role of each link in constructing the paths to the gateway we define the Utility metric for each link ($U(e)$). $U(e)$ will help us to estimate the probability of using a specific link by routing protocol.



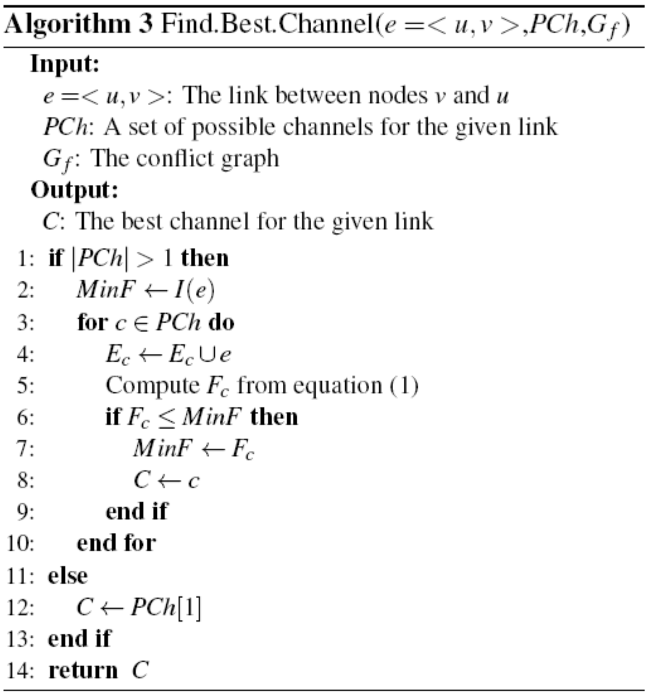
Algorithm 1 Utility Based Channel Assignment(G_t,G_f,C)

```
  Input:
    $G_t = (V,E)$: The topology graph
    $G_f$: The conflict graph of $G_t$
    $C$: The available frequency channels
Require: $C > 0$
  Channel Assignment Phase 1:
 1: Order potential edges of $G_t$ in non-increasing order of
    their priority
 2: for Unvisited and Potential edge $e = <v,u>$ in order do
 3:   $PCh \leftarrow Get.Possible.Channel(e,C)$
 4:   if $PCh$ is $\varnothing$ then
 5:     Mark $e$ as "Visited and Potential"
 6:   else
 7:     $c \leftarrow Find.Best.Channel(e,PCh,G_f)$
 8:     Assign $c$ to one radio of each endpoint nodes ($v$,$u$)
 9:   end if
10: end for
  Channel Assignment Phase 2:
 1: Order all "Visited and Potential" links in $G_t$ in an increas-
    ing order of their priority
 2: for Potential link $e = <v,u>$ in order do
 3:   if Any path $P$ between $u$ and $v$ and $<u,v> \notin P$ then
 4:     Remove $e$ from $G_t$
 5:   else
 6:     $C_v \leftarrow$ Channels that have already been assigned to
        node $v$
 7:     $C_u \leftarrow$ Channels that have already been assigned to
        node $u$
 8:     $c_v \leftarrow \{c \in C_v | F_c < F_i \forall i \in C_v\}$
 9:     $c_u \leftarrow \{c \in C_u | F_c < F_i \forall i \in C_u\}$
10:     $Merge.Channels(c_v, c_u)$
11:   end if
12: end for
```



Algorithm 2 Get.Possible.Channel($e = <u,v>$,C)

```
  Input:
    $e = <u,v>$: The link between nodes $v$ and $u$
  Output:
    $PCh$: A set of possible channels for the given link
Require: $C > 0$
 1: if $v$ and $u$ have free radio then
 2:   $PCh \leftarrow C$
 3: else if at least one node has free radio then
 4:   $Lim.Node \leftarrow$ The node which has no free radio
 5:   $PCh \leftarrow$ Current channels of the $Lim.Node$
 6: else if $v$ and $u$ have no free radio but common channels
    then
 7:   $PCh \leftarrow$ Common channels between $u$ and $v$
 8: else
 9:   $PCh \leftarrow \varnothing$
10: end if
11: return $PCh$
```



Algorithm 3 Find.Best.Channel($e=<u,v>$,PCh,G_f)

```
Input:
  $e = <u,v>$: The link between nodes $v$ and $u$
  $PCh$: A set of possible channels for the given link
  $G_f$: The conflict graph
Output:
  $C$: The best channel for the given link
if $|PCh| > 1$ then
    $MinF \leftarrow I(e)$
    for $c \in PCh$ do
        $E_c \leftarrow E_c \cup e$
        Compute $F_c$ from equation (1)
        if $F_c \leq MinF$ then
            $MinF \leftarrow F_c$
            $C \leftarrow c$
        end if
    end for
else
    $C \leftarrow PCh[1]$
end if
return $C$
```

Without any traffic profile, it is pretty hard to estimate the Utility precisely, but we can have a good estimation by considering a balanced traffic over the network. To compute $U(e)$, independent of specific traffic pattern, We use the shortest path first (SPF) and consider all shortest paths between the gateway and other nodes in the network. The shortest path, between two nodes, is a path with the lowest cost from one node to another.

The cost of a path is the total cost of its participant links while the cost of a link is equal to the inverse of its delivery probability ($1/p_d(e)$) i.e. the expected transmission count over the link, assuming Bernoulli trials (De Couto et al., 2003). We define $U(e)$ equal to the number of times that link e participates in constructing the shortest paths between the gateway and other nodes (Alg. 5).

In a network with $|V|$ nodes consisting one node as a gateway, we have $|V|-1$ paths from all nodes to the gateway. Thus, we estimate the probability of using a link for a transmission to the gateway as the utility of that link ($U(e)$) over the total number of paths ($\frac{U(e)}{|V|-1}$).

Since our channel assignment algorithm tries to prune the topology by deleting some links, it is important to start channel assignment from links with the higher utility. Through our simulation study, we found that in all topologies many links have the utility equal to zero or one. Therefore, after sorting all links based on their utility, links with the same utility must be ordered based on their delivery probability. This sorting can be done by defining the priority $P(e)$ of each link e as:

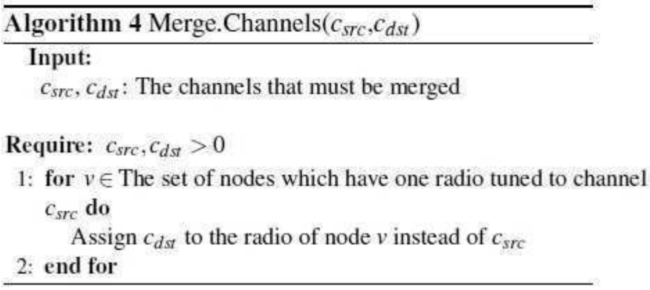
$$P(e)=\frac{U(e)}{|V|-1}+(1-\gamma)*p_d(e)\ \ \forall:e:\in:E \qquad (7)$$

where γ is a tuning parameter subject to $0<\gamma<1$. Big γ prefer links with higher utility while small γ is preferable for networks without any gateway. We use $\gamma=0.9$ throughout this work. Note that $\gamma=1$ means classifying links only considering their utility which may be equal for many links and thus results in an ambiguous classification.

4 Performance Evaluation

In this section, we study the performance of the proposed channel assignment algorithm using the R numerical tool (RDC Team, 2008) and the NS-2 (Fall and Varadhan, 2003) simulator. We use R to compare the capacity and interference properties of different multichannel algorithms (Marina et al., 2010; Subramanian et al., 2008; Draves et al., 2004; Dhananjay et al., 2009). Detailed NS-2 simulations are used to evaluate the performance of the channel assignment algorithms in 802.11-based multi-radio mesh networks. We have add the multi-radio functionality to the physical and MAC layer of 802.11 in NS-2 simulator based on the work done by Aguero-Calvo and Perez-Campo (2007). The routing tables in the NS-2 simulations are obtained using SPF, while considering $1/p_d(e)$ of any link e as its weight.

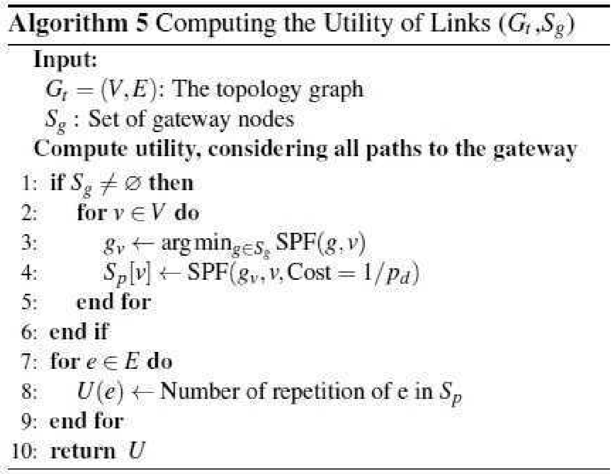
To assess the delivery probability ($p_d(e)$) of link e we use the shadowing propagation model (Rappaport, 2002) with a path loss $\beta=2.7$ and standard deviation $\sigma_{dB}=6$ dB. We have used the default NS-2 values for the other propagation model parameters (see Fall and Varadhan, 2003). In our model we consider that, links exist only when $p_d(e)\geq 0.5$. With the selected values, this delivery probability is achievable if the distance of the link is not longer than 131.5 m.



Algorithm 4 Merge.Channels(c_{src},c_{dst})

```
Input:
  $c_{src}$, $c_{dst}$: The channels that must be merged

Require: $c_{src}, c_{dst} > 0$
for $v \in$ The set of nodes which have one radio tuned to channel $c_{src}$ do
    Assign $c_{dst}$ to the radio of node $v$ instead of $c_{src}$
end for
```



Algorithm 5 Computing the Utility of Links (G_t,S_g)

```
Input:
  $G_t = (V,E)$: The topology graph
  $S_g$ : Set of gateway nodes
Compute utility, considering all paths to the gateway
if $S_g \neq \varnothing$ then
    for $v \in V$ do
        $g_v \leftarrow \arg\min_{g \in S_g} \mathrm{SPF}(g, v)$
        $S_p[v] \leftarrow \mathrm{SPF}(g_v, v, \mathrm{Cost} = 1/p_d)$
    end for
end if
for $e \in E$ do
    $U(e) \leftarrow$ Number of repetition of e in $S_p$
end for
return $U$
```

We consider a network with different number of nodes ($10 \leq N \leq 30$) which are randomly placed in a square field of 300×300 m^2.

We assume the protocol model for interference between wireless links with an interference range equal to 263.06 m (two times the communication-range). Throughout we assume that all nodes are equipped with two radios that can be tuned over 12 non-overlapping channels.

We compare UBCA with three relevant channel assignment algorithms that have been proposed in the literature: the Common Channel Assignment (CCA) (Draves et al., 2004); the Connected Low interference Channel Assignment algorithm (CLICA) (Marina et al., 2010); the distributed channel assignment (ROMA) (Dhananjay et al., 2009) and Greedy Channel Assignment (GCA) (Subramanian et al., 2008).

CCA applies the same channel assignment pattern for all nodes, i.e. the first radio of all nodes is tuned to the first channel, the second radio is tuned to the second channel and so on. Therefore if each node has *r* radio interfaces, regardless of the number of available channels, the network created by CCA always uses *r* channels.

CLICA is a centralized channel assignment which tries to reduce the interference weight of the links while preserving the network topology. CLICA visits the nodes based on their priority which is defined by their distance to a reference node and the number of free radios they have. Here the reference node is the gateway. While assigning a channel to a link, each end-point node will lose one of its free radios and thus during the channel assignment the priority of the nodes will change dynamically. CLICA selects a channel for a link in such a way that leads to the lowest amount of interference weight over that link and all other links which are interfering with it.

ROMA is a distributed algorithm proposed for a network with at least one gateway. At the beginning of the channel assignment each gateway produces a channel sequence $(c_1,c_2,\cdots,c_n)$, and broadcasts it. The node which is *i* hops far from the gateway will select the c_{i-1} and c_i elements of the channel sequence, and tune its radios to the selected channels. Therefore, at the end of the procedure each node will have a common channel with its previous node on the path to the gateway, and a common channel with its neighbors at the same and lower level.

GCA is a centralized algorithm which tries to minimize the interference on wireless links by assigning interfering links to different channels. The algorithm does not consider any gateway and gives the same priority to all links. The greedy algorithm runs in several rounds and in each round it checks all channel-link pairs to find the channel which must be assigned to a link and minimizes the total interference weight in the network. The algorithm ends if it can not decrease the total interference weight in the network anymore.

We also investigate the performance of our protocol UBCA with some changes; first without removing any links, which means that UBCA acts as a topology preserving algorithm (UBCA-TopologyPreserve); and second remove the links with zero *Utility* from the network topology before running the channel assignment and then apply UBCA (UBCA –RemoveFirst).

UBCA-TopologyPreserve is chosen to investigate the importance of link removal in channel assignment. On the other hand UBCA-RemoveFirst is chosen to proof that the link removal should be done in a controlled way to avoid decreasing the network performance.

Single channel network is also used as a base to compare the performance of multi channel networks.

4.1 Topology Properties

4.1.1 Capacity-Gain

We use the maximum number of concurrent transmissions as an estimation for network one-hop capacity (Balakrishnan et al., 2004). We calculate this metric in two steps: First, computing all independent sets in the conflict graph, and then selecting the set which gives us the maximum capacity factor, as we explain in the following.

The simplest way to determine the capacity factor ($C_f(C)$) of a multi-channel network with *C* orthogonal channels, is by considering the cardinality of the largest independent set, *S*, of the conflict graph of each channel $G_f(c)$ (Balakrishnan et al., 2004). To take into account the delivery probability in the capacity metric, we calculate the summation of the delivery probability of the links in each independent set (equation (8)).

$$C_f(c) = \max_{\forall S \subset G_f(c)} \sum_{\forall e \in S} p_d(e) \qquad (8)$$

Links over non-overlapping channels are able to transmit simultaneously, therefore the capacity factor of the network is the total capacity acquired over all channels (equation (9)).

$$C_f(C) = \sum_{\forall c \subset C} C_f(c) \qquad (9)$$

We define the capacity gain of the multi-channel network in relation to the single channel network in equation (10), where $C_f(C)$ and $C_f(1)$ represent the capacity factor of a multi-channel network with C orthogonal channels, and a single channel network respectively.

$$CapacityGain = \frac{C_f(C)}{C_f(1)} \qquad (10)$$

4.1.2 Network Interference

We use two metrics to show the interference characteristic in a multi-channel network: the collision domain size; and the link conflict weight.

Size of collision domains: A collision-domain is a subset of links in which all links collide each other if they transmit simultaneously. A collision-domain in the conflict graph is a complete subgraph or clique of vertices. All vertices in a clique are connected pairwise, therefore the set of their corresponding links in the topology graph make a potential collision domain.

Maximum average interference weight: The interference weight for a link is the number of links in its interference range (Section 2.1). We calculate this metric taking the maximum of equation (1) over C. This metric is important since a link with the maximum interference weight could be a potential bottleneck for the network. In CLICA (Marina et al., 2010) the authors use a similar metric as the objective function.

4.2 802.11 Based Multi-Radio Performance

We use NS-2 simulator to evaluate multi-radio networks created by different channel assignment algorithms in terms of aggregate throughput, packet delivery ratio, and average delay. In each network we produce random traffic flows. We use two types of sources: CBR traffic with fixed rate at 100 kbps and packet size equal to 1 kB; and TCP traffic with packet size equal to 1.4 kB. We consider two traffic profiles: *gateway profile* consisting of flows from the gateway to randomly selected nodes; and *random profile* consisting of flows between random pairs of nodes. The simulation time was set to 100 s. RTS/CTS mechanism is enabled. For each topology with different number of nodes, we repeated the simulation for 50 different random placements of the nodes, and report the average with the confidence intervals.

Aggregate Throughput: For TCP traffic we calculate the aggregate throughput (Mbps), dividing the total received traffic by the duration time of TCP flows.

Packet Delivery Ratio (PDR): We consider the number of correctly received packets with respect to the amount of sent packets.

Average Delay: For all received CBR packets, we calculate the average delay to verify the ability of the network to use non interfering channels to deliver data with less contention.

4.3 Results

As explained before, Fig. 2-4 have been computed with the R numerical tool (RDC Team, 2008), analyzing the properties of the topology graphs obtained by the channels assignment algorithms under study. Note that these properties do not reflect the fact of having a gateway. Fig. 5-17, on the other hand, show the results of analyzing the traces obtained using the NS-2 simulator (Fall and Varadhan, 2003) .

Fig. 2 shows the capacity gain of multi-channel networks (equation (10)). The results are produced by different channel assignment algorithms varying the number of nodes. Using CCA the capacity gain is bounded to two, since CCA only uses two channels throughout the network (Draves et al., 2004).

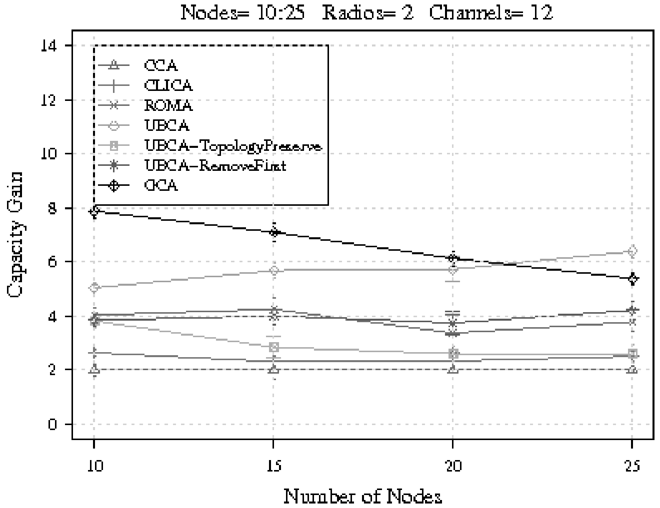


Figure 2: Network Gain

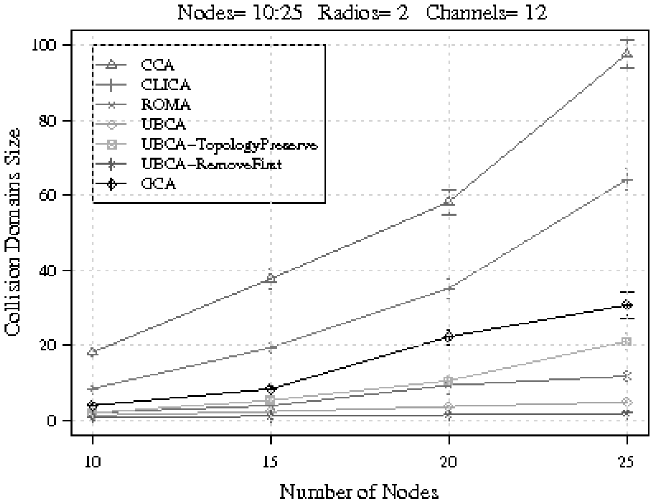


Figure 3: Size of the collision domains

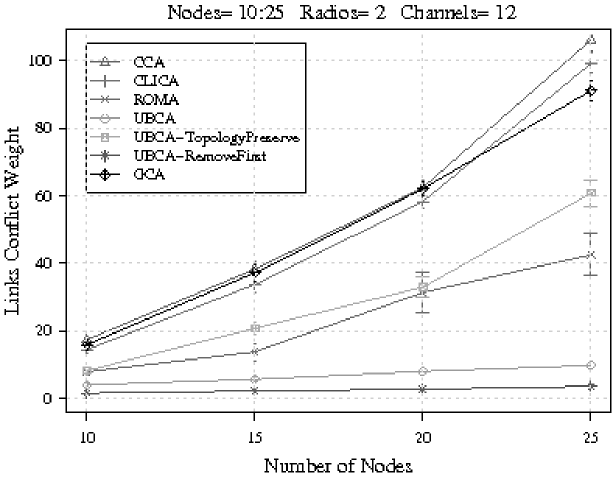


Figure 4: Maximum interference weight

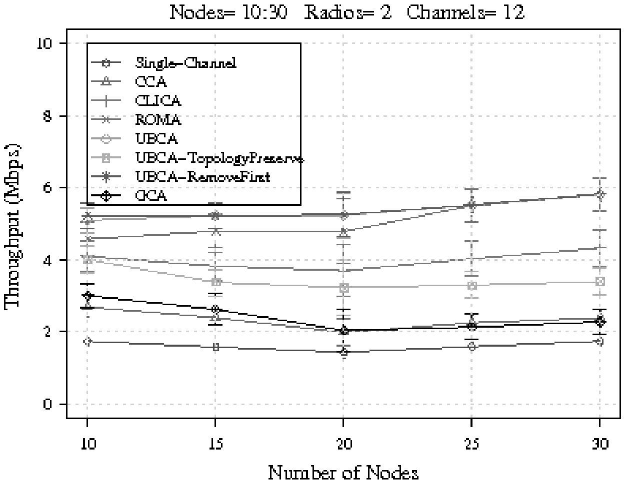


Figure 5: Aggregate throughput for TCP traffic (gateway profile)

The figure shows that GCA outperforms the other mechanisms, but it performs close to UBCA as the number of nodes increases because UBCA removes some links and reaches a more diverse solution. The good performance of UBCA is because two main reasons: first, considering the delivery probability of the links during the channel assignment; and second, removing some useless links from the original network. As expected, removing useless links will reduce the collision domain size, thus resulting a considerable increase in capacity gain, but it must be considered that removing many links from the topology as it has been done in UBCA-RemoveFirst does not necessarily improves the capacity gain since it decreases the number of possible concurrent transmissions. As expected UBCA-TopologyPreserve performance is close to the other topology preserving algorithm, CLICA.

Fig. 3 shows the average size of the collision domains for different number of nodes. CLICA and GCA are successful in reducing the size of collision domains compared to CCA, but the reduction is much higher with UBCA and ROMA. Note that, even after increasing the network density, the size of collision domains do not change much for the topologies created by UBCA and ROMA.

Fig. 4 depicts the maximum interference weight of the links for different network densities. The figure shows that by increasing the network density, the maximum interference weight do not change significantly for UBCA, while for the other mechanisms increases rapidly. Although CLICA and GCA are designed to minimize this metric, they couldn't achieve a reasonable result compared to UBCA since they try to preserve the topology and as explained in section 2.1, this approach couldn't achieve a diverse-channel assignment hence unsuccessful in eliminating the

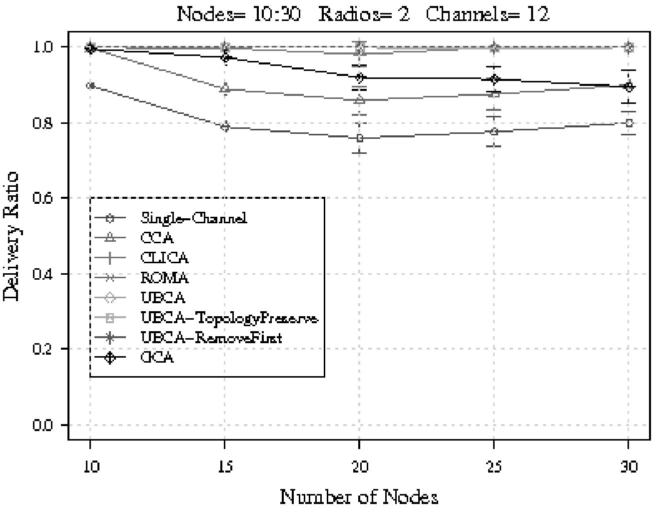


Figure 6: Average packet delivery ratio for CBR traffic (gateway profile) vs. number of nodes

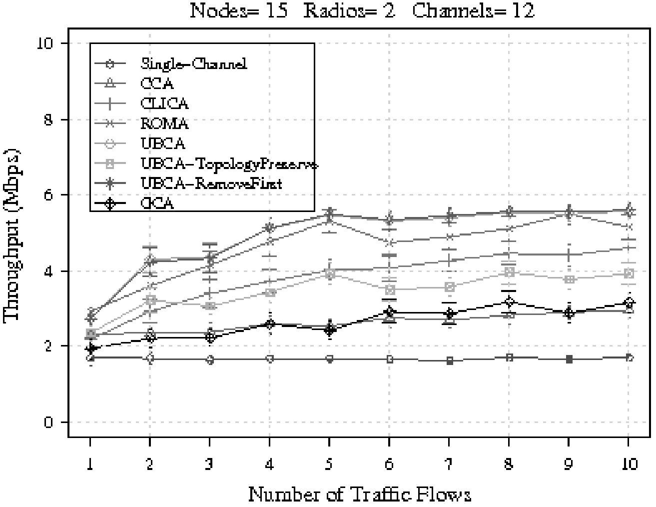


Figure 8: Aggregate throughput for TCP traffic (gateway profile) vs. number of flows

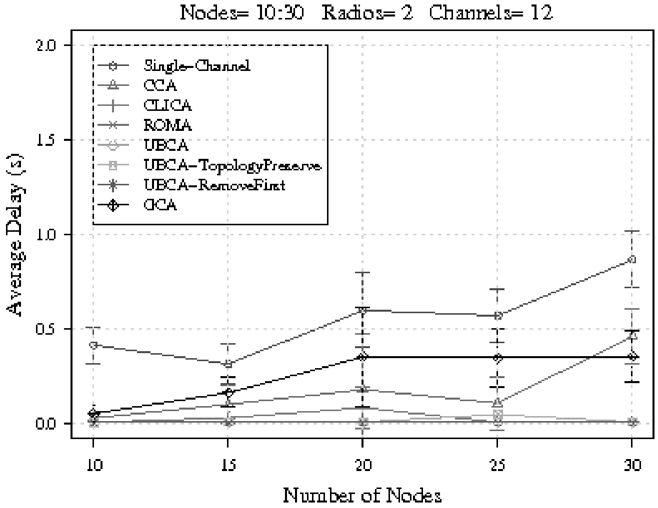


Figure 7: Average delay for CBR traffic (gateway profile) vs. number of nodes

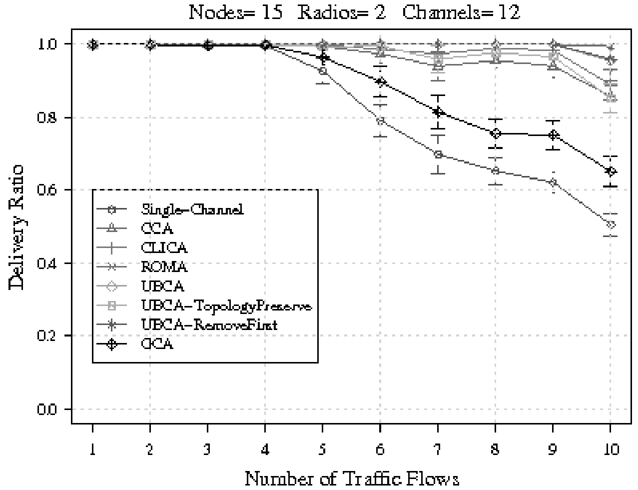


Figure 9: Average packet delivery ratio for CBR traffic (gateway profile) vs. number of flows

interference over links. The same reason justifies the higher value of maximum interference weight for UBCA-TopologyPreserve compared to the original protocol (UBCA) and UBCA-RemoveFirst.

To investigate the performance of the proposed channel assignment in a general situation, we first run the simulation for the gateway profile. Fig. 5 depicts the aggregate TCP throughput with different number of nodes and 5 TCP flows. The figure shows that due to a significant increase in the network capacity (see Fig. 2), UBCA and ROMA outperform the other mechanisms especially in dense topologies. Note that UBCA-RemoveFirst performs the same as UBCA, since it removes all links that never used to reach the gateway and the resulted CA solution will be same as UBCA for gateway traffic profile.

Fig. 6 shows that the packet delivery ratio in network topologies created by UBCA, CLICA and ROMA are also much better than CCA or GCA. Moreover, in contrast to other mechanisms, delivery ratio in UBCA, CLICA and ROMA is rather insensitive to the network density. GCA as expected does not perform well for gateway traffic profile since it does not consider the gateway placement for channel assignment.

Fig. 7 shows that by using UBCA, CLICA and ROMA the average delay for CBR traffic is significantly lower than with the other algorithms. This confirms the smaller size of collision domains, and lower interference weight obtained in Fig. 3 and Fig. 4 respectively.

We re-run the simulation by considering different number of traffic flows. The network consists of 15 nodes randomly distributed in a square field of 300×300 m^2. Fig. 8 shows the aggregate TCP throughput with different number of traffic flows. As expected, due to the proper channel assignment in ROMA and UBCA compared to CLICA and CCA, the aggregate throughput improves significantly. UBCA gets almost the same result as ROMA. Recall that ROMA is designed to optimize the gateway paths.

Fig. 9 shows that the packet delivery ratio in network topologies created by UBCA and ROMA is higher than the others. Additionally, in UBCA and ROMA the rate of decrease in the packet delivery ratio with respect to the increase of the traffic flows is much lower than the others. Finally, Fig. 10 shows that in UBCA and ROMA the average delay for CBR traffic is lower than the other algorithms.

These results are justifiable by taking into account the fact that ROMA optimizes the paths to the gateway, and UBCA estimates the utility of the links based on their frequency to access the gateway.

We repeat the simulation for random profile of traffic flows. The aggregate throughput is shown in Fig. 11. The result shows that UBCA performs better for random traffic compared to others. Note that unlike the results obtained for gateway traffic profile, the network aggregate throughput using UBCA-RemoveFirst for random traffic profile is worst than the original protocol (UBCA) which shows that removing all links with the Utility equal to zero from the network topology may decrease the network throughput.

We investigate the effect of increasing the number of radio interfaces on each node on the performance of the channel assignment protocols (except ROMA). Note that ROMA is proposed for a dual-radio network. Fig. 12 shows the aggergate network throughput for 5 TCP flows for different number of radio interfaces at each node. The figure shows that, increasing the number of radio interfaces at each node improves the performance of GCA significantly. It also shows that UBCA outperforms other CA mechanisms, but the performance of the network does not change a lot while using more radio interfaces for each node.

Fig. 13 is obtained for a network with two radio interfaces at each node while different number of channels are available.

The figure shows that as the number of channels increase the performance of multi radio

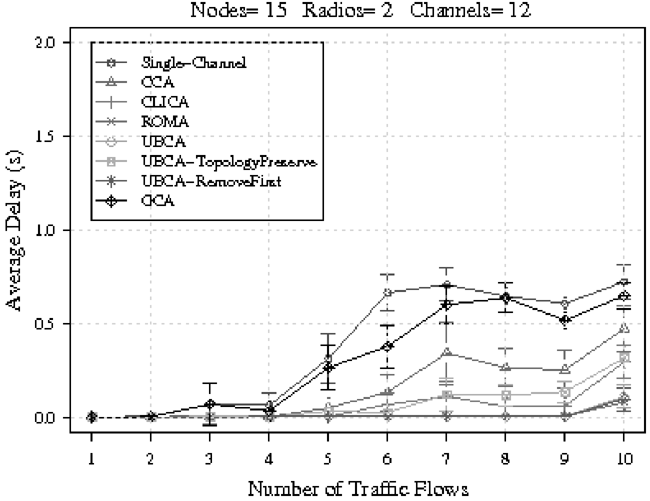


Figure 10: Average delay for CBR traffic (gateway profile) vs. number of flows

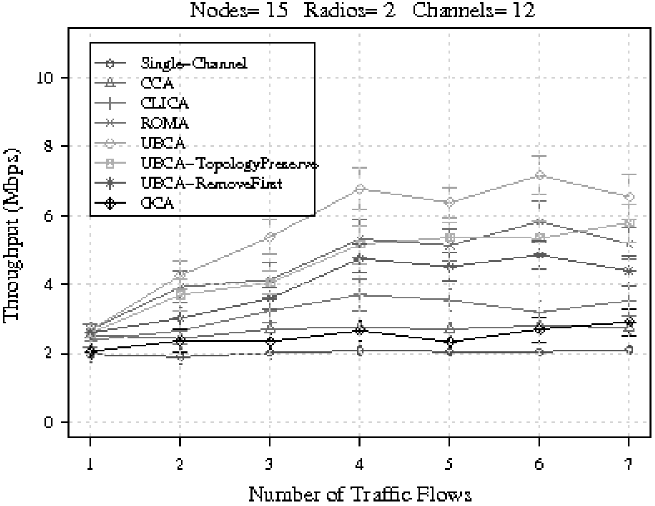


Figure 11: Aggregate throughput for TCP traffic (random profile) vs. number of flows

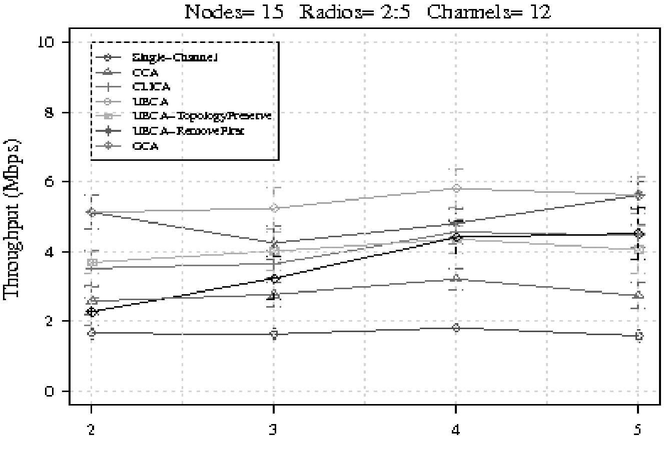


Figure 12: Aggregate throughput for TCP traffic (gateway profile) vs. number of radio interfaces per node

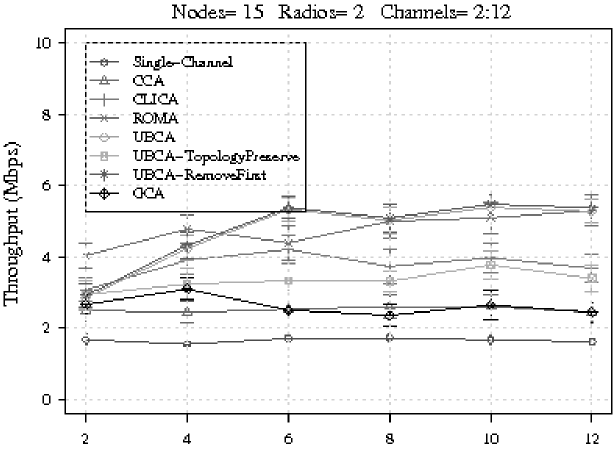


Figure 13: Aggregate throughput for TCP traffic (gateway profile) vs. number of available channels

networks improves but it does not improve as the number of channels increases to more than 6. ROMA and UBCA performs better than others.

We conclude that based on the simulation results, UBCA builds a network topology with low interference with a small number of radios per node. We showed that removing some links from the network topology will lead to a better performance, even if it may cause an increase in the length of some paths between nodes. The results confirm that performing channel assignment while disregarding the gateway placement does not improve the network performance in a scenario where the traffic is oriented toward the gateway.

It is also important to note that removing the wireless links from the topology is possible if the end point nodes do not share any common channel. Therefore it must be done during the channel assignment. In other words if the links are removed from the topology before channel assignment, the CA mechanisms might make those removed links to be available later by putting the neighboring nodes on a common channel. Fig.14 shows the number of removed links which are added to the topology after the channel assignment. The figure shows that if CA is not aware of the links which must be removed the links removal can not be performed correctly.

4.3.1 The Impact of γ

Recall that UBCA uses a weighted metric for assigning the priority to wireless links before channel assignment (Section 3.1). As we have explained previously, the priority metric is a weighted sum of the utility and the delivery probability of the wireless links. The weight given to the utility (U) is denoted by γ, while the weight given to the delivery probability (p_d) is $1-\gamma$.

To investigate the impact of γ on the performance of UBCA, we repeated the simulation with $0.1 \leq \gamma \leq 1$ for three different network densities (N=10,25,50).

Fig. 15 shows that by increasing γ, UBCA achieves better network throughput for TCP traffic generated in the gateway profile. Obviously the impact of different value of γ is more significant for dense networks (N=25,50), because big γ leads to decrease the interference over the links which are close to the gateway.

Fig. 16 - 17 show that a larger value of γ leads to better packet delivery ratio with less delay for CBR traffic.

These simulation confirms that, since a large γ forces the channel assignment to favor the links with higher utility, for the gateway profile UBCA achieves better performance with increasing γ. For the random profile we have not obtained significant differences for different values of γ.

5 Discussion

In this section, we discuss the importance of considering the quality of wireless links during the channel assignment process with more details.

Recall that we associate two properties with each wireless link e: the delivery probability ($p_d(e)$), defined in section 2.2; and the Utility ($U(e)$), defined and calculated in section 3.1.

We first study the relation between the quality of the path to the gateway with the throughput and bottleneck delay using queuing model.

We then evaluate the distribution of $U(e)$ over several random network topologies. Results confirm that links close to the gateway have much higher utility than other links. Therefore giving higher priority to links with higher utility to occupy better channels, leads to better resource allocation.

5.1 Bottleneck Delay and Throughput

Consider a mesh network with one gateway, we assume that all data requests from mesh routers aim to the gateway, therefore the gateway is treated as a service station in queuing system. We further assume that the gateway has an infinite buffer since the bandwidth of wired infrastructure, which connects the gateway to the Internet, is much more bigger than the wireless bandwidth between mesh nodes.

We define bottleneck delay, D_s, as the average delivery delay of data requests at the gateway while the service time of each request is identical and equal to T_s. We also define the throughput, λ, as the maximum data requesting frequency from all mesh nodes to the gateway.

We assume that all mesh nodes (excluded from gateway) generate data request at the constant rate of τ. Considering the delivery probability of a wireless link ($p_d(e)$), if τ is the rate of data

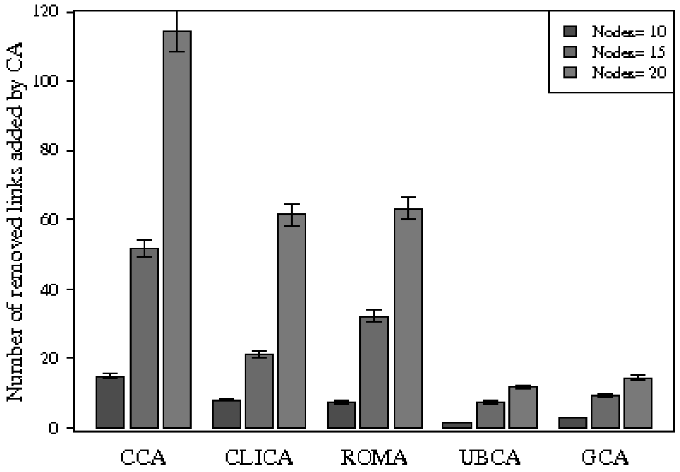


Figure 14: Number of zero utility links which are added by CA to the topology

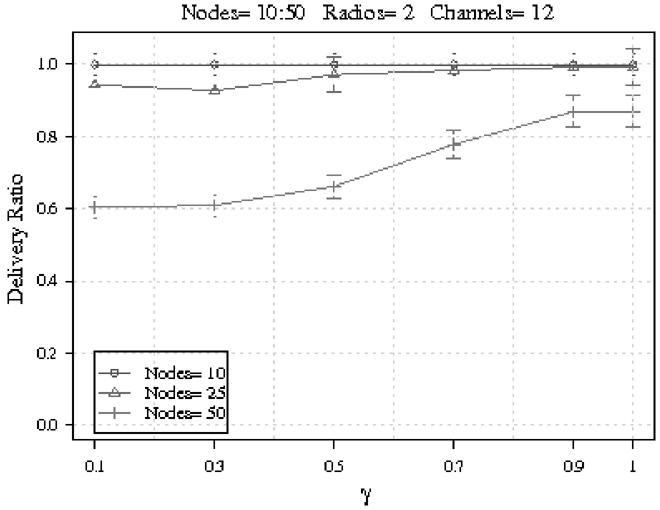


Figure 16: Average packet delivery ratio for CBR traffic using UBCA varying γ (gateway profile)

produced at one endpoint node then the data arrival at next node could be considered with the average rate of $p_d(e)\tau$.

Since the delivery probability is sufficiently smaller than τ we can model the data arrival at the next node as a Poisson distribution with mean $\lambda_e = p_d(e)\tau$.

Suppose that all nodes are placed over one path to the gateway, and as a result of the channel assignment mechanism, all links are active over non-interfering channels. Since all transmissions can be done simultaneously, according to the additivity property of Poisson distribution, data requests arrival at gateway will be of Poisson distribution with mean value of λ:

$$\lambda = \tau \sum_{e \in S_e} p_d(e) \qquad (11)$$

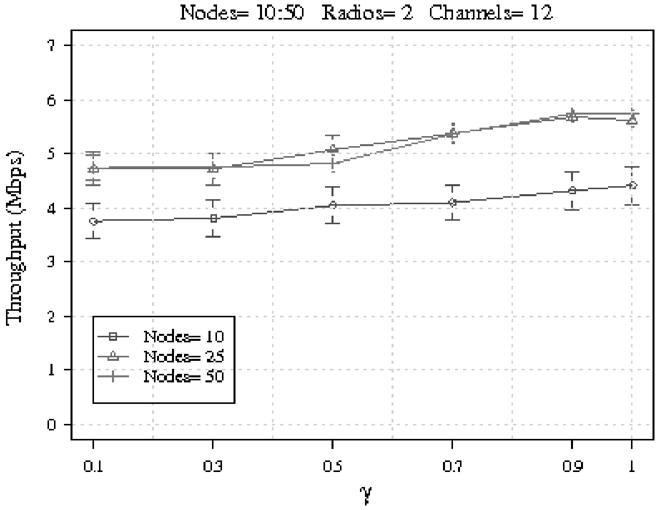


Figure 15: Aggregate throughput for TCP traffic using UBCA varying γ (gateway profile)

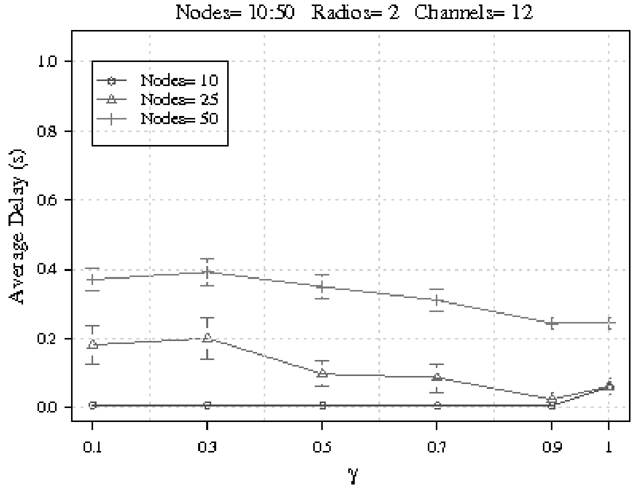


Figure 17: Average delay for CBR traffic using UBCA varying γ (gateway profile)

where S_e is the set of links which construct the path to the gateway. Equation (11) indicates that the quality of the shortest path to the gateway determines the upper bound for the throughput.

For a gateway with an infinite buffer; the constant service time of a data request; and data requests arrival at Poisson distribution, we can model the data incoming and outgoing as in an M/D/1 queue. Considering constant service time T_s, the expected value of T_s, is identical ($E[T_s]=T_s$) and the variation of T_s is zero. According to Wu et al. (2006), the bottleneck delay for a linear topology can be estimated as:

$$D_s = T_s + \frac{T_s^2 \lambda}{2(1 - T_s \lambda)} \qquad (12)$$

The data request arrival rate shows the bottleneck throughput and since the value of the *utilization factor* ($\rho=T_s\lambda$) should be smaller than one (in order to arrive at a steady state), the data arrival rate should be smaller than $\frac{1}{T_s}$. Therefore the maximum data rate for each mesh router is bounded by:

$$\tau=\frac{1}{T_s \sum_{e\in S_e} p_d(e)} \qquad (13)$$

We consider one path to the gateway which consists of ten mesh routers uniformly placed in a chain. We assume that after channel assignment each node can establish transmission only with its successive nodes on the path, moreover all links are established over non-overlapping channels. We change the distance between nodes to have different values for data delivery probability of wireless links, while the number of nodes is kept intact. As the distance between nodes gets larger the cost of the path to the gateway increases.

Note that, all nodes (except the gateway) produces data requests at the constant rate of τ. We run the experiment varying the value of τ in the range of 20 to 100. The service time, T_s, is assumed to be 1 ms for all requests.

Fig. 18 shows the maximum data arrival rate, λ, at gateway, for different value of the data rate at each node, τ. As expected, the throughput drops as the distance between nodes gets larger. Fig. 18 depicts that, the higher the data rate, the more sensitive to the distance.

Fig. 19, on the other hand, shows the bottleneck delay for different data rates of mesh nodes. As expected the higher data arrival rate the bigger delay it suffers at the gateway to get serviced. Note that for τ=100 in a scenario where the distance between nodes is 50 *m*, the bottleneck delay goes to infinity. This phenomenon happens because for this topology, according to equation (13), the bottleneck data rate value is about 100, which prevent the system from reaching the steady state.

5.2 Utility Distribution

In a WMN with a gateway, the traffic distribution is usually skewed, since most of the user traffic is directed to/from the gateway. Therefore links close to the gateway should be selected with higher probability by the routing protocol. We define utility ($U(e)$) to estimate the probability of using a specific link, *e*, for a random traffic request (see section 3.1).

Note that, since a weak link with $p_d(e)<0.5$ is not favorable for transmission, in our model we consider links with the delivery probability equal or greater than 0.5 (Section 2.2). In practice, routing protocols favor links which offer higher performance. Therefore most of the links with lower quality are likely useless.

We perform a simple experiment in our simulation area to verify the different types of links considering the delivery probability and the utility. We assume a network with one gateway and 24 nodes, which are distributed randomly in a 300*300 m^2 square field. The gateway is fixed and placed on the right edge of the field. We calculate the average utility of links in different area of the network over 1000 different network topologies. Fig. 20 shows a perspective view of average links' utility, for all networks. As expected, in the neighborhood areas of the gateway, the average utility of the links has a considerable bigger value compared to other regions of the network. According to our experiments, 79.54% of links have the utility equal to 0, it means that, none of the network nodes use them to access the gateway, while 0.2% of links participate in more than 12 paths ($U(e)\geq 12$), i.e. the probability of using those links, for a transmission, is more than 0.5.Note that, the average of links utility ($\overline{U}$), in this network, is 0.42. To generalize our result, we

Category	Links Delivery Probability	Links Utility	Considering the paths to the gateway	Considering all paths
1	$\geq \overline{p_d}$	$\geq \overline{U}$	8.56%	14.39%
2	$\geq \overline{p_d}$	$< \overline{U}$	43.01%	37.18%
3	$< \overline{p_d}$	$\geq \overline{U}$	11.89%	20.03%
4	$< \overline{p_d}$	$< \overline{U}$	36.54%	28.40%

Table 1: The percentage of links, distributed in four categories based on their delivery probability and utility in a network

calculate the utility of links regardless of the gateway, by considering not only the gateway-paths, but also, every other shortest paths between all pairs of nodes. Fig. 21 shows the empirical distribution function (ECDF) of links utility in this scenario. The figure shows that after considering all possible paths, a considerable amount of links have the utility close to zero.

We use the average of delivery probability ($\overline{p_d}$) and the average of utility ($\overline{U}$), to categorize the links based on their attributes: first the links with big delivery probability and utility; second the links with big delivery probability but small utility, third the links with small delivery probability but big utility; and forth the links with small delivery probability and small utility. Table 1 shows the percentage of links in four categories and for two mentioned scenarios: considering only paths to the gateway; and considering all paths between all pairs of nodes. As shown in table 1, the forth category contains at least 28% of the network links, which have the lowest amount of p_d, and are less important in the network (small U). Note that, this result is almost the same for a dense network.

From the result in this experiment, it is obvious that assigning high priority to the links in categories with higher U and p_d, to occupy better channels, will lead to better channel assignment, since these links are more probable to participate in traffic transmissions. On the other hand, it is important to disregard links in category four during the channel assignment. Removing weak links from the topology may lead to even better performance due to the following reasons:

- It reduces the potential interference over good links,
- It relaxes the restriction over channel assignment to make a common channel for weak links by considering finite number of radios per node,
- It doesn't have a big effect on the communication between nodes.

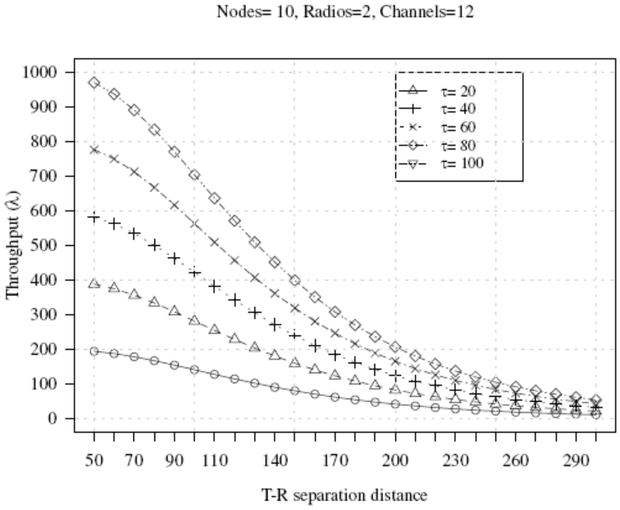


Figure 18: Maximum data request arrival at gateway for different data arrival rate at mesh nodes

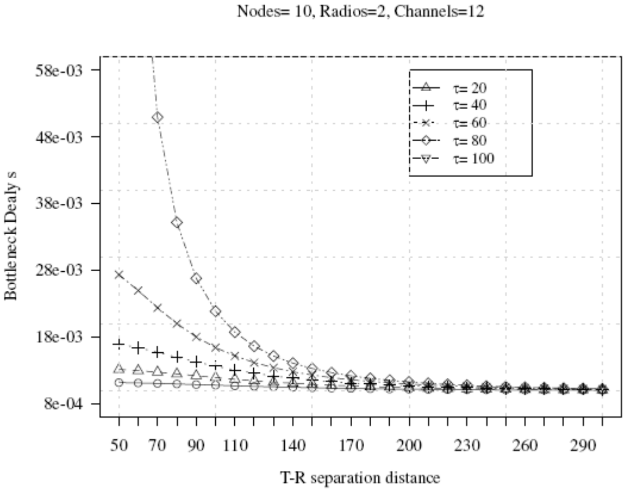


Figure 19: The bottleneck delay for different data arrival rate

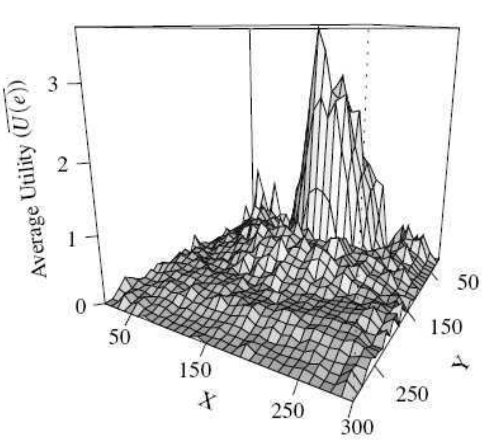


Figure 20: The distribution of links utility

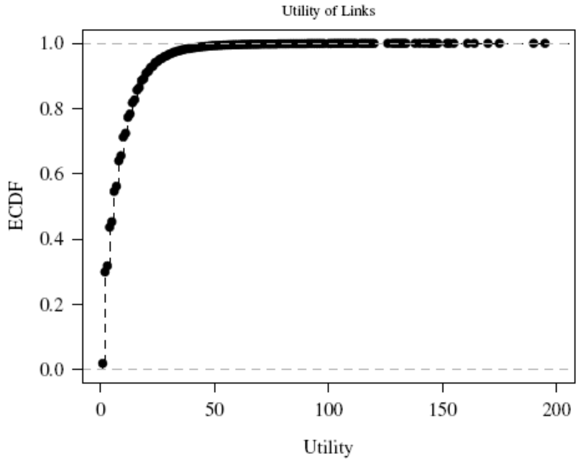


Figure 21: The ECDF of links utility

This simple experiment shows that; it is extremely unfair and illusive to threat all links identically.

5.3 Related Work

The static channel assignment problem is well studied in recent years and has been addressed in several proposals (Crichigno et al., 2008; Alicherry et al., 2006; Avallone and Akyildiz, 2008; Dhananjay et al., 2009; Marina et al., 2010. The detail classifications of channel assignment methods is presented in (Crichigno et al., 2008; Si et al., 2010. The core idea of all proposed algorithms is to use the available channels to eliminate the interference of neighboring transmissions. A simple approach for utilizing two channels in a dual-radio network is presented in Draves et al. (2004), while the main focus of authors is on modifying the routing parameters to benefit from multi-channel structure. Further investigations in Marina et al., (2010) and Raniwala et al. (2004) show that it is possible to increase the performance of the multi-channel network more than two factors by applying smart channel assignment algorithms. Raniwala et al. (2004), presented a traffic aware channel assignment. Given the network topology and the traffic profile, the channel assignment binds each radio to a channel such that the available bandwidth on each link is proportional to its expected load. If the traffic loads change over time the algorithm must perform channel reassignment. Skalli et al. (2007) also formulated the channel assignment problem considering the traffic load of each node. Their priority based scheme, uses a common channel on all nodes to exchange control messages and to maintain the network connectivity. It gives higher priority to nodes close to the gateway to occupy better channels. But using traffic dependent schemes in presence of dynamic traffic profile is very challenging since the channel assignment output affects the routing protocol decision and in return changes the traffic load over links. Marina et al. (2010) and Subramanian et al. (2008) formulated the static and traffic independent channel assignment as a topology control problem, and develop their approaches subject to minimize the link conflict weight. The channel assignment mechanisms proposed in (Marina et al., 2010; Subramanian et al., 2008) assign channel to all link to preserve the network topology and do not consider the different quality offered by wireless links. As shown in the earlier version of this paper (Amiri Nezhad and Cerdà-Alabern, 2010), this method leads to a suboptimal solution. Avallone et al., (2008) formulated the problem to reduce the size of collision domains by assigning links to non-overlapping channels. However they do not consider the delivery probability of the wireless links. Recently Dhananjay et al., (2009) proposed a distributed channel assignment and routing which takes the links delivery ratio into accounts to find the shortest path to the gateway. In the proposed algorithm each node follows the channel assignment pattern which is propagated by the gateway i.e. the algorithm optimizes the paths to the gateway, thus the big size of collision domains between links far from the gateway, especially in a dens network leads to a lower performance in an arbitrary traffic profile.

6 Conclusions and Future Work

In this paper we have studied the channel assignment problem in a multi-radio multi-channel mesh network. We have showed that considering the utility of the links makes it possible to estimate the usefulness of each link regardless of the traffic profile. We have presented a new algorithm, called UBCA, which assigns channel to links considering their utility without a tight constraint on preserving the topology. We have done a performance evaluation comparing UBCA with other relevant channels assignment algorithms proposed in the literature. In our study, we have used a numerical tool to analyze the properties of the topology graphs, and detailed NS-2 simulations. Our numerical results demonstrate the effectiveness of UBCA in exploiting channel diversity for reducing the interference over wireless links with a small number of radios per node. The simulation results show that our approach increases the performance of the multi-radio mesh network significantly. Additionally, UBCA provides a considerable decrease in the size of collision domain and thus significant increase in the network capacity. The NS-2 simulations proved that pruning the network from some useless links leads to a better channel utilization and thus reduces the average delay and increases in the packet delivery ratio and throughput.

Our future work will focus on formulating the proposed channel assignment problem to obtain the optimal bound for network capacity and interference. We will seek for an optimal solution while considering the delivery probability of wireless links and relaxing the constraint on topology preserving. We will consider the problem of estimating the delivery probability of wireless links while the quality of a link is different from each side. We will also investigate the distributed version of UBCA with the focus on traffic pattern and different data transmission rate on each node.

7 Acknowledgments

This work was partially supported by the EuroNF NoE and by Spanish grants TIN2010-21378-C02-01 and 2009-SGR-1167.

References

Aguero-Calvo R, and Perez-Campo J (2007). *Adding Multiple Interface Support in NS-2*. University of Cantabria, Cantabria, Spain.

Alicherry M, Bhatia R, and Li LE (2006). Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks. 24(11):1960–1971.

Amiri Nezhad M and Cerdà-Alabern L (2010). Utility based channel assignment mechanism for multi radio mesh networks. In *Proceedings of the 8th ACM international workshop on Mobility management and wireless access*, MobiWac '10, pages 68–74, New York, NY, USA. ACM.

Avallone S and Akyildiz IF (2008). A channel assignment algorithm for multi-radio wireless mesh networks. *Comput. Commun.*, 31(7):1343–1353.

Balakrishnan H, Barrett C, Kumar V, Marathe M, and Thite S (2004). The distance-2 matching problem and its relationship to the mac-layer capacity of ad hoc wireless networks. *Selected Areas in Communications, IEEE Journal on*, 22(6):1069 – 1079.

Crichigno J, Wu M, and Shu W (2008). Protocols and architectures for channel assignment in wireless mesh networks. *Ad Hoc Networks*, 6(7):1051–1077.

De Couto DSJ, Aguayo D, Bicket J, and Morris R (2003). A high-throughput path metric for multi-hop wireless routing. In *Proceedings of the 9th ACM International Conference on Mobile Computing and Networking (MobiCom '03)*, San Diego, California.

Dhananjay A, Zhang H, Li J, and Subramanian L (2009). Practical, distributed channel assignment and routing in dual-radio mesh networks. In *SIGCOMM '09: Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, pages 99–110, New York, NY, USA. ACM.

Draves R, Padhye J, and Zill B (2004). Routing in multi-radio, multi-hop wireless mesh networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 114–128. ACM New York, NY, USA.

Fall K, and Varadhan K (2003). The ns manual (formerly ns notes and documentation).

Gupta P, and Kumar P (2000). The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388 –404.

Jain K, Padhye J, Padmanabhan VN, and Qiu L (2003). Impact of interference on multi-hop wireless network performance. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 66–80, New York, NY, USA. ACM.

Marina MK, Das SR, and Subramanian AP (2010). A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks. *Computer Networks*, 54(2):241 – 256.

Mohsenian-Rad AH and Wong VWS (2007). Joint logical topology design, interface assignment, channel allocation, and routing for multi-channel wireless mesh networks. 6(12):4432–4440.

Ramachandran KN, Belding EM, Almeroth KC, and Buddhikot MM (2006). Interference-aware channel assignment in multi-radio wireless mesh networks. In *Proc. 25th IEEE Int. Conf. Computer Communications INFOCOM 2006*, pages 1–12.

Raniwala R and Chiueh T (2005). Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network. *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and*

Communications Societies. Proceedings IEEE, 3:2223–2234.

Raniwala A, Gopalan K, and Chiueh T (2004). Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 8(2):50–65.

Rappaport TS (2002). *Wireless Communications: Principles and Practice (2nd Edition)*. Prentice Hall PTR, 2 edition.

Shao B, Tao J, and Wang F (2010). Static channel assignment with the physical interference model for maximum capacity in multi-radio multi-channel wireless mesh networks. In *Proc. 9th Int Grid and Cooperative Computing (GCC) Conf*, pages 338–343.

Si W, Selvakennedy S, and Zomaya AY (2010). An overview of channel assignment methods for multi-radio multi-channel wireless mesh networks. *J. Parallel Distrib. Comput.*, 70:505–524.

Skalli H, Ghosh S, Das SK, and Lenzini L (2007). Channel assignment strategies for multiradio wireless mesh networks: Issues and solutions. *Communications Magazine, IEEE*, 45(11):86–95.

Subramanian A, Gupta H, Das S, and Cao J (2008). Minimum interference channel assignment in multiradio wireless mesh networks. *IEEE Transactions on Mobile Computing*, pages 1459–1473.

RDC Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Wu H, Yang F, Tan K, Chen J, Zhang Q, and Zhang Z (2006). Distributed channel assignment and routing in multiradio multichannel multihop wireless networks. 24(11):1972–1983.

Int. J. Ad Hoc and Ubiquitous Computing, Vol. 0, No. 00, 2011

A Semi-Dynamic, Game Based and Interference Aware Channel Assignment for Multi-Radio Multi-Channel Wireless Mesh Networks

Maryam Amiri Nezhad*

dept. Computer Architecture
Universitat Politecnica de Catalunya
Barcelona, Spain
Fax:34-93-4017055
E-mail: maryam@ac.upc.edu
*Corresponding author

Llorenç Cerdà-Alabern

dept. Computer Architecture
Universitat Politecnica de Catalunya
Barcelona, Spain
Fax:34-93-4017055
E-mail: llorenc@ac.upc.edu

Boris Bellalta

dept. Information and Communication Technologies
Universitat Pompeu Fabra
Barcelona, Spain,
E-mail:boris.bellalta@upf.edu

Manel Guerrero Zapata

dept. Computer Architecture
Universitat Politecnica de Catalunya
Barcelona, Spain,
Fax:34-93-4017055
E-mail: guerrero@ac.upc.edu

Abstract: Channel assignment has been extensively researched for multi-radio wireless mesh networks, but it is still very challenging when it comes to its implementation. In this paper we propose a semi-dynamic and distributed channel assignment mechanism called SICA (Semi-dynamic Interference aware Channel Assignment) based on game theory formulation. SICA is an interference aware, distributed channel assignment which preserves the network connectivity without relying on a common channel nor central node for coordination between mesh routers. SICA applies a real time learner algorithm which assumes that nodes do not have perfect information about the network topology. To the best of our knowledge this is the first game formulation of channel assignment which takes the co-channel interference into account. We have simulated SICA and compared against other channel assignment mechanisms proposed in the literature. Simulation results show that SICA outperforms other mechanisms.

Keywords: Channel Assignment; Multi-Radio; Multi-Channel; Wireless Mesh Network; Game Theory; Real-Time Learning;

Biographical notes: Maryam Amiri Nezhad received her MSc in Computer Engineering from Sharif University of Technology (SUT), in 2006. Currently she is doing her PhD in Computer Architecture at the Technical University of Catalonia (UPC). Her areas of interest are mesh networks and channel assignment protocols.

Llorenç Cerdà-Alabern is associate professor at the Computer Architecture Department, Technical University of Catalonia (UPC). He obtained his Ph.D. in Telecommunication Engineering in 2000 from the UPC. His areas of interest include protocols for wireless ad hoc and mesh networks (MAC and routing layers). He has published papers in international conferences and journals and participated in National and EU funded research projects.

Boris Bellalta received his MSc degree in Telecommunications from the Universitat Politecnica de Catalunya (UPC) in 2002 and the Ph.D. in Information and Digital Communications from the Universitat Pompeu Fabra (UPF) in 2007, where he combined the Ph.D. studies with a full-time assistant professor position. Since 2007, he is a post-doc researcher and full time lecturer at UPF. His main research interests are in the area of traffic engineering, queuing models and wireless communications.

Manel Guerrero Zapata is an Associate Professor (Profesor Contratado Doctor) in the Computer Architecture Department (DAC) at the Technical University of Catalonia (UPC). His research interests include network security, wireless networks, and routing protocols. He is the author of Secure Ad hoc On-Demand Distance Vector (SAODV) routing protocol and of Simple Ad hoc Key Management (SAKM) scheme. Manel Guerrero received his PhD, M.S. and B.S. in Computer Science from the Technical University of Catalonia (UPC) in 2006, 1999 and 1997 respectively. From 1998 to 2003 he worked at the Nokia Research Center in Helsinki (first as Assistant Research Engineer, then as Research Scientist, and finally as Senior Research Scientist). From 2003 to 2005 he worked as an assistant professor at the Pompeu Fabra University (UPF) in Barcelona.

1 Introduction

Wireless Mesh Networks (WMNs) are supposed to be the next Internet back-haul providing network connectivity for end users through multi-hop forwarding. However due to the increasing number of devices sharing the same spectrum brand, interference is one of the important factors that can degrade the performance of the mesh networks (Gupta & Kumar 2000). Interferences can be grouped in two types: external and internal. The external interference appears when two or more coexisting wireless networks work in the same frequency channel (Fig. 1). Although it can be eliminated by using different non-overlapping channels offered in IEEE 802.11 (IEEE-SA 2007), it requires that the different networks agree on the channel distribution. Additionally, as the number of non-overlapping channels is reduced, the scenario where the external interference can be completely avoided is unrealistic. By internal interference we refer to the time overlapping transmissions by nodes of the same network which can result in collisions or transmissions errors. Both external and internal interferences limit the system performance (Si et al. 2010, Ramachandran et al. 2006).

Multi-radio technologies are well known to offer a significant improvement in capacity through the use of multiple channels offered by IEEE 802.11 standards (Draves et al. 2004). The network capacity can be further enhanced if the network employs an intelligent channel assignment which seeks a proper mapping between the available channels and the radios at every node (Raniwala & Chiueh 2005, Raniwala et al. 2004, Marina et al. 2010).

Channel assignment (CA), was extensively researched but still challenging since many formulations of the problem turn to be NP-hard (Marina et al. 2010, Ramachandran et al. 2006, Si et al. 2010). Considering the time duration between consecutive runs of channel assignment algorithm, many channel assignment approaches fall under the *static* category, where mesh nodes tune an antenna to a specific channel permanently (see e.g., (Raniwala et al. 2004, Marina et al. 2010, Avallone & Akyildiz 2008, Amiri Nezhad & Cerdà-Alabern 2010) and references therein). Static approaches provide low complexity and low feedback overheads and can achieve performance improvements through minimizing the internal interference. However due to the variable nature of the wireless medium, the channel assignment mechanism must be flexible enough to adapt to the arbitrary traffic and both external and internal interference patterns (Wu & Mohapatra 2010).

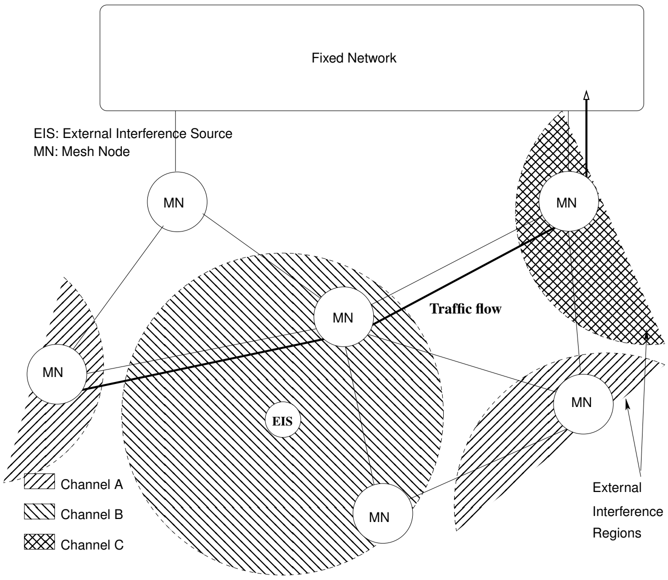


Figure 1: Wireless Mesh Network suffering from interference caused by external sources. If there are only three available channels (A, B and C), to avoid the external interference the mesh network has to be able to find the most suitable channel at each hop.

Dynamic channel assignments (Gong et al. 2009, Bahl & Chandra 2004), on the other hand, enforce nodes to switch their interface dynamically from one channel to another between successive data transmissions. Therefore they require tight synchronizations among nodes. Dynamic approaches are only applied for single radio nodes working over multiple frequencies, since they can not exploit the advantages of multi-radio networks (Crichigno et al. 2008).

Static CAs can easily be extended to *semi-dynamic* by refreshing the channel assignment at regular time intervals in response to changes in traffic pattern or co-channel interference from both external and internal sources (Ramachandran et al. 2006, Wu et al. 2006, Alicherry et al. 2006, Mohsenian-Rad & Wong 2007, Raniwala & Chiueh 2005, Kyasanur & Vaidya 2006, Wooseong Kim & Gerla 2010, Kim & Suh 2008, Felegyhazi et al. 2007, Fu & Agrawal 2008).

In this work we propose SICA, an interference aware channel assignment algorithm for IEEE 802.11 based WMNs. We estimate the amount of interference over channels, induced by any external wireless device, based on IEEE 802.11k standard (IEEE-SA 2008). We then use game theory to formulate the semi-dynamic channel assignation problem. Unlike previous game formulation in the literature (Felegyhazi et al. 2007, Gao et al. 2008, Chen & Zhong 2009, Shah et al. 2010, Gao & Wang 2008) we assume a more realistic scenario by explicitly considering the presence of external interferences from other networks and, we assume that nodes do not have perfect information about others' strategies. Then we apply a real-time learning method to design a distributed algorithm which assigns channels to radios while avoiding the ripple and channel oscillation effects (Si et al. 2010). The nodes continuously refine their decision accounting the changes in the wireless environment. The main contributions that sets our work apart from others are the following ones:

- A novel game theory formulation of the channel assignment problem, considering external and internal interferences.
- A decision making strategy assuming imperfect information at each mesh router that allows a fast network adaption to the changing wireless environment.
- A fully distributed channel assignment algorithm which preserves the network connectivity and supports any routing protocol.
- A new protocol which applies channel load estimation, interface switching, control message exchange and data delivery mechanisms in addition to channel assignment.

We evaluate SICA through simulations using ns-3 (ns-3 development team 2011) and compare it with other channel assignment mechanisms that have been proposed in the literature (Wooseong Kim & Gerla 2010, Ramachandran et al. 2006, Gao et al. 2008). Results demonstrate the effectiveness of SICA in exploiting channel diversity, hence reducing the interference over wireless links and improving the system performance in terms of capacity and supported nodes and networks.

The paper is organized as follows. We first describe the related work in Section 2. Then, in Sections 3–6 we explain SICA architecture, channel assignment algorithm and simulation details. The performance evaluation is done in Section 7, and Section 8 ends the paper with the concluding remarks.

2 Related Work

The channel assignment problem has been studied in deep during last years (Crichigno et al. 2008, Si et al. 2010) and many semi-dynamic solutions have been addressed in previous proposals (Ramachandran et al. 2006, Wu et al. 2006, Alicherry et al. 2006, Mohsenian-Rad & Wong 2007, Raniwala & Chiueh 2005, Kyasanur & Vaidya 2006, Wooseong Kim & Gerla 2010, Kim & Suh 2008, Felegyhazi et al. 2007), but few proposals consider the effect of the external interference (Ramachandran et al. 2006, Wooseong Kim & Gerla 2010).

A simple semi-dynamic approach is proposed in (Kyasanur & Vaidya 2006) for a mesh network with two radio interfaces per node. Although the authors introduce a new path metric which takes into account the interface switching cost in addition to the expected transmission time (Draves et al. 2004), the proposed mechanism considers only the internal interference.

Breath first search channel assignment (BFSCA) (Ramachandran et al. 2006) is the first interference aware CA mechanism. In BFSCA each node estimates the external interference through monitoring the wireless media and coordinates with a central node through a common channel. The central coordinator then assigns channels to links considering the distance of each link to the coordinator and the quality of the link in terms of transmission delay. The main drawbacks of BFSCA are: it needs tight synchronization between nodes and, the channel assignment algorithm is very slow and time consuming since it does an exhaustive search over all interfering links to find the best channel for each link.

Urban-X is another adaptive and semi-dynamic channel assignment (Wooseong Kim & Gerla 2010). Urban-X is proposed for a network where each node must have at least three radios. One radio of all nodes is tuned to a common channel and is used for control traffic. The channel assignment considers the external interference in addition to the number of flows at each node, to make decisions. The best channel is occupied by the receiving radio of a node which has more traffic to send, although it may not receive any traffic.

Channel assignment algorithms using game theory models have been studied recently in some works (Felegyhazi et al. 2007, Gao et al. 2008, Chen & Zhong 2009, Shah et al. 2010). None of the proposed algorithms considers the effect of co-channel interference. All approaches consider that nodes or players have information about all strategies and payoffs, it means that all nodes make decisions based on a global payoff table. However, in a scenario having external interference, it is difficult to have a perfect knowledge of the channel use, before making decisions.

Felegyhazi et al. (2007), formulate the channel assignment problem as a game where traffic flows compete for shared channels in a conflict situation. Although the algorithm converges to a stable Nash Equilibrium, their work is limited to a single hop single collision domain network, where each node participate in only one traffic flow. Further extensions of this work for multi hop networks but limited to one collision domain are presented in (Gao et al. 2008, Chen & Zhong 2009, Kim et al. 2009), where nodes are limited to communicate with devices in their transmission range. However, although it may be an unrealistic assumption, the authors assume that each node knows about the existence of all other nodes in the network and the channel they use.

Kim et al. (2009) did not put any constraints for the number of radios per node. The proposed game should be played sequentially and channels should be reallocated for any changes in the traffic profile. The approach proposed in (Gao et al. 2008) formulates the channel assignment as a cooperative game where nodes are cooperating with each other to improve the network throughput. The channel reallocation is necessary for any changes in traffic pattern. Shah et al. (2010) formulate the game for multiple collision domains, but they use a static game which is limited to find a Nash Equilibrium for competing flows.

Unlike all previous game models for channel assignment, SICA consider co-channel interference while assuming that nodes have imperfect information and the solution is independent from the traffic profile.

This paper is a revised and expanded version of the workshop paper (Nezhad & Cerdà-Alabern 2011). A new game theory model that handles better the internal and external interference is presented here. The new game theory model is based on a set of control parameters that improve the system adaptability to the changing environment conditions. Using the new game theory model, SICA achieves a gain equal to 11% compared with the former protocol (Nezhad & Cerdà-Alabern 2011). Finally, a new random topology is introduced to evaluate SICA, as well as we extend the protocols to which it is compared (see Section 7.1), showing that SICA outperforms all of them.

3 SICA Architecture

We present a multi-radio multi-channel mechanism which mitigates the impact of the interference, and improves the performance of the wireless mesh networks by driving the benefits of non-overlapping channels. The distributed multi-channel architecture considers the channel selection mechanism, describes the switching process of the antennas and controls data buffering and transmitting. Nodes use a distributed algorithm to occupy the best channel based on the information gathered during the channel sensing periods. Channel assignment is viewed as a lower layer mechanism which does not consider the traffic load and, therefore any routing protocol can be applied to the network.

We describe SICA in the specific case where nodes are equipped with two radio interfaces, each one being able to use a set C (with cardinality $|C| > 1$) of non overlapping channels. However, SICA could be easily extended to a network where nodes are equipped with a number of radios larger than two. The radios will be referred to as the *receiving radio* and the *transmitting radio*, and denoted by R and T, respectively (Fu & Agrawal 2008).

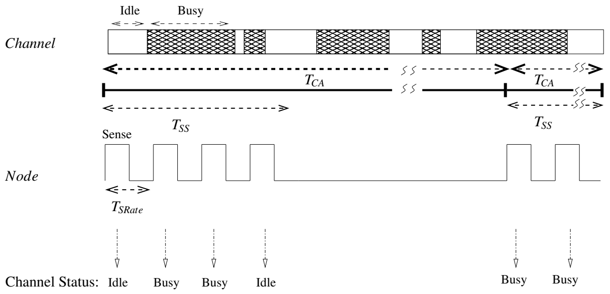


Figure 2: Sensing a channel by gathering samples

The distributed channel assignment mechanism selects and assigns the best channel to the R radio of each node. Then, nodes switch the T radio accordingly. For example, if a generic node A tunes its R radio over channel $c \in C$, each neighboring node, which aims to send traffic to A, will switch its T radio to channel c before start transmission. The T radio remains on channel c until all packets, which are addressed to node A, have been sent, or until a maximum period of time, called T_{max}.

In the following sections we explain the details of SICA. We explain the channel sensing mechanism and CA algorithm in sections 3.1, 3.2 and 3.4, respectively. The synchronization and switching of R and T radios are explained in sections 4, 5 and 6.

3.1 External interference estimation

To estimate the amount of external interference, mesh nodes use the clear channel assessment (CCA) mechanism for spectrum sensing (Kleinrock & Tobagi 1975). CCA is based on energy detection during a specific period of time. Since CCA is used to estimate the wireless interference of external devices, to exclude the interference of the nodes from the same network, at a given time all nodes on the same channel stop transmission and start sensing the channel. The required time synchronization for the common sensing period is achieved through sending messages (see Section 4). Since all nodes working on the same channel must remain silent during channel state assessment, a big sensing period will degrade the network throughput. However, a long enough sensing period is necessary to have a precise estimation (Wooseong Kim & Gerla 2010, IEEE-SA 2008, Nguyen et al. 2011). In SICA, each node senses only one channel at each sensing period (the channel of its receiving radio). Then, at the end of each sensing period, each node exchanges the channel state information with its neighbors.

As shown in Fig 2, during the sensing period (T_{SS}) every node monitors the channel by taking samples at the sense rate (T_{SRate}). T_{CA} is the period of time between consecutive runs of channel assignment algorithm.

The channel status would be monitored as either *idle* or *busy*. Define $T_{i,busy}(c)$ as the time that channel is sensed busy during the sensing period and $T_{i,idle}(c)$ as the amount of time that the channel is sensed idle.

IEEE 802.11k standard for radio resources measurement (IEEE-SA 2008) proposes a simple formulation to compute the channel load as the percentage of time that the node senses the medium as busy. At the end of the sensing period, node i estimates the *normalized bandwidth* (or duty cycle) consumed by external networks over a channel c, as:

$$B_{i,neig}(c) = \frac{T_{i,busy}(c)}{T_{i,busy}(c) + T_{i,idle}(c)} \tag{1}$$

The mesh node then uses the channel load to make decision in channel assignment algorithm (see Section 3.2).

3.2 Game Theory Based Channel Assignment Model

We use a game theory model for the distributed channel assignment in SICA, which is adaptive to the external interference.

In our model each node is a rational player which tries to occupy the best channel for its R radio. The best channel is a channel which suffers less external interference and it is not shared by many neighboring nodes in the same network. From this point forward, we use the terms node and player interchangeably.

Let N be the number of nodes of the network, and $f_{i,c}$ the number of R radios of player i using channel c ($f_{i,c} \in \{0,1\}$). Define the strategy of player i, s_i, as its channel allocation vector, given by:

$$s_i = (f_{i,1}, f_{i,2}, \ldots, f_{i,|C|}), \quad i = 1, \ldots, N \tag{2}$$

A player strategy describes whether it has a radio over a specific channel or not. Note that the total number of R radios employed by player i is given by

$$f_i = \sum_{k=1}^{|C|} f_{i,k} \tag{3}$$

In a dual radio network with one R radio for each node, $f_i = 1$.

We define the strategy matrix (strategy profile), S, as the strategy vector of all players at a given time:

$$S = \begin{pmatrix} s_1 \\ s_2 \\ \ldots \\ s_N \end{pmatrix} \tag{4}$$

By S_{-i} we refer to the strategy matrix consisting of all nodes' strategies except player i. Note that, node i may not know S_{-i} completely.

We formulate a game theory model where each player i chooses a channel c trying to minimize a *loss function*. Each mesh router uses two separate costs for selecting a channel. The first cost is according to the channel load estimated in Section 3.1 (Equation (1)). The second cost is according to internal interference induced from neighboring nodes. To estimate the internal interference over a channel, mesh routers compute how congested is the channel in the neighborhood. Let N_i be the number of nodes in the interference range of node i (two-hops neighbors based on the interference protocol model presented by Gupta & Kumar (2000)). We represent by $R_i(c)$ the number of nodes in the set N_i that have tuned their R radio to channel c at a given time:

$$R_i(c) = \sum_{k \in N_i} f_{k,c} \tag{5}$$

We define the density of interfering nodes over channel c by

$$\frac{R_i(c)}{N_i} \tag{6}$$

The mesh router then merges the costs by taking the weighted average of the individual cost as a *bandwidth loss function*:

$$M_{i,B}(c, S_{-i}) = \alpha \cdot B_{i,neig}(c) + (1-\alpha)\frac{R_i(c)}{N_i} \tag{7}$$

where $\alpha \in [0,1]$ is the control parameter.

However, the cost of one node's decision depends not only on the available bandwidth of the selected channel, but also on the switching delay penalty if a node's radio switches frequently (Yang et al. 2011). According to Wooseong Kim & Gerla (2010), Murray et al. (2007) current 802.11 commodities suffer a considerable switching delay (D_s), that ranges from 80 μs to 22 ms. We consider the magnitude of the switching delay related to the *Hello* interval, T_H (explained in Section 5).

If the hello interval is large enough the effect of the switching delay is negligible and nodes are allowed to switch frequently to other channels. On the other hand a considerable switching delay should result in a higher channel switching cost, making nodes to switch between channels less frequently.

Let c_i the channel being used by node i for the R radio, we assume that a *switching delay loss function*, for any channel, is given by:

$$M_{i,D}(c, S_{-i}) = \begin{cases} \frac{D_s}{T_H}, & c \neq c_i \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Finally, we combine bandwidth and switching delay costs in the loss function given by:

$$M_i(c, S_{-i}) = \gamma M_{i,B}(c, S_{-i}) + (1-\gamma) M_{i,D}(c, S_{-i}) \tag{9}$$

where $\gamma \in [0,1]$ is a tuning parameter. Note that the loss function codomain is $[0,1]$.

It is not feasible nor necessary for a player to compute $M_i(c, S_{-i})$ for all possible values of S_{-i}. Each player computes the *loss* value for one strategy profile at a time. In Section 3.3 we explain how this method solves the game effectively.

To sum up, we have defined a game with the following properties:

- Nodes are rational players and try to occupy the most vacant frequency channels.
- Nodes do not have knowledge about their neighbors criteria of making decision, beforehand.

- Each channel decision imposes a cost (in the range of 0 to 1) to a node, as a function of switching delay and available bandwidth on the selected channel.
- The game is played in several rounds, as the external parameters introduced by the environment may differ in each round, the environment is unpredictable and can remain permanently in the transient state.

3.3 Solving the channel assignment game

Due to the changes in the co-channel interference, the game outlined in the previous section has no deterministic loss matrix, therefore using common approaches to solve the game is impossible. Our solution is based on real-time learning approach proposed by (Freund & Schapire 1996, 1999).

As define before, $M_i(c,S_{-i})$ is the loss matrix of node i, i.e. the rows of $M_i(c,S_{-i})$ are the strategies of node i (the channels, $c \in C$, it can choose), and the columns are all possible strategies of the other players, S_{-i}.

Each node assigns non-negative weights ($w_i(c)$) to the rows of $M_i(c,S_{-i})$. We assume that, the number of rows in $M_i(c,S_{-i})$ is the same for all nodes and equal to the number of orthogonal channels ($|C|$).

Initially $M_i(c,S_{-i})$ is unknown to player i, but this game can be played repeatedly in a sequence of *game rounds* $(1,\dots,T)$. To avoid the channel oscillation in each round t ($t \in 1,\dots,T$), the player plays a mixed strategy based on the weights ($w_{i,t}(c)$) assigned to the rows of $M_i(c,S_{-i})$. The probability of selecting the channel c, is calculated as:

$$P_{i,t}(c) = \frac{w_{i,t}(c)}{\sum_{c \in C} w_{i,t}(c)} \tag{10}$$

Initially, all weights are set to 1, which means that the probability of selecting any channel is identical. After selecting a channel, the node gathers information from its neighbors and updates the loss that is suffered (equation (9)). Then, the weights are updated as:

$$w_{i,t}(c) = w_{i,t-1}(c)\, \beta^{M_i(c,S_{-i})} \tag{11}$$

where $\beta \in [0,1]$ is the game parameter (Freund & Schapire 1999). The main theorem (Freund & Schapire 1996, 1999) concerning this algorithm is:

Theorem 1: *For any matrix M with n rows and entries in $[0,1]$, and for any sequence of mixed strategies $Q_1,\dots,Q_T$ played by the environment, the sequence of mixed strategies $P_1,\dots,P_T$ produced by the algorithm satisfies*

$$\sum_{t=1}^{T} M(P_t,Q_t) \leq \frac{ln(\frac{1}{\beta})}{1-\beta} \min_P \sum_{t=1}^{T} M(P,Q_t) + \frac{1}{1-\beta} ln\, n \tag{12}$$

The proof can be found in (Freund & Schapire 1996, 1999). The theorem simply implies that, the excellency of the decisions made by the learner using the multiplicative weights scheme, depends on the value of β. A high β value introduces minor changes to the weights, and the learner follows the environment more accurately but slowly. Therefore it is applicable to a scenario where the environment changes less frequently. On the contrary, a low β value imposes big changes in the weights, which introduces a higher error to the decision but, it is adequate to a scenario with frequent changes. In our simulations we found that $\beta = 0.2$ leads to better results (see Section 7). We use the same β for all players. Note that the best solution reached by the learner is not necessarily the Nash Equilibrium. Since it has been shown that multiplicative weights update learning algorithm cannot work for Nash Equilibrium in general bi-matrix games (Daskalakis et al. 2010).

3.4 Channel Assignment Mechanism

Alg. 1 summarizes the channel assignment mechanism previously described. Recall that the main idea of SICA is to use all the available information at each node, which is gathered from its neighbors and sensing the channels, and selects the best channel by playing a game with mixed strategies. As explained in Section 3.3, the game is played in rounds that we refer to as *channel assignment periods*, and represent its duration by T_{CA}. Each node i runs Alg. 1 at every T_{CA}.

Algorithm 1 SICA(N_i)

```
Input:
  $N_i$: set of one and two-hops neighbors of node $i$.
if this is the first assignment then
    Set $w_{i,t}(c) = 1\ \forall c \in C$
    Assign a random channel $c_i$ to the $R$ radio
else
    Compute $P_{i,t}(c)\ \forall c \in C$ (Eq. (10))
    Compute the Cumulative Distribution Function (CDF) out of $P_{i,t}(c)\ \forall c \in C$.
    Select channel $c_i$ using Inverse Transform Sampling.
    Assign channel $c_i$ to the $R$ radio.
end if
Switch the $R$ radio to channel $c_i$ (Alg. 2)
Use $CCA$ to estimate $B_{i,neig}(c)$ (Eq. (1))
Inform other neighbors about $B_{i,neig}(c)$
for $c \in \{$channels used by $R$ radio of $N_i$ nodes$\}$ do
    Calculate $M_i(c,S_{-i})$ (Eq. (9))
    Update $w_{i,t}(c)$ (Eq. (11))
end for
```

Four main reasons make SICA to be an efficient channel assignment algorithm:

- Nodes are not required to have the perfect information about other players' strategies and loss functions.
- Nodes are supposed to be selfish players trying to occupy the best channels.
- It is not necessary for a node to estimate the external interference over all channels. Each node only estimates the interference over the channel of its R radio.
- The proposed channel assignment eliminates the *Channel Oscillation* problem. This problem happens

when some nodes find a channel empty and try to occupy it simultaneously. In such situation, the nodes will switch back as they will find it busy by the others that have switched to that channel too. Playing a mixed strategy, as previously described, avoids channel oscillation since each node selects the destination channel randomly with a predefined probability.

Table 1 summarizes the parameters of the channel assignment mechanism and the default value for each one of them.

4 Mesh nodes synchronization mechanism

Unlike most CAs proposed in the literature (see Section 2), in SICA there is not a common control channel shared by all nodes. In SICA, the synchronization is achieved through exchanging messages over the data channels. Since each node can assign a different channel to its receiving (*R*) radio, nodes must be aware of the channels used by their neighbors' *R* radios. In SICA, a node broadcasts *Hello* messages to report the channel of its *R* radio to its neighbors.

It is not necessary to send *Hello* messages over all available channels, except when a new node joins the network or when a node stops receiving *Hello* messages from any neighbor. Once a node knows the channels used by the *R* radio of its one-hop neighbors, the node switches the *T* radio to those channels and sends *Hello* messages every T_H seconds.

We refer to the period between *Hello* messages as T_H. This period must be long enough to minimize the overhead caused by the switching time of the *T* radio, and small enough to keep the information updated for all nodes.

A *Hello* packet contains the following information, besides a sequence number:

1. Channel and MAC address of the *R* radio,
2. Channels of the *R* radios of the node's one-hop neighbors,
3. Spectrum sensing information,
4. Channel switching attempt information.

Additionally, each node has to broadcast the the channel used by the *R* radios of its one-hop neighbors in the *Hello* messages. Therefore the neighbors are able to keep the information of two-hops nodes.

Spectrum sensing information contains: *i*) the estimated consumed bandwidth by external interferences over the receiving channel; and *ii*) the time units remaining before the start of the next sensing period. Neighboring nodes need information about the upcoming sensing period to avoid initiating any transmission over the channel that is going to be sensed. Nodes broadcast the remaining time units before their next sensing event periodically, therefore neighboring nodes are informed even though some of the messages get lost. We assume that re-sending the sensing information, which is done frequently via hello messages, reduces the error of miscalculating the sensing time for neighbors.

Channel switching attempt information is the expected time units before the moment that a node will switch its *R* radio to a new channel (See Section 6). Note that there is no need to have a tight synchronizations between nodes since nodes are aware of changes through *Hello* messages.

5 Data delivery mechanism

After gathering information from the neighbors, a node may start transmitting data. One node may have packets to deliver to different neighbors which have their *R* radio on different channels. In our model each channel is associated with a sequential first-in first-out (FIFO) queue. Packets are added to the corresponding queue according to the receiving channel of the neighbors. When the *T* radio switches to each channel, it sends all or some of the packets in the associated queue. We use a different queue for *Hello* messages, which has higher priority than data packets' queue.

A mesh node uses a Round Robin approach to visit all the channels for which it has data to sent, defined as the subset $C_i \subset C$. To avoid starvation, after switching, the *T* radio will stay in one channel for at most a specific period of time (T_{max}). We assume that the switching delay of the *T* radio is constant and equal to D_s. Therefore if a node has to send data over $C_i \subset C$ channels, it will take $|C_i|\, T_{max} + (|C_i| - 1)\, D_s$ to visit them. In order to have the same opportunities to transmit over C_i channels, the node computes T_{max} as:

$$T_{max} = \frac{T_H - (|C_i| - 1)\, D_s}{|C_i|} - D_t \tag{13}$$

where T_H is the period between *Hello* messages and D_t is the delay before the *T* radio starts the transmission. Note that C_i is given by the number of different channels used by node's neighbors.

Fig. 3 shows an example describing the whole process. In the example the node has $|C_i| = 2$ channels over which it has data to send. Every T_H a *Hello* packet is pushed at the front of the C_i queues. We show *Hello* and Data packets with black and shaded boxes, respectively. T_{max} shows the maximum duration of time that a node may remain in each channel (see Equation (13)). In detail, Fig. 3-a shows the node status before T_H starts. The mechanism starts from channel one (Fig. 3-b) and, after T_{max} it switches to channel two, where it is able to transmit only three packets before to switch again to channel one.

After switching the *T* radio to a channel, to avoid collision with any ongoing transmission on the channel, a node must wait at least for D_t units of time before it starts transmitting, (see Section 6.2). After D_t, the node sends packets during at most T_{max}, then it switches the *T* radio to channel two and the transmission process is repeated for this channel (Fig. 3-b). While there are packets in the queues, the node will round robin among them until the end of T_H (Fig. 3-c).

Note that the *T* radio can not switch to a channel and initiate any transmission when the channel is being sensed by any of the neighboring nodes. Moreover nodes must consider the switching attempt of the *R* radio which is announced to the neighbors before it starts. The switching mechanisms of *T*

Parameter Name	Description	Possible Value	Default
T_H	Hello interval	10 ms - 100 ms	20 ms
T_{CA}	Channel assignment interval	$T_H \ll T_{CA}$	10 s
T_{SS}	Channel sense interval	$T_H < T_{SS} < T_{CA}$	5 s
T_{SRate}	Channel sense rate	$T_{SRate} \ll T_{SS}$	1 ms
α	Bandwidth loss function tuning parameter	$0 \leq \alpha \leq 1$	1
β	Channel weight parameter	$0 < \beta < 1$	0.2
γ	Loss function tuning parameter	$0 \leq \gamma \leq 1$	0.8
$\|C\|$	Number of available orthogonal channels	$1 < \|C\|$	8
D_s	Switching delay of the radio	80 μs - 20 ms	300 μs

Table 1 Channel Assignment Parameters

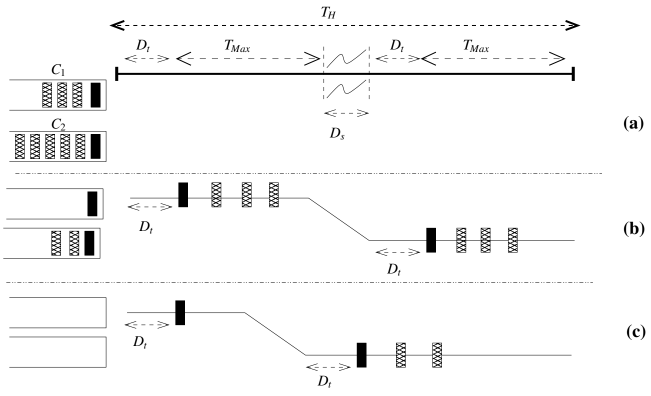


Figure 3: Channel queues and data delivery mechanism

Algorithm 2 Switch The R Radio (c)

```
Input:
  c: The target channel for switching
if there is any transmission on R radio then
    Wait until the end of transmission
else
    $T_S \leftarrow Random(2T_H : 3T_H)$
    Set the timer delay to $T_S$
    while the timer is running do
        Update the switching attempt information in Hello messages
    end while
    Switch the R radio to channel c
end if
```

and R radios are explained with more details in the following section.

6 Channel Switching Mechanism

The switching mechanism consists of two different protocols for switching T and R radios, respectively.

6.1 Switching the R radio

When a node decides to switch the R radio to any channel, it must announce the switching attempt through *Hello* messages. The switching attempt information in a *Hello* message, consists the following fields:

1. The destination channel to which the R radio will switch,
2. The time to switch (T_S).

The switching time (T_S) contains the remaining time in the current channel until the R radio switches to the new channel. This time must be longer than the *Hello* interval to make sure that all neighbors are informed. Therefore the neighbors will consider the new channel for upcoming transmissions. A node follows the steps in Alg. 2 to tune the R radio to the defined channel.

If a node misses any information about a switching attempt of a neighbor, the node would fail to send packets to it. The algorithm tries to prevent this by selecting a sufficiently large T_S (see Table1). Moreover the node always gets information about a lost neighbor from other common neighbors, thus, updating its information.

6.2 Switching The T Radio

The T radio switches channels more often than R. Alg. 3 describes the switching mechanism for the T radio. Here each node checks all queues sequentially (Round Robin) and if there is any data waiting for transmission, it switches the T radio to the corresponding channel and starts sending data after D_t units of time.

When a node switches to a new channel it may fail to hear an on-going transmission between any other nodes on the same channel, so it could be a hidden terminal for the ongoing transmission and must avoid transmitting immediately to prevent the collision. Consequently after switching, a node may wait for D_t before starting any transmission.

The node remains on the target channel until the end of the transmission or at most for T_{max}. Then it proceeds to check the other queues.

7 Performance Evaluation

In this section, we study the performance of the proposed channel assignment algorithm using ns-3 simulator (ns-3 development team 2011) for 802.11-based multi-radio mesh networks. We use a network where the mesh routers initialize their routing tables using Shortest Path First (SPF),

Algorithm 3 Switch The T Radio (Q_{C_i}, C_i)

```
Input:
  $Q_c$: The packet queue associated with the channel $c$
  $C_i$: Channels of $R$ radios of one-hop neighbors
for $c \in C_i$ do
    Set the timer delay to $T_{max}$
    if $|Q_c| > 0$ then
        Switch the $T$ radio to channel $c$
        Wait for $D_t$ period of time
        while the timer is running AND $|Q_c| > 0$ do
            $P \leftarrow$ Pop one packet from $Q_c$
            Send $P$
        end while
    end if
end for
```

minimizing the number of hops. We assume a two ray ground propagation model with a radio range of 250 m. Wireless nodes can tune their radio to any channel among 8 non-overlapping channels (according to IEEE 802.11a standard). RTS/CTS mechanism is disabled.

We compare SICA with three relevant channel assignment mechanisms proposed in the literature: the Centralized Breath First Channel Assignment (BFSCA) (Ramachandran et al. 2006); the semi-dynamic interference aware channel assignment (Urban-X) (Wooseong Kim & Gerla 2010); and the Nash Equilibrium Channel Assignment (NEMCA) (Kim et al. 2009). A baseline single radio network working over one channel is also introduced for completeness.

7.1 Description of BFSCA, Urban-X and NEMCA

Breath First Search Channel Assignment (BFSCA) (Ramachandran et al. 2006) is a centralized mechanism that considers one node in the WMN as a coordinator, which is responsible for assigning channels to all nodes' radios in the network. It also considers that one radio at each node is tuned to a common channel for control messages. Each node estimates the external interference through monitoring the wireless media, sending the results of such action to the central node through the common channel. The coordinator then assigns channels to links and informs nodes. Nodes redirect the traffic to the common channel before switching to a new channel and, therefore, they need to be tightly synchronized as, otherwise the channel switching mechanism would interrupt the traffic transmission. Channel assignment in BFSCA uses a graph theory based interference model to find the interfering links (Gupta & Kumar 2000). It sorts all links based on their distance to the central node and then tries to assign different channels to the interfering links but if such a channel is not found it assigns a random channel to the link.

Urban-X (Wooseong Kim & Gerla 2010) uses three radios for each node: one R and one T radio, as in SICA, and a third radio which is tuned to a common channel for all nodes. The common channel stays unchanged through the life time of the network. Channel assignment in Urban-X is distributed and takes into account the amount of flows a node has to send, and the estimated external interference over the channels. Nodes need to have information about the number of flows their neighbors have. Then Urban-X assigns a priority to each node based on the number of active flows it has. Nodes having higher priority have more chances to occupy the best channels. Nodes broadcast control messages over the common channel up to two-hops neighbors. After switching to a channel, the T radio remains there for a predefined period of time (40 ms).

Nash Equilibrium Channel Assignment (NEMCA) (Kim et al. 2009) is a game theory based channel assignment which only considers the internal interference. NEMCA is distributed and played by rational nodes. Nodes consider the internal interference over channels without keeping any memory. Each node selects a list of channels which are better than the current channel it has, from where it selects the next channel to use using a uniform random probability.

7.2 Simulation Parameters and Performance Metrics

We have evaluated the performance of the protocols for different number of nodes for two different node placement topologies: grid and random, and with different number of traffic flows. The traffic is generated by 100 kbps CBR flows with packet size of $L = 8000$ bits sent over vertical and horizontal directions in the grid topology and between random nodes in the random topology. Data queues are simulated in a way that they drop the old packets automatically to avoid saturation. We assume that the maximum time duration that a packet can remain in a queue is 1 s for data and 20 ms (T_H) for *Hello* messages.

The channels receiving interference from external networks, are chosen randomly. In the simulations, we consider that, at each time 50% of channels are busy due to the external interference. A channel with external interference is modeled as an on-off process, such that the channel is sensed busy and idle during the on and off states, respectively. Note that as the channel is detected busy due the external interference, nodes are not allowed to transmit during that state. The duration of the *busy* state has been fixed to a constant value, while the duration of the *idle* state is chosen exponentially distributed. The duration of the busy and idle periods have been varied to produce different interference loads.

The SICA parameters have been set using the values of Table 1. The specific parameters of other protocols are set according to the values given in (Ramachandran et al. 2006, Wooseong Kim & Gerla 2010, Kim et al. 2009).

We consider three network performance measures:

- *Data delivery ratio:* ratio of the total amount of data which is correctly received by the destinations, to the total amount of data packets transmitted by the sources.

- *Average end to end delay:* mean delay of the packets to reach the destination.

- *Control overhead:* ratio of the total number of control messages sent between nodes, to the total number of correctly received packets.

7.3 Results

7.3.1 Grid Topology

Figures 4-6 show the network performance for different number of nodes and two CBR traffic flows of 100 kbps. Every 50 s, external interference is introduced over 4 channels chosen randomly.

In these simulations the duration of the busy state of the external interference is fixed to 10 ms, while the mean duration of the idle state is 8 ms. The results have been obtained averaging over 10 runs of 1000 s simulation time with different seeds. The error bars in the figures show 95% confidence intervals.

Fig. 4 shows that the delivery ratio is 20% higher in SICA and BFSCA than in Urban-X or NEMCA. This is a significant improvement, since Urban-X and BFSCA use 3 radios and SICA uses only 2. This result shows that the game theory approach used in the channel assignment of SICA outperforms the optimized centralized algorithm used in BFSCA for large networks. The bigger error bars in NEMCA and the single-channel network shows that they suffer more unexpected external interference which is avoided by adaptive schemes. Moreover, NEMCA, as a non adaptive channel assignment, performs worse than a single channel network because of the radio switching penalty and channel assignment overheads for making inefficient decisions without considering the external interference.

In Fig. 5 we can see that the average end to end delay is lower in SICA and BFSCA than the other protocols. SICA leads to a lower delay thanks to the fast switching mechanism of the T radio over all channels (see Section 5) and because it avoids the channels which suffer external interference. The average delay in Urban-X is much higher than others because it keeps the T radio in each channel for a predefined period of time, regardless of having data to send, therefore forcing a considerable delay for data waiting in the queues of the other channels.

Fig. 6 shows that SICA and BFSCA have similar control overhead. BFSCA uses a common channel for control packets which should be forwarded until the coordinator node, in SICA on the other hand a node broadcasts control messages only over all channels which are used by its neighbors. The control overhead in NEMCA is much higher than others since it broadcasts many control messages for channel assignment but has a low data delivery ratio. The control messages of the single channel network consist of *Hello* messages that nodes send every T_H seconds to their neighbors to announce their presence.

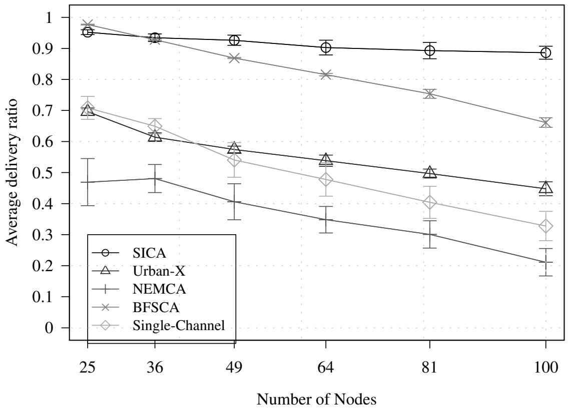


Figure 4: Data delivery ratio vs. number of nodes (Grid topology)

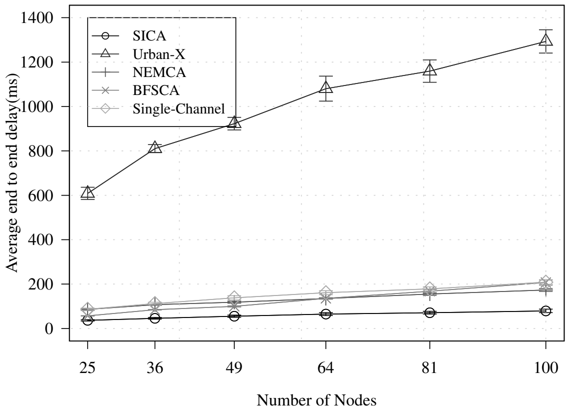


Figure 5: Average end to end delay vs. number of nodes (Grid topology)

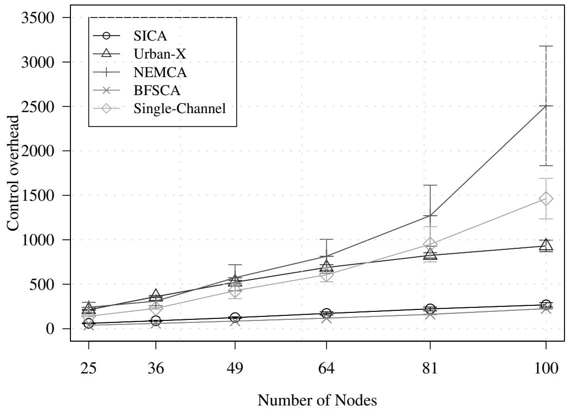


Figure 6: Control overhead vs. number of nodes (Grid topology)

7.3.2 Random Topology

Fig. 7- 9, show the packet delivery ratio, average end to end delay and control overhead for different number of nodes which are randomly distributed over a $1000 \times 1000\ m^2$ area. We check the network topology to be connected using R numerical tool (r development core team 2008). Each point in the figures shows the average over 50 runs. The error bars in the figures show 95% confidence intervals. Fig. 7 shows that the delivery ratio in SICA is higher than other protocols and it does not drop a lot as the number of nodes increases. For BFSCA and Urban-x on the other hand, the delivery ratio drops fast as the number of nodes increases. NEMCA performs the same as a single channel network following the reasons explained for Fig. 4.

Fig. 8 shows that the average end to end delay of all protocols is much lower compared to Urban-X due to the same reason explained for Fig. 5.

Fig. 9 confirms the reason explained for Fig. 6, showing that control overhead for NEMCA is higher than other

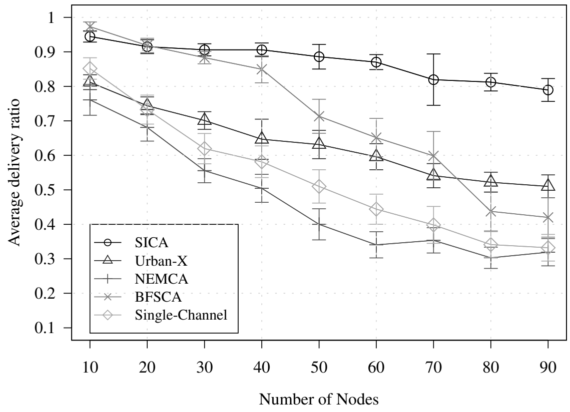


Figure 7: Data delivery ratio vs. number of nodes (Random topology)

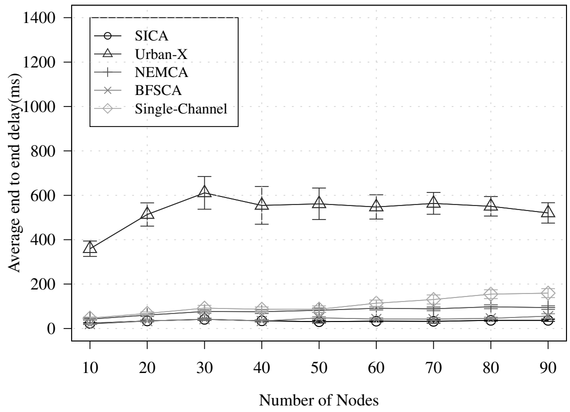


Figure 8: Average end to end delay vs. number of nodes (Random topology)

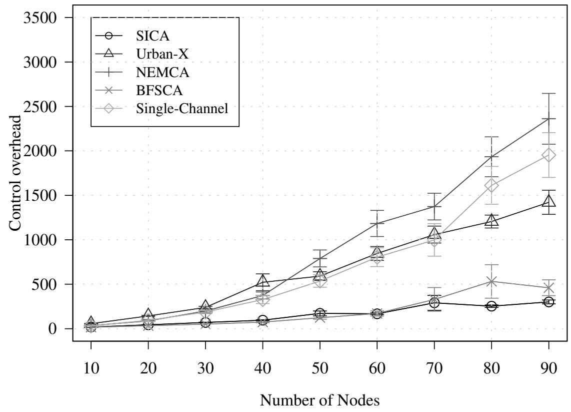


Figure 9: Control overhead vs. number of nodes (Random topology)

protocols, while SICA and BFSCA lead to lower control overhead.

7.3.3 Increasing the number of traffic sources

Fig. 10-12 are obtained using a 8×8 grid network while other parameters are the same as the parameters considered for the grid scenario.

Fig. 10 shows that the delivery ratio of SICA is higher than others in presence of high traffic load. The delivery ratio of BFSCA drops fast increasing the number of traffic flows, since any channel switching interrupts the data transmission and nodes are forced to deliver data packets through common channel which offers a high load over the common channel (Section 7.1).

Urban-X performs better in presence of high load traffic compared to BFSCA since it considers the traffic load for making decision, but it results in a considerable high end to end delay (Fig. 11).

Fig. 12 shows that the control overhead of SICA and BFSCA is almost better than others.

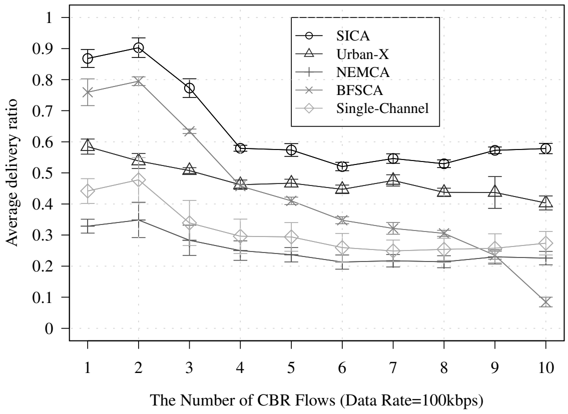


Figure 10: Data delivery ratio vs. number of CBR traffics

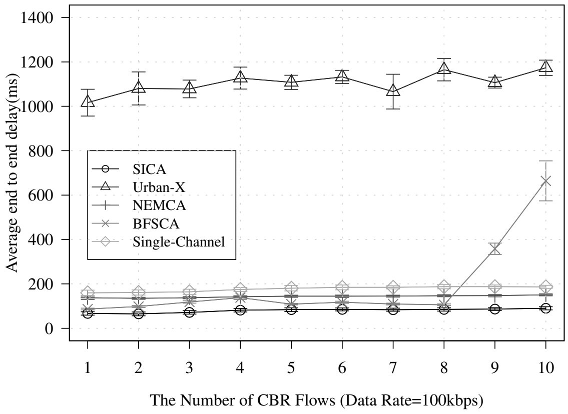


Figure 11: Average end to end delay vs. number of CBR traffics

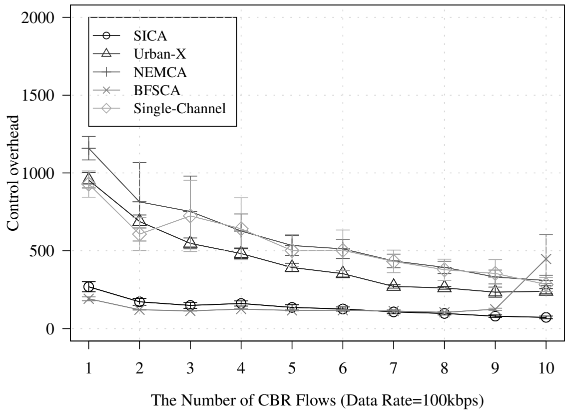


Figure 12: Control overhead vs. number of CBR traffics

7.3.4 Changing the interference load

Fig. 13 is obtained using a 8×8 grid network while other parameters are the same as the parameters considered for the grid scenario.

Fig. 13 compares the delivery probability obtained with all protocols, varying the load of the interference. The x-axis of these figures shows the amount of the external interference, which is varied by changing the duration of the busy state of the interference process between 5 ms and 20 ms, and maintaining the mean duration of the idle state equal to 8 ms. We introduced interference over 4 channels. Fig. 13 shows that, even with a high interference load, the delivery ratio in SICA and BFSCA does not change, when the interference is increased. On the other hand, delivery ratio in Urban-X and NEMCA drops fast. The result confirms that, SICA and BFSCA are much more robust and less sensitive to the external interference than Urban-X.

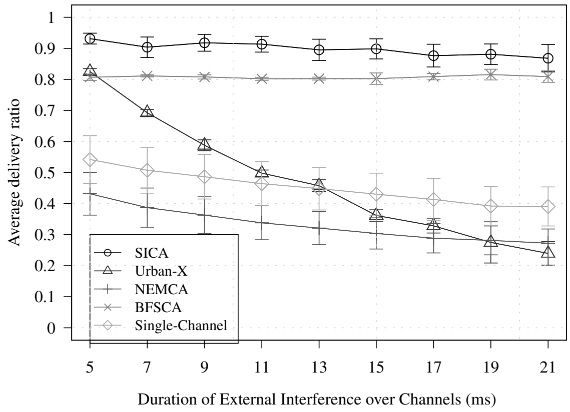


Figure 13: Data delivery ratio vs. Busy duration of external interference

7.3.5 Data delivery ratio: temporal evolution

In order to have a more detailed view of the protocol's behavior, Fig. 14 shows the time evolution of the delivery ratio obtained with different channel assignment protocols using a 8×8 grid network. Other parameters are the same as the parameters considered for the grid scenario. The values shown in the figure have been obtained repeating the simulation for 20 different random seeds and averaging the delivery ratio over 5 s periods.

Figure shows that in SICA the delivery ratio is kept more stable than in others. BFSCA is also able to offer a high packet delivery ratio with a few variations. Urban-x shows high variations in delivery probability and NEMCA is incapable to avoid the interference over channels.

7.3.6 SICA performance as function of α, γ and β

Finally, we investigate the sensitivity of SICA to the α, γ and β tuning parameters that characterize the response of the game theory model (see Section 3.2). Fig. 15-17 are obtained using a 8×8 grid network while other parameters are the same as the parameters considered for the grid scenario.

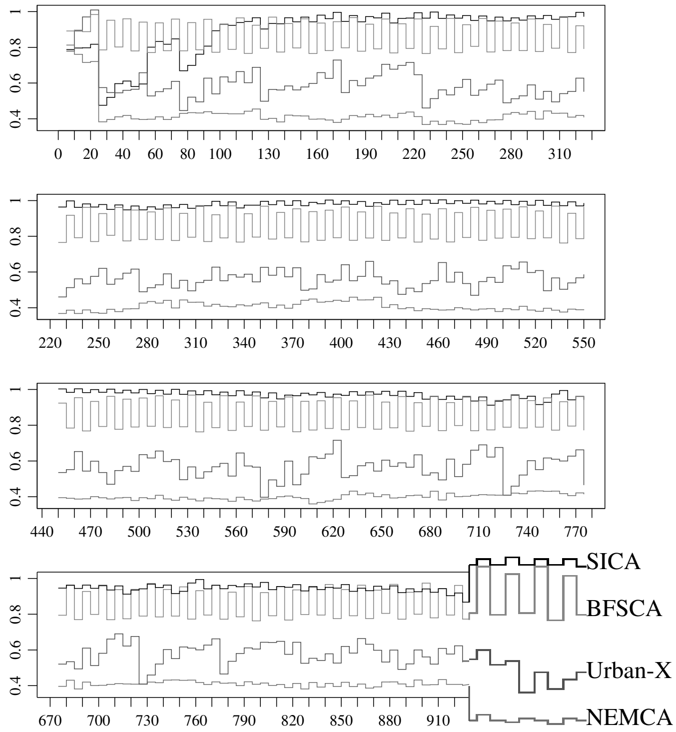


Figure 14: Data delivery ratio over time (zoomed in at the end of the graph in order to make it easier to see which line corresponds to each protocol)

Recall that α is the weight which controls the related magnitude of external and internal interference and γ is the weight between the *bandwidth* and *switching delay* loss functions, while β is used to update the mixed probabilities of the game. Fig. 15- 17 show the delivery ratio obtained with SICA varying α, γ and β, respectively. The figures show the values obtained for a grid network with different number of nodes (depicted with different line types), and using other parameters the same as the parameters considered for the grid scenario.

Fig. 15 shows that SICA performs better for the high values of α. This comes from the fact that in our simulation scenario, nodes are prevented from communicating during the time that a channel is occupied by external interference, which has a higher impact than the internal interference on the system performance.

Fig. 16 shows that the performance of SICA is not sensitive to γ but values of γ in the range of $[0.4, 1]$ give better results, which implies that it is more important to give a higher weight to the bandwidth loss functions rather than switching delay loss function. This comes from the fact that we have chosen a *Hello* interval ($T_H = 20$ ms) significantly larger than the switching delay ($D_s = 300\ \mu$s).

Regarding β, Fig. 17 shows that the best results are obtained with $\beta < 0.4$. Recall that choosing a lower value for β makes the algorithm to adapt faster to the changes of external interference.

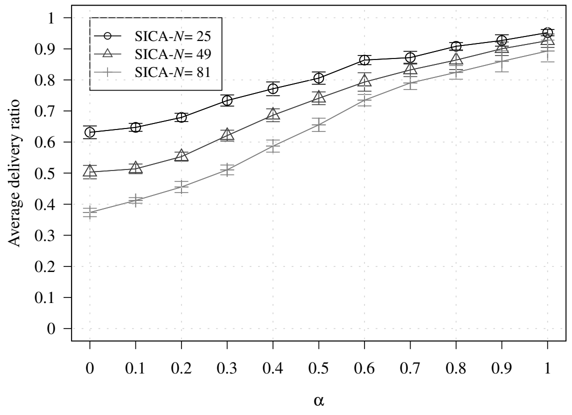


Figure 15: Data delivery ratio vs. α

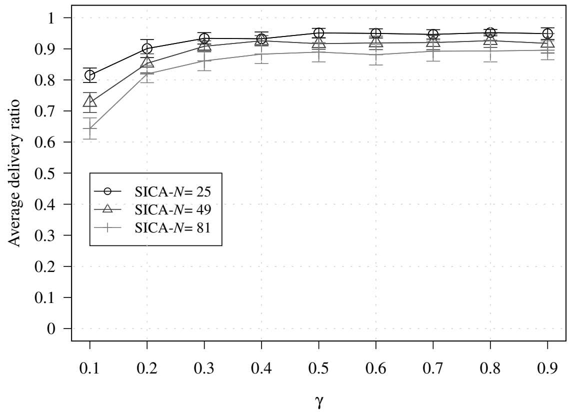


Figure 16: Data delivery ratio vs. γ

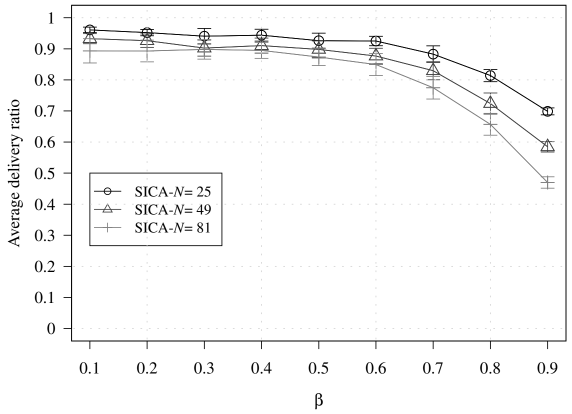


Figure 17: Data delivery ratio vs. β

8 Conclusions

In this paper we have investigated the channel assignment problem in multi-radio wireless mesh networks. We have proposed a new semi-dynamic channel assignment protocol called SICA. We presented a novel formulation for channel assignment problem using game theory and have solved the game using a real time learning mechanism. SICA is a distributed channel assignment and assumes that nodes do not have perfect knowledge about other's nodes strategies. We have done a performance evaluation comparing SICA with other channel assignments mechanisms proposed in the literature. Simulation results show the efficiency of SICA in assigning proper channels to radios by avoiding external interference.

Acknowledgments

This work was supported by the Spanish government and Generalitat de Catalunya through projects TIN2010-21378-C02-01 and 2009-SGR-1167, respectively, and by the European Commission through the NoE EuroNF.

References

Alicherry, M., Bhatia, R. & Li, L. E. (2006), 'Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks', *IEEE Journal on Selected Areas in Communications* **24**(11), 1960–1971.

Amiri Nezhad, M. & Cerdà-Alabern, L. (2010), Utility based channel assignment mechanism for multi radio mesh networks, *in* 'Proceedings of the 8th ACM international workshop on Mobility management and wireless access', MobiWac '10, ACM, New York, NY, USA, pp. 68–74.

Avallone, S. & Akyildiz, I. F. (2008), 'A channel assignment algorithm for multi-radio wireless mesh networks', *Proceedings of 16th International Conference on Computer Communications and Networks, ICCCN 2007* **31**(7), 1343–1353.

Bahl, P. & Chandra, R. (2004), Ssch: Slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks, *in* 'Proceedings of the 10th annual international conference on Mobile computing and networking', pp. 216–230.

Chen, T. & Zhong, S. (2009), 'Perfectly fair channel assignment in non-cooperative multi-radio multi-channel wireless networks', *Computer Communications* **32**(6), 1058 – 1061.

Crichigno, J., Wu, M. & Shu, W. (2008), 'Protocols and architectures for channel assignment in wireless mesh networks', *Ad Hoc Networks* **6**(7), 1051–1077.

Daskalakis, C., Frongillo, R., Papadimitriou, C. H., Pierrakos, G. & Valiant, G. (2010), On learning algorithms for nash equilibria, *in* 'Proceedings of the Third international conference on Algorithmic game theory. SAGT'10', pp. 114–125.

Draves, R., Padhye, J. & Zill, B. (2004), Routing in multi-radio, multi-hop wireless mesh networks, *in* 'Proceedings of the 10th annual international conference on Mobile computing and networking', ACM New York, NY, USA, pp. 114–128.

Felegyhazi, M., Cagalj, M., Bidokhti, S. & Hubaux, J.-P. (2007), Non-cooperative multi-radio channel allocation in wireless networks, *in* '26th IEEE International Conference on Computer Communications. IEEE INFOCOM 2007.', pp. 1442 –1450.

Freund, Y. & Schapire, R. E. (1996), Game theory, on-line prediction and boosting, *in* 'Proceedings of the ninth annual conference on Computational learning theory. COLT '96', ACM, New York, NY, USA, pp. 325–332.

Freund, Y. & Schapire, R. E. (1999), 'Adaptive game playing using multiplicative weights', *Games and Economic Behavior* **29**(1-2), 79–103.

Fu, W. & Agrawal, D. (2008), Effective radio partitioning and efficient queue management schemes in a wireless mesh network, *in* 'Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE', pp. 1 –5.

Gao, L. & Wang, X. (2008), A game approach for multi-channel allocation in multi-hop wireless networks, *in* 'Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing', MobiHoc '08, ACM, New York, NY, USA, pp. 303–312.

Gao, L., Wang, X., Xu, Y. & Chen, W. (2008), Distributed multi-radio channel allocation in multi-hop ad hoc networks., *in* 'IEEE International Conference on Communications, 2008. ICC '08', pp. 3156–3160.

Gong, M., Midkiff, S. & Mao, S. (2009), 'On-demand routing and channel assignment in multi-channel mobile ad hoc networks', *Ad Hoc Networks* **7**(1), 63–78.

Gupta, P. & Kumar, P. (2000), 'The capacity of wireless networks', *IEEE Transactions on Information Theory* **46**(2), 388 –404.

IEEE-SA (2007), 'Ieee standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications'. IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999).

IEEE-SA (2008), 'Ieee standard for information technology telecommunications and information exchange between systems local and metropolitan area networks specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 1: Radio resource measurement of wireless lans'. IEEE Std 802.11k-2008 (Amendment to IEEE Std 802.11-2007).

Kim, H.-K., Oh, C.-Y. & Lee, T.-J. (2009), 'Fair and Collision-Aware Multi-Channel Assignment Based on Game Theory for Wireless Multi-Hop Networks', *IEICE Transactions on Communications* **92**, 1282–1290.

Kim, S.-H. & Suh, Y.-J. (2008), Rate-based channel assignment algorithm for multi-channel multi-rate wireless mesh networks, *in* 'Proceedings of the IEEE Global Telecommunications Conf. IEEE GLOBECOM 2008', pp. 1–5.

Kleinrock, L. & Tobagi, F. (1975), 'Packet switching in radio channels: Part i–carrier sense multiple-access modes and their throughput-delay characteristics', *IEEE Transactions on Communications* **23**(12), 1400–1416.

Kyasanur, P. & Vaidya, N. H. (2006), 'Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks', *ACM SIGMOBILE Mobile Computing and Communications Review* **10**(1), 31–43.

Marina, M. K., Das, S. R. & Subramanian, A. P. (2010), 'A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks', *Computer Networks* **54**(2), 241 – 256.

Mohsenian-Rad, A. H. & Wong, V. W. S. (2007), 'Joint logical topology design, interface assignment, channel allocation, and routing for multi-channel wireless mesh networks', *IEEE Transactions on Wireless Communications* **6**(12), 4432–4440.

Murray, D., Dixon, M. & Koziniec, T. (2007), Scanning delays in 802.11 networks, *in* 'The 2007 International Conference on Next Generation Mobile Applications, Services and Technologies, 2007. NGMAST '07.', pp. 255 –260.

Nezhad, M. A. & Cerdà-Alabern, L. (2011), Adaptive channel assignment for wireless mesh networks using game theory, *in* 'Proceeding of the IEEE 8th Int Mobile Adhoc and Sensor Systems (MASS) Conf', pp. 746–751.

Nguyen, L. T., Beuran, R. & Shinoda, Y. (2011), 'An interference and load aware routing metric for wireless mesh networks', *International Journal of Ad Hoc and Ubiquitous Computing* **7**, 25–37.

ns-3 development team (2011), 'Manual: an in-depth coverage of the architecture and core of ns-3 (no. 3.9)'.

r development core team (2008), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *http://www.R-project.org*

Ramachandran, K. N., Belding, E. M., Almeroth, K. C. & Buddhikot, M. M. (2006), Interference-aware channel assignment in multi-radio wireless mesh networks, *in* 'Proceeding of 25th IEEE Int. Conf. Computer Communications INFOCOM 2006', pp. 1–12.

Raniwala, A. & Chiueh, T.-c. (2005), 'Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network', *Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies. INFOCOM 2005.* **3**, 2223–2234.

Raniwala, A., Gopalan, K. & Chiueh, T.-c. (2004), 'Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks', *SIGMOBILE Mob. Comput. Commun. Rev.* **8**(2), 50–65.

Shah, I., Jan, S. & Loo, K.-K. (2010), Selfish flow games in non-cooperative multi-radio multi-channel wireless mesh networks with imperfect information, *in* '6th International Conference on Wireless and Mobile Communications (ICWMC), 2010', pp. 219 –225.

Si, W., Selvakennedy, S. & Zomaya, A. Y. (2010), 'An overview of channel assignment methods for multi-radio multi-channel wireless mesh networks', *Parallel and Distributed Computing* **70**, 505–524.

Wooseong Kim, Andreas J. Kassler, M. D. F. & Gerla, M. (2010), Urban-x: Towards distributed channel assignment in cognitive multi-radio mesh networks, *in* 'IFIP Wireless Days', Venice, Italy.

Wu, D. & Mohapatra, P. (2010), From theory to practice: Evaluating static channel assignments on a wireless mesh network, *in* 'Proceedings of the IEEE INFOCOM, 2010', pp. 1 –5.

Wu, X., Liu, J. & Chen, G. (2006), Analysis of bottleneck delay and throughput in wireless mesh networks, *in* 'Proceedings of the IEEE Int Mobile Adhoc and Sensor Systems (MASS) Conf', pp. 765–770.

Yang, P., Qin, G., Yan, Y., Wang, H., Zhang, L., Chen, G. & Guo, D. (2011), 'On the channel usability of wireless mesh networks: when stability plays with you', *International Journal of Ad Hoc and Ubiquitous Computing* **8**, 64–77.

Simulation of Multi-Radio Multi-Channel 802.11 Mesh Networks in NS-3

Maryam Amiri Nezhad[1], Manel Guerrero Zapata[1], Boris Bellalta[2], and Llorenç Cerdà-Alabern[1]

[1]dept. Computer Architecture, Universitat Politecnica de Catalunya, Barcelona, Spain,
E-mail: maryam,guerrero,llorenc@ac.upc.edu
[2]dept. Information and Communication Technologies, Universitat Pompeu Fabra, Barcelona, Spain,
E-mail:boris.bellalta@upf.edu

***Abstract*—In the context of wireless network simulation, many simulators are capable of evaluating the performance of single channel network protocols, but they need many modifications to be able to simulate multi radio multi channel networks. We address the problem of simulating channel assignment protocols for multi radio wireless mesh networks in ns-3 simulator, providing the essential steps needed to simulate a channel assignment protocol. In addition, we explain the details of simulating the Semi-dynamic Interference aware Channel Assignment protocol (SICA) as an example. We validate the channel assignment model used in SICA using a Markov model and check the correctness of the simulation by doing an exhaustive search over the relevant events that occur during its runtime.**

***Index Terms*—Network Simulation; ns-3; Channel Assignment; Markov Model**

I. INTRODUCTION

Wireless Mesh Networks (WMN) will be the next self-configuring back-haul providing Internet access for last mile users through multi hop forwarding (Fig. 1). Mesh networks however suffer from wireless interference due to the broadcast nature of the wireless media that may degrade their performance significantly.

Multi radio WMNs are able to offer higher network capacity using different non-overlapping channels provided by IEEE 802.11 standards in the unlicensed bands [1]. Although there are several non-overlapping channels available, the number of channels that can be used simultaneously by a single node is limited by the number of radio interfaces installed on the node. Therefore a mechanism which selects the best channel, in terms of interference, among all available channels, is needed in order to achieve the maximum possible network performance.

Technical solutions for multihop wireless networks are being specified in IEEE 802.11s [2]. IEEE 802.11s is developed as an extension of the successful IEEE 802.11 standard for WLANs (Wireless Local Area Networks) [1]. IEEE 802.11s defines the mesh operation in a single channel although multi-radio mesh routers can form different meshes. The connection between different meshes is provided via bridging. Mesh routers can initiate the channel switching mechanism which moves the mesh, or part of it, to another channel. The routers which do not want to follow the channel switch request may join another mesh. Channel switching may help mesh routers to avoid the external interference but does not reduce the internal interference between routers which belong to the same Mesh Basic Service Set (MBSS), since it moves the MBSS to another channel. However frequent channel switching may degrade the mesh performance due to the high overheads that it implies [3].

In multi-radio mesh networks, the channel assignment (CA) is a mechanism which tries to find a feasible mapping between wireless channels and radio interfaces at each node with the aim of maximizing the capacity of the network.

A channel assignment solution must satisfy the following conditions to be feasible:

- The number of channels assigned to a node must be equal or less than the number of radio interfaces it has.
- The neighboring nodes must have at least a radio at a common channel to be able to communicate with each other.

Based on the time duration between consecutive runs of the channel assignment protocol, CAs are categorized as: static; dynamic; and semi-dynamic [3].

Most of CA proposed in the literature fall into static category [4–13], where nodes tune their radios to certain channels permanently. Static CAs are easy to deploy but unsuccessful to cope with the changes in the wireless environment [4].

Dynamic channel assignments [14; 15], on the other hand, enforce nodes to switch their interface dynamically from one channel to another between successive data transmissions. Therefore they require tight synchronizations among nodes. Dynamic approaches are only used for single radio nodes working over multiple frequencies, since they can not exploit the advantages of multi-radio networks [3].

Static CAs could be easily extended to be semi-dynamic [16–25] if the node refreshes the channels assigned to the radios on a regular time period. Semi-dynamic CAs adapt fast to the changes in the traffic pattern and the interference on the wireless medium from both internal and external sources. However to maintain the network connectivity, neighboring nodes are supposed to share a common channel [16; 26].

Hybrid CAs apply a semi-dynamic channel assignment to the fixed radio interface of each node while the other interface is controlled dynamically. Wireless nodes which use hybrid CAs, do not share common channel with their neighbors, since

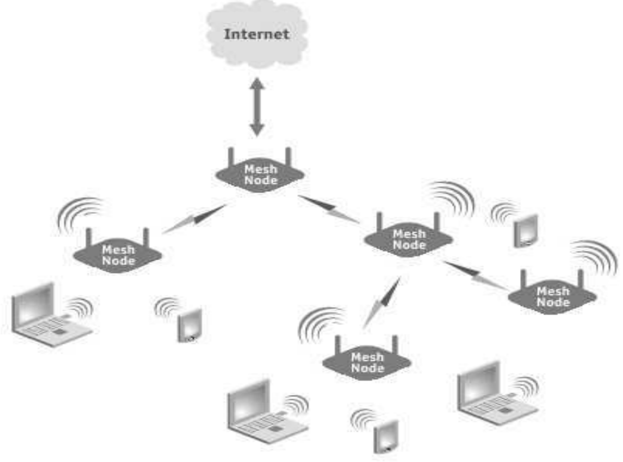


Fig. 1: Multi-Radio Wireless Mesh Network

the dynamic radio switches to the channel of the neighboring nodes to make the connection.

Implementing a multi-radio multi-channel mesh network is very challenging due to the following reasons:

- The transmitting and receiving antennas installed on a single mesh node should be separated enough to reduce the noise of the transmitter on the own receiver [26; 27].
- The current 802.11 commodity devices suffer interference caused from non-overlapping frequencies over each other due to lack of prefect filtering at the antenna [27].
- The MAC layer of IEEE 802.11 standard should be modified to support channel assignment mechanisms [3].
- The routing protocol must be modified to take the advantage of the underlying channel assignment [6; 16; 17; 26].

Channel assignment mechanisms are usually evaluated via simulations due to the multiple considerations and complex scenarios required to evaluate them [7–10; 13; 22–25]. However most of current network simulators do not support multi-radio routers nor dynamic re-configuration of the radio interface, requiring modifications in to their core-level to allow the evaluation of CA mechanisms [28].

Ns-3 [29] is a young simulator which allows the wireless mesh nodes to be equipped with more than one radio interface, and makes it possible to change the frequency of the radio during the simulation runtime. Although those features are necessary in order to simulate a multi-radio multi-channel network, they are not enough for simulating hybrid or dynamic CA mechanisms. Multi radio mesh networks simulated in ns-3 based on the recently published standard, IEEE 802.11s [2], are assumed to share common channels. The ns-3 modules for IEEE 802.11s can be used to simulate static or semi-dynamic CAs but they are insufficient for dynamic or hybrid CA. As in IEEE 802.11s, the peer links are formed over common channels, the Peer Management Protocol must be changed to be able to send beacons for neighboring mesh points which do not share any common channel. This modification is incompatible with IEEE 802.11s standard which assumes that mesh points operate in a single channel [2]. According to IEEE 802.11s standard, multi radio stations form different meshes on different channels, which are unified in a single LAN using the layer two bridging. We consider the problem of simulating hybrid CAs without restrict our work to IEEE 802.11s mesh networks.

In this work we show in detail how to simulate a hybrid channel assignment protocol using ns-3 simulator [29] without any need to modify the simulator's source code (Section II). We use the simulator version 3.9 released on August 2011. As a specific example, we present the simulation of the channel assignment proposed in [25] (Section III), followed by the simulation verification and the validation of the proposed CA model using a Markov chain (Section IV- V). A brief summary of other CA protocols proposed in the literature which could also be implemented following the presented approach, is described in Section VI and conclusions are presented in Section VII.

II. SIMULATION COMPONENTS FOR MULTI RADIO MESH NETWORKS

In this section, we introduce the required extensions to ns-3 simulator [29] for simulating hybrid CA mechanisms. These extensions include the basic functions on top of which the CA mechanism can be implemented. In addition, a CBR traffic generator and a specific routing strategy are presented as required components, to evaluate the performance of the CA mechanism.

A. Multi Radio Wireless Nodes

Multi radio wireless mesh nodes are the required basic building blocks of the simulation scenarios. Fig. 2 shows a wireless mesh node equipped with several radio interfaces.

The figure depicts a general cross-layer channel assignment protocol that interacts with the new components which must be added to the simulation: traffic generators; routing protocol; and the channel sensing mechanism.

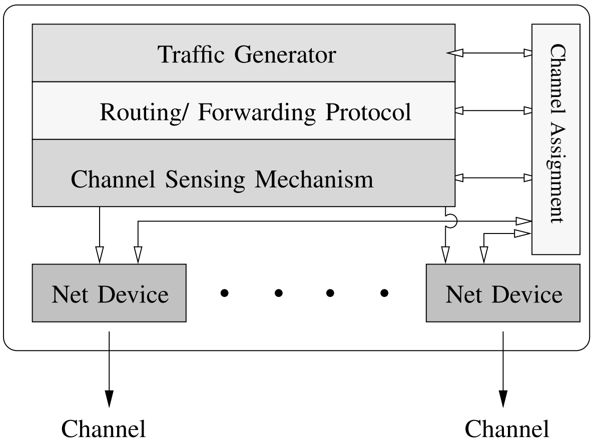


Fig. 2: Node object

In Program 1 it can be seen how to make a multi radio node (*MRNode*) in ns-3. Where *I* is the number of radio interfaces which must be installed on a node. Note that in ns-3 a unique ID starting from 0 is automatically assigned to each radio interface.

We configure the MAC and physical layers according to Table I which makes IEEE 802.11a based radio interfaces. The radio propagation model is set to the fixed range propagation model with a maximum range equal to 250 *m*. We also define a single transmission rate equal to 6*Mbps* for data and control packets transmission. Note that this configuration is not fixed and it could be changed to any of the propagation and channel loss models that ns-3 supports [29].

Program 1 Creating a multi radio wireless node

```
1  Ptr<Node> MRNode = CreateObject<Node> (); ///< Create a
       node
2  Ptr<WifiNetDevice> radio;
3  for (int i=1; i< I ; i++ ){
4    radio=CreateObject<WifiNetDevice> ();///< Create one
         radio interface
5    radio->SetMac (macConfiguration);///< Configure the MAC
         layer of the interface
6    radio->SetPhy(phyConfiguration);///< Configure the
         physical attributes of the interface
7   MRNode->AddDevice(radio);} ///< Add the device to the node
8   Ptr<NewClass> NewObject = CreateObject<NewClass>
9   ();///< Create another object, i.e. CA
10  MRNode ->AggregateObject(NewObject);///< Attach the
        object to the node
```

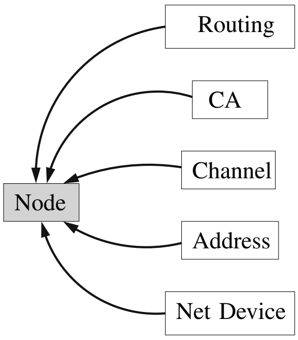


Fig. 3: Class Diagram of the Node

Lines 8- 10 of Program 1 shows how any other object (CA mechanism, routing protocol, etc.) is created and aggregated to the wireless mesh node. Fig. 3 shows the class diagram of a node in ns-3.

Sections II-B1- II-C provide a brief explanation of the components which must be added to the multi radio node to make it possible to simulate the channel assignment mechanism.

B. Network Devices

1) Configuring The Radio Interface: Ns-3 indicates each channel with a unique *ID*, which starts from 0. By default the channel assigned to all radios in a node is set to 0. To reconfigure the channel assigned to a radio interface, a node has to change the ID of the default channel in the physical layer of the radio interface. All channels is ns-3 are supposed to be non-overlapping.

Program 2 shows the process of assigning channel *c* to the first radio interface (*netDevice*) installed on *MRNode*.

Program 2 Assign channel *c* to a radio interface (netDevice)

```
1    Ptr <NetDevice> netDevice= MRNode->GetDevice(0); ///< Get
         the address of the radio interface
2    Ptr <WifiNetDevice> wifiNetDevice=
3    netDevice->GetObject<WifiNetDevice >();
4    Ptr<WifiPhy> netDevicePhyicalLayer = wifiNetDevice->
       GetPhy();///< Get the access to the physical layer
5      if (!netDevicePhyicalLayer->IsStateBusy())
6      netDevicePhyicalLayer->SetChannelNumber(c); ///<
           Change the channel of the radio interface
7      else
8      {
9        Time delayToIdle =
10      netDevicePhyicalLayer->GetDelayUntilIdle(); ///< Get
            the time left until the device get free
11       Simulator::Schedule(delayToIdle,&SwitchInterface,
            this, c); ///< Schedule the timer to change the
            channel later
12     }
```

Note that, if the interface is busy due to sending or receiving, it is not possible to switch to another frequency. Program 2 checks the status of the device (Line 5) before setting the new channel.

It is also possible to acquire the remaining time until the device gets idle (Line 9) and set the channel afterward (Line 11).

2) Data Service Components: To transmit packets in a multi radio multi channel architecture, where wireless nodes use a dynamic or hybrid CA, some extra tasks must be done to acquire the channel on which the packets must be sent, in addition to computing the time for switching a radio interface to the desired channel. Fig. 4 shows the main process that a node follows to deliver packets on the presented multi-radio multi-channel architecture. It comprises three steps: Destination Channel Query, Time Stamp Assignment and Packet Queuing

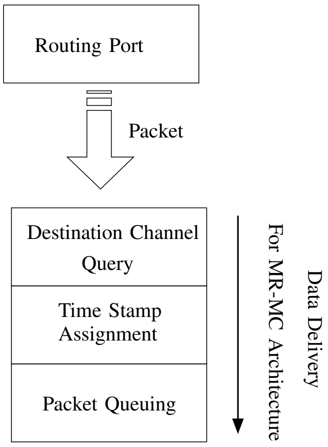


Fig. 4: Data Services For Channel Assignment Mechanism

Unlike single channel networks, a node in a multi channel architecture needs to consider the possibility of having next

Parameter Name	Description	Value
Standard	MAC and Physical layer standard	ns3::WIFI_PHY_STANDARD_80211a
PropagationDelay	The propagation delay between the specified source and destination	ns3::ConstantSpeedPropagationDelayModel
PropagationLoss	Modalize the propagation loss through a transmission medium	ns3::RangePropagationLossModel
MaxRange	Radio range	250 m
RemoteStationManager	Data and control packets transmission rate	ns3::ConstantRateWifiManager

TABLE I: Physical and MAC layer configuration

hop neighbors over different channels. Therefore, to forward a packet, a wireless mesh node needs to know the channel over which it can send the packet, in addition to the the address of the next hop mesh node.

The *Destination Channel Query* function finds the channel on which the packet must be sent. A *Packet Queuing* mechanism buffers packets and keeps them until the sender switches one radio interface to the corresponding destination channel. To avoid saturation in a given queue, the packets that have been waiting for a period of time exceeding a certain threshold, are dropped from the queue. For such purpose, a time stamp is assigned to each new arriving packet and used as a reference to drop the packets.

a. Destination Channel Query: Multi radio nodes keep the following information about their neighbors in addition to their address:

- The number of radio interfaces a node has,
- The channel assigned to each radio interface,
- At which time each node will switch from one channel to another,
- At which time each node can receive packets on each channel.

The mentioned information is kept in the *Neighbor Table*. To forward a packet, a node gets the receiving channel of the next hop neighbor. With this information the node can tune a radio to that channel and send the pending packets.

Nodes use control messages to update the information about their neighbors in the *Neighbor Table*. This is necessary for the adaptive channel allocation mechanisms, where a node changes the channel of its radio interface frequently.

Every time that a node informs its neighbors about switching the channel of its radio interface, the neighbors will update the entry in the *Neighbor Table*.

In a multi channel architecture, nodes may need to monitor the available channels to acquire the list of busy channels. When forwarding a packet, the node should make sure that the receiver is not in the monitoring mode. Therefore, the node needs to keep the next monitoring period for all of its neighbors to avoid initializing any transmission.

Program 3 shows an entry of the *Neighbor Table*.

b. Timestamp Assignment: Nodes must time stamp the packets waiting to be sent over a certain channel, in the case the destination channel is busy, the older packets must be discarded to avoid saturation. Therefore, it is possible to control the length of the buffers.

Program 3 An entry of the Neighbor Table

```
struct NeighborTableEntry
{
  uint32_t m_id; ///< The ID of the neighbor
  uint32_t m_neighborRadioNo;///< The number of radio
       interfaces that the neighbor has
  uint32_t m_neighborChannel; ///< The channel used by
       neighbor's radio interface
  Address m_Addr;///< MAC address of the neighbor's radio
       interface
  Time m_switchTime;///< Shows the remaining time that a
       neighbor stay on the current channel
  uint32_t m_neighborNewChannel;///< New channel for the
       receiving radio
  Time m_monitorTime; ///< Shows the remaining time until
       the neighbor starts monitoring a channel
  Time m_updateTime;///< Time stamp, which shows the
       moment that the information is updated
  bool close; ///< If the entry expired or not
}
```

The timestamp assigned to each packet is equal to the time at which the packet was received from the routing agent ($T_{enqueu}(p)$). We define $T_{wait}(p)$ as the maximum amount of time that a packet is allowed to stay in the buffer before it is transmitted over the wireless media. To remove the old packets, a node checks the current time of the system ($T_{current}$), and deletes those packets that $T_{current} > T_{enqueu}(p) + T_{wait}(p)$.

To avoid the queues from getting saturated in high traffic rate, nodes may select a smaller value for $T_{wait}(p)$.

c. Data Queuing: In multi-radio multi-channel networks, the number of available frequencies is bigger than the number of radio interfaces at each router ($|C| > I$). Therefore neighboring nodes may set their antennas over different channels. A wireless mesh node which has traffic for more than one neighbor, might need to switch to different channels to be able to deliver its packets to the next hop. During the time that a node transmits data over a channel, packets destined for other channels must be buffered to be sent later.

We create sequential First-in, First-out queues for buffering packets in ns-3 for different channels. Each queue corresponds to one of the available channels. The defined queues have the ability to eliminate the old packets automatically as described in Section II-B2.

Table. II shows the fields of each element of the queue with a brief description.

Element	Description
Packet Type	The type of the packet: Data or Control
Packet	A pointer to the packet
Expire Time	The time at which the packet expires and should be dropped

TABLE II: Packet buffers elements

C. *External Interference and Channel Sensing*

Simulating two separate wireless networks which interfere each other is not possible in ns-3. Separated wireless networks simulated in the same scenario do not have any effect over each other since ns-3 treated them separately. Note that, Signal to Noise plus Interference (SNIR) is defined as part of the physical layer of the WiFi device [30]. Therefore setting the noise figure weight for a radio interface affects all packets which have been received over different channels. For an adaptive channel assignment we have a scenario where each channel is occupied by different amounts of interference. Nodes estimate the amount of interference through sensing or monitoring all channels at the same time [16; 22]. Simulating this scenario is possible only if the interference threshold is set separately for each channel.

Therefore, to simulate the external interference on wireless channels, we have created an *Interference Emulator* for each channel (Program 4). The *interference emulator* is based on a semi Markov model of the possible channel status (Fig. 5), where the status of a channel is considered as either *Busy* or *Idle*.

The emulator keeps the channel *Busy* for predefined period of time (*BusyDuration*). The duration of *Idle* state is determined using an exponential random variable with mean equal to *MeanIdleTime*.

In the constructor (Line 8 of Program 4), a timer is initialized to call the *ChangeStatus* function when it expires. Initially the delay of the timer is set to an exponentially distributed random variable with mean equal to 8 *ms*.

ChangeStatus changes the status of the channel from *Idle_State* to *Busy_State* or vice versa. It sets the timer to *BusyDuration* after setting the channel status to *Busy_State*. Whenever it changes the channel status to *Idle_State* it sets the timer to a randomly selected duration.

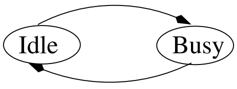


Fig. 5: Channel Status

To monitor a channel, a node checks the *Interference Emulator Status* attached to the channel during the sensing period.

The amount of interference over a channel, induced by an external network, can be varied by setting different values for *MeanIdleTime* and *BusyDuration*.

Program 4 External Interference Emulator

```
1  ChannelEmu::ChannelEmu():
2
3  busyDuration(<MilliSeconds> BusyDuration), ///< The
       duration of Busy status
4  nextTime(ExponentialVariable(MeanIdleTime)), ///< The
       random variable which determines the duration of
       Idle status
5  Status(Idle_State),///< Initial status of the emulator
6  statusTimer(Timer::CANCEL_ON_DESTROY)
7
8  { statusTimer.SetDelay(nextTime.GetValue());
9    statusTimer.SetFunction(&ChannelEmu::ChangeStatus,
         this); ///< Set the function which is called when
          the timer is expired
10   statusTimer.Schedule();}
```

D. *Routing Modifications*

When simulating multi radio multi channel WMNs, considering a proper routing protocol is challenging. The routing have to be considered as a joint problem with the channel assignment [16; 18–20], since any change in the channel radio mapping affects the quality of the links between nodes and may trigger the routing protocol to re-route the traffic. On the other hand, the changes in the traffic pattern have an impact on the next decision of the channel assignment protocol. Therefore, the routing protocol and the routing metric must be modified to adapt to the channel assignment [6].

In ns-3 the available routing protocols are not applicable for multi radio wireless networks when using dynamic or hybrid channel assignment. Nevertheless, as we are only focusing on fixed WMNs, where nodes are fixed or have limited mobility, the static routing is a simple way to avoid complexity related to the dynamic routing protocols. The Global Routing simulated in ns-3, provides static routing only for wired networks by filling the routing tables at the beginning of the simulation. In the case of wireless networks there is no static routing since the topology of a wireless network is determined by the propagation model and other parameters at run time [29].

Therefore, we have attached a static routing table to each node and initialized it using Shortest Path First (SPF) algorithm, minimizing the number of hops between each node and any other destination. Each node knows about the next hop node on the paths to all other nodes in the network. Program 5 shows an entry of the *Routing Table*.

The *Routing Table* is filled using a file containing all shortest paths between all nodes. We have created this file using R numerical tool [31] feeding the position of nodes in the network.

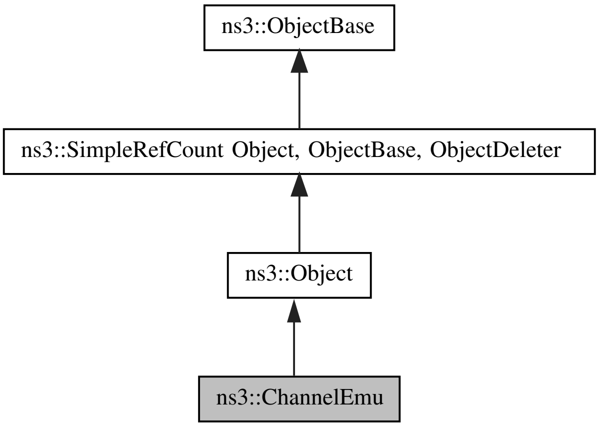


Fig. 6: Class diagram of Interference Emulator

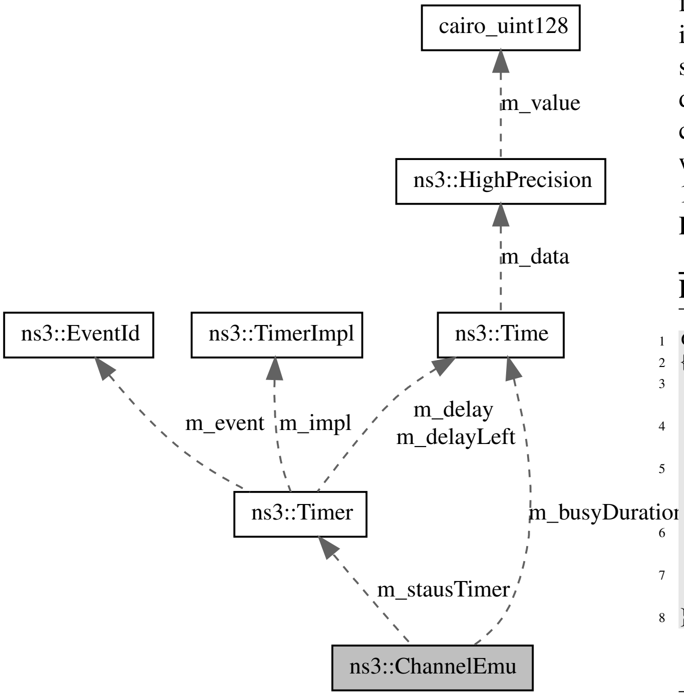


Fig. 7: Collaboration class diagram of Interference Emulator

Program 5 A routing table entry

```
struct RoutingEntry
  {
    uint32_t m_dst;///<The Id of the destination node
    uint32_t m_nextHop; ///<The Id of the next hop node
        toward the destination
    double  m_metric; ///<The metric value of the path
}
```

The routing header which is attached to each packet has the following elements:

- Sequence number,
- Node ID of the source,
- Node ID of the destination,
- Node ID of the next hop device to the destination,
- The time of originating the packet.

The relay nodes on the path from the source to the destination, update the ID of the next hop node in the header and froward the packet.

E. Traffic Generator

Channel assignment problem can be formulated in a way to reduce the time overlapping of low rate traffics and high rate traffics by using wireless mesh nodes to communicate over different channels based on their respective data rates [23; 32]. This approach increases the throughput of high rate traffics.

The traffic generated in ns-3, must be provided by an intermediate agent to interact with the dynamic CA in the case that, there is the need to control the traffic rate. Moreover for dynamic or hybrid CA, the packets can not be sent immediately through the media after the next hop has been selected, since the radio interface should first switch to the destination channel. For simplicity, we have created a simple constant bit rate (CBR) packet source that generates traffic with a specific packet size (1000 *b*) at the constant rate (i.e., 100 *kbps*). The packet generator is attached to the source node. Program 6 shows our CBR traffic generator.

Program 6 Constant Bit Rate (CBR) Packet Generator

```
1 GenerateCBRTraffic( uint32_t pSize, uint32_t pRate)
2 {
3   Ptr<Packet> p= Create<Packet>(pSize); ///< Create a
        packet with the defined size
4   RoutingPort(p);///< Deliver packet to the routing
        protocol
5   Time packetInterval= MilliSeconds(pSize/pRate); ///<
        Compute the time duration between creating each
        packet
6   Simulator::Schedule (packetInterval, &GenerateCBRTraffic,
        this
7   ,pSize,pRate);///< Call the function to create the next
        packet after the interval
8 }
```

The packet is created (Line 3) and delivered to the routing agent (Line 4) which adds the routing header to it and sends it to the data service (Fig 4). Then, the packet is enqueued to the corresponding queue (Section II-B2).

III. AN EXAMPLE OF CA IMPLEMENTATION: SICA

Fig. 8-9 show the class diagram and the collaboration diagram of using the defined classes to model a channel assignment protocol. In the following section we show with more detail the simulation of a channel assignment mechanism, called SICA, in ns-3 using the components introduced before. Moreover we provide the details of some important process regarding the *Data Forwarding* and *Channel Selection*.

Semi dynamic Interference aware Channel Assignment mechanism (SICA), is a protocol proposed and simulated in ns-3 [29], for wireless mesh networks [25]. The source code of SICA is available and can be accessed at [33].

SICA is implemented in a network where nodes are equipped with two radio interfaces, each one is able to use a

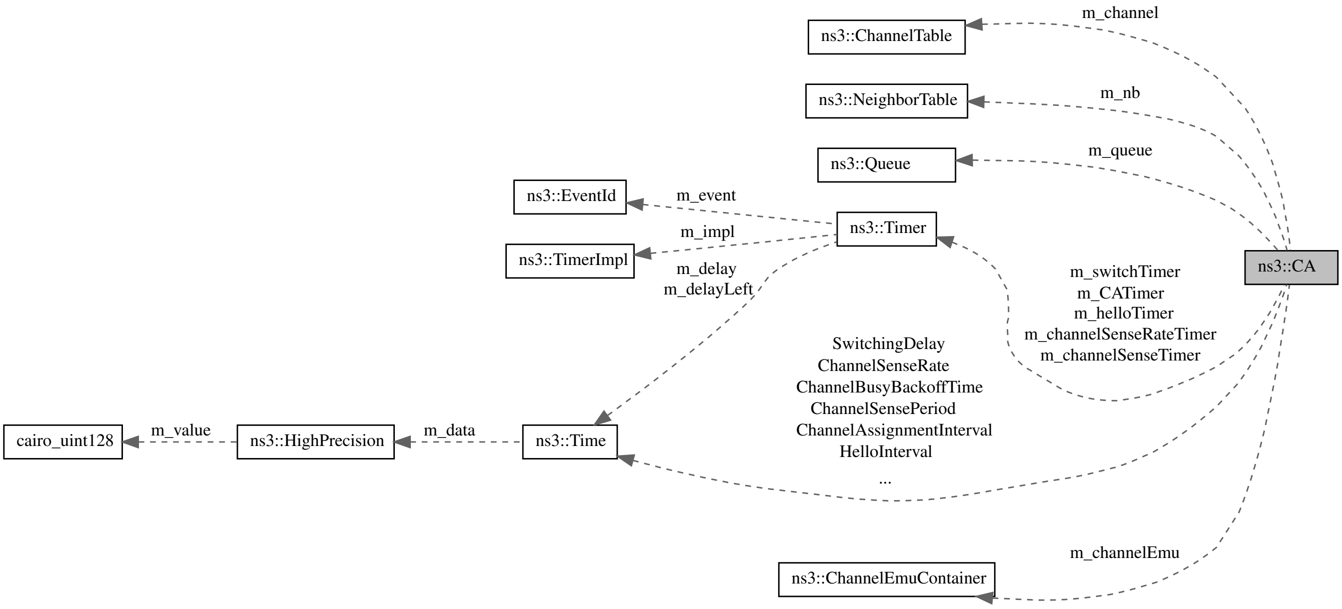


Fig. 9: Collaboration Class Diagram of the CA Class

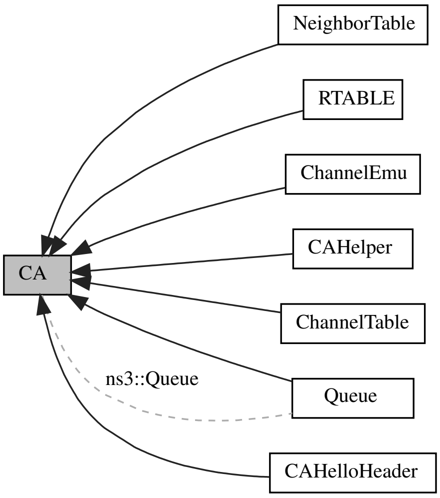


Fig. 8: Class Diagram of the CA Class

set C (with cardinality $|C| > 1$) of non overlapping channels. The radios will be referred to as the *receiving radio* and the *transmitting radio*, and denoted by R and T, respectively.

The aim of the channel assignment mechanism is to select the channels which suffer less interferences in terms of both internal and external interference. The channel assignment mechanism selects and assigns a channel to the R radio of each node. Then, the node switches the T radio according to the receiving channel of its neighbors to start transmission. After a channel switch, the T radio remains on the same channel until all packets, which are addressed to the same destination node, have been sent, or until a maximum period of time has expired [25].

Nodes estimate the amount of external interference on a channel via sensing the channel. Then, they use control packets called *Hello* to exchange channel sensing information and inform their neighbors about upcoming interface switching for the receiving radio (R).

A. External Interference Estimation

Using the *Channel Emulators* (Section II-C), the status of a channel can be monitored as *Busy* or *Idle*. Each node monitors the channel's status for its receiving radio (R) once in each sensing period of length T_{SS} seconds. For monitoring a channel c, the node checks the status of the corresponding *Channel Emulator* (c_e) at a pre-defined rate (T_{SRate}) (Program 7).

Program 7 Monitoring Channel

```
1 void SenseChannel(uint32_t c)
2 {
3   Ptr<ChannelEmu> emu=GetChannelEmu(c); ///< Get the interference emulator attached to the channel
4   if (emu->IsBusy())
5   T_busy+=ChannelSenseRate; ///< Increase the channel busy time if the channel is busy
6   else
7   T_idle+=ChannelSenseRate; ///< Increase the channel idle time if the channel is idle
8   m_channelSenseTimer.Schedule(); ///< Reschedule the sense timer for the next sensing period
9 }
```

T_{busy} (Line 5) and T_{idle} (Line 7), are global variables which save the duration of the *Busy* and the *Idle* state respectively.

At the end of the monitoring period the node estimates the amount of external interference over a channel (B_{ext}) using Equation (1).

$$B_{ext}(c) = \frac{T_{busy}(c)}{T_{busy}(c) + T_{idle}(c)} \tag{1}$$

Note that, each node senses only one channel during the monitoring period. It then sends this information to its neighbors and as its neighbors do the same, it gathers information about other channels via the control packets received from its neighbors (see Section III-C).

The internal interference is estimated based on the *Interference Protocol Model* proposed in [34]. Each node is informed by its neighbors about the current channel of their receiving radio. Therefore a node i can calculate the number of neighboring nodes over channel c ($R_i(c)$). Then the node estimates the density of interfering nodes on channel c by $\frac{R_i(c)}{N_i}$ where N_i represents the total number of neighbors of node i.

In SICA, nodes use the average of the estimated external and internal interference over a channel as a metric to select a channel that has more available capacity.

B. Channel Selection Mechanism

The channel selection mechanism is developed as a repeated game which uses the interference estimation information and selects the best possible channel for the receiving radio of a node.

Initially the channel selection mechanism calculates and assigns a weight to all available channels based on the amount of internal and external interference estimated over a channel. Then, the weights are updated using the multiplicative weight update technique proposed in [35; 36].

We define $w_t(c)$ as the weight assigned to channel c at time t. The channel assignment computes the probability of selecting a channel as :

$$P_t(c) = \frac{w_t(c)}{\sum_{c \in C} w_t(c)} \tag{2}$$

Initially all weights are set to 1, thus, the probability of selecting any channel is identical. After selecting a channel, the node gathers information from its neighbors and updates the loss that it is suffered. Then, the weights are updated as follows:

$$w_{i,t}(c) = w_{i,t-1}(c)\,\beta^{M_i(c)} \tag{3}$$

where $M_i(c)$ is the loss suffered by node i considering the external and internal interference over channel c [25], and β is the game parameter in the range of $(0,1)$ [36].

Then SICA selects a radom channel considering the probability of each channel as shown in Program 8.

Program 8 computes and sorts the probability of selecting a channel according to Equation. (2) (Line 11- 18 and Line 20). Then it sorts the list of available channels based on the weights (probabilities) assigned to each channel (Line 19). Line 21 computes the cumulative probability vector by adding up each probability value with all probabilities lower than it. Then the probabilities are fed to the empirical random variable generator (Line 22-23) which generates a random number as an index for the available channels' list (Line 24 of Program 8).

Program 8 Channel Selection Mechanism

```
1  uint32_t SelectRandomChannel(std::vector<uint32_t> channels
         )
2  {
3    /// channels contains the list of available channels
4    std::vector<double> prob; ///< A vector to keep the
           probability of selecting any channel
5    std::vector<double> weight;///< A vector to keep the
           weights assigned to  each channel
6    std::vector<double> cProb;///< A vector to keep the
           cumulative probability  of channels
7    EmpiricalVariable  emRnd;///< An empirical random
           variable to select a random number based on
           cumulative probabilities
8    double totalWeight=0;
9    double tempW;
10   /// read the weights of channels
11   for (std::vector<uint32_t >::iterator i=channels.begin();
           i!=channels.end(); ++i)
12     {
13       tempW=m_channel.GetChannelWeight(*i);
14       weight.push_back(tempW);
15       totalWeight+=tempW;
16     }
17   for (uint32_t i=0; i<weight.size(); i++)
18     prob.push_back(weight[i]/totalWeight);///< Compute the
           probabilities using the weights
19   sort(channels.begin(),channels.end(),
           CompareChannelsWeight(*this));///<  Sort channels
           considering their weights
20   sort(prob.begin(),prob.end());///< Sort the probabilities
21   cProb=ComputeCumulativeProbability(prob);///< Adds up the
            probabilities to have the cumulative probability
           vector
22   for (uint32_t i=0; i< cProb.size(); i++)
23    emRnd.CDF(i,cProb[i]);///< Feeds the probabilities to
           the random variable
24   return  channels[emRnd.GetInteger()];///< Select a
           random channel considering the assigned probabilities
25 }
```

C. Control Packet Elements

Unlike most CAs proposed in the literature [16; 20; 22], in SICA there is no common control channel shared by all nodes. In SICA, the synchronization is achieved through exchanging packets over the data channels. Since each node can assign a different channel to its receiving (R) radio, nodes must be aware of the channels used by their neighbors' R radios. In SICA, a node broadcasts *Hello* packets to report the channel of its R radio to its neighbors. Fig 10 shows the content of a *Hello* packet.

In addition to the receiving channel announcement, *Hello* packets are used to inform about the channel sensing information and the receiving channel of the neighboring nodes, so a node can compute the external and internal interference for each channel. In order to do that, there is a field in each packet which contains the amount of external interference over the receiving channel estimated by the node. The number of neighbors and the channels of their receiving radio is also added at the end of each *Hello* packet.

The remaining time before a node switches its R radio to another channel and the remaining time before the node starts sensing the current channel is also announced via *Hello* packets. The announcements inform the neighboring nodes about upcoming channel switching or sensing events.

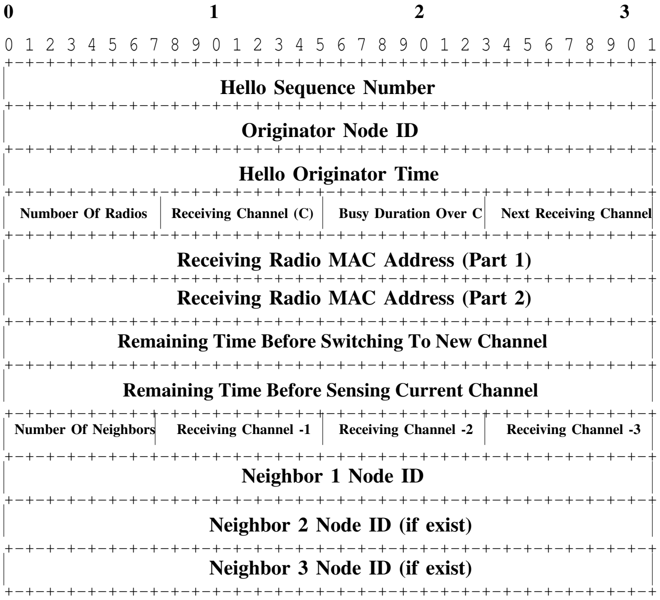


Fig. 10: Hello Packet Elements

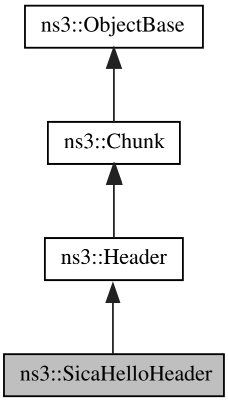


Fig. 11: Hello Header Class Diagram

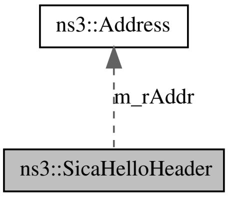


Fig. 12: Hello Header Collaboration Class Diagram

D. Data And Control Transmission

For each channel $c \in C$, SICA maintains two sequential First In First Out queues for buffering packets (Section II-B2): control packets queue ($Q_{ctrl}(c)$); and data packets queue ($Q_d(c)$). $Q_{ctrl}(c)$ has higher priority than $Q_d(c)$. The source and each node in the middle of the path to the destination, push each packet to the corresponding queue attached to each channel. Each packet is time stamped when it is pushed to the queue. Using the timestamp as a reference, packets that remain in the queue for a longer time than a certain threshold are discarded to avoid saturation (see Section II-B2).

When the transmitter radio switches to a channel, it fetches the stored packets from the queues and sends them until there are no packets left in the queues or until a maximum amount of time has elapsed [25]. Program 9 shows how packets are sent over a channel c.

Program 9 Send packets over channel c

```
1   uint32_t helloQueueSize=Q_ctrl.GetSize(c); ///< Get the
        size of the queue which keeps control messages
2   uint32_t dataQueueSize=Q_d.GetSize(c); ///< Get the size
        of the queue which keeps data messages
3   Time txEstimation=EstimateTxDuration(maxPacketSize,
        wifiphy);
4   while ((helloQueueSize>0 || dataQueueSize>0 )&&
        TInterfaceReadyToSend(c,txEstimation))
5   {
6     if (helloQueueSize>0)
7     {
8     protocolNumber=SICA_HELLO_PORT; ///< Set the port
          number to control port number
9     qEntry=Q_ctrl.Dequeue(c); ///< Fetch a queue entry of a
          control message
10    }
11    else
12    {
13      protocolNumber=SICA_DATA_PORT; ///< Set the port
            number to data port number
14      qEntry=Q_d.Dequeue(c); ///< Fetch a queue entry of a
            data message
15    }
16    if ( qEntry)
17    {
18      SendPacket(qEntry->GetPacket()->Copy(),m_tInterface
            ,protocolNumber); ///< Send data or control
            messages using different port number
19    }
20    helloQueueSize=Q_ctrl.GetSize(c);
21    dataQueueSize=Q_d.GetSize(c);
22  }
```

TInterfaceReadyToSend (Line 4) is a function which gets the ID of the channel and the estimation of the required time to send a packet by the T radio, and checks whether it is possible or not to send a packet over the channel.

In the following cases the *TInterfaceReadyToSend* returns *False*, which prevents the node from sending packets over channel c:

- The channel is being sensed by any of the neighbors.
- The channel is busy due to the external interference.
- The T interface did not switch to the channel successfully.
- The estimated transmission time is less than the remaining time over the current channel.

IV. VALIDATING THE GAME MODEL

The channel assignment in SICA is modeled as a repeated game, where nodes compete to occupy the most vacant channel (as explained in detail in Section III-B). The game is played repeatedly in a sequence of *game rounds*. In each round, the player plays a mixed strategy based on the weights assigned to each channel.

In the following section we explain the Markov model we use to validate the game theory model used in SICA. The Markov model is used to proof that, the game theory based channel assignment algorithm converges to a steady state if the

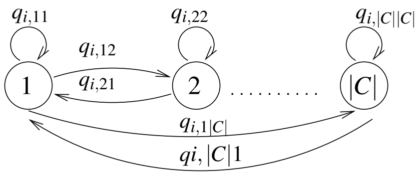


Fig. 13: Markovian process of changing channel at node i

algorithm is played in sufficiently enough rounds in the case that the environment does not change. We simulate the game model using R numerical tool [31] and compare the result with the results obtained using the Markov model to compute the probability of selecting channels. We define the process of selecting a channel for a radio interface at each node as a Markov chain.

Fig. 13 shows the Markov chain that models how a node selects the channels. The channels set (strategy set) was mapped to the state space of the Markov model. The payoff $P_i = 1 - M_i$, is considered as the reward for the transition between channel c and $\acute{c}$ by node i.

We define the state transition matrix (Q) by using the Boltzmann distribution (equation (4)) [37]. Boltzmann distribution is used to solve the Markov model, by defining the transition from state c to $\acute{c}$ related to the reward obtained on state $\acute{c}$.

We assume that the probability of transmission from channel c to $\acute{c}$ is related to the reward of the destination channel normalized by the total reward of all channels:

$$Q_{i,c\acute{c}} = \frac{e^{\frac{P_i[\acute{c}]}{\lambda}}}{\sum_{k \in C} e^{\frac{P_i[k]}{\lambda}}} \tag{4}$$

Here λ is the learning parameter. For our model we define $\lambda = 1 - \beta$, where β is the learning parameter used for the Game model of SICA (see Section III). Note that small values of λ enable the player to choose the optimal strategy more accurately.

The stationary transition matrix (i.e the eigenvalues of Q) is computed using the global balance equation for ergodic Markov chains:

$$\begin{cases} \rho(Q - I) = 0 \\ \rho E = e \end{cases} \tag{5}$$

Where I is the identity matrix, E is the square unit matrix and e is the unit row matrix.

By solving equation (5) we get the stationary channel transition vector for node i as $\rho = e(Q + E - I)^{-1}$. Using the R tool we've simulated the Game based learning algorithm of SICA for a node. We define $\acute{\rho}$ as the final probability vector of selecting each channel equal to the mixed strategy vector obtained for the node. We compare $\acute{\rho}$ obtained for different rounds of SICA algorithm, with ρ obtained from the Markov model (Fig. 14- 15). Moreover we run SICA in ns-3 simulator and report the probability distribution of selecting channels over 200 s of simulation run. (Fig. 16- 17).

Fig. 14 shows the probability of selecting a channel at each node. For clarifying, we show the available bandwidth of each channel with a red line at the same figure with the probability distribution of selecting a channel. As shown, the available bandwidth in each channel decreases linearly from channel 1 to 8. The right y-axis shows the available bandwidth related to the total bandwidth of a channel while the left y-axis shows the probability of selecting the channel.

Fig. 14 shows that the weighted learning algorithm matches quite closely the result from solving the Markov model after 100 rounds for $\beta = 0.9$.

We examine SICA and the Markov model for the case that 50% of the channels (channels with the IDs from 5 to 8) are occupied by the external interference. The interference occupies 80% of the bandwidth of those channels, while channels with the IDs from 1 to 4 are almost free.

Fig. 15 shows that, in this specific scenario, when the decision about selecting the best channels, is more clear, the learning algorithm of SICA converges to the Markov solution much faster, in this case only after 15 rounds.

To study the behavior of SICA over time, we simulate SICA using ns-3 simulator for a grid network of size 5×5 nodes and 2 CBR traffic flows of 100 kbps. The results are obtained averaging over 10 simulation runs and the error bars show the 90% confidence interval.

Fig 16 shows how the probability of selecting a target channel changes over time, when a different amount of external interference is introduced on each channel. Specifically, the channel with the higher ID suffers more external interference. The figure shows that, the nodes tend to select the channel which offers more bandwidth, which is the expected behavior of any CA.

Fig 17 shows the average probability of selecting each channel over time when the external interference is introduced on the first four channels (channels with the IDs from 1 to 4) while other channels are free. The figure shows that the probability of selecting any of the busy channels tends to 0 over time, while the probability of selecting any of the free channels (channels with the IDs $5 - 8$) increases to 0.25 which confirms the results shown in Fig 15. Moreover, it can be observed how SICA achieves a proportional utilization of the four free channels.

V. VERIFYING THE SIMULATION MODEL

To verify the simulation in ns-3, we design a state space explorer (SSE) which is invoked by the major events that happen in the simulations and performs state checking to verify the correctness of the simulation [38].

We assume that the simulation, as a system, consists of a group of entities joined together to accomplish the packet delivery goal and hence virtually, represents the operation of the real system.

We consider each radio interface as an *entity* with the following attributes:

- The frequency channel it is using.
- The number of packets it has received.

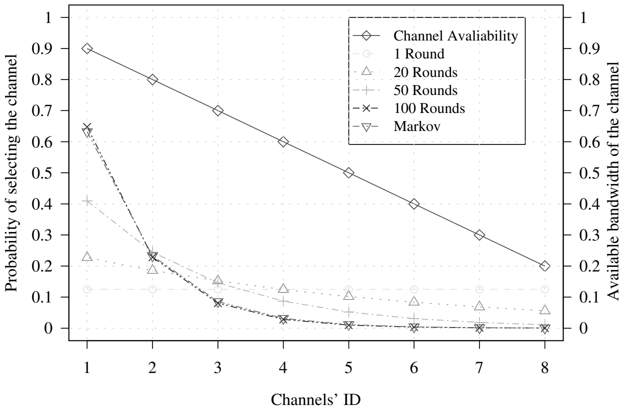


Fig. 14: The probability of selecting a channel for a node regarding the channel occupancy for different rounds of the game

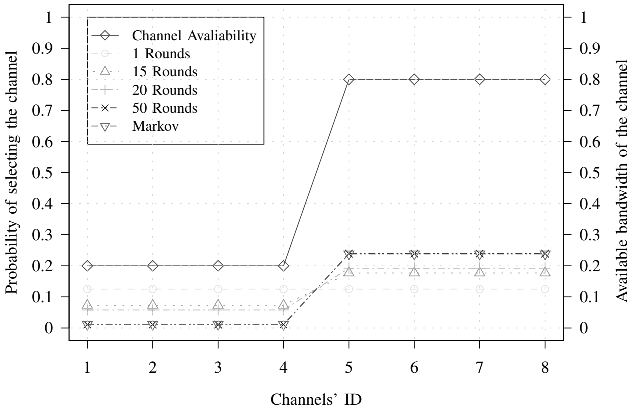


Fig. 15: The probability of selecting a channel for a node regarding the channel occupancy for different rounds of the game

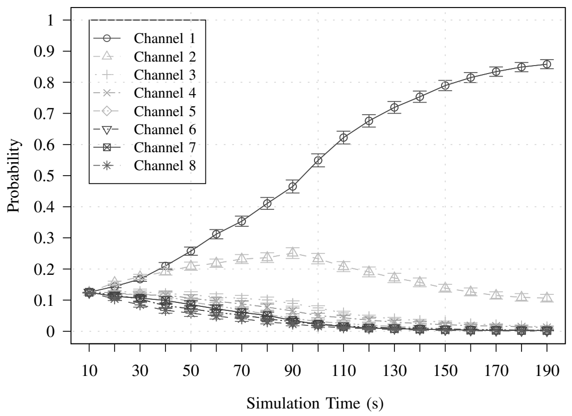


Fig. 16: The probability of selecting a channel for a node over time

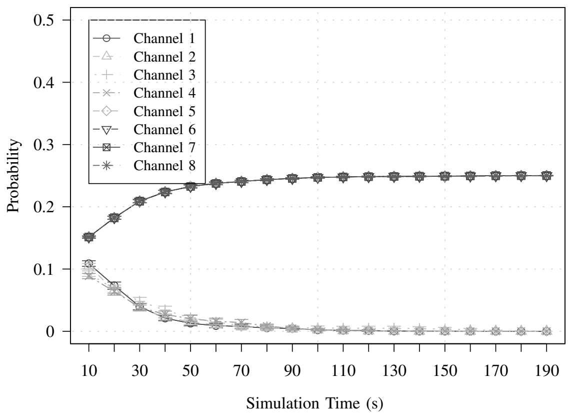


Fig. 17: The probability of selecting a channel for a node over time

- The number of packets that has sent.
- The number of packets that has dropped.

The *State* of the system (s_t) is the complete description of the system at any time which includes all entities and values of their attributes. We consider those attributes which are relevant to the objective of interest. s_t represents a function of all entities which assigns a value to all attributes, we consider that, ξ contains all possible states of the system, and call it *state space*.

The *Event* (e_t) is any occurrence which changes the state of the system. In our model we consider the following events:

- Sending a packet (*Send*, unconditional event).
- Receiving a packet (*Receive*, conditional, $e_t = Receive|\ e_{t_0} = Send\ \ t_0 < t$).
- Dropping a packet (*Drop*, conditional , $e_t = Drop|\ e_{t_0} = Send\ \ t_0 < t$).
- Changing the channel of the radio interface (*Channel-Change*, unconditional event).

We define $\mathcal{E}$ as the set containing all above events. The first three events occurred between two entities: the sender and the receiver.

An ***Assertion*** is a property which must be hold for all states of the system to be able to proof the correctness of the simulation.

For the simulation of the channel assignment protocol we define the following assertions:

- Only one channel must be assigned to a radio,
- There must be a common channel between the sender radio and the receiver radio at any given time.

We've designed SSE in ns-3, which starts from the initial state of the system and reaches the successor state following the simulation events, and checks for any violation from the assertions.

s_0 is defined as the initial state of the simulation. We define $\acute{s}$ is explored by s_0 if $\acute{s}$ is the successor state of the system after the triggering event e ($e \in \mathcal{E}$) and we notate it as: $s_0 \xrightarrow{e} \acute{s}$. The event handler is triggered by any of the events in $\mathcal{E}$ and it calls the *state control* procedure to check the assertion of the successor event.

$f_{i,c}$ is defined as the number of receiving radios of node i which are tuned to channel c. For any node i the assertion condition is:

$$\sum_{k=1}^{|C|} f_{i,k} = I_i \ , \ \ \forall i \in N \tag{6}$$

where I_i represents the number of radio interfaces of node i.

For any transmission pairs (i, j) the assertion condition is:

$$\exists c \in C \ , \ \ f_{i,c} \cdot f_{j,c} = 1 \tag{7}$$

Alg. 1, shows the state check algorithm which is triggered by sending, receiving or dropping packet events and checks if any violations happen from the assertions.

Algorithm 1 StateCheck1(S_t,e_{t-1},P)

```
Input:
  $S_t$: The current state of the system at time $t$.
  $e_{t-1}$: The last event which moved the system to state $S_t$.
  $P$: The pointer to the packet in the case of sending and receiving events.
if $S_t == S_0$ then
    exit(0)
else
    $src = e_{t-1} -> Sender()$
    $dst = e_{t-1} -> Receiver()$
    for $c \in C$ do
        if $f_{src,c} \cdot f_{dst,c}$ then
            Write('Assertion holds')
            exit(0)
        end if
    end for
    Write('Assertion violated')
    exit(1)
end if
```

Algorithm 2 StateCheck2(S_t,e_{t-1})

```
Input:
  $S_t$: The current state of the system at time $t$.
  $e_{t-1}$: The last event which moved the system to state $S_t$.
if $S_t == S_0$ then
    exit(0)
else
    $N = e_{t-1} -> Node()$
    $C = N -> Channels()$
    $I = N -> Radios()$
    if $C <= I$ then
        Write('Assertion holds')
        exit(0)
    end if
    Write('Assertion violated')
    exit(1)
end if
```

Alg 2 shows the process of checking *Channel-Change* event. The state checker gets the node which selects the new channels and checks whether the number of channels selected by the node is equal or less than the number of radio interfaces it has. In SICA each node selects only one channel at a time and all nodes are equipped with two radio interfaces, thus the violation from the assertions never happens for the *Channel-Change* events.

Fig. 18-19 are obtained running *State Space Check* algorithm (Alg. 1) for checking the simulation events of SICA. The simulation parameters are the same as the parameters considered for Fig. 17.

Fig. 18, shows the state space control for *Send*, *Receive* and *Drop* events. The figure depicts the number of events that happened during 200 *s* of simulation run, and the number of events for which the assertion holds. The figure shows that, SICA does not violate any of the assertions.

Fig. 19 shows the number of packets which have been sent, received or dropped on each channel when the external interference is introduced over channels with the IDs from 1 to 4. The figure confirms that SICA makes the network to use the capacity of the free channels efficiently.

Fig. 20 shows the average number of radio interfaces over each channel during the simulation time. Note that the simulation parameters are the same as Fig. 19. The figure shows that the number of radio interfaces over the set of channels that are occupied by external interference decreases over time. It proves that, the channel assignment successfully detects and avoids the channels which are occupied by external interference as it is expected.

Fig. 21 shows the average number of packets (with a size of 1kb) in each data queue over the time. The figure shows that the proposed data delivery mechanism ensures that the number of waiting packets in each channel is lower than the default maximum queue size in ns-3 (100 packets) [29]. In addition the maximum number of packets waiting in a queue at the beginning of the simulation (marked by red), happened over those channels that are kept busy by external interference.

VI. RELATED WORK

Channel assignment is a problem that must be solved toward the implementation of wireless mesh networks and it has been studied extensively during the last years. Although many solutions have been considered for channel assignment [3–13; 16–24; 39], few proposals are adaptive to the changes in the wireless environment such as the interference induced from other wireless networks [16; 22; 25].

The first interference aware CA mechanism is Breath First Search Channel Assignment (BFSCA) [16]. BFSCA is a priority based and centralized algorithm which assigns better channels to the links that are close to the gateway. The channel assignment assumes that there is a common channel between nodes for coordination. Another relevant adaptive and semi-dynamic CA proposal is Urban-X [22]. Urban-X is a traffic and interference aware CA which relies on a common channel between nodes for exchanging control packets.

Semi-dynamic Interference aware Channel Assignment (SICA) [25] is a Game based CA mechanism that considers both the internal and external interference. SICA is distributed and assumes that, in wireless networks, nodes do not have perfect information about their neighbors. This is the main contribution that differentiates SICA from other Game based CAs [24; 40–43]. Moreover, nodes do not need to tune one

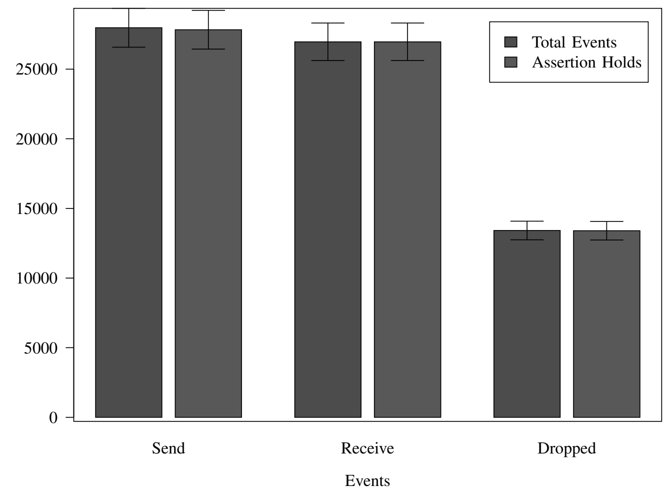


Fig. 18: Protocols events and the assertions

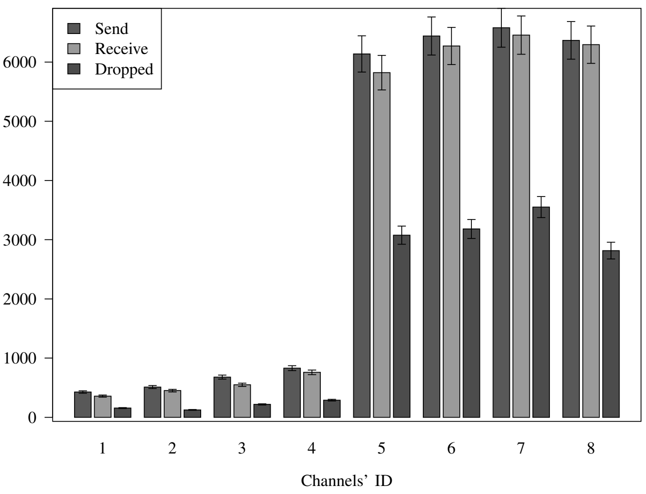


Fig. 19: Number of events over each channel

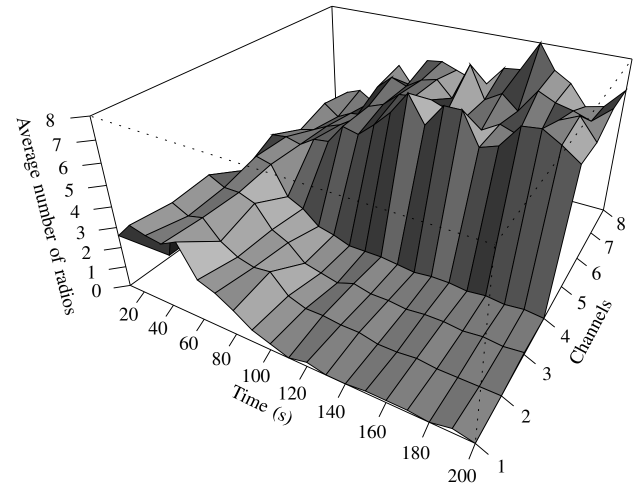


Fig. 20: The average number of radio interfaces over each channel during the simulation time

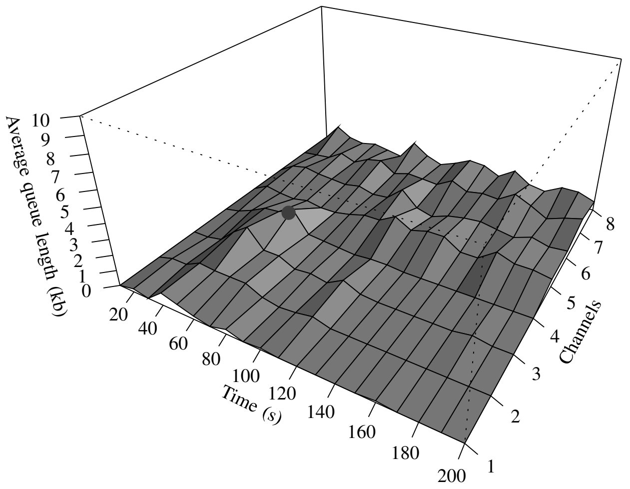


Fig. 21: The average length of queues during the simulation time

radio interface over a common channel. The implementation of SICA in ns-3 has been detailed in previous section.

Simulating SICA and any other hybrid or dynamic CA in the current network simulators needs some general components which are missing in all simulators. The necessary modifications for ns-2 simulator [44], for evaluating multi radio wireless networks, are presented in [28]. The manual is restricted to the static channel assignment, which assigns a channel to the radio interface of a node before the simulation starts and keeps the configuration until the end of the simulation. Unlike the previous work, we provide the essential steps toward simulating channel assignment protocols in ns-3 simulator in this paper, including the validation of the game model used in SICA with a Markov model.

Furthermore, to the best of the authors knowledge, this is the first time that the correctness of the channel assignment simulation is controlled through checking the simulation events.

VII. CONCLUSIONS

In this paper we have presented the extensions which must be done in ns-3 simulator to simulate channel assignment mechanisms for multi radio wireless networks. We provided the simulation details of the Semi-dynamic Interference aware Channel Assignment (SICA) mechanism which is proposed in [25] as an example. In addition the source code of SICA is published and available at [33]. We verified the simulation of SICA using a state space checker which checks the relevant simulation events for any violation from the feasibility conditions of channel assignment solution. The results prove the correctness of the simulation and show that SICA is capable of utilizing the channels in a fair and efficient way. Moreover we justified the results obtained by the Game theory based model of SICA, using a Markov chain model.

In the future, we plan to expand our implementation by improving the data delivery mechanism to incorporate differentiated priority scheduling, such that higher priority traffic can be transmitted in preference to lower priority traffic. It is also desired to consider traffic rate adaptation for channel assignment in addition to designing a method to avoid saturation of data queues for high rate traffic. We also plan to investigate the necessary modifications to the routing protocol for making it able to work along hybrid channel assignment protocols.

ACKNOWLEDGMENTS

This work was supported by the Spanish Government and Generalitat de Catalunya through projects TIN2010-21378-C02-01, TEC2008-0655, and 2009-SGR-1167, respectively, and by the European Commission through the NoE EuroNF.

REFERENCES

[1] "Ieee standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," 2007, iEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999).

[2] "Ieee standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 10: Mesh networking," 2011, iEEE Std 802.11s-2011 (Amendment to IEEE Std 802.11-2007).

[3] 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, 2008.

[4] L. Zhang, X. Wang, and C. Liu, "Channel assignment in multi-radio multi-channel wireless mesh network by topology approach," in *Proc. WRI Int. Conf. Communications and Mobile Computing CMC '09*, vol. 2, 2009, pp. 358–362.

[5] L. Xu, Y. Xiang, and M. Shi, "A novel channel assignment algorithm based on topology simplification in multi-radio wirelesss mesh networks," in *Proc. 25th IEEE Int. Performance, Computing, and Communications Conf. IPCCC 2006*, 2006.

[6] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM New York, NY, USA, 2004, pp. 114–128.

[7] M. K. Marina, S. R. Das, and A. P. Subramanian, "A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks," *Computer Networks*, vol. 54, no. 2, pp. 241 – 256, 2010.

[8] A. Subramanian, H. Gupta, S. Das, and J. Cao, "Minimum interference channel assignment in multiradio wireless mesh networks," *IEEE Transactions on Mobile Computing*, pp. 1459–1473, 2008.

[9] S. Avallone and I. F. Akyildiz, "A channel assignment algorithm for multi-radio wireless mesh networks," *Comput. Commun.*, vol. 31, no. 7, pp. 1343–1353, 2008.

[10] H. Skalli, S. Ghosh, S. K. Das, and L. Lenzini, "Channel assignment strategies for multiradio wireless mesh networks: Issues and solutions," *Communications Magazine, IEEE*, vol. 45, no. 11, pp. 86–95, 2007.

[11] B. Shao, J. Tao, and F. Wang, "Static channel assignment with the physical interference model for maximum capacity in multi-radio multi-channel wireless mesh networks," in *Proc. 9th Int Grid and Cooperative Computing (GCC) Conf*, 2010, pp. 338–343.

[12] A. Raniwala, K. Gopalan, and T.-c. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 8, no. 2, pp. 50–65, 2004.

[13] M. Amiri Nezhad and L. Cerdà-Alabern, "Utility based channel assignment mechanism for multi radio mesh networks," in *Proceedings of the 8th ACM international workshop on Mobility management and wireless access*, ser. MobiWac '10. New York, NY, USA: ACM, 2010, pp. 68–74.

[14] M. Gong, S. Midkiff, and S. Mao, "On-demand routing and channel assignment in multi-channel mobile ad hoc networks," *Ad Hoc Networks*, vol. 7, no. 1, pp. 63–78, 2009.

[15] P. Bahl and R. Chandra, "Ssch: Slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks," in *in IEEE 802.11 Ad-Hoc Wireless Networks, in ACM Mobicom*, 2004, pp. 216–230.

[16] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," in *Proc. 25th IEEE Int. Conf. Computer Communications INFOCOM 2006*, 2006, pp. 1–12.

[17] X. Wu, J. Liu, and G. Chen, "Analysis of bottleneck delay and throughput in wireless mesh networks," in *Proc. IEEE Int Mobile Adhoc and Sensor Systems (MASS) Conf*, 2006, pp. 765–770.

[18] M. Alicherry, R. Bhatia, and L. E. Li, "Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 11, pp. 1960–1971, 2006.

[19] A. H. Mohsenian-Rad and V. W. S. Wong, "Joint logical topology design, interface assignment, channel allocation, and routing for multi-channel wireless mesh networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 12, pp. 4432–4440, 2007.

[20] A. Raniwala and T.-c. Chiueh, "Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network," *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3, pp. 2223–2234, August 2005.

[21] P. Kyasanur and N. H. Vaidya, "Routing and link-layer protocols for multi-channel multi-interface ad hoc wire-

less networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 10, no. 1, pp. 31–43, 2006.

[22] M. D. F. Wooseong Kim, Andreas J. Kassler and M. Gerla, "Urban-x: Towards distributed channel assignment in cognitive multi-radio mesh networks," in *IFIP Wireless Days*, Venice, Italy, 2010.

[23] S.-H. Kim and Y.-J. Suh, "Rate-based channel assignment algorithm for multi-channel multi-rate wireless mesh networks," in *Proc. IEEE Global Telecommunications Conf. IEEE GLOBECOM 2008*, 2008, pp. 1–5.

[24] M. Felegyhazi, M. Cagalj, S. Bidokhti, and J.-P. Hubaux, "Non-cooperative multi-radio channel allocation in wireless networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, May 2007, pp. 1442 –1450.

[25] M. A. Nezhad and L. Cerda-Alabern, "Adaptive channel assignment for wireless mesh networks using game theory," in *Proc. IEEE 8th Int Mobile Adhoc and Sensor Systems (MASS) Conf*, 2011, pp. 746–751.

[26] A. Dhananjay, H. Zhang, J. Li, and L. Subramanian, "Practical, distributed channel assignment and routing in dual-radio mesh networks," in *SIGCOMM '09: Proceedings of the ACM SIGCOMM 2009 conference on Data communication*. New York, NY, USA: ACM, 2009, pp. 99–110.

[27] J. Robinson, K. Papagiannaki, C. Diot, X. Guo, and L. Krishnamurthy, "Experimenting with a multi-radio mesh networking testbed," in *in (WiNMee Workshop*, 2005.

[28] R. Aguero-Calvo and J. Perez-Campo, *Adding Multiple Interface Support in NS-2*, University of Cantabria, Cantabria, Spain, 2007. [Online]. Available: http://personales.unican.es/aguerocr/

[29] ns-3 development team, "ns-3 manual (release ns-3.9)," 2011.

[30] M. Lacage and T. R. Henderson, "Yet another network simulator," in *Proceeding from the 2006 workshop on ns-2: the IP network simulator*, ser. WNS2 '06. New York, NY, USA: ACM, 2006. [Online]. Available: http://doi.acm.org/10.1145/1190455.1190467

[31] R. D. C. Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2008. [Online]. Available: http://www.R-project.org

[32] S.-C. Yip, S.-W. Tan, and T.-C. Chuah, "Data rate-aware channel assignment algorithm for multi-rate multi-channel wireless mesh networks," in *Proc. 6th Int Wireless and Pervasive Computing (ISWPC) Symp*, 2011, pp. 1–6.

[33] M. A. Nezhad, "Semi-dynamic interference aware channel assignment (sica)," Feb. 2012. [Online]. Available: http://personals.ac.upc.edu/maryam/

[34] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388 –404, Mar 2000.

[35] Y. Freund and R. E. Schapire, "Game theory, on-line prediction and boosting," in *COLT '96: Proceedings of the ninth annual conference on Computational learning theory*. New York, NY, USA: ACM, 1996, pp. 325–332.

[36] Y. Freund and R. Schapire, "Adaptive game playing using multiplicative weights," *Games and Economic Behavior*, vol. 29, no. 1-2, pp. 79–103, October 1999. [Online]. Available: http://ideas.repec.org/a/eee/gamebe/v29y1999i1-2p79-103.html

[37] L. Weibing, W. Xianjia, and H. Binbin, "Evolutionary markov games based on neural network," in *Proceedings of the 6th International Symposium on Neural Networks: Advances in Neural Networks - Part III*, ser. ISNN 2009. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 109–115. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-01513-7_12

[38] A. Sobeih, M. D'Amorim, M. Viswanathan, D. Marinov, and J. C. Hou, "Assertion checking in j-sim simulation models of network protocols," *Simulation*, vol. 86, pp. 651–673, November 2010. [Online]. Available: http://dx.doi.org/10.1177/0037549709349326

[39] W. Si, S. Selvakennedy, and A. Y. Zomaya, "An overview of channel assignment methods for multi-radio multi-channel wireless mesh networks," *J. Parallel Distrib. Comput.*, vol. 70, pp. 505–524, May 2010. [Online]. Available: http://dx.doi.org/10.1016/j.jpdc.2009.09.011

[40] L. Gao, X. Wang, Y. Xu, and W. Chen, "Distributed multi-radio channel allocation in multi-hop ad hoc networks." in *ICC'08*, 2008, pp. 3156–3160.

[41] T. Chen and S. Zhong, "Perfectly fair channel assignment in non-cooperative multi-radio multi-channel wireless networks," *Computer Communications*, vol. 32, no. 6, pp. 1058 – 1061, 2009.

[42] I. Shah, S. Jan, and K.-K. Loo, "Selfish flow games in non-cooperative multi-radio multi-channel wireless mesh networks with imperfect information," in *Wireless and Mobile Communications (ICWMC), 2010 6th International Conference on*, 2010, pp. 219 –225.

[43] H.-K. Kim, C.-Y. Oh, and T.-J. Lee, "Fair and Collision-Aware Multi-Channel Assignment Based on Game Theory for Wireless Multi-Hop Networks," *IEICE Transactions on Communications*, vol. 92, pp. 1282–1290, 2009.

[44] K. Fall and K. V. (eds.), "The ns manual (formerly ns notes and documentation)," 2003.

Game Theory Formulation of Channel Assignment for Multi-Radio Multi-Channel Wireless Networks with Unbalanced Resources

Maryam Amiri Nezhad, Boris Bellalta, Manel Guerrero Zapata, Llorenç Cerdà-Alabern and Enric Monte Moreno

***Abstract*—The channel assignment problem in a dynamic wireless environment is investigated using game theory. Unlike previous game models, we consider diverse available bandwidth for different channels in addition to different data rate requests at wireless routers. Users are assumed to be selfish and try to obtain the same share of bandwidth as their requested data rate. The game is considered repeated with incomplete information. We develop a distributed real time multiplicative weights update learner algorithm to solve the game for a multi collision domain network. The optimization problem for finding the best response is solved to compare with the learner answer. The simulation results show that our proposal outperforms the previous proposals.**

***Index Terms*—Channel Assignment; Multi-Radio; Multi-Channel; Wireless Mesh Network; Game Theory; Real-time Learning**

I. INTRODUCTION

CHANNEL assignment (CA) [1] deals with selecting the best channels for an individual wireless node or the entire network. This is a special challenging task in distributed wireless networks, as each channel may be shared by many wireless devices from the same or different networks, which makes it difficult to predict the amount of available bandwidth in each channel.

Game theory is considered to solve the non-cooperative and distributed channel assignment problem where nodes have conflict of interests for the wireless bandwidth [2]–[4]. However the proposed models do not consider the dynamic nature of wireless environment, which results in a different amount of available bandwidth in each channel due to presence of external devices. Moreover, they assume perfect information at each player, and in some cases there is no constraint for the number of radios installed on nodes [4]. However, in a scenario with external interference, it is not possible to have the perfect knowledge of the channel use before moving to that channel. The other shortage of the previous proposals is assuming all nodes to seek for a maximum identical amount of bandwidth while in a wireless network, nodes may reduce their data rate to avoid congestion or to save energy. The proposed games are solved disregarding the effect of past events on future decisions.

In this work we present a game theory formulation for the channel assignment problem with incomplete information in a dynamic wireless environment. The size of the network is not a limitation for our model neither the strategy of other users for channel assignment. We propose a real time learning algorithm to solve the game. We have evaluated our model using R numerical tool [5].

II. PROBLEM FORMULATION

A. Scenario

We consider a multi radio wireless network that consists of N wireless nodes which are equipped with several radio interfaces. We assume that there are $|C|$ non-overlapping channels available. We define $I_i < |C|$ as the number of radio interfaces installed on node i. Each wireless channel suffers external interference caused from external devices which belong to other networks. The goal of a node is to use the channels which have more available bandwidth in order to satisfy its required data rate.

We define the normalized available bandwidth of a channel c, $B_{free}(c) = \frac{BW(free,c)}{BW(total)}$, as the amount of bandwidth which is free over channel c related to the total bandwidth of the channel, where $BW(total)$ is identical for all channels. We assume that nodes can estimate the occupancy of channels using any of the bandwidth estimation methods proposed in the literature [1]. We define $r_{i,c}$ as the number of radios that node i has placed on channel c. Since the node does not get any benefit from placing more than one radio on the same channel, $r_{i,c}$ is limited to 1 ($r_{i,c} \leq 1$). Therefore, the bandwidth share available for node i over channel c is equal to:

$$B_{i,quota}(c) = \frac{B_{free}(c) r_{i,c}}{max(\sum_{n \in N} r_{n,c}, 1)} \quad (1)$$

We define ϑ_i as the bandwidth that node i wants to acquire. This minimum bandwidth can be satisfied using one or more channels at the same time.

M.Amiri Nezhad (corresponding author), M.Guerrero Zapata and L.Cerdà-Alabern are with Dept. of Computer Architecture, Universitat Politecnica de Catalunya, Barcelona, Spain e-mail:maryam,guerrero,llorenc@ac.upc.edu.

B. Bellata is with the Dept. of Information and Communication Technologies, Universitat Pompeu Fabra, Barcelona, Spain e-mail:boris.bellalta@upf.edu.

E. Monte Moreno is with Dept. of Signal Theory and Communications, Universitat Politecnica de Catalunya, Barcelona, Spain e-mail: enric@gps.tsc.upc.es.

This work was supported by the Spanish Government and Generalitat de Catalunya through projects TIN2010-21378-C02-01, TEC2008-0655, TEC2009-14094-C04-01, and 2009-SGR-1167, respectively, and by the European Commission through the NoE EuroNF.

B. Channel Assignment Game

We consider the problem of finding the best channel assignment as a game played between each node and the environment. The environment consists of all other nodes of the same network, as well as the devices from other networks that also operate in the same set of channels, that is, all wireless nodes which work in the interference range of node i. From now on we shall use the terms node and player interchangeably. We assume that, players are rational and selfish. However they do not know exactly the payoff matrix of the game, but they are able to measure the current strategy's payoff at the time. The game is played over several rounds and players can modify their decision in order to obtain a higher payoff.

We define s_i as the strategy of player i, which is the channel allocation vector of the player, given by $s_i = (r_{i,1}, r_{i,2}, ..., r_{i,C})$.

A node strategy, s_i, describes whether it has a radio over a specific channel or not which can be either a pure strategy or a mixed one.

We define the strategy matrix (strategy profile), S, as the strategy vector of all players at a given time. By S_{-i} we shall refer to the strategy matrix which consists all nodes' strategies except player i.

We define the payoff of playing the strategy vector s_i equal to the bandwidth share that node i gets from all channels, on which it has radios, to the total available bandwidth of all channels in its neighborhood.

$$F(s_i, S_{-i}) = \frac{\sum\limits_{c\in C} B_{i,quota}(c)}{\sum\limits_{c\in C} B_{free}(c)} \tag{2}$$

We define φ_i as the minimum payoff that node i expects to gain considering its required bandwidth (ϑ_i).

$$\varphi_i = \frac{\vartheta_i}{\sum\limits_{c\in C} B_{free}(c)} \tag{3}$$

Nodes are not able to know the payoff of all possible strategies but they can choose a channel by measuring the payoff in the current strategy profile. However they must avoid *channel oscillation*. Channel oscillation happens when nodes detect that a given channel is occupied by many nodes and they decide to switch to another channel at the same time, eventually they will find the target channel occupied again. To avoid channel oscillation we use a Markov model to compute the expected payoff of selecting any channel discounting the expectation of the neighboring nodes (Section II-C).

C. Markov Model

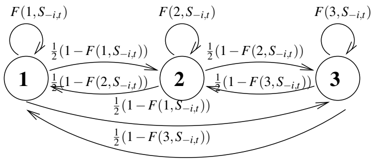


Fig. 1: Markov model of transition from one strategy to another

The behavior of a node can be modeled as a Markov chain, when the node keeps the current strategy with the probability proportional to the payoff it gets and changes to other strategies with the probability proportional to the complementary of staying in the current strategy. Fig. 1 shows a chain with 3 strategies or states available for a node with one radio interface while $S_{-i,t}$ is the current strategy of the environment. The probability of staying in strategy s_i for a node is self transition over s_i and is equal to $p_{s_i s_i} = F(s_i, S_{-i,t})$. We assume that, the node has the same tendency to leave its strategy and occupy any other strategies. Therefore the probability of departure from strategy s_i, to any other strategy $\acute{s_i}$, is considered as $p_{s_i \acute{s_i}} = \frac{1-F(s_i,S_{-i,t})}{|S_i|-1}$. Note that for a node with I_i radios, $|S_i| = \binom{|C|}{I_i}$ is the number of possible channel selections (strategies) while $|C|$ channels are available. The transition matrix of the Markovian process is:

$$P_i(S_{-i,t}) = \begin{bmatrix} F(1,S_{-i,t}) & \frac{1-F(1,S_{-i,t})}{|S_i|-1} & \frac{1-F(1,S_{-i,t})}{|S_i|-1} & \dots \\ \frac{1-F(2,S_{-i,t})}{|S_i|-1} & F(2,S_{-i,t}) & \frac{1-F(2,S_{-i,t})}{|S_i|-1} & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \tag{4}$$

We define the predicted payoff of selecting any strategy as the stationary vector, that is, the probability of selecting a strategy after a hight number of steps while $S_{-i,t}$ remains unchanged ($\rho_{i,t} = \rho_{i,t} P_i(S_{-i,t})$).

III. ONE COLLISION DOMAIN SOLUTION

We first investigate the scenario where nodes have complete information and the game is played simultaneously at each stage. Later, we will provide a new algorithm to solve the game in a general scenario. This section is a benchmark to justify the results of the general algorithm.

We define the goal of the channel assignment as maximizing the total bandwidth that all nodes acquire in the network. We further assume that there is only one collision domain.

The best response is the strategy profile that leads to the maximum achievable bandwidth considering the limited number of radios (I_i) and the bandwidth required by each node (φ_i).

We define best response as the strategy that minimizes the difference between the bandwidth that a node gains and the bandwidth it requires. Therefore, for a given S, the best response can be stated as the following optimization problem:

Minimize:

$$G(S) = \sum_{i\in N} \left[\frac{\vartheta_i}{\sum\limits_{c\in C} B_{i,quota}(c)} \right] \tag{5}$$

subject to:

$$\sum_{c\in C} r_{i,c} \le I_i \quad \forall i \in N$$

$$\sum_{i\in N} \xi_i r_{i,c} \le B_{free}(c) \quad \forall c \in C$$

Note that, ξ_i shows the bandwidth demand of a node for each channel (6), since we assume that, a node which has radios over more than one channel uses all channels proportionally.

$$\xi_i = \frac{\vartheta_i}{\sum_{c \in C} r_{i,c}} \tag{6}$$

To obtain the best response, the minimization problem is solved using Lagrangian multipliers:

$$L(G(S), \lambda_1, \ldots, \lambda_{N+|C|}) =$$

$$\sum \left[\frac{\vartheta_i}{\sum\limits_{c \in C} B_{i,quota}(c)} \right] - \lambda_1 \left[\sum_{c \in C} r_{1,c} - I_1 \right] - \ldots$$

$$-\lambda_N \left[\sum_{c \in C} r_{N,c} - I_N \right] - \lambda_{N+1} \left[\sum_{i \in N} r_{i,1} \xi_i - B_{free}(1) \right] - \ldots$$

$$-\lambda_{N+|C|} \left[\sum_{i \in N} r_{i,|C|} \xi_i - B_{free}(|C|) \right]$$

In detail, the equations that are solved are:

$$\frac{\partial L}{\partial r_{i,c}} = 0 \quad \forall i \in N; \quad \forall c \in C$$

together with the complementary slackness conditions:

$$\lambda_i \left[\sum_{c \in C} r_{i,c} - I_i \right] = 0 \quad \forall i \in N$$

$$\lambda_{N+c} \left[\sum_{i \in N} r_{i,c} \xi_i - B_{free}(c) \right] = 0 \quad \forall c \in C$$

To solve the optimization problem we must assume that a node knows the payoff of all possible strategies, which is only possible if all nodes are in one collision domain. In the next section we provide an algorithm which solves the game without any need to deal with the optimization problem.

IV. GENERAL MULTI COLLISION DOMAIN SOLUTION

We propose a real-time learner algorithm which uses the local information at each node and is based on a multiplicative weights update scheme [6]. The main idea of the real-time learner is as follows: 1) each player assigns a weight to each possible strategy considering its payoff, 2) if the current strategy does not provide the desired payoff, the player selects a random strategy considering the weights, 3) the player updates the weights based on the local observation of the channel occupancy.

The algorithm minimizes the loss that players suffer without knowing the payoff matrix of all strategy combinations.

Let $E[F(S_{-i,t})]$ be the *expected payoff vector* of node i. Therefore considering the strategy profile $S_{-i,t}$, the expected payoff of selecting strategy s_i is $E[F(s_i, S_{-i,t})]$:

$$E[F(s_i, S_{-i,t})] = \delta E[F(s_i, S_{-i,t-1})] + (1-\delta) F(s_i, S_{-i,t}) \tag{7}$$

where δ is the memory tuning parameter that takes values in the range $[0,1)$. Equation (7) works as a low pass filter with an exponential impulse response, that approximates the mean value of $F(s_i, S_{-i,t})$.

A node assigns a non-negative weight ($w(s_i)$) to each element of $E[F(s_i, S_{-i,t})]$. Initially $E[F(s_i, S_{-i,t})]$ is unknown to player i, but as the game is played repeatedly in a sequence of game rounds $(1, \ldots, T)$, it is updated at each round. In each round $t \in [1, \ldots, T]$, the player plays a mixed strategy based on the weights assigned to the elements of $E[F(s_i, S_{-i,t})]$. The probability of selecting the strategy s_i, is calculated as the related magnitude of its weight to the total weights of available strategies:

$$p_t(s_i) = \frac{w_t(s_i)}{\sum\limits_{s \in S_i} w_t(s)} \tag{8}$$

Initially all weights are set to 1, thus, the probability of selecting any strategy is identical following a uniform distribution. Once one strategy has been selected, the node observes the network state, that is, the external interference and the channels occupied by its neighbors. Based on that, it measures its obtained payoff and updates the weight assigned to strategy s_i using the following formula:

$$w_t(s_i) = \begin{cases} 1, & F(s_i, S_{-i,t}) \geq \varphi_i \text{ OR} \\ & F(s_i, S_{-i,t}) \geq max(\rho_{i,t}) \\ w_{t-1}(s_i)\, \beta^{(1-E[F(s_i, S_{-i,t})])}, & \text{otherwise} \end{cases} \tag{9}$$

Here β is the game parameter in the range of $(0,1)$ which yields a multiplicative update [6]. The player selects a new strategy only if it does not obtain the expected payoff (3) and the current payoff is lower than the maximum payoff computed by the Markov model (Section II-C). This implies that, a node stops oscillating between channels when the bandwidth it gets with the current strategy is adequate. Note that, reaching a stable solution is a key point since oscillating between channels reduces the network performance due to the switching delay of the current wireless antennas.

The performance of the decision made by the learner using the multiplicative weights update scheme, depends on the value of β. With β near 1, the algorithm introduces minor changes to the weights and the learner, follows the environment more accurate but slowly. On the contrary, a β close to zero, imposes big changes in the weights, and introduces a higher error to the decision. Therefore, it is adequate to a scenario with frequent changes. Note that the best solution reached by the learner is not necessarily the Nash Equilibrium, since it has been shown that multiplicative weights update learning algorithm cannot work for Nash Equilibrium in general bi-matrix games [7].

V. PERFORMANCE EVALUATION

The performance of the real-time learner channel assignment (RTLCA) algorithm (Section IV) is evaluated and compared with the best response solution and two other game models proposed in the literature.

We first consider a scenario with $N = 5$ nodes, where each node has $I_i = 2$ radio interfaces. The number of available channels is $|C| = 5$. All channels have the same total bandwidth and it is normalized to 1. The amount of external interference in each channel follows a uniform random variable which picks values in the range between 0.2 and 0.8. The values of external interference are kept unchanged during the simulation.

The learner algorithm is examined for different values of β, δ and φ and compared with the best response solution. Remember that δ is the memory tuning parameter for computing the payoff (7), β is the game parameter (9), and φ is the minimum payoff that a node expects to gain (3).

Fig. 2 shows the average distance between the payoff obtained by RTLCA and the maximum achievable payoff from the best response for different values of δ. Each point in the chart shows the result of running the algorithm for $T = 50$ rounds and is averaged over 50 runs with different random seeds. The error bars show 95% of confidence interval. We have run the simulation for $\beta \in \{0.1, 0.9\}$ and $\varphi \in \{0.2, 0.8\}$. The figure shows that, changing φ has a big impact on the results. With small φ the small difference between the best possible payoff and the payoff obtained by the algorithm shows that it reaches the expected results easily. For a network where nodes have high expectation for payoff (big value of φ) with a small value of β, the best result is achieved when δ is high (and vice versa). This comes from the fact that a small β imposes a big change to the weights assigned to the strategies, and nodes are prone to select different strategies that are less dependant to the past, this is get adjusted by having a big value for δ as a memory tuning parameter.

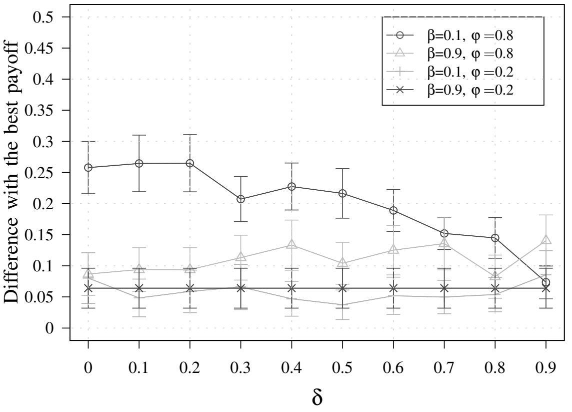


Fig. 2: The performance of the learner related to the best response vs. different values of δ

We evaluate the performance of RTLCA with two other game based channel assignments proposed in the literature: Centralized Nash Equilibrium Channel Assignment (CNECA) [8]; and Centralized Nash Equilibrium Multi Collision domain Channel Assignment (CNEMCCA) [3]. CNECA considers all nodes interfere each others while CNEMCCA is designed for multi collision domain networks. Both algorithms distribute the interfering radios on different channels fairly. However none of the proposed algorithms considers the different available bandwidth for different channels.

Fig. 3 is obtained for grid networks with different number of nodes ($9 \leq N \leq 100$). The number of available channels is $|C| = 12$. Nodes' requested payoff follows a uniform random variable which picks values in the range between 0.1 and 1 (3). We obtain results for RTLCA with $\beta = 0.2$ and $\gamma = 0.8$. The other parameters are the same as for Fig 2.

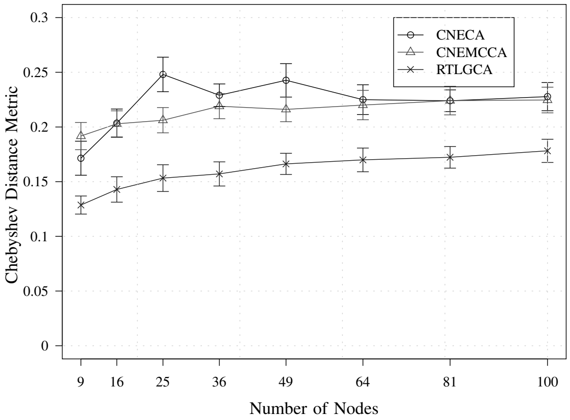


Fig. 3: Chebyshev distance between the obtained and the demand payoff vs. different number of nodes

Fig. 3 shows the Chebyshev distance metric between the obtained payoff and the payoff which is required by the players for different game based channel assignments. The results show that, RTLCA is more successful in providing the required bandwidth, for players, in unpredictable wireless environment.

VI. CONCLUSION

In this paper we have presented a game theory based formulation for the channel assignment problem in multi-radio wireless networks. The considered problem is formulated using a repeated game with incomplete information which is general enough to be applicable for any wireless network. We solve the game using a multiplicative weights update learning algorithm which adapts to the changes in the environment and reaches a desired solution in a limited amount of time.

REFERENCES

[1] W. Si, S. Selvakennedy, and A. Y. Zomaya, "An overview of channel assignment methods for multi-radio multi-channel wireless mesh networks," *J. Parallel Distrib. Comput.*, vol. 70, pp. 505–524, May 2010.

[2] M. Felegyhazi, M. Cagalj, S. Bidokhti, and J.-P. Hubaux, "Non-cooperative multi-radio channel allocation in wireless networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, May 2007, pp. 1442 –1450.

[3] R. D. Vallam, A. A. Kanagasabapathy, and C. S. Murthy, "A non-cooperative game-theoretic approach to channel assignment in multi-channel multi-radio wireless networks," *Wirel. Netw.*, vol. 17, no. 2, pp. 411–435, Feb. 2011. [Online]. Available: http://dx.doi.org/10.1007/s11276-010-0288-5

[4] S.-H. Kim and Y.-J. Suh, "Rate-based channel assignment algorithm for multi-channel multi-rate wireless mesh networks," in *Proc. IEEE Global Telecommunications Conf. IEEE GLOBECOM 2008*, 2008, pp. 1–5.

[5] R. D. C. Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2008. [Online]. Available: http://www.R-project.org

[6] Y. Freund and R. E. Schapire, "Game theory, on-line prediction and boosting," in *COLT '96: Proceedings of the ninth annual conference on Computational learning theory*. New York, NY, USA: ACM, 1996, pp. 325–332.

[7] C. Daskalakis, R. Frongillo, C. H. Papadimitriou, G. Pierrakos, and G. Valiant, "On learning algorithms for nash equilibria," in *Algorithmic Game Theory*, 2010, pp. 114–125.

[8] J. P. H. Mark Felegyhazi, "Game theory in wireless networks: A tutorial," EPFL Technical report: LCA-REPORT-2006-2, 2 2006.