Conception et optimisation de performance inter-couches dans les réseaux maillés radio multi-canal multi-interface

Carina Teixeira de Oliveira Teixeira de Oliveira

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Carina TEIXEIRA DE OLIVEIRA

Thèse dirigée par **Andrzej Duda**
et codirigée par **Fabrice Theoleyre**

préparée au sein de **UMR 5217 - LIG - Laboratoire d'Informatique de Grenoble**
et de **École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique (EDMSTII)**

Cross-Layer Design and Performance Optimization of Multi-Channel Multi-Interface Wireless Mesh Networks

Thèse soutenue publiquement le **26 octobre 2012**,
devant le jury composé de :

M. Hervé Martin
Professeur, Université Joseph Fourier, Président
M. Guy Juanole
Professeur Émérite, Université de Toulouse, Rapporteur
M. Marcelo Dias de Amorim
Directeur de Recherche CNRS, Université Pierre et Marie Curie, Rapporteur
M. Juliusz Chroboczek
Maître de Conférences, Université Paris Diderot, Examinateur
M. Andrzej Duda
Professeur, Grenoble INP – Ensimag, Directeur de thèse
M. Fabrice Theoleyre
Chargé de Recherche CNRS, Université de Strasbourg, Co-Directeur de thèse

Acknowledgments

First and foremost, I would like to express my sincere gratitude and deep appreciation to my supervisors, Dr. Andrzej Duda and Dr. Fabrice Theoleyre, for their valuable advice and guidance during the course of my research. I appreciated all their contributions of time, ideas, and support to make my Ph.D. experience productive and stimulating.

I offer my sincere regards to members of the thesis committee: Dr. Guy Juanole from University of Toulouse, Dr. Marcelo Dias de Amorim from Pierre and Marie Curie University, Dr. Hervé Martin from Joseph Fourier University, and Dr. Juliusz Chroboczek from Paris Diderot University, who have given me many valuable suggestions and comments.

I wish to thank all past and present members of Drakkar team for creating a pleasant environment for carrying this work. Also, I would like to thank the administrative and technical staff members of Grenoble Informatics Laboratory and Joseph Fourier University, who have been more than kind to advise and help in their respective roles.

I would like to thank Dr. Mark Davis for his supervision during my Ph.D. internship at Dublin Institute of Technology in Ireland. I really appreciated his scientific advice and insightful comments in my work.

My time at Grenoble was made enjoyable in large part due to the many friends; special thanks to Serge, Gisele, Nicole, Jacqueline, Bernard, Aida, Isabel, Ana, Maru, Nazim, Azzeddine, Asif, Bogdan, Maciej, Ghalem, Mouna, Betul, Jun-Young, Aastha, Naeem, Sofia, Raquel, Elton, Samuel, Hyane (in memoriam), Fany, Patrick, Fabi, Lucas, Bia, Cesar, Shirley, Isaac, Guri Lucas, Guria Letícia, Renato, Clayton, Soraia, Elizandra, Luiz Carlos, Talitha, Cony, André, Alex, Nathalie, Tales and Jose.

A very special thank to my parents, Mauro and Lidu, sisters, Karol and Carolina, and my whole family, who have been extremely understanding and supportive of my studies. I feel very lucky to have a family that shares my enthusiasm for academic pursuits.

Last but not least, I would like to thank my husband, Reinaldo, who has encouraged me so much over the years. His love and fellowship are my constant source of inspiration and aspiration.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Context

Wireless networks have gained great popularity in recent years due to their convenience in setup, usage and maintenance. Their reduced wired infrastructure and large-scale commercialization combined with increasing data rates have revolutionized the way we communicate. Added to this fact the proliferation of mobile computing and communication devices such as laptops, Personal Digital Assistants (PDAs), smartphones, and tablets through which mobile users can access all the required information whenever and wherever needed.

All these facilities boosted the development of a set of different wireless technologies including the Wireless Local Area Network (WLAN) that offers a flexible way of communication while allowing for user mobility. The standard communication technology for WLANs is IEEE 802.11 [1], commercially known as Wireless-Fidelity (Wi-Fi). IEEE 802.11 describes the PHYsical (PHY) layer and the Medium Access Control (MAC) layer for wireless communication frequencies in the ranges of 2.4 gigahertz (GHz) and 5 GHz. Since its release in 1997, the standard was amended many times to include different physical layers (e.g., IEEE 802.11a [2], IEEE 802.11b [3], IEEE 802.11g [4], IEEE 802.11n [5]) or introduce new capabilities such as security (IEEE 802.11i [6]) and quality of service (IEEE 802.11e [7]). It has been deployed in a variety of environments such as university campus, enterprise buildings, public places, and homes [8, 9]. It is therefore a natural choice for research and innovation to improve the performance of wireless networks using the 802.11 technologies.

Networks operating in the IEEE 802.11 standard can perform in two modes: infrastructured and ad hoc. The first, and by the far most common mode, is characterized by the presence of an Access Point (AP) that intermediates all communication between stations (i.e., end users devices) associated to it. The AP acts as a centralized element of traffic. Thus, the coverage to stations is limited within the transmission range of the AP. Usually, the APs are connected to a wired network (typically the Internet) to extend access to the wireless stations. Thus, extend the wireless coverage becomes costly and impractical due to extensive cabling requirements.

On the other hand, the ad hoc mode consists only of stations that communicate directly with each other in a peer-to-peer manner (i.e., in a single-hop) without any AP. In this case, some stations might not be able to communicate directly to each other because of their limited transmission range. This drawback can be overcome by multi-hop networking: the communication goes through multiple hops/stations before reaching its final destination [10]. In such cases, stations act as forwarders for other stations in the network.

Since IEEE 802.11 was designed for single-hop wireless network [1], many different protocols were proposed to add routing capabilities to stations and surpass the lacking multi-hop communication. Destination-Sequenced Distance-Vector (DSDV) [11], Ad hoc On-demand Distance Vector (AODV) [12], Optimized Link State Routing (OLSR) [13] and Dynamic Source Routing (DSR) [14] are examples of such protocols. Nevertheless, multi-hop routing solutions still present limitations. In general, these protocols present high routing overhead and limited scalability [15]. Besides that, the ad hoc network may become disconnected with a set of isolated networks due to the dynamicity of the wireless medium condition (e.g., path loss, fading, interference) and the mobility of stations [16].

A Wireless Mesh Network (WMN) is a promising wireless technology that combines the advantages of infrastructured and ad hoc modes [17, 18]. WMNs have attracted increasing attention in recent years because of their low-cost, ease of deployment, robustness, and reliable service coverage. These networks provide broadband wireless coverage to large areas without infrastructure requirements while guaranteeing connectivity even with the mobility of users and the dynamicity of the medium. WMN have been used in numerous application scenarios such as broadband home networking, community networks, metropolitan area network, enterprise networking, and intelligent transportation systems [17].

The popularity of WMNs has lead to the recent IEEE 802.11 standardization of a mesh networking solution for broadcast and unicast packet delivery over a self-configured multi-hop topology. The standard is called IEEE 802.11s [19] Mesh Networking. It proposes, among other *mesh services*, path selection and forwarding with routing capabilities at the MAC layer, interworking with external networks and security facilities.

WMNs are composed of wireless routers, called *mesh routers*, interconnected to form a multi-hop backbone. Mesh routers are dynamically self-organized, self-configured and self-healing without any centralized control. With these capabilities, WMN can be incrementally deployed, one node at a time, according to the demand. Besides that, mesh routers are usually stationary and, consequently, they can be permanently power-supplied and benefit from resources such as memory, energy, and computation power. To extend the wired network access beyond the transmission range of a single AP, mesh routers interconnect wirelessly to establish and maintain *mesh connectivity* among themselves. They forward across multiple hops the traffic received from stations as well as the traffic received from other mesh routers. Therefore, the wireless mesh backbone combines advantages of both infrastructured and ad hoc modes.

Mesh routers can also play a role of an gateway/bridge. This functionality enables the integration of WMN with different wired and wireless networks like Ethernet, cellular, Wi-Fi, and sensors. Some of the mesh routers also act as gateways toward the Internet via high-speed wired links. Through an integrated WMN, the users of these wired or wireless networks can benefit from otherwise impossible services of these networks.

Note that, WMN involves a set of challenging research characteristics such as compatibility between different wireless technologies, integration between wired and wireless networks, stations mobility, and ubiquity.

In this thesis, we focus on the issues of mesh routers[1], leaving the Internet and the WLAN issues (e.g., stations mobility, communication between mesh routers and stations) to other areas of research. Moreover, we focus on IEEE 802.11-based backbone wireless mesh networks.

Since energy saving and mobility are no longer issues for mesh routers, most of the research so far has focused on the optimization of MAC and routing layers in terms of achievable throughput, end-to-end delay and scalability [20].

In the initial design of WMN, mesh routers were typically equipped with a single IEEE 802.11 wireless network interface card [2] that operates over only a small portion of the available spectrum (a channel). It is well known that such *single-channel single-interface networks* present a limited scalability (i.e, the increase of the network size) [21]. Besides, the performance degrades when the node density and the number of hops increase, which is mainly due to interference between nearby flows (known as inter-flow interference) and also between the nearby hops in a

[1]In this thesis, we also refer to the *mesh router* as *node* or *Mesh Station* (Mesh STA) hereafter.
[2]Interface, radio, and transceiver are synonymous.

single flow (known as intra-flow interference) [22]. Then, increase the scalability and performance gains of such wireless network has been the challenge of much recent research.

One way of improving the performance is to use multiple orthogonal channels (non-overlapping / free of inter-channel interference) so that different mesh routers may simultaneously operate on different channels. For instance, IEEE 802.11b/g standards provide 3 orthogonal channels (1, 6, 11) in the 2.4 GHz spectrum, while IEEE 802.11a provides 12 orthogonal channels in the 5 GHz spectrum [23].

The main goal of *multi-channel single-interface networks* is to distribute the load across the available orthogonal channels to permit simultaneous communication flows, while minimizing interference and channel congestion problems. Throughput increases and delay reduces with the parallelism of transmissions on distinct channels. However, the aforementioned advantages cannot be fully realized without a multi-channel protocol to negotiate first, how nodes agree on the channel for transmitting data, secondly, how nodes resolve potential contention for a channel [24].

However, note that when a node is listening on a particular channel, it cannot hear communication taking place on a different channel. An interface can only listen or transmit on one channel at a time. For this reason, the operation on multiple channels has to be carefully designed to avoid the *deafness problem* [25]: the transmitter must know if the receiver is tuned to the same channel at the same time to succeed communication. Moreover, a new type of hidden terminal problem can occur called *multi-channel hidden terminal problem* [26]. Another important issue is the *broadcast problem* [27]. The transmitter cannot be certain that all neighbors within the physical communication range received a broadcast packet because nodes can be tuned to different channels.

To fully take advantage of multiple orthogonal channels, nodes may have multiple interfaces so that different mesh routers may operate on different channels at the same time and simultaneously communicate with many neighbors. In other words, full-duplex operation is possible at each node. Equipping mesh routers with multiple interfaces then leads to efficient spectrum utilization and increases the actual bandwidth available to nodes in the network. Such networks are often designated as Multi-Channel Multi-Interface (MCMI) WMN [28]. It has been shown that network capacity would increase with the number of channels and interfaces [29].

1.2 Problem Statement

Although deafness, hidden terminal and broadcast problems are mitigated by equipping nodes with more than one interface, MCMI systems introduce new concerns.

Firstly, due to the limited number of available orthogonal wireless channels, more than one node in a given region could contend for the same channel at the same time, thereby resulting in interference and collisions. Hence, a key challenge in MCMI WMN consists in achieving an effective *channel and interface assignment* strategy (mapping) to decide "*when to switch interfaces*" as well as "*which channel to assign*" [28]. We must maintain the network connectivity while multiplexing efficiently the transmissions across orthogonal channels.

Secondly, routing protocols, together with routing metrics and path computation algorithms, rely on the network topology formation. Accordingly, if the assignment of channels and interfaces results in a multi-interface network with high density of connections (the number of radio links that exists in the network), more choices of paths will be available to the routing protocol. In turn, a routing protocol that achieves better path diversity could make the network more robust to failures and also improve capacity in terms of throughput and latency [30]. On the other hand, the larger the number of links, the greater the chance of nodes being in the same physical communication range of each other and then increase interference. Thus, there is a trade-off between connectivity, interference, and routing.

Last but not the least, wireless link quality varies over time in spite of the stationary characteristic of mesh routers. This variation is due mainly to environmental factors such as interference, multi-path effects and even weather conditions. Frequent variations on link conditions can influence how routing metrics evaluate the quality of the path, and thus lead to oscillations, which have an adverse effect on the overall network performance, including network throughput, resource usage, and reliability [31]. More precisely, the way the quality of a wireless link is evaluated has a highly dependency on the *network stability*, which actually minimizes the fluctuation of the path after being declared as an efficient one. Thusly, the accurate measurement of link quality is indispensable to improve the performance of MCMI WMN.

While there has been significant work on MCMI WMN [20, 28], realizing the full potential of multi-interface mesh networks has remaining a challenging problem. As explained above, to take advantage of the increased capacity in MCMI WMN, a number of issues has to be handled properly. In general, these issues include channel and interface assignment, connectivity, topology formation, broadcast, interference, throughput, latency, fault tolerance, stability, and routing [20]. Some of the issues have mutually beneficial relationships such as "*an optimal channel and interface assignment strategy can minimize interference and increase throughput*" or "*connected topologies offer better tolerance fault*". On the contrary, there also exist trade-off relationships such as "*assign interfaces to different channels can reduce interference but results in less connectivity and worse tolerance fault support*" or "*high level of fairness can sacrifice the overall throughput*".

All things considered, the design of a MCMR WMN capturing together all above mentioned issues is a challenging research problem.

1.3 Goals and Contributions of this Thesis

The first goal of this thesis is to present a novel classification and formal evaluation of different *channel and interface assignment* strategies addressing three main connectivity issues: topology formation, density of connections, and neighbor discovery. We classify the existing work that addresses the channel and interface assignment problem into one of the proposed strategies. We compare the different strategies based on a probabilistic analysis corroborated by simulations. In the meantime, advantages and limitations of each strategy are discussed regarding the issues such as interference, routing, load balancing, and stability. The results of this comparison and performance evaluation provide insight into the state-of-art approaches proposed in the channel and interface assignment research area, as well as guidelines for network designers to select the most suitable solution to guarantee connectivity with a certain probability for a given network density and the number of channels and interfaces.

The second goal of this thesis is to propose broadcast algorithms able to handle any of the multi-channel multi-interface assignment strategies mentioned above. As some high layer protocols rely on the broadcast support at the MAC layer (e.g., routing protocols), the development of such algorithms is essential for the proper functioning of the network as a whole. If the interfaces of the neighboring mesh routers are tuned to different channels, a single broadcast transmission on one specific channel cannot reach all the neighboring mesh routers simultaneously. Transmit a copy of the broadcast packet on each channel is inefficient because it incurs higher overhead. Therefore, we have designed algorithms that guarantee a broadcast packet to be delivered with a minimum probability to all neighbors. Simulation results show that the proposed broadcast algorithms efficiently limit the overhead.

The third goal of this thesis concern the evaluation of network capacity associated to any channel and interface assignment strategy. We define three Mixed Integer Linear Programming (MILP) formulations to evaluate network capacity in MCMI WMN. Formulations model the routing and bandwidth sharing constraints in presence of interference. We consider the objective function that maximizes throughput while maintaining fairness. These formulations permit to extract what we could obtain with an optimal centralized assignment, constituting an upper bound. First, we present numerical results that demonstrate that MCMI WMNs can achieve near optimal performance with static interfaces (regardless the number of channel switching). Next, we present extensive simulation results that take into account aspects neglected in the numerical evaluation (realistic MAC layer, routing, traffic load). We investigated the impact of the channel and interface assignment strategy onto IEEE 802.11s mesh networking. We highlight the drawbacks that limit the performance of the standard (e.g., broadcast storm, queue contention), while at the same time stating solutions capable to overcome these problems.

The fourth and last goal of this thesis is the development of a novel cross-layer routing solution for multi-interface networks. To achieve this objective, we propose a link-quality aware metric to estimate the residual bandwidth of a link. Among other measures, the metric consider those obtained from the physical layer to make them available to the network layer. In this way, inter-flow interference and traffic-load are dynamically incorporated into the metric. Next, the metric is incorporated into a new on-demand path selection protocol that operates over the link layer protocol. To reduce intra-flow interference, our path selection protocol consider the *channel diversity*: it gives higher weights to paths with consecutive links using the same channel. Finally, we evaluate the performance of our cross-layer routing solution via extensive simulations. We find that in multi-channel multi-interface wireless mesh networks, our solution significantly outperforms previously-proposed routing metrics and path selection protocols.

This thesis is structured in seven chapters divided into two parts: the state of the art and contributions. The state of the art part presents background on Multi-Channel Multi-Interface Wireless Mesh Networks. The contribution part contains our solutions to design and optimize MCMI WMN. In Chapter 3, we present the comparison and performance evaluation of channel and interface strategies, addressing the connectivity issues [32]. In Chapter 4, we describe and evaluate our broadcast algorithms [33, 34]. Chapter 5 presents three models to evaluate network capacity [35, 36]. In Chapter 6, we describe and evaluate our routing protocol. Finally, Chapter 7 concludes this thesis and present our view on future research directions.

Part I

State of the Art

Chapter 2

Multi-Channel Multi-Interface Wireless Mesh Networks

Contents

2.1 Wireless Mesh Networks

IEEE 802.11s [19] is an IEEE 802.11 (Wi-Fi) amendment for WLAN mesh networks. In 2004, the IEEE 802.11s Task Group (TG) was created to meet the growing demand for a mesh network standard. At the beginning, diverse industrial organizations recommended practices to resolve different issues involved in mesh networking [37, 38]. In 2006, after the consolidation and merge of these recommendations, the first draft was released. The draft has undergone several changes until its final approval as an IEEE 802.11 amendment in September 2011.

IEEE 802.11s describes protocols to support unicast, multicast and broadcast packet delivery in- and outside of the Mesh Basic Service Set (MBSS) [39, 40], referred to a Wireless Mesh Network (WMN) in the following.

2.1.1 Architecture

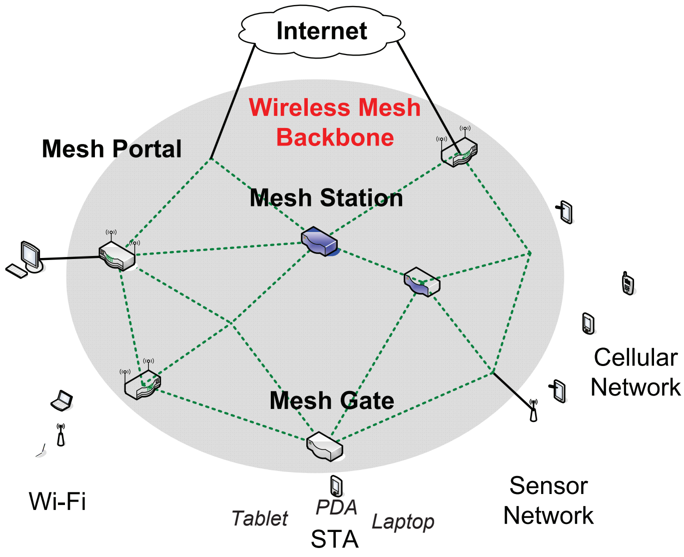
The architecture of a WMN consists of a backbone of autonomous nodes that provides large coverage, connectivity and robustness to the network. Figure 2.1 gives an example of WMN, where dashed and solid lines indicate wireless and wired links, respectively. Nodes fall into one of the following categories:

- **Station (STA)** is the basic entity in an 802.11 network. It corresponds to a node that requests services but does not forward frames, nor participates in path discovery mechanisms. STAs can significantly differ in terms of degree of mobility and energy autonomy. However, STAs are usually mobile and battery constrained. In addition, STAs are usually equipped with a single-interface. Laptops, cell phones, smartphones, and tablets are examples of STAs.

- **Mesh Station (Mesh STA)** is a quality-of-service STA that implements *mesh facilities* such as topology construction, path selection and data forwarding. A Mesh STA can establish *mesh peering* with multiple neighbor Mesh STAs, but it does not offer support to STAs.

- **Mesh Gate** is an Mesh STA with Access Point (AP) functionalities. Consequently, mesh Gates can support non-mesh wireless stations (STA). A mesh network may contain zero or more Mesh Gates. Note that a STA must first associate with a Mesh Gate before accessing the mesh network, as illustrated in Figure 2.1. STAs do not have awareness of the mechanisms working within the mesh network (e.g. discovery and routing procedures) so that each Mesh Gate shall then act as a proxy for its associated STAs [19]. In other words, the mesh network must be completely transparent from the STA point of view.

- **Mesh Portal** is a Mesh STA integrated with gateway functions to interoperate with external networks (non-IEEE-802.11 LAN) such as the Internet. Every 802.11s mesh network may have zero or more Mesh Portals. It is up to each Mesh STA to choose which Mesh Portal to use to get access to the external network. As an Mesh STA, Mesh Portal does not have access point functionalities and then does not offer support to STAs.


Internet
Wireless Mesh
Backbone
Mesh Portal
Mesh Station
Mesh Gate
Cellular
Network
Sensor
Network
Wi-Fi
Tablet
PDA
Laptop
STA

range called *Contention Window* (CW). A backoff counter is decreased by one unit for every time slot the channel is sensed to be idle, and frozen if the channel is sensed to be busy. When the backoff counter reaches zero, the Mesh STA can start the transmission. Upon the successful reception of a frame, the receiver waits for a Short Inter-Frame Space (SIFS) and then sends an ACKnowledgment frame (ACK). If the sending Mesh STA receives the ACK, the transmission is considered successful. Otherwise, the sending Mesh STA assumes that a collision occurs. Then, it doubles the value of its current CW, randomly resets its backoff counter, and retransmits the frame if the backoff counter reaches zero. If another collision occurs, CW size is doubled again until a maximum size (CW_{max}). Once a transmission is successfully transmitted, the CW range is reduced to its minimum (CW_{min}) value for the next transmission.

Using the above scheme to coordinate access to the wireless medium, collisions are already avoided in most cases. However, the *hidden terminal problem* can still happen [43]. More specifically, this problem occurs when two nodes A and B are too far from each other to sense their transmission and thus both detect the medium as idle. Then, if both nodes attempt to send to a third node C located in between, their transmissions will interfere and packets will be lost.

The IEEE 802.11 standard suggests the use of the Request-To-Send/Clear-To-Send (RTS/CTS) handshake to solve this problem. Following the example above, assume that node A has data to send to node C. Then, node A initiates the process by sending the RTS control frame. Then, the destination node C replies with the CTS control frame. When node A receives the CTS, it sends data. After successful reception, node C replies with an ACK. Note that by setting the duration fields of both control frames, the two nodes A and C set up a NAV that prohibits node overhearing the RTS and/or CTS frames (e.g., node B receives the CTS in our example) to send for a time interval that is used to transmit the data frame and return its acknowledgement. However, the RTS/CTS mechanism is shown to be ineffective in eliminating the hidden terminal problem in some scenarios. For example, the RTS/CTS mechanism does not take into account that nodes out of the transmission range of both the transmitter and the receiver may still interfere with the receiver [44].

To enhance the Quality of Service (QoS) support, IEEE 802.11s introduces two schemes: Enhanced Distributed Channel Access (EDCA) and MCF Controlled Channel Access (MCCA) [19].

EDCA is an improved variant of DCF that differentiates four traffic categories (or access categories): voice, video, best effort and background. Compared with DCF that uses DIFS as the common Inter-Frame Space (IFS) for a station access the channel, EDCA uses different Arbitration Inter-Frame Space (AIFS) for each traffic category to achieve medium access differentiation. In this case, lower priorities use a larger AIFS. Additionally, each traffic category contends channel access with different CW_{min} and CW_{max} settings. Besides the prioritization scheme, EDCA also introduces the concept of *transmission opportunity limit* (TXOP Limit). In contrast to a common restriction for one packet as in DCF, EDCA allows a Mesh STA to transmit multiple frames whose total transmission duration does not exceed the TXOP Limit [39]. In this case, the TXOP Limit is granted according to the traffic category [45].

MCCA is a distributed medium access method that allows Mesh STAs to access the wireless medium at selected times with lower contention [19]. More specifically, mesh STAs can reserve TXOPs in the future called *MCCA opportunities* (MCCAOPs). Each MCCAOP has a precise start time and duration measured in multiples of 32 μs slots. MCCA defines a set of management frames to allow Mesh STAs to negotiate the reservation for transmissions. For example, the Mesh STA sends an MCCA Setup Request frame to the intended receiver to initiate a reservation. Once established a reservation, the transmitter and the receivers of this frame advertise their neighbors via an MCCA advertisement. In this way, at the beginning of an MCCA reservation, Mesh STAs other than the MCCAOP owner refrain from channel access. The owner of the MCCAOP uses EDCA to access the medium.

According to the IEEE 802.11s standard, EDCA is a mandatory scheme, while MCCA is optional. In this thesis, we do not consider different traffic categories. In particular, we focus on a single traffic category that uses the same inter-frame space, CW_{min}, CW_{max} and TXOP Limit parameters as DCF. Besides, we do not consider MCCA.

2.1.3 Characteristics

WMNs exhibit some unique characteristics that differentiate them from other wireless and wired networks. These characteristics are explained as follows:

Lack of Mobility and Energy Limitations

Mesh routers (Mesh STAs, Mesh Gates and Mesh Portals) are usually stationary and do not have energy constraints. They may profit from resources such as multiples interfaces, memory, storage, computation power, and so on.

Table 2.1 shows an outline of the specific characteristics of each element according to the energy constraints, mobility, number of interfaces, support to STAs and gateway functions involved.

Type of Node	Energy Constraints	Degree of Mobility	Number of Interfaces	Support to STAs	Gateway Functions
STA	Yes	High/Low	Single	-	No
Mesh STA	No	Low	Multiple	No	No
Mesh Gate	No	Low	Multiple	Yes	No
Mesh Portal	No	Low	Multiple	No	Yes

Table 2.1: Characteristics of IEEE 802.11s nodes.

Multi-hop Communication

As shown in Figure 2.1, mesh routers establish and maintain *wireless mesh connectivity* among themselves to form a multi-hop *Wireless Mesh Backbone* able to extend the coverage range of current wireless networks. Similarly to nodes in Mobile Ad hoc NETworks (MANETs) [46], mesh routers forward across multiple hops the traffic generated by other nodes (STAs and other mesh routers) that may not be within direct wireless transmission range of their destinations. However, unlike MANETs, node mobility in the multi-hop mesh backbone is not frequent. In this thesis, we focus on static WMN.

The data is forwarded from one mesh router to another until it reaches the destination. Thus, mesh routers tend to connect with each other through shorter link distances rather than long direct connections. As a result, the WMN can cover the same area with less transmission power than a traditional wireless router and thus experience less interference and achieve a higher throughput [17]. This feature gives the impression that all Mesh STAs are directly connected at the MAC layer [19], while in fact they are not within the transmission range of each other.

Self-configuration, Self-organization

The features of the multi-hop wireless mesh network give rise to self-configuration and self-organization properties [47]. The WMN can be incrementally deployed, one node at a time, without any special administrative intervention. This characteristic makes WMN attractive for novice users who can quickly join an existing mesh network by setting up their own mesh router (e.g, desktops, laptops). The mesh routing protocols allow mesh routers to learn about their neighbors and dynamically route data among themselves as the nodes enter and leave the network [48]. As a result, WMNs have low upfront investment requirement, especially when compared with IEEE 802.11-based AP.

Self-healing

The self-healing feature allows the WMN to continue operating even if one or

Type of Network	Network Entities	Energy Constr.	Transmision	Traffic Pattern
WMN Backbone	Static Routers	No	Multi-hop	Gateway Oriented
Cellular Network	Base Stations and Mobile Clients	Yes	One-hop	Base Station Oriented
Infrastructured WLAN	Access Points and Mobile Clients	Yes	One-hop	Access Point Oriented
Ad Hoc WLAN	Mobile Nodes	Yes	Multi-hop	Distributed
WSN	Static Nodes	Yes	Multi-hop	Gateway Oriented

Table 2.2: Comparison of Wireless Network Technologies.

more mesh routers fail (e.g., software or hardware failures, power outage) or a connection is interrupted (e.g., physical obstacle). Mesh routers are able to find alternative routes to their destinations because routers are "meshed" together and have multiple paths available in the multi-hop backbone. Indeed, the extent of self-healing capability depends on the number of available paths (i.e., degree of "meshing"). Adding more routers can increase reliability as more alternate paths become available. However, a large number of nodes sharing the wireless medium can result in increased contention and bottleneck. To provide sufficient self-healing capability and maintain network performance at an acceptable level is necessary to obtain equilibrium between the contention levels and the number of alternative paths.

Traffic pattern

Another peculiarity of WMN is the traffic pattern. In fact, users normally want to access the resources available on the Internet, which resides in the wired infrastructure (Mesh Portal). Thus, the traffic is primarily between the Mesh Portal and an end user. Consequently, a lot of traffic has to traverse the mesh backbone through long paths.

Mesh routers in close proximity to the Mesh Portal are more likely to become congested and suffering from quickly buffer overflow than mesh routers far from the Mesh Portal. Consequently, this traffic pattern may result in congestion in areas in close proximity to the Mesh Portals, leading to significant performance degradation in terms of the achievable throughput and the end-to-end delays.

Table 2.2 presents a comparison of different network technologies. In particular, we compare the WMN backbone with Cellular Networks, Infrastructured WLANs,

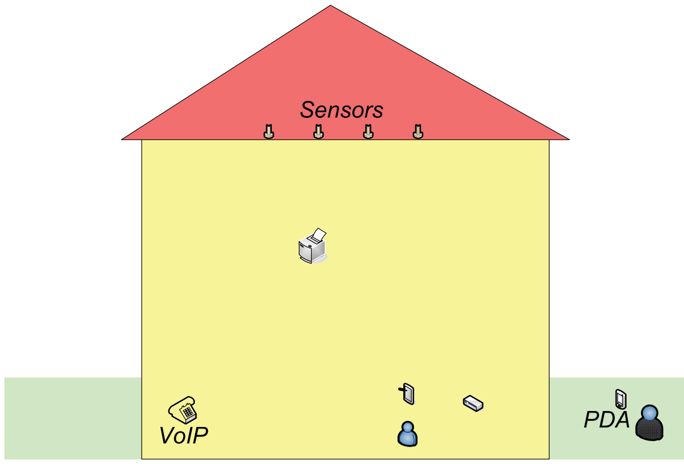


Figure 2.2: Scenario of Broadband Home Networking.

Ad Hoc WLANs and Wireless Sensor Networks (WSN). We consider only the parts of the networks involving wireless communications.

2.1.4 Application Scenarios

WMNs have the potential to provide support for a variety of applications that cannot be supported directly by other wireless networks such as cellular, sensor, Wi-Fi, and ad hoc networks. Research and development have been made by many commercial companies [49, 50, 51, 52], demonstrating the promising market of WMN for indoor and outdoor applications. At the same time, a large number of research centers and universities are undertaking the deployment of experimental testbeds [8, 53, 54, 55].

In this section, we focus on the most common applications that benefit from wireless mesh networks [17].

Broadband home networking

A single AP may not cover a larger area like a whole house due to its relatively short transmission range in indoor environments. Additionally, walls, doors, ceilings, and other structural works may reduce the transmission range in indoor placements. To extend wireless coverage, the traditional approach is to lay down cables and interconnect more APs. However, this is not a cost-effective solution.

WMN is a practical and effective solution to resolve the location of the APs in home networking [9]. Conventional APs must be replaced by wireless mesh routers with mesh connectivity established among them, thereby forming a backbone to extend the wireless coverage. Changing the locations of the mesh routers, or else by

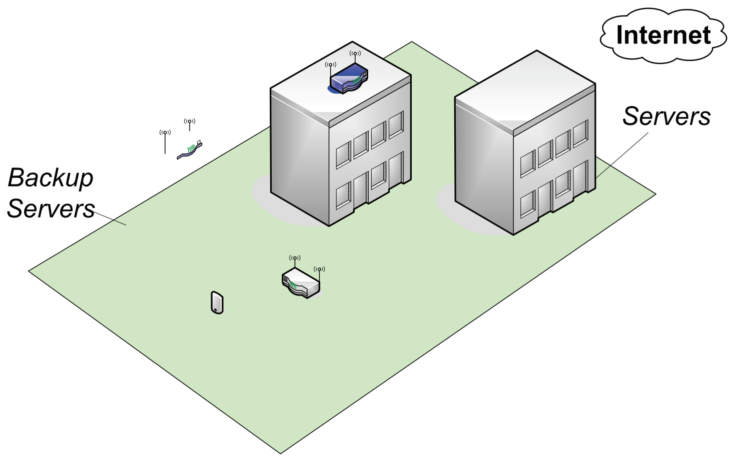


Figure 2.3: Scenario of Enterprise Networking.

adding new mesh routers in the house, can easily eliminate *dead zones* (i.e., zones without service coverage).

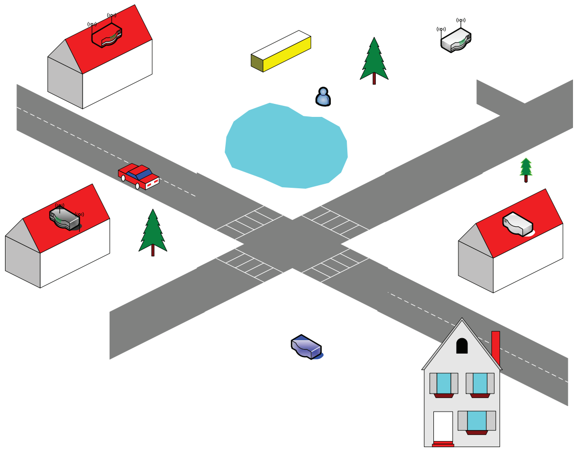
The usage scenario for broadband home network is not limited to Internet access. As illustrated in Figure 2.2, home media server (e.g., music, video/movies, images/photos), shared storage, monitoring systems (e.g., cameras, sensors), and gaming are other examples of services. In summary, WMN allows digital interaction anywhere, anytime from any device in the house.

Enterprise networking

As shown in Figure 2.3, enterprise buildings often consist of several floors and many offices that have to be interconnected. Although modern buildings are usually constructed with support for wired networks, when the enterprise grows and/or network technologies evolve, the existing cabling may become insufficient. Moreover, when it comes to bridge several enterprise buildings, implement cabling becomes very expensive.

WMNs mitigate the above disadvantages. In enterprise scenarios, the mesh network can be easily extended by installing additional mesh routers and upgraded by simply replacing them. For example, the network can greatly improve robustness utilizing mesh routers with multiple interfaces. This feature improves bandwidth to support internal services (e.g., servers backups, terminal services, security procedures, software maintenance).

The enterprise networking model can be applied to other service networking scenarios such as shopping malls, hotels, airports, convention centers, trade fairs, sports stadiums, etc [17].



Intelligent transportation networking

Recently, government agencies and public transportation companies are interested in practical networking solutions for intelligent transportation systems [56]. The main idea is to implement traffic management as well as to integrate public transportation systems such as buses, trains, ferries, etc. For instance, provide real-time travel information to passengers. In this regard, WMNs can provide flexible wireless networking solutions to implement the required information delivery system. With the use of WMNs, it is possible to address and alleviate transportation congestion problems, control pollution, and improve transportation safety and security [57].

In addition to the above applications, WMNs can also be applied for several other purposes: spontaneous networking (e.g., disaster, emergency), peer-to-peer communications, public safety (e.g., fire departments, police, first responders, and emergency services), health and medical systems, security surveillance systems, building automation networking, etc [18, 17].

2.2 Design Issues in Multi-Interface Networks

The IEEE 802.11 standard divides the wireless spectrum into different spectral bands, called *channels*, in the ranges of 2.4 GHz and 5 GHz [1]. For instance, IEEE 802.11b/g use the 2.4 GHz band [3, 4] and IEEE 802.11a uses the 5 GHz band [2]. As shown in Figure 2.5, channels have a center frequency of 5 MHz apart from each other and an overall channel bandwidth (or frequency occupation) of 22 MHz. The level of radio frequency energy that crosses between these channels determines *interference*. Thus, any pair of channels separated in frequency by 25 MHz (2.4 GHz) / 20 MHz (5 GHz) can be used simultaneously without mutual interference. These channels are said orthogonal or “non-overlapping”. IEEE 802.11b/g provides a triple of orthogonal channels (1, 6, 11), as shown in Figure 2.5, and the 5 GHz provides 12 orthogonal channels [23].

In fact, the availability of channels varies according to regulatory bodies worldwide. As depicted in Figure 2.5, the 2.4 GHz frequency band is broken down into 14 distinct channels: 11 channels for the North American domain [23], 13 channels for the European domain [58]. The last channel is designed specifically to Japanese regulations.

Each interface of a Mesh STA is associated with its own PHY and MAC layers. Various PHY layers technologies are available after subsequent amendments of the base version of the IEEE 802.11 standard. Most of these amendments propose new PHYs in order to increase the aggregate throughput of a IEEE 802.11 network, while

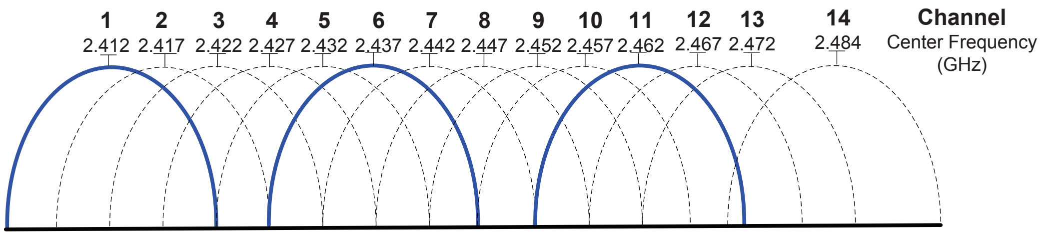


Figure 2.5: Orthogonal channel selection according to IEEE 802.11b/g.

Standard/ Amendment	Release (Year)	Data Rate (Mbit/s)	Frequency Band (GHz)	Orthogonal Channels
Original	1997	1, 2	2.4	3
802.11a	1999	6, 9, 12, 18, 24, 36, 48, 54	5	12
802.11b	1999	1, 2, 5.5, 11	2.4	3
802.11g	2003	6, 9, 12, 18, 24, 36, 48, 54	2.4	3

Table 2.3: Characteristics of IEEE 802.11 PHY layers.

preserving the MAC layer. Table 2.3 presents some amendments for PHY layers that have been standardized during the last years with different combinations of data rates, frequency bands, and number of orthogonal channels.

Since IEEE 802.11 supports multiple orthogonal channels, the above mentioned categories of mesh routers (Mesh STAs, Mesh Gates and Mesh Portals) can take advantage of multiple operating interfaces, unlike nodes of other wireless networks that suffer with high mobility (e.g., MANETs) or energy constraints (e.g., WSNs). The cost of multiple network interface cards is no longer a prohibitive factor with the proliferation of wireless networks.

Equipping nodes with multiple interfaces is known to be beneficial to improve the capacity of WMN [59]. First, it enables full-duplex operation at each node. Second, neighboring links assigned to different orthogonal channels can carry traffic free of interference and reduce link-layer delay. Third, multiple interfaces increase the number of paths available to nodes, which provide abundant choices for recovering from faults. All things considered, Multi-Channel Multi-Interface (MCMI) mesh network can improve the network capacity.

From a general standpoint, a MCMI WMN is subject to the following requirements:

1. The number of available orthogonal channels is limited by the use of a specific standard (e.g., IEEE 802.11a/b/g) and government regulations [23, 58].

2. The interfaces may have different transmission ranges and data rates (see Table 2.3).

3. A node equipped with I interfaces can only communicate on I orthogonal channels at a time, which can cause the *deafness problem* [25]. Deafness occurs when the transmitter fails to communicate to its intended receiver because the receiver's interfaces are tuned to different channels.

4. The interfaces at each node are capable of switching between channel, with a switching cost overhead.

5. The number of interfaces at each node is generally less than that of the available channels.

In a MCMI WMN, two neighbor nodes wishing to communicate establish a wireless link between them by tuning at least one of their interfaces to the same channel. Each mesh router has to be able to handle more than one channel and implement specific mechanisms to coordinate between channels to efficiently use the available channels.

In order to bring up some concerns that can impact the performance of MCMI WMN, we discuss here the most relevant challenges and design trade-offs that have to be faced.

2.2.1 Modeling Interference

Due to the shared nature of the wireless medium, a wireless link in a mesh network does not have a dedicated bandwidth since nodes in the vicinity may also compete for the same bandwidth and hence interfere with the transmission on the other links. The level of interference depends on factors such as the network topology, traffic on neighboring links, etc.

To address the interference issue, we must use a model to describe the interference impact on the success of a given transmission. There are three main interference models that are widely adopted in the literature: Protocol Model, Logical Model and Physical Model [20, 21].

These models can be influenced by the concept of three types of radio ranges: *transmission range*, *carrier sensing range* and *interference range* [44, 60]. The *transmission range* (R_{tx}) corresponds to the range (with respect to the transmitter node) within which a radio frequency signal can be successfully received if there is no interference from other nodes. In Figure 2.6, the transmission ranges of nodes A and B are represented as dashed-line circles. Note that node B is within the transmission range of node A, and vice versa. The *carrier sensing range* (R_{cs}) is the range (with respect to the transmitter node) within which other nodes are able to detect the signal, even though correct packet reception may not be available. In

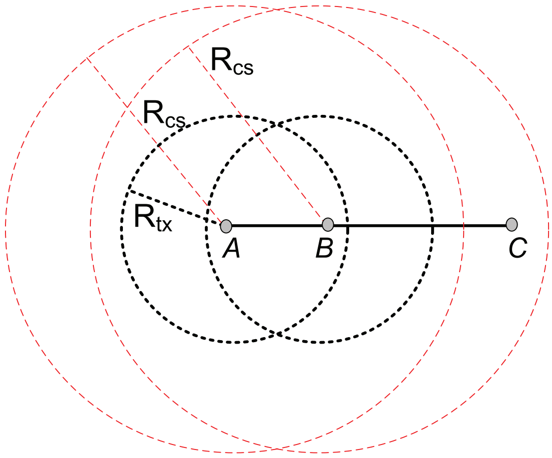


Figure 2.6: Example of Transmission and Carrier Sensing Ranges.

Figure 2.6, the carrier sensing ranges are depicted as dashed-line circles. Node C is within the carrier sensing range of node B, but not within the carrier sensing range of node A. The *interference range* (R_i) is the range within which nodes in receive mode will be "interfered with" by a transmiter node, and thus suffer a loss. It is generally assumed that $R_{tx} < R_i < R_{cs}$ [61]. Based on these ranges definitions, the hidden terminals refer to nodes within the interference range of the intended receiver node and out of the carrier sensing range of the transmiter node [44]. For example, assume that node A is transmitting to node B in Figure 2.6. If a hidden node C wants to transmit at the same time, it senses the medium and finds it free because it is not able to hear A's transmission (i.e., node C is out of the carrier sensing range of node A). Therefore, collision will happen at the receiver node B.

The three main interference models are described as follows:

Protocol Model: determines that a transmission from a node A to a node B is successful if (i) B is in the transmission range of A and (ii) any other node within the interference range R_i is neither transmitting nor receiving in the channel used by A and B. This model is designed to guarantee that the links do not interfere with each other through the particular channels assigned for each one. The Protocol Model assumes interference to be an "all-or-nothing phenomenon" [62]. Besides, note that this model forms the interference that occurs *before* the transmission [63].

Logical Model: takes into account the interference in the MAC layer. It is also known as the *Channel Contention Interference* because it stems from the medium access protocol (e.g., CSMA/CA). A node waits until the channel is free before starting a transmission. Hence, this model includes the deferred access time to the wireless medium as the shared channel may be occuped by the transmission of other nodes that are using the same channel within the carrier sensing range of the node, which represents interference [64]. Likewise the Protocol Model, the Logical Model displays the interference that occurs *before* the transmission [63, 64].

Physical Model: captures the interference experienced by the wireless links in the physical layer. This physical interference represents the superposition of waves that changes the original signal and causes bit alterations. It determines that a transmission from a node A to a node B is successful if the signal strength at the receiver B is above a certain threshold and this depends on the desired transmission characteristics such as the channel data rate. Thus, the Physical Model displays the interference that occurs *during* the transmission [63, 64].

There are several ways of measuring interference based on the physical layer [65]. The main measures are described as follows:

- **Received Signal Strength Indication (RSSI)** captures the signal strength observed on the receiver antenna during packet reception. Three main considerations must be taken into account when measuring RSSI. First, the RSSI calculation is solely based on correctly received packets, which implies that RSSI will not record packets that failed because of interference [20]. Second, the RSSI is not the average of the signal strength measured through the reception of the whole packet. In fact, the RSSI value represents the received signal strength captured only during the reception of the preamble and header of the Physical Layer Convergence Protocol (PLCP) [1]. The PLCP allows the receiver to synchronize with the transmitter for correct reception. Thus, in cases where the interference affects only the data portion of the frame, this effect of interference will not be captured in the RSSI measurement [65]. Finally, PLCP is sent at the basic (lowest) transmission rate (e.g., 6 Mbit/s with 802.11a/g) [1, 2, 4]. In conclusion, RSSI is not a good candidate to represent interference precisely.

- **Bit Error Rate (BER)** is the ratio between the number of bits with errors and the total number of bits that have been received over a specific time period. While the concept is simple, measuring BER is a non-trivial task in real systems. BER is a finer grain indication, which means that repeated computations of this measure are required to measure interference [65]. Consequently, significant overhead is introduced. Besides, this measure is of little value when the network conditions are changing quickly over a period of time [20] .

- **Frame Error Rate (FER)** is the ratio between the number of frames with errors and the total number of frames that have been received over a specific time period. If the bit errors are independent identically distributed events, FER is calculated as follows:

$$FER = 1 - (1 - BER)^S \tag{2.1}$$

where $(1 - BER)$ is the probability that each bit is correct and S the frame length in bits. Thus, FER is a coarser grained measure than BER. However, PER calculation still requires the processing of an amount of previously known data.

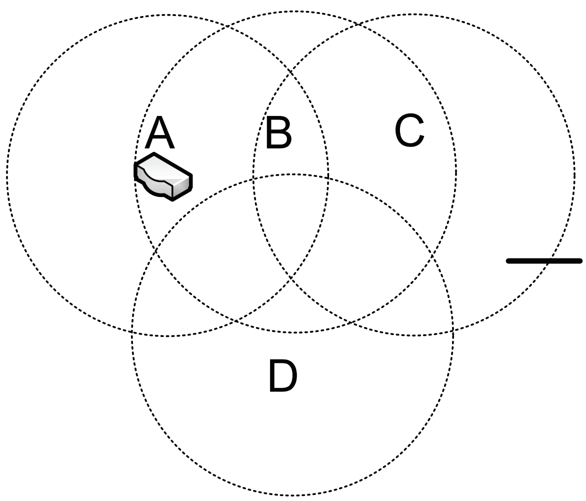
- **Signal to Interference plus Noise Ratio (SINR)** represents the extent to which the power of the received signal exceeds the sum of noise plus interference at the receiver. Recent studies have considered the SINR the most appropriate metric for quantify the quality of a link [63, 66]. However, having an accurate measure of SINR is extremely hard, if not impossible, in practice because commercial wireless cards do not usually record this measure during the reception of a packet [20, 67, 65]. In general, the SINR is estimated on the basis of RSSI [67] and therefore it presents the similar problems as RSSI described above.

Table 2.4 presents an overall summary of physical measurements for MCMI WMN. Advantages and limitations of each measure are presented. The recent related work suggests that each measure can reveal interesting behavioral aspects of a link, but generally none of the these measures on their own accurately characterize the quality of a link [65].

From another perspective, we also distinguish between two types of interference: intra-flow and inter-flow [63, 68]. Let us first consider the single-channel single-interface case depicted in Figure 2.7. We represent the transmission range of nodes by dashed-line circles. Intra-flow interference arises when nearby nodes transmitting packets belonging to the same flow interfere. Nodes $A-B-C$ experience intra-flow interference because they are forwarding data from the same flow from source A until the gateway. In consequence, if every node operates its single interface on the same channel, a forwarding node will interfere with the two subsequent nodes along any multi-hop path, which drastically reduce the end-to-end throughput. However, depending upon the transmission power and the carrier sensing range, the intra-flow interference can also be between links two or more hops away. Inter-flow interference is caused by the interference between nearby flows sharing the same channel and competing for the medium access. For instance, node D interferes on the transmission of node B. Inter-flow interference can result in bandwidth starvation for some nodes since these nodes may always experience busy channels.

Measure	Definition	Assets & Limitations
RSSI	Signal strength observed on the receiver's antenna during packet reception.	⊕ recorded by commercial wireless cards ⊖ bad indicator of link interference ⊖ based solely on packets correctly received ⊖ does not consider the signal strength measured through the reception of the whole packet ⊖ calculated at the basic transmission rate
BER	Ratio of the number of bits with errors to the total number of received bits over a given time period.	⊕ fine-grained measure ⊖ introduces significant overhead ⊖ requires the processing of a large amount of previously known data
FER	Ratio of the number of frames with errors to the total number of received frames over a given time period.	⊕ coarse-grained measure ⊕ simpler to implement than BER ⊖ requires the processing of an amount of previously known data ⊖ takes a long time to capture interference
SINR	Extent to which the power of the received signal exceeds the sum of noise plus interference at the receiver.	⊕ appropriate to quantify the link quality ⊖ not recorded by commercial wireless cards ⊖ acquires the deficiencies of RSSI

Table 2.4: Characteristics of Different Physical Measures.



2.2.2 Throughput and Latency Optimization

Throughput and latency are two important measures for the MCMI WMN performance. They have a strong relationship and thus are generally addressed together by multi-interface approaches [28]. Much work acknowledges that reducing interference is the most effective method to obtain optimal throughput and latency [68, 69]. Besides, it is recommended to make this method adaptive to the traffic load [63, 70].

In this context, two challenging issues have to be addressed [20, 28]: (1) assignment of channels to interfaces and (2) routing. On one hand, different assignments may alter the network topology (e.g. some links are created while other may disappear). Since the network topology is the basic factor for making routing decisions, we can conclude that routing is dependent on assignment. On the other hand, routing can change the traffic load distribution in the network, which is one of the primary factors considered by assignment approaches to reduce interference [71]. So in this case, assignment is also dependent of routing. To handle such a relationship is not trivial.

2.3 Channel and Interface Assignment

Although multiple interfaces have the potential to significantly improve the performance of WMN, the decision of which channel to assign to each interface at every mesh router is still a significant challenge [72]. How to build this efficient mapping between channels and interfaces is known as the *Channel and Interface Assignment* problem. Note that this problem is non-trivial in the typical case where the number of interfaces per node is generally less than that of available channels. A key factor is achieving an optimal strategy able to minimize interference while maintaining the connectivity of the network.

2.3.1 Connectivity vs. Interference

We use a simple mesh scenario to discuss these two key design issues: connectivity versus interference. Figure 2.8(a) shows five mesh routers and their respective transmission ranges for a particular propagation model. Note that an identical transmission range is assumed for all nodes. We consider that each mesh router is equipped with two interfaces. Let c_1, c_2, c_3, c_4 be the four available orthogonal channels.

We distinguish between two main concepts:

- **Topology induced by the *Unit Disk Graph* (UDG)** is used to model wireless networks under the Protocol Model described above. This topology is an undirected graph[1] $G = (V, E)$, where V represents the set of nodes in the network and E the set of edges corresponding to two nodes able to directly communicate. For example, Figure 2.8(b) shows the Topology induced by the UDG of the mesh scenario illustrated in Figure 2.8(a). Accordingly, the Topology induced by the UDG is independent of the Channel and Interface Assignment.

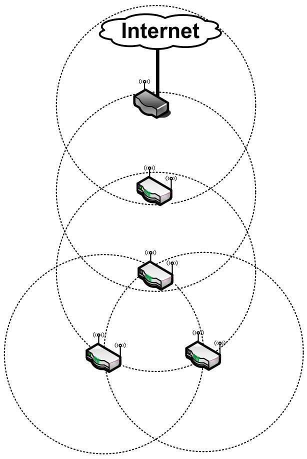
- **Network Topology** models the fact that any two nodes actually share a common channel as their communication link. Network Topology is an undirected graph $T = (V, E)$, where V represents the set of nodes and E the set of actual communication links in the network. For instance, $\forall v, u \in V$, the link $(v, u) \in E$ if the following conditions are satisfied:
  (i) v and u are within the communication range of each other;
  (ii) one of the v'interfaces and one of the u'interfaces share a common channel.
  Thus, the Network Topology is only known after the Channel and Interface Assignment. Figures 2.8(c), 2.8(d) and 2.8(e) show three examples of Network Topology for the scenario of Figure 2.8(a). The channel assigned to each interface is shown inside the square brackets nearby the nodes.

According to the above two definitions, two types of discrepancies can occur between the Topology induced by the UDG and the Network Topology [28]. First, a link between two nodes in the Topology induced by the UDG may not be present in the Network Topology if the interfaces on these two nodes are not assigned to a common channel, such as link CD in Figure 2.8(d). Second, multiple edges (i.e., links) exist between two nodes in the Network Topology if multiple common channels are assigned to the interfaces on these two nodes, as illustrated in Figure 2.8(c).

These two models are especially important for the Channel and Interface Assignment approaches. Topology induced by the UDG, because it is usually used as the base to perform the assignment, since it gives the distance relationships between nodes in the wireless network. The Network Topology because it is usually used to specify the connectivity requirements for assignment [28]. Therefore, Channel and Interface Assignment approaches must be aware of which assignment decisions can change the Network Topology, which is a key difference between the single-channel and multi-channel networks.

For example, Figure 2.8(c) shows a Network Topology in which all nodes have their interfaces assigned to an identical set of channels ($[c_1, c_2]$). In this Network Topology, the number of radio links is maximized (ten radio links in total). However, such assignment may result in increased intra-flow and inter-flow interference

[1] An undirected graph is one in which edges have no orientation.


Internet

because the load is not efficiently distribute among all available orthogonal channels. In this example, only 50% of available channels are used. On the contrary, Channel and Interface Assignment with lower interference may lead to network partitions. As shown in Figure 2.8(d), the partition resulted in two subnetworks ABC and DE. Similarly, the assignment may also impact upper layers, especially routing protocols. As we can see from Figure 2.8(e), the Network Topology avoids intra-flow and inter-flow interference, but it limits to one the number of available paths between any two node.

Hence, there is an inherent trade-off between connectivity and interference that directly affects throughput and latency. The balance between connectivity preservation and interference reduction makes the Channel and Interface Assignment an optimization problem in which "some interference measure" defined over the whole network according to a interference model is optimized with the constraint that "some notion of connectivity" is preserved [73].

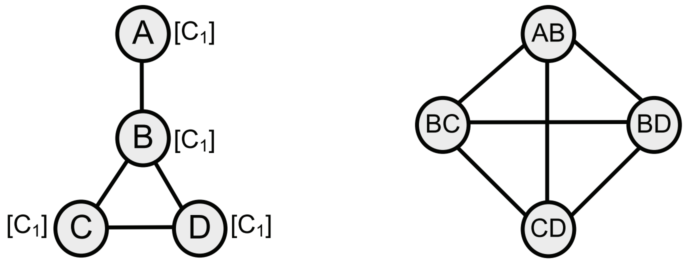
2.3.2 Conflict graph

The *conflict graph* concept is a promising approach to incorporate connectivity and interference into the Channel and Interface Assignment problem. This graph theoretic model was originally proposed by Jain *et. al* [62] in 2003. Since then, it has been widely used in the literature because it offers a flexible and fine-grained approach to model wireless interference under various conditions (e.g., multiple nodes, multiple channels, multiple interfaces, etc.) [73].

The conflict graph $G_C(V_C, E_C)$ is derived from the Network Topology $T = (V, E)$ and models the fact that two links in E interfere or not with each other. Each vertex V_C represents a link in the Network Topology. If two links in the Network Topology interfere, an edge connecting the two corresponding vertices in V_C is included in E_C to represent their conflict.

Let us consider the Network Topology depicted in Figure 2.9(a), in which nodes have a single-interface tuned to the same channel ($[C_1]$). In particular, links are pairwise interfering. The Network Topology in Figure 2.9(a) has four links, so four corresponding vertices exist in the corresponding conflict graph illustrated in Figure 2.9(b). For instance, since links AB and BC interfere with each other, there is one edge connecting the two corresponding vertices in the conflict graph.

The conflict graph can be extended to multi-channel multi-interface networks [74]. In this case, the resulting conflict-graph is the union of conflict-graphs for each individual channel. Both Protocol and Physical models can be expressed as a conflict graph [62, 75].



(a) Network Topology (b) Conflict Graph G_C

Figure 2.9: A Network Topology and its Conflict-graph

2.3.3 Stability

Guaranteeing a stable network topology after a Channel and Interface Assignment is an important issue. In general, the Channel and Interface Assignment operation can cause two phenomena that undermine network stability [25]: ripple effect and channel oscillation.

We use the following single-interface scenario described by Si *et al.* [28] to exemplify the ripple effect phenomenon. Assume node X originally at channel c_1 wants to communicate with node Y at channel c_2. Thus, X switches to channel c_2. At the same time, assume that a third node Z is currently communicating with X using channel c_1. This node Z has to switch to channel c_2 to mantain communication. Note that such channel change may continue to propagate along the network, if another node W is currently communicating with Z using channel c_1. Another problem associated with the ripple effect is that, say in the above single-interface example, when X switches to channel c_2, some packets may be lost in the communication between X and Z on channel c_1, before Z switches to channel c_2.

The channel oscillation phenomenon occurs when the Channel and Interface Assignment does not converge and changes back and forth among several choices. This phenomenon usually happens when the assignment is based on a dynamic metric such as interference and traffic load [71]. For instance, if two nodes discover that a channel c_1 is under-utilized according to such a dynamic metric, they may simultaneously switch to this channel and both begin transmission on it. As the result, the two nodes now contend for channel c_1, and then switch back because this channel is now overloaded, as indicated by the dynamic metric. These oscillations might indefinitely continue and lead to a non-convergent behavior that severely impairs the network performance.

2.3.4 Review of Channel and Interface Assignment Approaches

In this section, we provide a description of Channel and Interface Assignment approaches. Specifically, we study how to efficiently perform *rendezvous* between nodes when a set of channels and interfaces are available [76].

This area of research started around 2003 with the development of multi-channel single-interface protocols [25, 77]. Although many of these protocols can be adapted to multiple interfaces scenarios, new solutions have been specifically designed for MCMI WMN [28].

Our goal here is to present an overview of different coordination mechanisms. Besides, this section aims to examine the effects of a number of parameters in the performance of multi-interface approaches, such as the number of interfaces, synchronization, channel switching cost, interference, traffic pattern, etc. The Channel and Interface Assignment approaches have followed three main trends: multi-channel single-interface, distributed multi-channel multi-interface and centralized multi-channel multi-interface, as outlined in Sections 2.3.4.1, 2.3.4.2 and 2.3.4.3, respectively.

2.3.4.1 Multi-Channel Single-Interface Approaches

Here, we present some protocols originally proposed for multi-channel single-interface environments that can be extended to use multiple interfaces as well.

Slotted Seeded Channel Hopping (SSCH)

SSCH [78] is a *hopping sequence* distributed protocol. The main idea is for each interface to hop/switch across multiple channels according to its own hopping pattern. In fact, each interface time-multiplexes multiple sequences (e.g., set to 4 in paper simulations) uniquely determined by the seed of a pseudo-random generator. The objective is to match its own hopping sequence with its receiver's current hopping sequence to allow rendezvous. In other words, each interface dynamically adapts its hopping sequence as traffic demand changes. Nodes learn about each other's hopping sequence by periodically broadcasting their sequence. When a node wants to transmit data to another node, it waits until its interface is tuned on the same channel as the receiver's interface. Then, SSCH does not require channel negotiations before data transmission. SSCH preserves the 802.11 MAC protocol, but still requires link-layer techniques for time synchronization. Besides, it also requires fast channel switching capability. SSCH may also suffer from the deafness and the multi-channel hidden terminal problems because interfaces switch between different channels over time, as well as because the transmitter nodes have to pause the switching for some time in order to have rendezvous on the receiver's channel.

McMAC [79] follows an approach similar to SSCH. The main difference is that interfaces in McMAC hops over all available channels in a pseudo-random fashion using its own MAC address as the seed. Besides, an interface never changes its hopping sequence once it is established. It avoids the constant realignment of hopping sequences typical of SSCH [77]. In addition, McMAC allows senders to temporarily deviate from its sequence to send data on the receiver's channel.

Multi-channel MAC (MMAC)

MMAC [26] is a *split phase* multi-channel protocol in which time is divided into an alternate cycle of control and data exchange phases. During the control phase, all nodes listen to a common control channel to negotiate the channel to be used during the data exchange phase. Each node maintains a classification of channels called Preferable Channel List (PCL), that indicates which channel is preferable for communication according to the traffic scheduled for each channel. In this way, a pair of nodes can dynamically selects an appropriate channel for best immunity from interference. If two nearby source-destination pairs choose the same channel for data transmission, they will contend with each other just as in original IEEE 802.11. As SSCH and McMAC, this protocol assumes tight time synchronization among nodes and channel switching capability. Another drawback is that all the channels (except the one used for control purposes) remain idle during the control phase.

MAP [80] is another example of split phase protocol. Unlike MMAC, this protocol does not fixe the duration of the data phase. In MAP, the data phase duration depends on the agreements conducted during the control phase.

2.3.4.2 Distributed Multi-Channel Multi-Interface Approaches

In this section, we present distributed protocols especially designed for multi-interface environments. Each node must run the assignment algorithm locally, since no central entity is assumed to perform this task.

Multi-radio Unification Protocol (MUP)

MUP [81] is a *common assignment* solution for multi-interface wireless networks. As an example, Figure 2.8(c) shows a common assignment where the i^{th} interface of a node listens to channel c_i. In MUP, each node locally estimates the quality of channels to select the best channel to communicate with each neighbor. Once the channel is assigned to an interface, it does not change. The main idea is to select channels currently used by nearby nodes instead of available idle channels. The technique used to estimate channel quality is to send probe packets over each interface and measure the round-trip latency of channels. Although common assignment is simple to implement and avoid deafness, it leads to a sub-optimal throughput in multi-channel networks because of the inappropriate use of channel resources [25].

Although the importance of Channel and Interface Assignment in WMN, the current IEEE 802.11s mesh networking standard does not specify any Channel and Interface Assignment algorithm [19]. However, the previous drafts advocated the use of a common assignment to preserve connectivity. Even a *simple channel unification* protocol was specified to allow nodes to converge to the same set of channels after initialization [39, 82]. When a mesh node bootstraps, it must perform a scan to discover existing neighbors. If the node does not discover neighbors, it creates a new network randomly choosing one channel per interface and assigns a *channel precedence value* to each channel (the number of microseconds since the boot time plus a random number). If two disjoint nodes are discovered, the channels are chosen according to the highest *channel precedence value*. Hop by hop, the protocol removes the channels with the lowest *channel precedence value* progressively converging to common channels and forming a set of Unified Channel Graphs (UCG) [38].

Hyacinth

Hyacinth [71] is a distributed algorithm that uses local topology and traffic load to perform assignment and route computation. This approach assumes that most of the traffic is directed to/from the Internet via gateways (Mesh Portals). Besides, Hyacinth establishes a priority mechanism to give higher priority during the assignment to nodes close to the gateways over those nodes far from the gateway. The result is a tree network architecture in which links close to the gateway are given higher bandwidth.

Hyacinth classifies Network Interface Cards (NICs) in two types: UP-NICs to connect with parents and DOWN-NICs to connect with children. The UP-NIC selects the same channel as the one used by its parent, as illustrated in Figure 2.10. Thus, node only needs to evaluate the channels to assign to its DOWN-NICs. To achieve this goal, measures are made periodically to calculate the *total load* of each channel. This measure is a weighted sum of two parameters:
(i) the number of links using the channel within the interference range;
(ii) the aggregate traffic load from all links within the interference range.
Then, the node chooses the least-used channel among all its channels that is not used by a higher priority node within the interference range. In this way, interference can be reduced, especially near the gateways. In addition, Hyacinth prevents the *ripple*

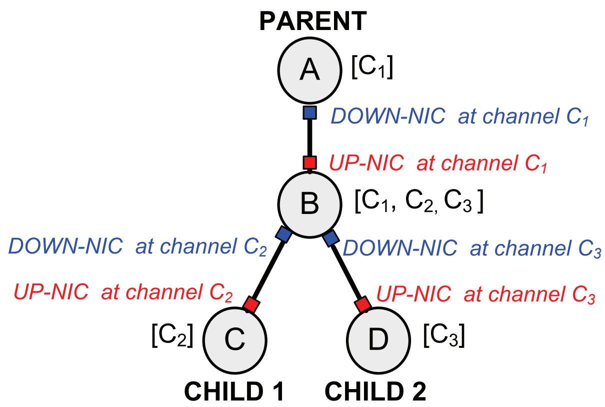


Figure 2.10: Hyacinth Operation.

effect. In other words, when a node dynamically re-assign a channel to a DOWN-NIC interface, it is not necessary to propagate this information in the entire network. Only the nodes into the interference range must be aware of this modification.

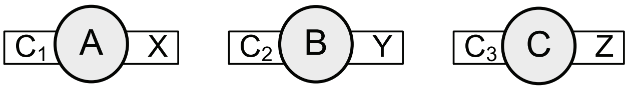
A major drawback of Hyacinth is the long period of channel scanning process to a new node join the network discover the DOWN-NIC channels of its potential parents [28].

Probabilistic Channel Usage (PCU)[2]

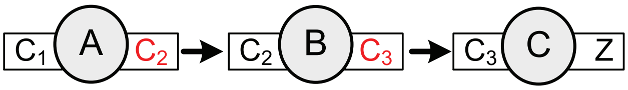
PCU [83, 84] is a multi-interface protocol that categorizes interfaces into fixed and switchable interfaces. Fixed interfaces are used to receive traffic and thus stay on specified channels for long intervals of time. On the other hand, switchable interfaces frequently switch between the remaining channels in order to transmit traffic. In the example of Figure 2.11(a) each node has two interfaces, one fixed and one switchable. If node A want to transmit traffic to C via B, A has to switch its switchable interface (originally at channel X) to channel c_2 so as to transmit to B. In turn, B switches to channel c_3 the switchable interface (originally at channel Y) so as to transmit to node C. Figure 2.11(b) depicts the resulting assignment. PCU prevents the ripple effect through the use of fixed interfaces.

A co-ordination protocol is necessary to decide what channel to assign to fixed interfaces. To balance the load across the available channels, it is advantageous if other nodes in the neighborhood use a different channel for their fixed interfaces. PCU proposes a localized protocol to measure the *channel usage*, which represents the number of nodes in the interference range that are using the same fixed channel. A particular issue is that the channel usage measure ignores channel load information, which is a drawback of PCU. If a node detects that a number of neighboring

[2]This protocol is not named, so we use this acronym for the convenience of later reference.



(a) Before Switching



(b) After Switching

Figure 2.11: Probabilistic Channel Usage (PCU) Operation.

nodes is also tuned to one of its fixed channels, it can switch to a less used channel with some probability p (set to 0.4 in paper simulations). This probabilistic approach is used to avoid the channel oscillation phenomena. Additionally, it is fundamental to inform neighboring nodes about the channel being used by each fixed interface. Periodically, each node broadcasts a *hello packet* on every channel announcing its fixed channels as well as the fixed channels being used by its neighbors.

Although the network is not fully connected at the same instant of time, the routing protocols can still assume the existence of "fixed links", since they can be dynamically established. However, adaptations are necessary to withstand frequent channel switches and network disruptions that may cause severe packet losses.

Adaptive Dynamic Channel Allocation (ADCA)

ADCA [85] is a hybrid multi-interface architecture that combines the advantages of both static and dynamic interfaces. Each mesh router uses a single interface to switch between channels frequently (i.e., dynamic interfaces), while the other interfaces stay on fixed channels (i.e., static interfaces).

Static interfaces aim at maximizing the throughput allocating channels according to the proximity of mesh routers to the gateway in a tree topology. Higher priority is given to mesh routers closer to the gateway. On the other hand, dynamic interfaces work in an on-demand fashion. Similarly to MMAC, time is split into fixed intervals of control and data exchange. A node with data to transmit to a neighbor tries to negotiate a common channel during the control phase. Each dynamic interface maintains one queue in the link layer for each neighbor. To consider some level of fairness, the service time already allocated to the queue and the queue length are used as criterion for selecting neighbors to communicate during the data exchange phase.

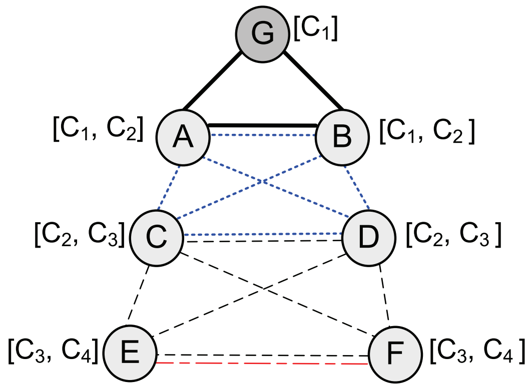


Figure 2.12: Dual-Interface Mesh Network (DIMN) Operation.

While MMAC only allows the negotiation between pairs of nodes in each interval, ADCA extends the number of nodes that can participate in a negotiation. As a result, ADCA achieves a smaller packet delay than MMAC, without degrading network throughput.

2.3.4.3 Centralized Multi-Channel Multi-Interface Approaches

Finally, we present centralized protocols especially designed for multi-interface environments. Centralized protocols assume the existence of a central authority with complete knowledge of the network. Thus, the assignment formulation is solved at a single place.

Dual-Interface Mesh Network (DIMN)[3]

DIMN [86] is a protocol for gateway paths in dual-interface mesh networks. Each node uses two distinct channels to communicate with its previous hop neighbor (i.e., its father in the tree topology) and next hop neighbor (i.e., its child in the tree topology). The assignment is defined by the gateway, which chooses a sequence of orthogonal channels (e.g., $c_1, c_2, c_3, ...$) that will be use to guide other nodes' channel assignment along the tree. As illustrated in Figure 2.12, a node k hops away from the gateway assigns channels c_k and c_{k+1} of the defined channel sequence.

This approach avoids intra-path interference due to the use of orthogonal channels along the channel sequence. However, inter-flow interference is not considered. As the authors focus mostly on single-interface gateways, they defend that assign the same channels to nodes at the same level is valid since all paths ultimately compete with each other at the first hop of the gateway. Additionally, the authors argue that many cross-links (e.g., $C - F$, $D - E$) are useful to routing protocols. They can

[3] This protocol is not named, so we use this acronym for the convenience of later reference.

profit from multiple paths to quickly adapt paths to the variations of the wireless links. For instance, if path $G - A - C - E$ degrades, node E can use a different path such as $G - A - D - E$ without having to change the assignment. Multiple interfering gateways try to use different channels in their channel sequences.

The authors propose some heuristics to minimize the inter-flow interference at different gateways. Specifically, a gateway chooses sequences whose first hop channels differ from those of potentially interfering gateways. It is considered that the first hop is typically the bottleneck with multiple flows.

DIMN does not consider the actual topology in choosing channel sequences. For instance, it may happen that nodes k hops away from the gateway are not at the same level (i.e., physical location) in the tree topology. Thus, the resulting channel sequence may not be optimal.

Connected Low Interference Channel Assignment (CLICA)

CLICA [73] is a polynomial time heuristic algorithm to assign channels to interfaces based on the Topology induced by the UDG and conflict graph. The main goal is to reduce interference over the whole network in an effort to minimize the maximum *conflict weight* among all links in the resultant Network Topology. Besides, it aims to preserve any link in the Topology induced by the UDG. The link conflict weight for a link is the sum of the number of edges incident to the vertex representing this link in the conflict graph.

When the algorithm starts, each node is given a priority based on some criterion (e.g., randomness, closeness to gateway, traffic load). Channel decisions are then made in the order of this priority. For each node, it is selected the channel to its interfaces as well as for its adjacent nodes in order to set a channel to all its links incident in the Topology induced by the UDG. These channel decisions are made in a greedy fashion. A node faced with a decision to pick a channel for an incident link makes a locally optimal choice from among the feasible set of channels. For example, the node can pick the channel that minimizes the maximum link conflict weight over all interfering links. Alternatively, the channel that minimizes the link conflict weight for the link. At the end of the algorithm, it is possible that interfaces at two neighboring nodes share more than one common channel (i.e., multiple links exist between such nodes in the Network Topology). Also, there may still remain some nodes with unassigned interfaces because they have more interfaces than their respective degree. In this case, assign channels to them can increase the potential interference of the Network Topology.

Although this algorithm overcomes link revisits, it does not incorporate the role of traffic patterns in Channel and Interface Assignment for WMNs.

Note that CLICA is dependent of a set of inputs such as the Topology induced by the UDG, the number of interfaces at each node, the number of available channels, and the interference information described by the conflict graph. However, although CLICA corresponds to a centralized algorithm, it can be used as a benchmark when evaluating distributed Channel and Interface Assignment algorithms.

Table 2.5 summarizes some features of the protocols and architectures of Channel and Interface Assignment approaches. If a propriety cannot be applied on an approach, the corresponding table entry is marked by N/A (Not Applicable).

Approach	Year	Type of Interface	Control Channel	Clock Synch.	Gateway Oriented	Deafness Problem	Ripple Effect	Channel Oscillation	Load-Aware	Ref.
SSCH	2004	Dynamic	No	Yes	No	Unsolved	Unsolved	N/A	Yes	[78]
McMAC	2007	Dynamic	No	Yes	No	Unsolved	Solved	N/A	No	[79]
MMAC	2004	Dynamic	Yes	Yes	No	Solved	N/A	N/A	Yes	[26]
MAP	2003	Dynamic	Yes	Yes	No	Solved	N/A	N/A	Yes	[80]
MUP	2004	Static	No	No	No	Solved	Solved	Solved	No	[81]
Hyacinth	2005	Dynamic	No	No	Yes	Solved	Solved	Unsolved	Yes	[71]
PCU	2006	Hybrid	No	No	No	Solved	Solved	Solved	No	[84]
ADCA	2009	Hybrid	Yes	Yes	Yes	Solved	N/A	N/A	No	[80]
DIMN	2009	Static	No	No	Yes	Solved	N/A	N/A	No	[86]
CLICA	2010	Static	No	No	No	Solved	N/A	N/A	No	[73]

Table 2.5: Comparison of Channel and Interface Assignment Approaches.

2.4 Routing

Routing can be referred to as a process of selecting paths to send network traffic between a source and destination nodes. To select optimal paths, routing metrics are assigned to each path and then used by the routing algorithm to select one, or more, out of paths discovered by the routing protocol. Furthermore, routing is a key functionality for controlling communication in large networks such as wireless mesh networks.

Although routing has been thoroughly studied in conventional networks (i.e., wired infrastructure) and ad hoc networks, the characteristics inherent to WMN called for better adapted routing metrics and protocols [87]. For instance, several ad hoc routing protocols were especially designed to overcome the frequent topology changes and/or the high flooding overhead caused by nodes mobility [46]. Since mesh routers are usually stationary, ad hoc routing must then be adapted to deal with the peculiarities of WMN, for example, by considering link quality variations [48].

This section identifies different routing metrics and protocols for Multi-Channel Multi-Interface Wireless Mesh Networks.

2.4.1 Review of Routing Metrics

A routing metric is a component or a combination of several components that depict characteristics of the path in the network. Traditionally, routing metrics are calculated only from information available directly in the network layer. Today, however, many researchers acknowledge the need for cross-layer approaches for designing routing metrics [20].

A large number of routing metric has been recommended in the literature for Wireless Mesh Networks [20, 88]. Mesh routers can adopt different *monitoring mechanisms* to gather the information (i.e., measurements) they need for the calculation of the routing metric. Generally, the two main factors that determine the choice of the monitoring mechanism are accuracy and overhead.

According to Baumann *et al.*, monitoring mechanisms are classified as follows [89]:

- **Local monitoring:** measurements required by the metric are available locally at the node, such as number of orthogonal channels, number of interfaces and length of queues.

- **Passive monitoring:** measurements are gathered by observing the traffic coming in and going out of a node. This mechanism is widely employed to collect the cross-layer measures because it does not cause overhead in the network. The interference measures mentioned in Section 2.2.1 (e.g., RSSI, BER, PER and SINR) are examples of metrics that can be captured through passive monitoring.

Passive monitoring can gather inconsistent measures in some specific situations, such as:
(*i*) when there is a small amount of processed data (e.g. BER, FER);
(*ii*) when the reception of the entire packet is not considered (e.g. RSSI);
(*iii*) when only control packets are used, as they are small in size and can fail to depict the real conditions of the channel.

- **Active monitoring:** probe packets are generated and included in the traffic to measure the link characteristics. Consequently, this mechanism introduces overhead in the network, which is directly dependent of the frequency of measurements. Active monitoring can gather inaccurate cross-layer information. For instance, occasional losses of probe packets due to wireless medium instability can lead to an overestimation of the link quality [20].

- **Piggyback monitoring:** measurements are done by including probing information into regular traffic or routing protocol packets. Thus, no additional packets are generated for metric computation proposes. Piggyback mechanism is a common method to measure delay [89].

Although the measurements represent the current state of the link, it is usually desired that short-term variations do not influence the value of routing metrics. This is a fundamental issue to ensure routing stability. Statistical functions can be used to smooth out the value of metrics and thus avoid unnecessary route oscillations. Fixed History Window (FHW) is an example of statistical function adopted by some routing metrics [63, 90]. In summary, an average value is calculated, either from a fixed number of previous measurements or from the measurements captured during a fixed time interval. Other approaches favor to give more weight to recent measurements while not entirely discarding older ones [70]. The weight given to old measurements is decreased exponentially. This statistical function is called Exponential Weighting Moving Average (EWMA).

Next, the most recent and relevant routing metrics for multi-channel multi-interface WMN are described [91, 92]. We have grouped the metric into three main categories: basic, interference aware, and interference and load aware routing metrics, as outlined in Sections 2.4.1.1, 2.4.1.2 and 2.4.1.3, respectively.

2.4.1.1 Basic Routing Metrics

Basic routing metrics describe measures that directly influence the traditional performance parameters such as throughput, latency and packet loss ratio. Most of the cross-layer routing metrics presented in the following subsections have employed at least one of these basic metrics.

Hop Count (Hop)

Hop count is a measure of the number of hops between the source and destination of a path. Link quality for this metric is a binary concept: either the link exists or it doesn't [93]. The main advantage of this metric is its simplicity since it is easy to compute and minimize the hop count between a source and a destination once the topology is known. Moreover, computing the hop count requires no additional measurements, unlike the other metrics we will describe below.

Traditional MANET protocols typically find routes with the minimum hop count [12, 14]. This metric is appropriate for these networks because new paths must be found rapidly. For instance, Draves *et al.* [93] demonstrated that hop count metric outperforms link-quality metrics in mobile scenarios because hop count metric reacts more quickly to fast topology change.

The primary disadvantage of this metric is that it maintains a very limited view of links, ignoring issues such as packet loss or bandwidth. When two paths of same hop count are found, the arbitrary choice of paths does not take into account the quality of the link. This can lead the hop count metric to choose paths with slow or lossy links. Furthermore, minimize the hop count means maximizing the distance traveled by each hop, which is likely to minimize signal strength (e.g., RSSI) and maximize the loss ratio, leading to poor throughput. For example, De Couto *et al.* [94] showed that a route with a higher number of *short links* can outperform a route with a smaller number of *long distance links* (i.e., lower quality links). Hence, a metric to effectively capture the quality of the wireless links is necessary.

Expected Transmission Count (ETX)

ETX [90] is a measure of link and path quality. This metric calculates the expected number of MAC layer transmissions, including retransmissions, a node requires to successfully deliver a unicast packet through a wireless link. The weight of a path is the sum of the ETX values for each link along the path. So, this metric comprises both packet loss ratio and path length.

To compute the derivation of ETX, each node periodically broadcasts probes packets (e.g., set to 1 seconds in paper simulations) of a fixed size containing the number of received probes from each neighbor. The number of received probes is calculated at the least τ time interval in a sliding-window fashion (e.g., set to 10 seconds in paper simulations). Thus, a node A calculates the ETX of a link l to a node B by using the delivery ratio of probes sent on the forward (data) and reverse (acknowledgement) directions. The forward delivery ratio (d_f) is the fraction of successfully received probes from A announced by B at interval τ. The reverse delivery ratio (d_r) is the fraction of successfully received probes from B at the same interval τ. Assuming d_f and d_r independents, the probability of a successful transmission, including acknowledgement, is $d_f * d_r$. ETX of a link l is the inverse of this value:

$$ETX_l = \frac{1}{d_f * d_r}. \tag{2.2}$$

Although ETX outperforms hop count in static single-rate single-interface networks [94], it presents shortcomings that affects the overall performance. First, ETX does not distinguish links with different data rates. Second, ETX assumes that the probability that a given packet is lost in transmission is independent of its size, which introduces inaccuracies in the estimation of the loss rate. These inaccuracies are caused by the smaller size of the probe packets when compared with data packets. Third, the metric does not account for channel diversity. Hence, ETX does not depict the extent of intra-flow interference. It can only detect inter-flow interference indirectly because the high level of interference will probably result in higher packet loss ratio and therefore in a higher ETX value [20].

Koksal and Balakrishnan [95] extended ETX to propose two metrics called Modified ETX (mETX) and Effective Number of Transmissions (ENT). These new metrics are also calculated by broadcasting link-layer probes packets. However, unlike ETT, mETT and ENT metrics are aware of probe size and data rate. Besides, they estimate the losses at the bit level rather than considering probe losses at the packet level. In particular, the bit error estimation uses the position of the erred bits in each probe packet. In ENT, when a specific link presents a number of expected transmissions higher than a certain threshold (i.e., tolerable loss rate), ENT assigns an infinity metric to this link in order to exclude the link from the routing computation. As ETX, these metrics do not take into account the intra-flow interference.

Expected Transmission Time (ETT)

ETT [30] measures the total amount of time it would take to send a data packet along a path, while taking into account the transmission rate of each link and its delivery probability at that transmission rate. This metric extends ETX by taking account of the differences in link data rates and data packet sizes. The weight of a path is the sum of the ETT values for each link along the path. The ETT of link l

is calculated as follows:

$$ETT_l = ETX_l * \frac{S}{B_l}, \tag{2.3}$$

where S is the packet size and B_l the bandwidth of link l (raw data rate). The relation $\frac{S}{B_l}$ estimates the *expected time* to successfully transmit a packet over link l. Packet loss ratio is also comprised because ETX is a part of ETT. For this reason, ETT maintains a number of drawbacks of ETX such as not being able to capture link load explicitly. Furthermore, ETT was not designed for multi-interface networks and therefore does not attempt to minimize inter-flow and intra-flow interference.

Airtime Link Metric (ALM)

ALM is the default link metric of IEEE 802.11s [19], which is calculate for each pairwise link within the mesh network. The path selection protocol accumulates all the link metric values included in the selected multi-hop path to obtain the overall cost of the path. This metric estimates the amount of channel resources consumed when transmitting a frame over a link l. It is defined in terms of overhead, data rate, and transmission errors:

$$ALM_l = \left(O_{ca} + O_p + \frac{S}{r}\right) * \frac{1}{1 - e_f}, \tag{2.4}$$

where O_{ca} is the channel access overhead, O_p is the protocol overhead, and S is the number of bits in the test frame. The parameter r is the data rate in megabits per second. The frame error rate e_f is the probability that when a frame of standard size S is transmitted at the current transmission bit rate r, the frame is corrupted due to transmission error.

A closer look at the ALM reveals that this metric is analogous to ETT [96]. The first part of Equation 2.4 reflects the transmission time and the second part measures the number of retransmissions required, like ETX.

2.4.1.2 Interference Aware Routing Metrics

In this section, we describe the most relevant interference aware routing metrics.

Weighted Cumulative ETT (WCETT)

WCETT [30] is an ETT extension to reduce the intra-flow interference in MCMI WMN. Unlike the previous presented metrics, WCETT is an end-to-end metric and then its outcome is the final cost of the path. This change occurs because WCETT must consider all channels used along the path to avoid intra-flow interference.

The WCETT metric for a h-hops path p is defined as follows:

$$WCETT_p = (1-\beta) * \sum_{\text{link } l \in p} ETT_l + \beta * \max_{1 \leq c \leq \mathcal{C}} X_c, \tag{2.5}$$

where β is a tunable parameter subject to $0 \leq \beta \leq 1$. $\mathcal{C}$ is the total number of available orthogonal channels. X_c considers the number of times channel c is used along path p. It is calculated as the sum of transmission times (i.e., ETT) of links in the same channel c:

$$X_c = \sum_{\text{link } l \text{ is on channel } c} ETT_l \ , \ 1 \leq c \leq \mathcal{C}. \tag{2.6}$$

Thus, the maximum X_c in Equation 2.5 considers the maximum number of times that the same channel appears along a path. The rationale is that the total path throughput is determined by the bottleneck channel (i.e., the busiest channel on the path), which corresponds to the largest X_c.

Accordingly, the first part of WCETT metric (Equation 2.5) helps in finding paths with links having less ETT. The second part will favor paths that have greater channel diversity and helps in finding paths with less intra-flow interference. The parameter β permits to balance between delay and channel diversity/throughput.

One limitation of WCETT is that it does not capture the traffic load. Hence, this metric may route flows to dense areas where congestion is more likely and overall network throughput degrades. Another limitation is the way interference range is defined. WCETT assumes that, if links on a path are on the same channel, these links always interfere with each other independently of the distance between them. In other words, the interference is considered so large that it covers the entire path. This assumption is usually true for short paths, but is somehow pessimist for longer paths.

Also, WCETT lacks of *isotonicity* property due to the X_c component [97]. The isotonic property means that a metric should ensure that the order of the weights of two paths is preserved if they are linked to a common third path, as illustrated in Figure 2.13. More formally, assuming that $W(a)$ denotes the weight defined by a routing metric for a path a. Denoting $a \oplus b'$ the concatenation of two paths a and b', a routing metric $W(\cdot)$ is isotonic if $W(a) \leq W(b')$ implies $W(a \oplus c) \leq W(b \oplus c)$ and $W(c' \oplus a) \leq W(c' \oplus b)$ for all a, b, c, c' paths. Given this definition, Sobrinho's work [98, 99] has shown that isotonicity is a sufficient and necessary condition for both the Bellman-Ford and Dijkstra's algorithm to find minimum weight paths and to ensure loop-free routing. Network performance may degrade with the resulting sub-optimal paths. In conclusion, if a routing metric is not isotonic, only algorithms with exponential complexity will be capable to calculate minimum weight paths.

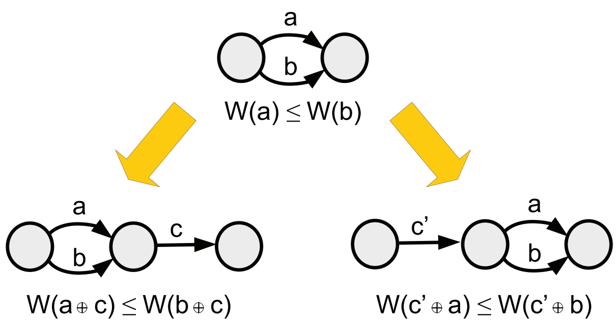


Figure 2.13: Example of Isotonicity.

One solution is use on-demand routing, source routing or distance-vector routing with non-isotonic routing metrics, since these protocols do not require isotonicity to ensure loop-free routing [97].

Metric of Interference and Channel-switching (MIC)

MIC [100] improves WCETT by overcoming its inability to capture inter-flow interference. The MIC metric for a h-hops path p is defined as follows:

$$MIC_p = \frac{1}{N * \min(ETT)} * \sum_{\text{link } l \in p} IRU_l + \sum_{\text{node } v \in p} CSC_v, \tag{2.7}$$

where N is the total number of nodes in the network and the $\min(ETT)$ the smallest ETT in the network. The two components are the Interference-aware Resource Usage (IRU) and the Channel Switching Cost (CSC).

IRU depicts the inter-flow interference as follows:

$$IRU_l = ETT_l * N_l, \tag{2.8}$$

where N_l denotes the set of neighbors that can interfere with the transmission on link l. Essentially, IRU_l represents the aggregate channel time spent by the transmission of neighboring nodes in the link l [97].

The CSC depicts the intra-flow interference as follows:

$$CSC_v = \begin{cases} w_1, & \text{if} \quad c(prev(v)) \neq c(v), \\ w_2, & \text{if} \quad c(prev(v)) = c(v), \end{cases} \tag{2.9}$$

where $0 \leq w_1 < w_2$, $c(v)$ is the channel assigned to node v and $prev(v)$ represents the previous hop of node v along the path p. This component of MIC gives more weight to paths with consecutives links using the same channel, favoring paths with more diversified channel assignments.

It is worth noting that MIC is non-isotonic because of CSC. Another limitation of MIC is that IRU assumes that all links in the interference range have the same degree of interference, even though a neighbor is not involved in any transmission, whether it occurs simultaneously with that link or not. In fact, the degree of interference depends on the amount of traffic generated by the interfering node. Consequently, MIC does not take account of traffic load measures.

In addition, CSC in a non-scalable component that may become impracticable because the run-time complexity increases significantly with the number of interfaces [20].

Interference Aware Routing (iAWARE)

The iAWARE [68] metric addresses intra-flow and inter-flow interference by means of signal strength values. It was the first metric to employ a measurement of inter-flow interference based on the Physical Model. Unlike MIC, iAWARE continuously captures the degree of interference caused by each interfering node on a link.

The iAWARE metric for a path p is defined as follows:

$$iAWARE_p = (1-\beta) \sum_{\text{link } l \in p} iAWARE_l + \beta * \max_{1 \le c \le \mathcal{C}} X_c, \tag{2.10}$$

where $iAWARE$ depicts the inter-flow interference, X_c captures the intra-flow interference, and β represents a trade-off between the inter-flow and intra-flow interference.

The $iAWARE$ metric captures the inter-flow interference of a link l as follows:

$$iAWARE_l = \frac{ETT_l}{IR_l}. \tag{2.11}$$

The Interference Ratio (IR) component estimates the interference level in the network through the Signal to Interference plus Noise Ratio (SINR) and Signal-to-Noise-Ratio (SNR):

$$IR_l = \frac{SINR_l}{SNR_l}, \tag{2.12}$$

where $0 \le IR_l \le 1$. In turn, SNR_l and $SINR_l$ are calculated as follows:

$$SNR_l = \frac{P_l}{Noise}, \tag{2.13}$$

$$SINR_l = \frac{P_l}{Noise + \sum_{u \in N_l - v} \tau_u * P_u}, \tag{2.14}$$

where P_l is the signal strength of the link l, $Noise$ the background noise and N_l the set of neighbors that can interfere with the transmission on link l. τ_u is the amount of time that node u occupies the channel. When there is no interference (i.e., no traffic generated by interfering neighbors or no interfering neighbors), $IR_l = 1$

because $SINR_l$=SNR_l. Accordingly to Equation 2.11, $iAWARE_l$ is simply the ETT_l.

To address intra-flow interference, X_c is employed to take advantage of the diversity of available orthogonal channels $\mathcal{C}$:

$$X_c = \sum_{\text{conflicting links } l \text{ on channel } c} iAWARE_l \tag{2.15}$$

Note that iAWARE takes full account of the maximum sum of iAWARE over the links, while WCETT accounts for the maximum sum of ETT (Equation 2.6).

As WCETT, iAWARE is non-isotonic because of the second component X_c. Besides, it does not take account of traffic load measures. Consequently, this metric does not always provide paths with less congestion.

2.4.1.3 Interference and Load Aware Routing Metrics

Finally, we present routing metrics that together depict interference and load.

Resource Aware Routing for mEsh (RARE)

RARE [70] uses passive monitoring techniques to measure the links characteristics. In particular, the capacity of a link (C), available bandwidth (B_l), Received Signal Strength Indication (RSSI) and average contention (C_o) are combined in the same link cost function:

$$RARE_l = \alpha * \frac{C - B_l}{B_l} + \beta * \frac{RSSI_{\max} - RSSI}{RSSI} + \gamma * C_o, \tag{2.16}$$

where α, β and γ are weights associated with the bandwidth, RSSI and contention components, respectively. $RSSI_{\max}$ is the maximum value of RSSI and depend on the chipset of the wireless card.

To measure the traffic load, B_l is based on the duration of busy and idle intervals, which are normalized and combined with the transmission rate as follows [101]:

$$B_l = \frac{T_{idle}}{T_{idle} + T_{busy}} * TX_{rate}, \tag{2.17}$$

where TX_{rate} is the transmission rate, T_{busy} is the busy time on the medium associated with the transport of traffic load and T_{idle} is the complementary time intervals.

RARE is an isotonic routing metric in which all the parameters are captured through a passive monitoring. Thus, is does not introduce measurements overhead.

RARE has two main drawbacks. First, it is the inability to depict channel diversity and hence, does not result in paths with less intra-flow interference. Second, as discussed in Section 2.2.1, RSSI is not an accurate measure to depict interference, especially at high transmission rates.

Expected Link Performance (ELP)

ELP [64] proposes three main components to calculate the routing metric.

First, the metric proposes a simple solution to address the problem of link asymmetry in ETX: the packet size on the forwarded link (d_f) and the reverse link (d_r) is asymmetric. To improve the link delivery ratio calculations, ELP proposes to assign a higher weight to the forward link in order to give more importance to data packets. The authors argue that the reverse link is only meant for the ACK packets (for the path being calculated) that are loss resistant and would probably be successfully received almost regardless of estimated reverse delivery ratio. More specifically, the link loss probability expressed in terms of probe delivery probabilities is equal to the sum of the probability that the data transmission fails in the forward direction $(1 - d_f)$ plus the probability that a data transmission is successfully received in the forward direction but the corresponding ACK is lost $d_f * (1 - d_r)$. Based on this definition, ELP introduces the following constant:

$$ELP_{LinkLoss} = \beta * (1 - d_f) + d_f * [(1 - \beta) * (1 - d_r)], \tag{2.18}$$

where β represents the corrective term $(0.5 < \beta < 1)$.

The second component addresses the link interference in order to capture the logical interference present in the link. In particular, the *Average Interference Ratio (AIR)* at a node v during the time interval T is defined as:

$$AIR(v) = \frac{T_{Receive} + T_{Occupied} + T_{Backoff}}{T}, \tag{2.19}$$

where $T_{Receive}$, $T_{Occupied}$ and $T_{Backoff}$ represent the fraction of time that node v is unable to transmit on the channel. These three channels states are measured through IEEE 802.11 interfaces used in promiscuous mode. Based on the local view of nodes located at the two ends of a link (e.g., nodes v and u), the AIR for a link l is calculated as the maximum of $AIR(v)$ and $AIR(u)$. Thus, the link interference portion of the ELP metric becomes:

$$ELP_{LinkInterference(l)} = AIR(l) = Max(AIR(v), AIR(u)), \tag{2.20}$$

The third component takes into consideration the link capacities. In particular, links with higher bandwidth are given a lower link cost, as follows:

$$ELP_{LinkCapacityFactor(l)} = \frac{1}{Bandwidth(l)}, \tag{2.21}$$

Through Equation 2.21, links with higher capacity are preferable: they can transmit data at a higher rate and therefore occupy the medium for a shorter period of time compared to the low capacity link which will take longer time and create interference for nodes in vicinity.

Based on these three components, the ELP metrics for a link l is calculated as follows:

$$ELP(l) = ELP_{LinkLoss(l)} + ELP_{LinkInterference(l)} + ELP_{LinkCapacityFactor(l)}, \quad (2.22)$$

Then, the weight of a path p is the sum of the ELP values for each link along the path:

$$ELP(p) = \sum_{\text{link } l \in p} ELP(l), \quad (2.23)$$

The path with minimum $ELP(p)$ is selected.

Although this metric was proposed for single-channel single-interface networks, it can also be applied for multi-channel multi-interface networks. However, ELP does not favor paths with greater channel diversity.

Metric for INterference and channel Diversity (MIND)

MIND [63] combines inter-flow interference based on signal strength measures with intra-flow interference based on channel diversity. Besides, it considers traffic load estimation through passive monitoring. The MIND metric for a path p is expressed as follows:

$$MIND_p = \sum_{\text{link } l \in p} InterLoad_l + \sum_{\text{node } v \in p} CSC_v. \quad (2.24)$$

The first component $InterLoad_l$ captures inter-flow interference and traffic load simultaneous:

$$InterLoad_l = ((1 - IR_l) * \tau) * CBT, \quad (2.25)$$

where CBT is the Channel Busy Time, τ is a configurable parameter used to provide higher weight to interference in the $InterLoad$ component, and $0 \leq IR \leq 1$ and $0 \leq CBT \leq 1$.

The Interference Ratio (IR) component captures the inter-flow interference based on the Physical Model. In fact, MIND extends the IR defined by iAWARE (Equation 2.12) with the difference that here $SINR$ does not take into consideration the amount of time a node occupies the channel (i.e., parameter τ in Equation 2.14) because CBT is already used as a component of $InterLoad$.

The estimation of traffic load is based on CBT as follows:

$$CBT = \frac{TotalTime - T_{idle}}{TotalTime}, \quad (2.26)$$

where $TotalTime$ is the measure of time between the first attempt to send the packet and the reception of its acknowledge. T_{idle} is the measure of backoff times (i.e., node finds the medium busy when it tries to transmit and then waits for a random period of time before trying to transmit again), and the time in which the node senses medium free for access and it has no data to transmit.

Instead of using the current value of a single packet, smoothing out functions are used to avoid oscillations of CBT and IR. For example, authors use the CBT average of the last twenty packets, including both data and control packets.

Similar to MIC, MIND uses the Channel Switching Cost (CSC) to reduce intra-flow interference (Equation 2.9). Consequently, MIND is also non-isotonic.

Table 2.6 presents an overview of the characteristics supported by the presented routing metrics. If a propriety cannot be applied on an approach, the corresponding table entry is marked by N/A (Not Applicable). It is worth noting that most routing metrics combine measures, methods or metrics provided by other routing metrics. For instance, ETT and ETX are reutilized by WCETT, MIC, and iAWARE metrics; MIND uses the CSC proposed by MIC to measure intra-flow interference; SINR are used by both iAWARE and MIND to measure inter-flow interference; etc. Despite the fact that routing metrics in MCMI WMN has been widely addressed in the literature, there are still a number of important research issues that need to be further analyzed and improved.

Metric	Year	Load-Aware	Data Rate	Packet Size	Interfer. Type	Interfer. Model	Isoto-nicity	Monitor. Mechan.	Stability Mechan.	Ref.
Hop	-	No	No	No	None	None	Yes	None	N/A	[93]
ETX	2003	No	No	No	None	None	Yes	Active	FHW	[90]
mETX	2006	No	Yes	Yes	None	None	Yes	Active	EWMA	[95]
ENT	2006	No	Yes	Yes	None	None	Yes	Active	EWMA	[95]
ETT	2004	No	Yes	Yes	None	None	Yes	Local, Active	FHW	[30]
ALM	2006	No	Yes	Yes	None	None	Yes	Active	FHW	[19]
WCETT	2004	No	Yes	Yes	Intra	None	No	Local, Active	FHW	[30]
MIC	2005	No	Yes	Yes	Intra, Inter	Logical	No	Local, Passive	FHW	[100]
iAWARE	2006	No	Yes	Yes	Intra, Inter	Physical	No	Local, Passive	FHW	[68]
RARE	2007	Yes	Yes	Yes	Inter	Physical	Yes	Passive	EWMA	[70]
ELP	2008	Yes	Yes	Yes	Intra, Inter	Logical, Physical	Yes	Active, Passive	FHW	[64]
MIND	2009	Yes	Yes	Yes	Intra, Inter	Logical, Physical	No	Passive	FHW	[63]

Table 2.6: Characteristics of Routing Metrics for Multi-Channel Multi-Interface Wireless Mesh Networks.

2.4.2 Review of Routing Protocols

Generally, mesh routing protocols can be classified into one of the following strategies [48, 103]:

- **Proactive Routing** establishes paths regardless of the willingness of a node to transmit data. Mesh routers keep at least one path to any destination in the network through a constant exchange of routing information. When a node has data to send to a certain destination, the path is selected from the paths stored in the routing table. Although this strategy permits every node to have information about the whole network, it results in a large overhead since the routing information is periodically being disseminated. DSDV [11] and OLSR [13] are commonly used proactive routing protocols.

- **On-demand (Reactive) Routing** processes are initiated upon request of a node that has traffic to send. AODV [12] and DSR [14] are examples of ad hoc reactive protocols. In comparison to proactive routing, this strategy can significantly reduce the routing overhead when the traffic is lightweight, since reactive routing does not need to update route information periodically, neither need to find and maintain routes on which there is no traffic. However, this strategy has some initial latency, since traffic can only be sent after path discovery process has finished.

- **Hybrid Routing** combines the advantages of proactive and reactive routing strategies. Here, the main challenge is to find the correct trade-off between two strategies.

Next, we present an overview of routing protocols for WMN. Some of them are extensions of existing ad hoc routing protocols, while others are specifically designed for WMN.

Hybrid Wireless Mesh Protocol (HWMP)

HWMP [19] is the mesh path selection protocol recommended by IEEE 802.11s. This protocol is based upon AODV protocol [12] with adaptations for MAC address-based path selection and link metric awareness. HWMP combines the flexibility of a reactive path selection with a proactive tree-based approach. These two modes of operation provide different levels of functionality as follows:

- **HWMP On-demand Mode** allows mesh routers (Mesh STA, Mesh Gates, Mesh Portals) to communicate with each other using peer-to-peer paths. This mode is always available, independent of whether a Mesh Portal is configured in the network or not. Specifically, a source node that does not have a valid path to the destination can use Path REQuest (PREQ) broadcasting mechanism to discover link metric information to destination. Upon receiving a PREQ, the destination sends a unicast Path REPly (PREP) message back to the source. Intermediate nodes which have forwarded PREQs and PREPs messages create or update their own routing table to relay data packets between the source and the destination.

- **HWMP Tree-based Proactive Mode** maintains a tree path from the gateway (Mesh Portal) to all other nodes. This mode can be seen as an additional functionality of on-demand mode. As discussed before, Internet access is one of the main services in WMNs, so it is expected that paths towards the gateway (Mesh Portal) are most frequently utilized.

  The IEEE 802.11s standard defines two mechanisms for proactively disseminating path selection information for reaching the Mesh Portal:

  - **Proactive PREQ mechanism**: is intended to create paths between Mesh STAs and the Mesh Portal in the network. The Mesh Portal periodically propagates proactive PREQs, with the Target Address set to all ones and the Target Only subfield set to 1. HWMP uses increasing sequence numbers to ensure that Mesh STAs can distinguish current path information from stale path information at all times in order to maintain loop-free connectivity. The PREQ contains the path metric and the sequence number.

    A Mesh STA receiving a proactive PREQ creates or updates its forwarding information to the Mesh Portal, updates the metric and hop count of the PREQ, records the metric and hop count to the Mesh Portal, and then transmits the updated PREQ [19].

    As the proactive PREQs are disseminated in the entire network, a Mesh STA may receive multiple copies of a proactive PREQ. Thus, a Mesh STA updates its current path to the Mesh Portal if and only if:
    (i) the new PREQ contains a greater HWMP sequence number, or
    (ii) the HWMP sequence number is the same as the current path, but the new PREQ offers a better metric than the current path.

    If the proactive PREQ is sent with the Proactive PREP subfield set to 0, every Mesh STA may respond to the received PREQ by sending a PREP back to the Mesh Portal. On the contrary, if the proactive PREP subfield set to 1, the response is mandatory. In this case, the network overhead for sending the PREPs may be significant.

Mechanism	Handshaking	Message Sequence
Proactive PREQ	2-way	Proactive PREQ → PREP
Proactive RANN	3-way	RANN → unicast PREQ → PREP

Table 2.7: Tree-based Proactive Mechanisms defined by IEEE 802.11s.

– **Root ANNouncement (RANN) mechanism**: is used to announce the presence of a Mesh Portal in the network, but there is no forwarding information created. In other words, the information contained in the RANN disseminates path metrics to the Mesh Portal, but reception of a RANN does not establish a path.

  The Mesh Portal periodically broadcasts RANN messages into the entire network. A node that wants to create or refresh the path sends a unicast PREQ to the Mesh Portal via the mesh router from which it received the RANN. Then, the Mesh Portal responds by sending a PREP in response to each PREQ. In summary, the addressed PREQ creates the reverse path from the Mesh Portal to the originator Mesh STA, while the PREP creates the forward path from the mesh STA to the Mesh Portal.

A Mesh Portal sends either proactive PREQ or RANN elements periodically. Table 2.7 summarizes the features of these two mechanisms.

Note that on-demand and tree-based proactive modes are not exclusive. In particular, they are used concurrently, since the tree-based proactive mode is an extension of the on-demand mode.

Path ERRor (PERR) messages can be used by any of the available HWMP modes for announcing one or more unreachable destinations. The announcement is sent to all traffic sources that have an active path to the destination(s). The PERR will reach the corresponding sources, which will start a new path discovery cycle in order to find an alternative path.

Unfortunately, HWMP is unsuitable for MCMI WMN because it does not take into consideration the high overhead incurred by the replication of control messages on multiple interfaces [104]. Specifically, when multiple channels and multiple interfaces are used, control messages such as PREQ are retransmitted by every interface. As a result, the amount of transmitted control messages exponentially increases with the number of interfaces, which causes severe performance degradation due to heavy contention and collisions. We refer to these problems associated with flooding as the *broadcast storm problem* [105]. We discuss the impact of this misbehavior of the HWMP in Chapters 5 and 6.

As HWMP, AODV-Spanning Tree (AODV-ST) [106] is a hybrid routing protocol designed for WMN. The proactive strategy is used to discover routes between the mesh routers and the gateway, while reactive strategy is used to find paths between mesh routers. In the proactive strategy, the gateway periodically broadcasts PREQ messages to initiate the creation of spanning trees. Each mesh routers creates a reverse route entry for the gateway if the received PREQ is the best known path. Then, it sends a gratuitous PREP back to the gateway. The reactive strategy works as HWMP on-demand mode.

Multi-Radio AODV (AODV-MR)

AODV-MR [107] is an AODV extension to support multiples radios/interfaces. When a source node wants to found a path in AODV-MR, it simultaneously broadcasts a PREQ message on all its interfaces. Each neighbor sharing at least one common channel will receive the message and create a reverse route that points towards the source node. To allow multi-interface support, the routing table is adapted to indicate the interface number via which a next hop node, for a particular path, can be reached. This information allows futures PREP and data traffic to be sending on the correct interface. After updating the routing table, the intermediate node re-broadcasts the PREQ message on all its interfaces, except the one on which the PREQ was initially received. The PREQ propagation continues until the message reaches the destination or an intermediate node with a fresh route to the destination. While PREP message is send to the source node, intermediate nodes profit to establish a forward path to the destination.

Link-Quality Source Routing (LQSR)

LQSR [108] is an ad hoc routing protocol based on DSR [14]. Consequently, LQSR implements the basic functionalities of DSR, including path discovery (PREQ and PREP messages) and path maintenance (PERR messages). However, LQSR improves DSR behavior to perform routing based on link quality metric such as ETX [90] rather than traditional hop count metric. To support link-quality metrics, LQSR is implemented at layer 2.5 instead of layer 3.

Fundamentally, LQSR is a link-state routing protocol, but it combines the advantages of proactive routing with reactive routing from ad hoc networks. Since periodic flooding results in high overhead, link-state proactive information is limited in such a way that hello messages are sent only to one hop neighbors. On the other hand, reactive routing is used to new path discovery procedures. The source node piggy-backs the link metric of its adjacent nodes on PREQs messages. In this way, intermediate nodes can use overheard PREQ to update link-state information.

Protocol	Year	Metrics	Strategy	Features	Ref.
HWMP	2006	ALM	Hybrid	Gateway-oriented	[19]
AODV-ST	2005	ETT	Hybrid	Gateway-oriented	[106]
AODV-MR	2008	Packet loss, delivery rate, delay, overhead	Reactive	Ad hoc based, Multi-interface	[110]
LQSR	2004	ETX	Hybrid	Ad hoc based	[108]
SrcRR	2005	ETX	Reactive	Ad hoc based	[109]

Table 2.8: Comparison of Wireless Mesh Routing Protocols.

In LQSR, only the target of a PREQ is allowed to send a PREP with the up-to-date link metrics from the arriving source route.

The SrcRR protocol [109] is used by the RoofNet mesh testbed [8]. As LQSR, the general design of SrcRR is inspired by DSR. However, SrcRR does not adopt a hybrid routing approach. It corresponds to a reactive protocol with source routing traffic. On the one hand, the lack of proactive information reduces the overhead of link-quality updates. On the other hand, it obliges nodes to calculate paths with a limited vision of the network topology.

Table 2.8 presents a summary with the main characteristics of the presented routing protocols.

2.5 Conclusion

In this chapter, we provide guidelines to contextualize the research conducted in this thesis. This chapter has sought to provide a thorough analysis of the state-of-art of Multi-Interface Multi-Channel Wireless Mesh Networks. We introduced WMNs along with their architecture, characteristics and application scenarios. The similarities and differences of mesh networks from traditional wireless networks were emphasized. Next, the chapter brings up some concerns that impact the development of MCMI WMN approaches with a special focus on interference and optimization issues. The most important issues and approaches related to the Channel and Interface Assignment were also presented. For these approaches, we extract their basic ideas and identify their advantages and limitations. In particular, we have focused on the trade-off between connectivity and interference (e.g., the more interfaces assigned to the same channels, the better connectivity, but the more interference is induced). We also present the conflict graph concept as a promising way to incorporate connectivity and interference into the Channel and Interface Assignment problem. Finally, the chapter gave a general description of routing metrics and protocols for MCMI WMN. We have shown that despite the fact that routing metrics and protocols have been widely addressed in the literature, there is still a number of important research issues that need to be further analyzed and solved.

Part II

Contributions

CHAPTER 3

Channel and Interface Assignment Framework

Contents

3.1 Introduction

There are many related papers that study the benefit of using multiple channels and interfaces in Wireless Mesh Networks (WMN) [25, 72]. These approaches may adopt different strategies to decide *when to switch interfaces* as well as *which channel to assign*. The key challenge consists in achieving an effective Channel and Interface Assignment. As discussed in Section 2.3, there is an inherent trade-off between connectivity and interference. The more interfaces are assigned to the same channels, the better connectivity. However, interference and contention have oppositely a negative impact on network capacity [28].

Surprisingly, taking into account the impact of Channel and Interface Assignment on connectivity has received little attention so far. In particular, the problem of network partitions when a network starts up was not well studied. Similarly, to the best of our knowledge, the impact of Channel and Interface Assignment strategies on neighbor discovery process has never been studied, although it is a major component of network operation.

We propose here a formal evaluation of different assignment solutions, which we classify into five different Channel and Interface Assignment (CIA) strategies. The proposed classification is flexible enough to accommodate any existing work that addresses the assignment problem. For each strategy, three main connectivity issues are studied: topology formation, density of connections, and neighbor discovery. We compare the CIA strategies based on probabilistic analysis, corroborated by simulations. In the meantime, advantages and limitations of each CIA strategy are discussed regarding issues such as interference, routing, load balancing and stability. As a result, our framework provides guidelines for network designers in planning MCMI network deployments. They can choose the most suitable CIA strategy to obtain the desired properties.

3.2 Network Model

We model a WMN as an undirected graph $G = (V, E)$, where V represents the set of nodes in the network and E the set of edges corresponding to two nodes able to directly communicate. The wireless network offers $\mathcal{C}$ orthogonal channels. Each mesh router v is equipped with I_v interfaces such as:

$$\forall v \in V, I_v = I_v^S + I_v^D, \tag{3.1}$$

where I_v^S is the number of *Static* interfaces and I_v^D the number of *Dynamic* interfaces. A *Static* interface stays tuned to a particular channel, which prevents deafness on reception. A *Dynamic* interface switches between different channels. It may suffer from deafness on reception: the transmitter must know the channel to use at a given instant.

We consider the situation in which the number of interfaces is smaller than the number of channels: $\forall v \in V, I_v < \mathcal{C}$. Otherwise, the assignment problem becomes much simpler. We can assign one interface per channel to maintain a fixed topology.

Two nodes in the radio range of each other can communicate directly if they use the same channel at the same time. More formally, $\forall v_1, v_2 \in V$, $(v_1, v_2) \in E$ if:

$$\exists i \in Intf(v_1), \exists j \in Intf(v_2), \mathcal{S}(i) \cap \mathcal{S}(j) \neq \emptyset; \tag{3.2}$$

where $Intf(v)$ is the set of interfaces of node $v \in V$, and $\mathcal{S}(i)$ is the schedule of interface i and corresponds to a list of tuples $\{channel, timeStart, timeStop\}$.

Moreover, if multiple interfaces at v_1 and v_2 share l common channels, there are l links $(v_1, v_2) \in E$, where l is a positive integer. We use $N_{links}(v_1, v_2)$ to denote the number of links between two nodes.

Table 3.1 presents the parameters of the network model and their corresponding definition. Next, we introduce a classification of CIA strategies we further use in our analysis.

Parameter	Definition
$\mathcal{C}$	Number of orthogonal channels
I_v	Number of interfaces of node v
I_v^S	Number of static interfaces of node v
I_v^D	Number of dynamic interfaces of node v
$Intf(v)$	Set of interfaces of node v
$\mathcal{S}(i)$	Schedule of interface i
$N_{links}(v_1, v_2)$	Number of links between nodes v_1 and v_2

Table 3.1: Parameters of the Network Model.

3.3 Interface Assignment

We define three types of interface behavior in the network. Figure 5.2 illustrates an example of each behavior with 2 interfaces and 4 channels. The x-axis is the time and the y-axis is the channel id.

- **Static Interfaces**: all interfaces are static and remain on the same channel for a long period of time. Thus,

$$I_v = I_v^S. \tag{3.3}$$

  Figure 3.1(a) shows an example of behavior with static interfaces. Note that each interface remains on the same channel regardless of time. So, interfaces do not explore all the available channels.

- **Dynamic Interfaces**: all interfaces are dynamic and frequently switch from one channel to another.

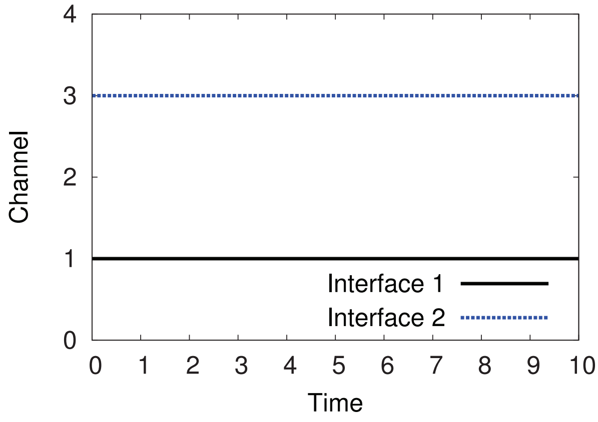
$$I_v = I_v^D. \tag{3.4}$$

  As illustrated in Figure 3.1(b), interfaces switch between channels over time tuning to all available channels.

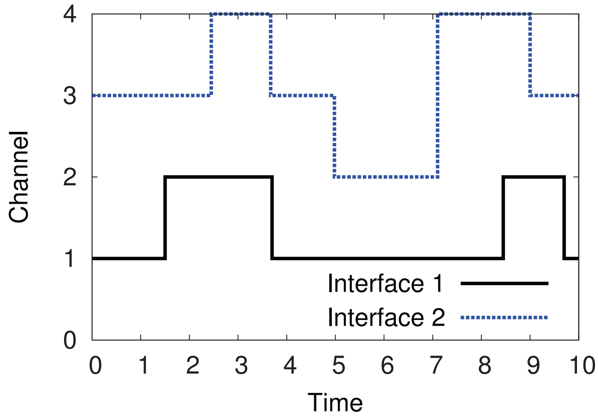
- **Mixed Interfaces**: I_v^S static interfaces permanently stay on a channel and I_v^D dynamic interfaces frequently switch from one channel to another. Thus,

$$I_v^S \geq 1, I_v^D \geq 1. \tag{3.5}$$

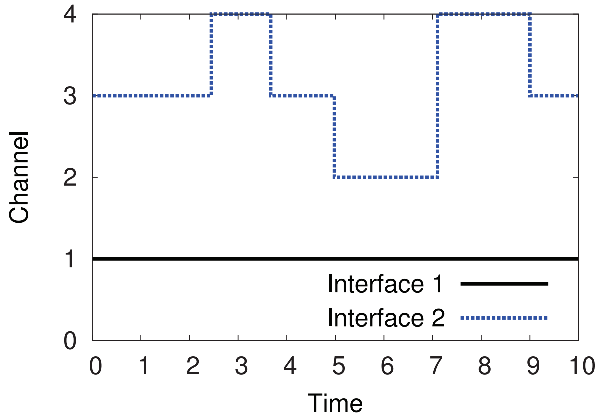
  In Figure 3.1(c), *Interface 2* switches from one channel to another and *Interface 1* remains tuned to channel 1. Mixed Interfaces combine the flexibility of dynamic interface assignment with the simplicity of static interface assignment.



(a) Static Interfaces.



(b) Dynamic Interfaces.



(c) Mixed Interfaces.

Figure 3.1: Example of Interface Behaviors.

3.4 Channel Assignment

Channel assignment decides which channels to assign for both static and dynamic interfaces. It is classified as follows:

- **Common**: the nodes may agree on using the same (common) channel set for all their static interfaces. For example, the i^{th} interface uses the i^{th} orthogonal channel. This common channel allocation is easy if the assignment is defined ahead of time [30].

- **Pseudo-Random**: each node assigns pseudo-randomly a set of channels to its static interfaces. A simple solution is to choose channels at random [111]. Another solution is to define a well-know function f of the node identifier to obtain the channel assigned to static interfaces [83]. Neighbors of node $v \in V$ use function f to compute the channel used by v.

- **Adaptive**: the nodes use some criteria (local or global) to dynamically adapt the set of channels used by its interface, such as: the time (pre-defined or random), the channel visiting order [78, 76], the interference level [74], or the available bandwidth [112].

3.5 Channel and Interface Assignment Strategies

A strategy is a combination of *Interface Assignment* (Section 3.3) and *Channel Assignment* (Section 3.4). Table 3.2 shows which combination of assignments forms a strategy. Besides, examples of related work are indicated for each strategy.

Static Interfaces/Common Channel Assignment

This strategy assigns a channel to each interface for permanent use (i.e., all interfaces are static). Besides, the common channel assignment is applied: the same channel is used for all i^{th} interfaces [30, 81].

Figure 3.2 shows an example of *Static Interfaces/Common Channel Assignment*. Each node has two interfaces. Four orthogonal channels are available. Static links are represented by bold lines. The channel assigned to each interface is shown inside the square brackets nearby the nodes. Note that any two neighbors have always multiple independent links to communicate with each other, resulting in a stable network topology without partitions. However, the network capacity decreases when the number of nodes increases as more contention and interference may occur [74].

The MUP protocol [81] and the previous drafts of IEEE 802.11s standard [39, 82] assume this strategy, as described in Section 2.3.4.2. Besides, the authors of the routing metrics ETT and WCETT adopt a *Static Interfaces/Common Channel Assignment* [30].

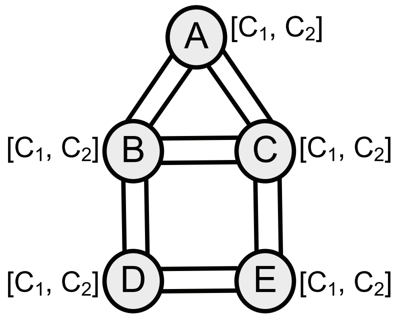


Figure 3.2: Example of *Static Interfaces/Common Channel Assignment.*

Static Interfaces/Pseudo-Random Channel Assignment

Similar to the previous strategy, the *Static Interfaces/Pseudo-Random Channel Assignment* strategy assigns a static channel for each interface. However, this assignment is independent between different nodes, as depicted in Figure 3.3. Consequently, this strategy does not guarantee connectivity. Two nodes may choose different channels for their interfaces leading thus to deafness. For example, nodes D and E have no common channel. To allow communication between them, the four-hops path $(D - B - A - C - E)$ has to be used instead of direct communication. As explained in Section 2.3.4.3, the DIMN protocol [86] considers this strategy in dual-interface mesh networks. The assignment is centralized at the gateway to ensure connectivity.

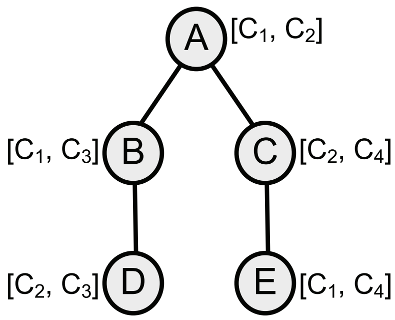


Figure 3.3: Example of *Static Interfaces/Pseudo-Random Channel Assignment.*

This strategy can also be modeled as a graph coloring problem [111]. The vertices of the graph represent the nodes and the "colors" correspond to channels. CLICA [73] is an example of graph coloring algorithm that results in a *Static Interfaces/Pseudo-Random Channel Assignment.* These algorithms would preserve the graph connectivity, but they usually require to know the topology in advance and/or to capture the interference variations.

Dynamic Interfaces/Adaptive Channel Assignment

In this strategy, all interfaces are dynamic. As illustrated in Figure 3.4, network topology continuously changes over time (e.g. t_1, t_2, t_3). We use dashed lines to indicate dynamic links. Often, nodes need to use a rendezvous mechanism to avoid deafness. For example, the nodes may adopt a schedule such that statistically a pair of node has common timeslots [76]. Nodes can also have their own *hopping sequence* schedules and adapt them according to the neighboring schedules like in SSCH [78] and McMAC [79]. *Split phase* protocols such as MMAC [26] and MAP [80] are another alternative solution to deal with dynamic interfaces. As time is divided into an alternate cycle of control and data exchange phases, nodes can negotiate the best channel to be used during the data exchange phase.

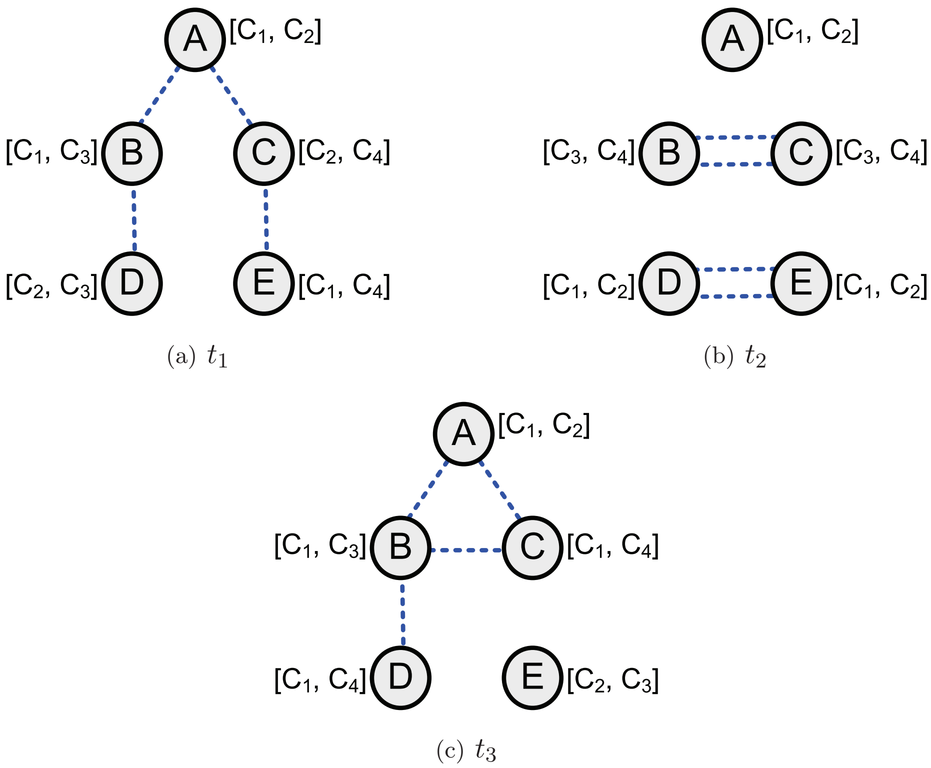


Figure 3.4: Example of *Dynamic Interfaces/Adaptive Channel Assignment.*

Mixed Interfaces/Common and Adaptive Channel Assignment

In this strategy, each node has static interfaces using a common channel assignment, while the dynamic interfaces act in an on-demand manner. As illustrated in

Figure 3.5, the static interface maintains global connectivity (i.e., bold lines) and the dynamic interface creates links over time (i.e., dotted lines).

For instance, one static interface may be tuned to a *dedicated control channel* to isolate control packets from data packets [113, 114, 115]. The dedicated control channel is used to reserve the channel that will be used further by a pair of dynamic interfaces for the data exchange. One advantage of this approach is that nodes can overhear all the agreements made on the control channel by other nodes and avoid busy channels. Besides that, the requirement of strict synchronization is relieved in comparison to the *Dynamic Interfaces/Adaptive Channel Assignment* strategy. However, the control channel can become a bottleneck if the number of nodes in the mesh network keeps increasing.

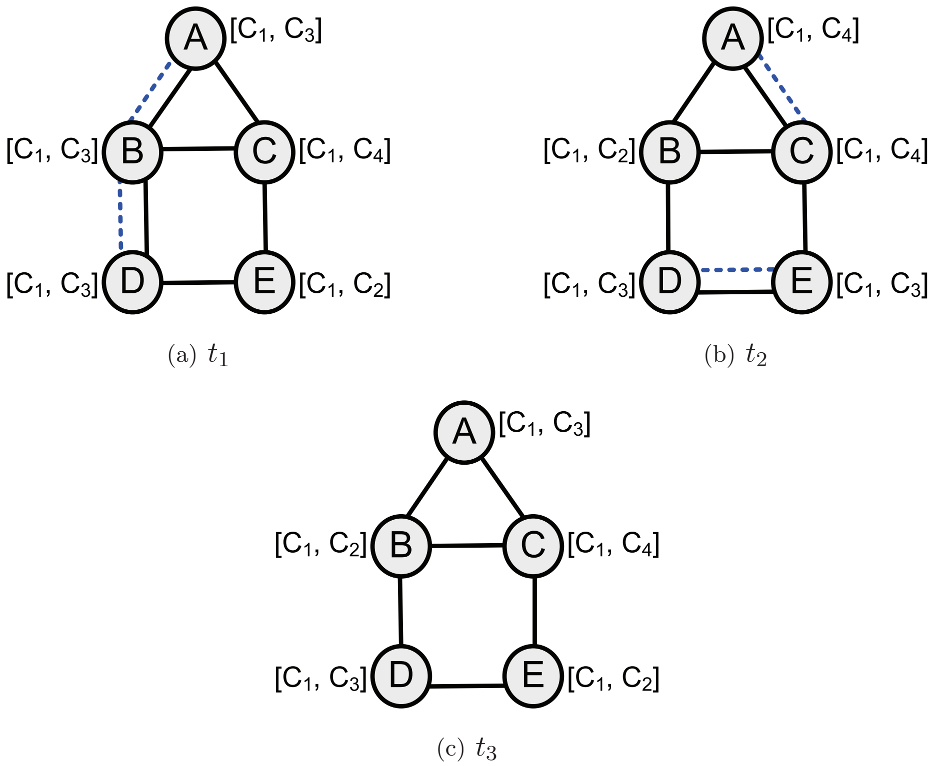


Figure 3.5: Example of *Mixed Interfaces/Common and Adaptive Channel Assignment*.

Mixed Interfaces/Pseudo-Random and Adaptive Channel Assignment

Different nodes assign their static interfaces to different channels, while the remaining interfaces switch channels in an adaptive manner. To find the optimal number of static and dynamic interfaces is a very complex task, leading often to a sub-optimal solution. Generally, two types of scenarios are found in the literature.

The first scenario is illustrated in Figure 3.6. It consists in using static interfaces for reception. Normally, a single static channel ($I_v^S = 1$) is adopted [83, 84]. In Figure 3.6, we use the first channel inside the square brackets to represent the static channel of each node. Note that the channel assigned to the static interface does not vary over time. A key issue is how to select the channel for the static interface. A simple solution is to select channels at random. On the other hand, PCU [83, 84] proposes a *channel usage protocol* to select the least used channel in the neighborhood. To send data, the transmitter node switches one of its dynamic interfaces to one of the static channel of the receiver node: no deafness occurs. This dynamic behavior is represented by dotted arrows in Figure 3.6. In Figure 3.6(a), nodes C and D switch their dynamic interface to channel C_3 to send data to node B. After some time (e.g., t_2 in Figure 3.6(b)), they switch their dynamic interfaces to send data to other neighbors: node D to node E, and node C to nodes A and E, both through C_1. Next (e.g., t_3 in Figure 3.6(c)), node C continues to send data to node A through channel C_1, while node D switches its dynamic interface to channel C_3 in order to send data to node B.

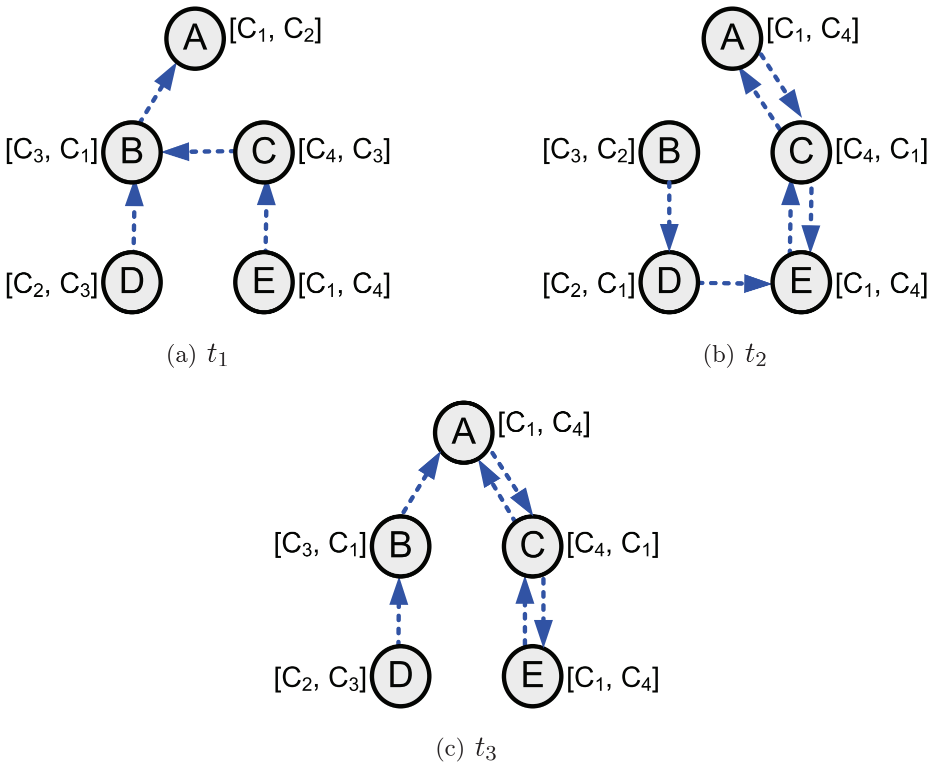


Figure 3.6: Example of *Mixed Interfaces/Pseudo-Random and Adaptive Channel Assignment.*

The second scenario consists in maximizing the network capacity with static interfaces, while dynamic interfaces work in an on-demand manner. In this case, the number of static interfaces is greater than that of dynamic interfaces ($I_v^S > I_v^D$) [85]. In this way, while static interfaces guarantee some degree of connectivity between nodes in the network, dynamic interfaces switch between channels frequently in order to relieve congested links.

Strategies	**Interface Assignment**			**Channel Assignment**			**Ref.**
	Static	Dynamic	Mixed	Common	Pseudo-Random	Adaptive	
Static/ Common	X			X			[30] [81] [39] [82]
Static/ Pseudo-Random	X				X		[73] [86] [111]
Dynamic/ Adaptive		X				X	[26] [78] [79] [80] [76]
Mixed/ Common and Adaptive			X	X		X	[113] [115] [114]
Mixed/ Pseudo-Random and Adaptive			X		X	X	[83] [84] [85]

Table 3.2: Channel and Interface Assignment (CIA) Strategies.

3.6 Performance Evaluation

We propose now to quantify the impact of the CIA strategies on the network performance. We will study in particular the following characteristics:

1. **Network connectivity**: the size of the largest connected component of the multi-channel graph;
2. **Density of connections**: the ratio of number of radio links that exist respectively between the single-channel and multi-channel network;
3. **Neighbor discovery**
   (a) **Probability of Rendezvous ($P(R)$)**: the probability of two neighbors selecting at least one common channel among the $\mathcal{C}$ available channels

Strategies	Density (%)
Single Channel	100
Static/Common	300
Static/Pseudo-Random	60
Dynamic/Adaptive	60
Mixed/Common and Adaptive	110
Mixed/ Pseudo-Random and Adaptive	200

Table 3.3: Density of Connections.

in any time slot τ. The event of interest is in this case:
"$R =$ *at least one channel in common*".

(b) **Expected Time to Rendezvous ($E[TR]$):** the expected time to achieve a rendezvous.

We present the results of our evaluation based on a probabilistic analysis, corroborated by simulations on MATLAB 7.0 [116]. We have implemented a custom-built simulator to evaluate the impact of CIA strategies on the network topology and the density of connections. We assume ideal PHY and MAC layers: no packet is lost and no collision occurs. We assume a fixed radio range leading to random Unit-Disk Graphs (UDG). The results are presented with a confidence interval of 95%.

Table 3.3 presents the percentage of maintained links in a network with 500 nodes. We show the average number of radio links: if two nodes have l common channels, we count l radio links. The measure is normalized according to the average number of neighbors in the single channel case. We consider an average density of 10 (number of neighbors in the single channel case) with nodes having 3 interfaces and 8 channels ($I_{v_1} + I_{v_2} < \mathcal{C}$). For *Mixed Interface Assignment*, we consider 1 static interface [26, 83]. We will discuss the results separately for each strategy.

Static Interface/Common Channel Assignment

Each node tunes its i^{th} interface to the i^{th} available channel. As shown in Table 3.1, $N_{links}(v_1, v_2)$ denotes the number of multi-channel links between nodes v_1 and v_2. One link exists for each channel (and transitively for each interface). For this strategy, we obtain:

$$N_{links}(v_1, v_2) = min(I^S_{v_1}, I^S_{v2}) \tag{3.6}$$

In other words, any pair of nodes that would be a neighbor in the single channel network would also be a neighbor in the multi-channel case leading to a connected network. Besides, the number of common radio links with one neighbor equals the minimum of the number of interfaces of both nodes. We can note that the density of connections is maximum in Table 3.3. The density of connections attains 300% because we consider that each node has 3 interfaces. Therefore, the efficiency of neighbor discovery is maximum: the neighbor will always be discovered ($P(R) = 1$) after only one single *hello*. Consequently, the expected time to achieve rendezvous is null ($E[TR] = 0$). If a node sends a *hello*, any neighbor will receive it *discovering* the corresponding source. Thus, the neighbor discovery process does not depend on the relation between the number of interfaces per node.

More contention and interference may occur on channels pre-defined in the *Common Channel Assignment* when the number of nodes increases [74]. In particular, when nodes have a different number of interfaces some pre-defined channels will be used by less nodes leading to less contention. Thus, this strategy does not fully distribute traffic uniformly over all channels.

We can notice that this strategy is optimal when the number of interfaces equals the number of available channels, which is seldom the case. On the contrary, the radio bandwidth is wasted if two nodes do not have the same number of interfaces. In particular, each node must have as many interfaces as at least one of its neighbors.

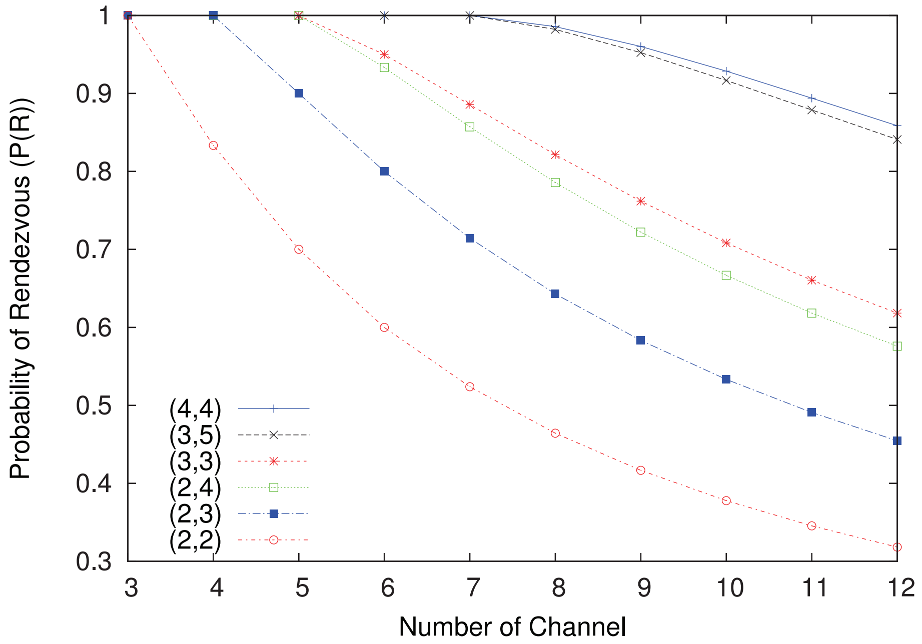
Static Interface/Pseudo-Random Channel Assignment

One way to reduce contention when all interfaces are static is to apply *Pseudo-Random Channel Assignment* to static interfaces. However, the network is more likely to be disconnected and with less available links than under *Static Interface/Common Channel Assignment* strategy. It presents the lowest density of connections among all strategies (c.f., Table 3.3). In the same way, the length of routes (number of hops) may increase.

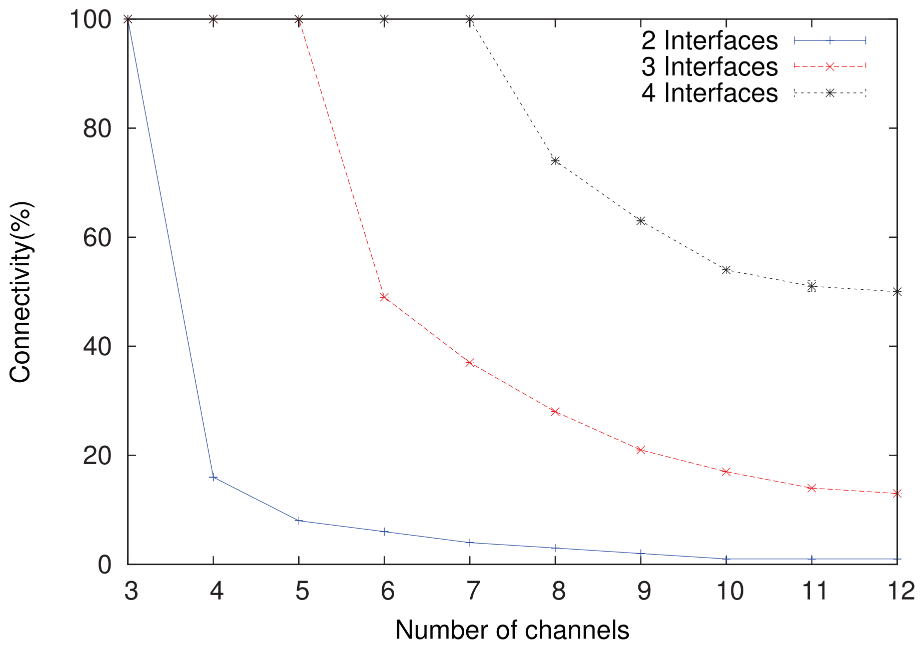
When the sum of interfaces of both nodes is strictly greater than the number of channels ($I^S_{v_1} + I^S_{v_2} > \mathcal{C}$), at least one channel is common to two neighbors. Thus, $P(R) = 1$ and $E[TR] = 0$.

Otherwise, if $I^S_{v_1} + I^S_{v_2} \leq \mathcal{C}$, $P(R)$ and $E[TR]$ depend on the relation between the number of interfaces per node and the number of available channels. It results in a combinatorial problem of unordered samples without replacement [117]. In this case, we can compute the rendezvous probability by means of Equation 3.7 when channels are selected at random:

$$P(R) = 1 - \frac{\binom{\mathcal{C}-I^S_{v_1}}{I^S_{v_2}}}{\binom{\mathcal{C}}{I^S_{v_2}}}. \tag{3.7}$$



(a) Probability of rendezvous of two neighbors in function of the number of available channels.



(b) Network connectivity in function of the number of available channels for a network of 300 nodes (average 10 single-channel neighbors)

Figure 3.7: Strategy *Static Interface/Pseudo-Random Channel Assignment*.

Figure 3.7(a) shows the rendezvous probability as the number of channels increases. The number of orthogonal channels varies from 3 (as in IEEE 802.11b/g) to 12 (as in IEEE 802.11a). To examine the effect of the number of interfaces per node, we plot six different cases in Figure 3.7(a). Each case corresponds to a pair of numbers representing the number of interfaces of any two neighbors v_1 and v_2: $(I^S_{v_1}, I^S_{v_2})$.

The results show that $P(R)$ depends more on the relation between the number of channels and the sum of interfaces than on the difference of the number of interfaces. When $I^S_{v_1} + I^S_{v_2} << \mathcal{C}$, the network is more likely to be disconnected. When it is sufficiently close to $\mathcal{C}$, the network is more likely to be connected. Note that the probability of a rendezvous is locally optimal when the radio interfaces are uniformly distributed between nodes. If a rendezvous occurs $E[TR] = 0$, otherwise $E[TR] = \infty$.

Figure 3.7(b) shows the impact of the number of channels on the network connectivity. Note that the network is connected when the sum of the number of interfaces is strictly superior to the number of channels (i.e., one channel at least is common between both nodes). However, connectivity quickly decreases when the difference between the number of channels and the number of interfaces increases. According to Figure 3.7(a), two nodes have a radio link with each other in 82% of the cases ($3 + 3$ interfaces, 8 channels). However, the global network connectivity is in this case only 28% (c.f., Figure 3.7(b)).

Dynamic Interface/Adaptive Channel Assignment

In this case, all interfaces are dynamic with *Adaptive Channel Assignment*. The main motivation is the use of all available channels to alleviate interference and channel congestion problems. However, the reassignment of dynamic interfaces in this strategy constantly alters the network topology and negatively impacts the network connectivity, as illustrated in Figure 3.4. Consequently, it presents the lowest connectivity similarly to *Static Interface/Pseudo-Random Channel Assignment* (c.f., Table 3.3). Each network topology generated over time by the *Dynamic Interface/Adaptive Channel Assignment* strategy can be seen as a particular network topology generated by the *Static Interface/Pseudo-Random Channel Assignment* strategy.

In this strategy, a rendezvous is important both for data exchange and neighbor discovery: two nodes can communicate if they have a channel in common. In particular, a pair of nodes with frequent rendezvous will be able to obtain a higher throughput.

The rendezvous probability depends on how channels are assigned, except when $I^D_{v_1} + I^D_{v_2} > \mathcal{C}$, which implies $P(R) = 1$ and $E[TR] = 0$.

If $I^D_{v_1} + I^D_{v_2} \leq \mathcal{C}$, we assume that each node randomly chooses its own hopping sequence. We obtain the following probability of a rendezvous:

$$P(R)_{k|t} = \sum_{p=k}^{t} \binom{t}{p} [1 - \frac{\binom{\mathcal{C}-I^D_{v_1}}{I^D_{v_2}}}{\binom{\mathcal{C}}{I^D_{v_2}}}]^p [\frac{\binom{\mathcal{C}-I^D_{v_1}}{I^D_{v_2}}}{\binom{\mathcal{C}}{I^D_{v_2}}}]^{t-p}, \tag{3.8}$$

where t is the number of channel switches done by dynamic interfaces and k the number of a successful rendezvous between v_1 and v_2.

This random process is a sequence of *Bernoulli trials* since it is a sequence of t independent repetitions [117]. The number of channel switches is relative and not absolute: we count the cumulative number of pairs of channels explored by the nodes. If node v_1 switches its channel at t and node v_2 at $t + \Delta_t$, we count 2 channel switches whereas if they change their channel at the same time, we count 1 channel switch. Thus, the channel switching process may or may not be synchronized.

If two nodes have k common slots, they will take on average $\frac{t}{2k}$ to discover each other:

$$E[TR] = \sum_{k \in [1..t]} \frac{t}{2k} P(R_{k|t}) + P(R_{0|t}) * \infty = \infty \tag{3.9}$$

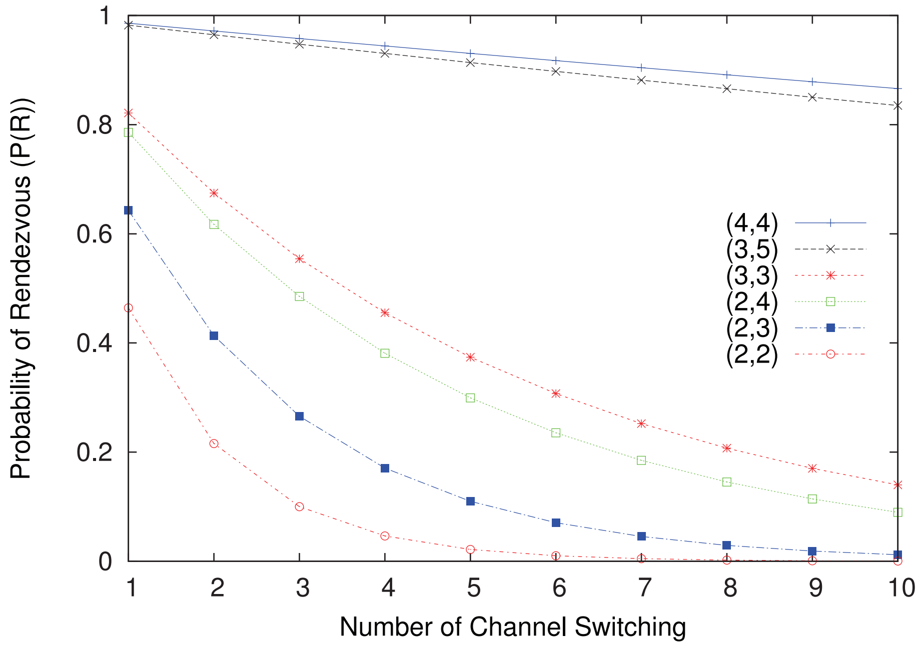
Figure 3.8(a) presents the probability that a pair of nodes is always connected after each channel switching (i.e. the radio link always exists). We consider 8 available channels. Clearly, the radio link is mostly intermittent when the number of channel switches increases and the number of interfaces decreases.

We have also represented the probability that a pair of nodes has at least one rendezvous after t switches (c.f., Figure. 3.8(b)): the nodes can communicate at least once during *channel scheduling*. As expected, $P(R)$ increases with the number of channel switches. In both cases, $P(R)$ presents a better result when radio interfaces are uniformly distributed between nodes.

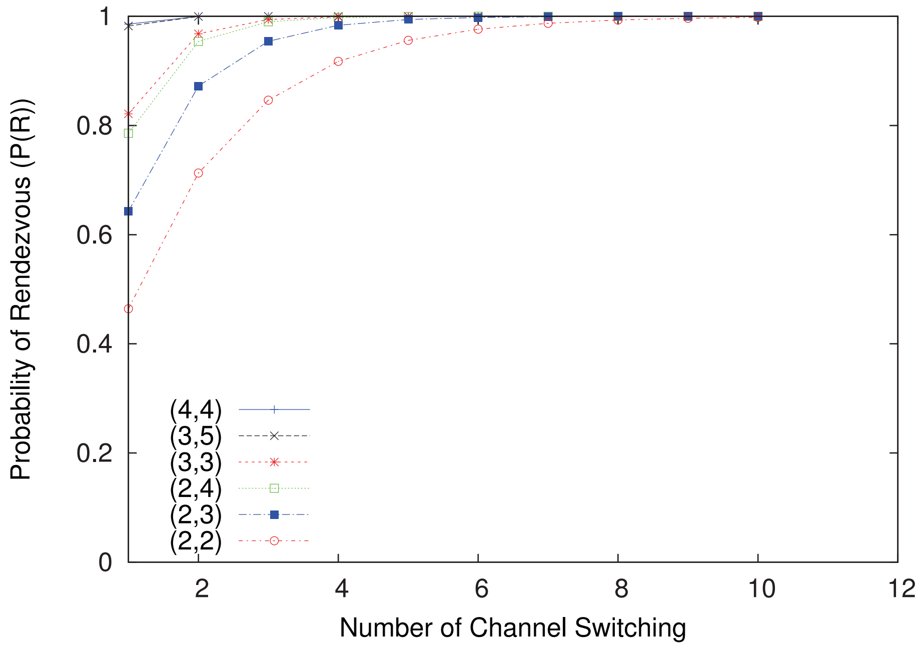
Network partitions may be avoided if nodes agree on deciding on which channel(s) exchange data. Nodes may publish their hopping sequences to make a future rendezvous easier [78] or reserve a predefined channel during at least one slot per sequence time. DaSilva *et al.* reduced the expected rendezvous time with pre-defined sequences [76]. Nevertheless, the scheme still requires a synchronization mechanism.

Mixed Interface/Common and Adaptive Channel Assignment

Strategies with *Mixed Interface Assignment* combine the advantages of the static and dynamic interfaces. In particular, rendezvous is simplified through static interface(s) while maintaining the flexibility coming from dynamic interface(s).



(a) Probability of two neighbors stay connected after each channel switching (i.e., all rendezvous are successful $k = t$).



(b) Probability of two neighbors have at least one rendezvous after t switches (i.e., one successful rendezvous $k = 1$).

Figure 3.8: Strategy *Dynamic Interface/Adaptive Channel Assignments.*

Because of the common control channel with a dedicated interface, the network is always globally connected even if it is not synchronized. In Table 3.3, 100% of the connections correspond to the static assignment (i.e., *Common Channel Assignment*). The remaining connections (10% on the average for 3 interfaces and 8 channels) arise from dynamic behavior (i.e., *Adaptive Channel Assignment*).

Likewise static interfaces in *Static Interface/Common Channel Assignment* strategy, common assignment preserves connectivity in this strategy. Thus, $P(R) = 1$ and $E[TR] = 0$. Static interfaces in *Mixed Interface/Common and Adaptive Channel Assignment* can be seen as a guaranteed way to make agreements for data exchange.

Mixed Interface/Pseudo-Random and Adaptive Assignment

This strategy also combines static and dynamic interfaces with the difference that static interfaces have *Pseudo-Random Channel Assignment*.

If a new node enters the network, it first assigns random channels to its static interfaces. Then, it scans all available channels through its dynamic interfaces. The transmitter has to use one of its dynamic interfaces to send its packets through one channel used by one of the static interfaces of the receiver. Since the dynamic interfaces are only used for transmissions, deafness never occurs. Therefore, a dynamic interface will meet all the neighboring static interfaces leading to $P(R) = 1$. However, despite the fact that network connectivity is guaranteed, the density of connections is reduced compared to *Static Interface/Common Channel Assignment* strategy (c.f., Table 3.3).

A node has to find with its dynamic interfaces one neighboring static interface. If both nodes have common static channels, they discover each other immediately. Otherwise, a node has to scan all non-static channels $(\mathcal{C} - I_v^S)$ and to stop as soon as it finds the first static interface. Let T_{sense} be the interval length during which each dynamic interface senses a channel. Thus, we obtain the average neighbor discovery time:

$$E[TR] = \lceil \frac{1}{2} * \frac{\mathcal{C} - I_{v_1}^S}{I_{v_1}^D * I_{v_2}^S} \rceil * T_{sense} * (1 - \frac{\binom{\mathcal{C} - I_{v_1}^S}{I_{v_2}^S}}{\binom{\mathcal{C}}{I_{v_2}^S}}) \tag{3.10}$$

where function $\lceil\rceil$ rounds to the upper integer. Therefore, the lower the number of channels $(\mathcal{C})$ to scan and the higher the number of dynamic interfaces (I_v^D) to scan, the lower is $E[TR]$.

If static interfaces are not used in the bootstrapping phase, only dynamic interfaces may discover each other. This clearly leads to the strategy *Dynamic Interface/Adaptive Channel Assignment* (Equation 3.8) in which all interfaces are dynamic.

3.7 Conclusion

In this chapter, we have focused on the connectivity problem in Multi-Channel Multi-Interface (MCMI) Wireless Mesh Networks (WMN). The analytical framework permits to quantify network connectivity and to study more finely the neighbor discovery process and its consequences on the WMN. Table 3.4 presents an overall summary of Channel and Interface Assignment (CIA) strategies for MCMI WMN. It provides the equations for the rendezvous probabilities: the network designer can select the most suitable solution to guarantee connectivity with a certain probability for a given network density and number of channels and interfaces. Furthermore, advantages and limitations of each strategy are presented.

Static approaches provide suitable stability for routing protocols without path changes, re-ordering, channel switches, etc. Once the CIA algorithm is performed, the interfaces remain on the assigned channels regardless of time. Thus, variations due to traffic load and interference are not considered to perform the CIA. In general, these two measures are used after the CIA in order to select the best channel to communicate with a neighbor [81]. The main advantages of static approaches are the ease of implementation and the fact that no strict synchronization is required. The main concern refers to the waste of channel resources: while some of the selected static channels may become overloaded, other channels remain idle.

Dynamic approaches have the ability to cover channels with few interfaces, thereby offering the potential to balance the load over different channels, to minimize interference, and to improve the capacity under heavy load. However, dynamic approaches may alter the network topology: some links are created while other may disappear. These changes can impact upper layers, especially routing protocols.

To combine the advantages of both approaches, a mixed solution can be applied. While connectivity is preserved with static interfaces, flexibility is achieved with dynamic interfaces. Anyway, static interfaces lead to the channel bottleneck problem. Furthermore, the way of performing dynamic channel assignment is still a challenge.

All things considered, a key issue is how to build an efficient mapping between all available channels and the interfaces at every mesh router when key design issues are addressed together: connectivity, interference, throughput, latency, stability, and fairness.

We believe that a promising approach consists in perform this mapping in an adaptive way. As nodes can be aware of network conditions (e.g., number of available channels and interfaces, traffic behavior, neighborhood, interference, QoS requirements, etc), they can profit from these information to choose the better

CIA strategy for a given situation. Also, if the network conditions change, nodes may choose to change the type of CIA strategy adopted in order to perform another CIA strategy more indicated for this new situation. For example, nodes may choose to adopt a *Static Interface/Common Channel Assignment* when the number of nodes and the network traffic are low. However, if the number of neighbors increases and exceeds a certain threshold, the nodes may choose to change from a *Static Interface/Common Channel Assignment* strategy to a *Mixed Interface/Common and Adaptive Channel Assignment* strategy to add flexibility to the network. Moreover, if the number of neighbors transmitting also increases, a *Dynamic Interface/Adaptive Channel Assignment* may be more appropriate in order to fairly schedule timeslots between neighbors. In this context, one research direction consists in determine the better CIA strategy for a given situation, as well as which network conditions/parameters have to be observed by the nodes to decide when to change from one strategy to another.

Strategy	Connectivity	Neighbor Discovery	Assets & Limits
Static/ Commom	Network is connected	$P(R) = 1$ $E[TR] = 0$	⊕ multiple links between any pair of nodes ⊖ more contention and interference on pre-defined channels ⊖ bandwidth is wasted if nodes have different number of interfaces
Static/ Pseudo-Random	Network can be partitioned if $I^S_{v_1} + I^S_{v_2} \leq \mathcal{C}$	$\triangleright$ if $I^S_{v_1} + I^S_{v_2} > \mathcal{C}$ $P(R) = 1$ $E[TR] = 0$ $\triangleright$ else $P(R)$: Equation 3.7 $E[TR] = \infty$	⊕ reduce contention and interference on predefined channels ⊖ lower connectivity ⊖ rendezvous is not always guaranteed due to deafness
Dynamic/ Adaptive	Network can be partitioned if $I^D_{v_1} + I^D_{v_2} \leq \mathcal{C}$	$\triangleright$ if $I^D_{v_1} + I^D_{v_2} > \mathcal{C}$ $P(R) = 1$ $E[TR] = 0$ $\triangleright$ else $P(R)$: Equation 3.8 $E[TR]$: Equation 3.9	⊕ all available channels are used ⊕ balances the load among channels ⊖ requires synchronization and a rendezvous if connectivity has to be guaranteed
Mixed/ Common and Adaptive	Network is connected	$P(R) = 1$ $E[TR] = 0$	⊕ all available channels are used ⊕ simplicity: common control channel for a rendezvous ⊖ contention on the control channel that will negatively impact performance
Mixed/ Pseudo-Random and Adaptive	Network is connected	$P(R) = 1$ $E[TR]$: Equation 3.10	⊕ all available channels are used ⊕ nodes can establish a radio link in a dynamic way ⊖ network capacity is not maximized

Table 3.4: Comparison of Channel and Interface Assignment (CIA) Strategies

Chapter 4

Broadcast Algorithms for Multi-Channel Multi-Interface Wireless Mesh Networks

Contents

4.1 Introduction

In this chapter, we propose to focus on the *broadcast problem* in Multi-Channel Multi-Interface (MCMI) Wireless Mesh Networks (WMN). The broadcast problem results from the fact that some higher layer protocols rely on layer-2 broadcasting [118]. For example, most routing protocols rely on broadcast to perform path discovery procedures (e.g., propagate PREQ messages in the network such as in HWMP [19]). Also, nodes usually rely on broadcasting to disseminate topological information (e.g., periodically *hello* packets to discover and maintain a list of neighbors).

In single-channel networks, all packets transmitted on a channel can potentially be received by all neighboring nodes listening to that channel. This capability is called *local broadcast*. A key challenge with the introduction of multiple channels and interfaces is to continue to provide efficient local broadcast [119]. Given that channel assignment protocols allow nodes to be tuned to different channels, broadcast packets transmitted in any channel are likely to reach only some of the nodes within the physical communication range. In other words, if the interfaces of the

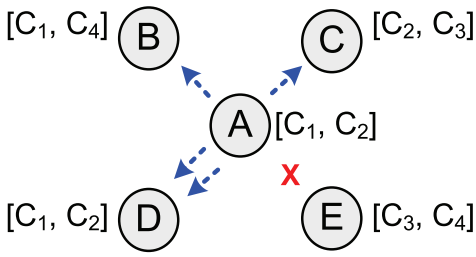


Figure 4.1: Broadcast Problem in Multi-Channel Multi-Interface Networks.

neighboring nodes are tuned to different channels, a single transmission on one channel cannot reach all the neighboring nodes simultaneously.

As a simple example, consider the scenario depicted in Figure 4.1, in which nodes are equipped with two interfaces. The channel assigned to each interface is shown inside the square brackets nearby the nodes. Assume that node A wants to perform a local broadcast. If A sends the broadcast message on its first interface (through channel C_1), only nodes B and D will be able to receive it, since they also have one interface tuned to channel C_1. On the other hand, if A decides to send the broadcast message on its second interface (through channel C_2), only node C and D will receive it. Also, if node A decides to perform broadcast on both interfaces, node D will receive duplicate copies of the broadcast message because it presents the same channel assignment as A. In all cases, note that node E is not *covered* by the broadcast messages sent by node A, since its interfaces are tuned to different channels than those of node A (i.e., nodes are neighbors in the Topology induced by the UDG, but not in the Network Topology).

In multi-channel multi-interface networks, the *local broadcast problem* is handled by explicitly transmitting a copy of the broadcast packet on all channels [120]. However, this broadcast scheme incurs high overhead when compared to single-channel networks. Besides, as the number of interfaces per node is usually much smaller than that of orthogonal channels, broadcast packets will be sent at slightly different times on some channels because of interface switching through all channels. An increase in the overall delay is also expected if the interfaces have to switch among a long range of channels (e.g., IEEE 802.11a offers twelve orthogonal channels) [121].

Another solution for the local broadcast problem in MCMI WMN is through periodic rendezvous on a common channel dedicated to broadcasting. This broadcast scheme does not increase the cost of broadcast when compared to single-channel networks. Nevertheless, it increases the transmission delay of broadcast packets. Besides, if the broadcast load is high, the common channel can become a bottleneck, even while there are plenty of other channels free. The drawbacks of both schemes are a call to arms for research on broadcasting solutions in MCMI scenarios.

To the best of our knowledge, the local broadcast problem has not been addressed in depth for MCMI WMN. Qadir *et al.* [27, 122] focus mainly on the Minimum Latency Broadcasting (MLB) in multi-rate networks. In particular, the authors proposed heuristic algorithms that transmit the same packet more than once, but at different transmission rate (i.e., different latencies) in order to achieve different subset of neighbors and form a broadcast tree. The concept of broadcast tree aims to reduce the impact of broadcast flooding in the network rather than solve the local broadcast problem. Han *et al.* [123] present a jointly channel assignment and routing algorithm to build a broadcast tree according to the number of interfaces of each node. Following a similar approach, Chiu *et al.* [124, 125] also propose a channel assignment and routing solution to construct several broadcast trees in MCMI WMN. Xing *et al.* [120, 121] propose channel assignment algorithms based on superimposed codes. In summary, the algorithms compute a matrix that allows a node to locate a common channel shared with its one-hop neighbors to perform broadcast. However, the authors do not consider the problem of minimizing broadcast redundancy in multiples interfaces scenarios.

Note that none of these few studies in the area of *multiple interface broadcasting* is sufficiently generic to deal with different channel assignment strategies. Besides, compute the channel assignment based on broadcast issues can substantially affect the overall network performance. For instance, a channel assignment that performs well for broadcasting does not necessarily perform well for unicast communication [27].

In this context, we present broadcast algorithms that fit any of the Channel and Interface Assignment (CIA) strategies described in Section 3.5. The proposed algorithms guarantee that a packet is delivered with a minimum probability to all neighbors. Simulation results show that the proposed algorithms efficiently limit the overhead.

4.2 Probabilistic Delivery Guarantee

When a node transmits a packet, it cannot be certain that all neighbors will receive it. Packet losses may arise due to effects at PHY and MAC layers, such as attenuation, interference, fading, multipath propagation, synchronization errors, and collisions. We propose to implement broadcast algorithms hiding this complexity, but still guaranteeing with a certain probability that each neighbor receives the broadcast packet.

We denote by p_b the bit error probability and by p_p the packet error probability.

Parameter	Definition
p_b	Bit error probability
p_p	Packet error probability
p_{deliv}	Probability of successful packet delivery
$p_{cover}(v_1 \to v_2)$	Probability that v_2 is covered by the broadcast of v_1
$p_{cover_{min}}$	Minimum probability tolerated for p_{cover}
$p_{p_{max}}$	Maximum probability tolerated for p_p

Table 4.1: Parameters of the Probabilistic Delivery Guarantee Model.

They are related by the following relation:

$$p_p = 1 - (1 - p_b)^S, \tag{4.1}$$

where $(1 - p_b)$ denotes the probability that each bit is correct and S denotes the size in bits of a packet.

Given p_p, we can calculate p_{deliv} as the probability of successful packet delivery as follows:

$$p_{deliv} = 1 - p_p. \tag{4.2}$$

Since this value depends on a given radio link, we use the notation $p_{deliv}(v_1, v_2)$ for the transmission from node v_1 to node v_2.

We consider that a local broadcast is successful if all the neighbors are *covered*. In turn, a neighbor is covered by a broadcast if it receives at least one copy of the corresponding packet with a probability superior or equal to $p_{cover_{min}}$. The higher layers may specify this *reliability threshold* when they want to transmit a broadcast packet, such that $p_{cover_{min}} > 0$.

Let $N(v)$ represent the neighbors of v. We denote by $p_{cover}(v_1 \to v_2)$ the probability that node v_2 correctly receives the broadcast of node v_1 when the broadcast algorithm has terminated its operation (i.e., v_2 is *covered*). Formally, the protocol implies that:

$$\forall v_2 \in N(v_1), p_{cover}(v_1 \to v_2) \geq p_{cover_{min}} \tag{4.3}$$

To provide guarantees, we limit the links to those with packet error probability of at least $p_{p_{max}}$. Thus, when a node has $p_p > p_{p_{max}}$, the other extremity is not considered a neighbor. This allows a node to maintain radio links of good quality.

Table 4.1 presents the parameters of the probabilistic delivery guarantee and their corresponding definition.

4.3 Broadcast Algorithms

In this section, we introduce the broadcast algorithms based on the Channel and Interface Assignment (CIA) framework described in Section 3.5.

Static Interfaces with Common Channel Assignment

With the *Common Channel Assignment*, the i^{th} channel is assigned to the i^{th} interface [25]. Thus, broadcast can be implemented simply: a node has just to broadcast a packet through any of its static interfaces and all its neighbors will receive it. In other words, no deafness arises.

A node has to send as many copies of the packet as required to cover each of its neighbors with the expected probability. If we consider packet losses uncorrelated among the different copies, the probability the node v_2 receives at least one of the k copies from v_1 is:

$$p_{cover}(v_1 \to v_2) = 1 - (1 - p_{deliv}(v_1, v_2))^k \tag{4.4}$$

Indeed, the probability that the broadcast algorithm performs successfully (i.e. one copy at least is received) is 1 minus the probability that all the copies are dropped.

Finally, a node v_1 has to send the following number of copies so that v_2 receives the packet with a probability superior to $p_{cover_{min}}$:

$$k = \left\lceil \frac{log(1 - p_{cover_{min}})}{log(1 - p_{deliv}(v_1, v_2))} \right\rceil \tag{4.5}$$

where . The link with the smallest p_{deliv} will determine the lower bound of the number of copies to transmit.

When a single static interface is available, this interface can be used to send broadcast packets. However, the whole control traffic is concentrated on the control channel thus leading to its high utilization for large broadcast load.

This approach can be applied to CIA strategies that use the *Common Channel Assignment*:

– *Static Interfaces/Common Channel Assignment* and;

– *Mixed Interface/Common and Adaptive Channel Assignment.*

Static Interfaces with Pseudo-Random Channel Assignment

With *Pseudo-Random Channel Assignment*, a single transmission is not sufficient for local broadcast because not all neighbors use the same channel, as illustrated in the example of Figure 4.1. A node may have to send several packets so that all its neighbors become covered through different channels. In this strategy, each node knows the list of its neighbors and their static channels as a feature of

Algorithm 1 Greedy Selection for *Static Interfaces* with *Pseudo-Random Channel Assignment*

```
/* Input: list of neighbors, their corresponding static channels, and
   their corresponding p_deliv                                          */
/* neighs ≡ {(neighbor, channel)}                                       */
1  neighs [] ← getListNeighStaticIntf()
   /* Initially, no neighbor is considered as covered                   */
2  for i ∈ [0..|neighs|] do
3  |  p_cover [i] ← 0
   /* Execute while at least one uncovered neighbor exists              */
4  while (∃v ∈ neighs such that p_cover[v] < p_cover_min) do
   |  /* Count the number of uncovered neighbors per channel            */
5  |  for c ∈ [0..|channels|] do
6  |  |  nbCovered [c] ← nbNeighsCovered(c)
   |  /* Select the channel that cover the largest number of uncovered
   |     neighbors                                                      */
7  |  bestChannel ←getChannelMaxNbUncoveredNeighs(nbCovered)
   |  /* Update p_cover for each newly covered neighbor                 */
8  |  for v ∈ neighUsing(bestChannel) do
9  |  |  if p_cover [v] = 0 then
10 |  |  |  p_cover [v] ← p_deliv [v]
11 |  |  else
12 |  |  |  p_cover [v] ← 1 − (1 − p_cover[v]) · (1 − p_deliv[v])
   |  /* Send one broadcast packet                                      */
13 |  sendBroadcast(bestChannel)
/* Output: all neighbors covered with the smallest number of broadcast
   replications                                                         */
```

the unicast protocol. Besides, each node knows the p_{deliv} of its neighbors. A node will also use this information for its broadcast transmissions.

We propose a greedy approach inspired by multipoint relays [126]. A node chooses the minimum number of channels that cover the largest number of neighbors. As represented in Algorithm 1, a node proceeds in the following way:

- a node constructs the list of its neighbors (i.e. all the nodes with which it has a common channel) and their corresponding static channels (Algorithm 1, line 1);

- initially, a node considers that all its neighbors are *uncovered* (Algorithm 1, lines $2-3$);

- while at least one neighbor is covered with a probability inferior to $p_{cover_{min}}$, the node continues to replicate broadcast as follows:
  - it counts the number of uncovered neighbors for each channel (Algorithm 1, lines $5-6$);
  - it chooses the best channel to replicate the broadcast (Algorithm 1, line 7). Note that the best channel corresponds to the channel that cover the largest number of uncovered neighbors. If more than one best channel is detected, a random function can be used to select one of them (e.g., to balance the load among channels);
  - for each neighbor reachable through this channel, it updates the corresponding p_{cover} (Algorithm 1, lines $8-12$). It corresponds to the probability of delivery for the link (v_1, v_2) if v_1 did not yet schedule a packet for v_2. Else, it applies recursively Equation 4.6:

$$p_{cover}(v_1, v_2) = 1 - (1 - p_{cover}(v_1, v_2))\,(1 - p_{deliv}(v_1, v_2))\,, \tag{4.6}$$

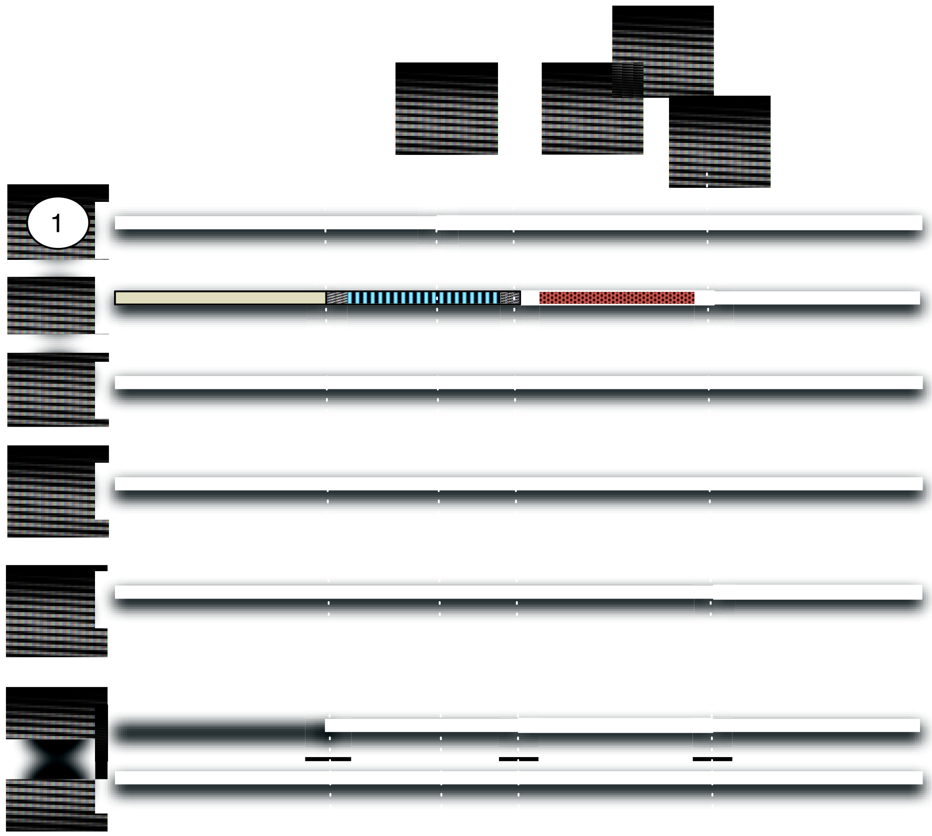
    where $(1 - p_{cover}(v_1, v_2))$ is associated to the previous broadcast replications and $(1 - p_{deliv}(v_1, v_2))$ is associated to the new broadcast replication.
  - it sends the broadcast packet on the selected channel (Algorithm 1, line 13);

Algorithm 1 can be applied to strategies that use static interfaces to receive packets:
– *Static Interface/Pseudo-Random Channel Assignment* and;
– *Mixed Interface/Pseudo-Random and Adaptive Assignment.*

Dynamic Interfaces with Adaptive Channel Assignment

When a node only uses dynamic interfaces, it needs to avoid deafness by correctly choosing a schedule of timeslot and channel/interface.

As illustrated in Figure 4.2, our solution aims to be flexible enough to allow nodes to switch between channels at different times. In the example of Figure 4.2, timeslots are delimited by vertical dashed lines. We consider four nodes: node v_1 with 3 interfaces, node v_2 and v_3 with one interface, and v_4 with 2 interfaces. The horizontal color bars represent different channels. If we take the first node v_1 as an example (i.e., the first three bars on the horizontal represents its three interfaces), note that its first interface performs a single switch at the end of timeslot 2, its second interface performs multiple switches, while its third interface stays on the


1

Algorithm 2 Greedy Selection for *Dynamic Interfaces* with *Adaptive Channel Assignment*

```
/* Input: list of neighbors, the channel used by each neighbor during each
   timeslot, and their corresponding p_deliv                               */
/* neighs ≡ {(neighbor)}                                                   */
/* schedule ≡ {(neighbor, channel, t_start, t_end)}                        */
1  neighs [] ← getListNeigh()
2  schedule [] ← constructSchedule()
   /* Initially, no neighbor is considered as covered                      */
3  for i ∈ [0..|neighs|] do
4  |   p_cover [i] ← 0
   /* Execute while at least one uncovered neighbor exists                 */
5  while (∃v ∈ neighs such that p_cover[v] < p_cover_min) do
   |   /* Count the number of uncovered neighbors per timeslot and channel */
6  |   for t ∈ [1..|timeslots|] do
7  |   |   for c ∈ [0..|channels|] do
8  |   |   |   nbCovered [t][c] ← nbNeighsCovered(t,c)
   |   /* Select the timeslot and channel that covers the largest number of
   |      uncovered neighbors                                              */
9  |   bestTimeslotChannel ←getTimeslotChannelMaxNbUncoveredNeighs(nbCovered)
   |   /* Update p_cover for each newly covered neighbor                   */
10 |   for v ∈ neighDuring(bestTimeslotChannel) do
11 |   |   if p_cover [v] = 0 then
12 |   |   |   p_cover [v] ← p_deliv [v]
13 |   |   else
14 |   |   |   p_cover [v] ← 1 − (1 − p_cover[v]) · (1 − p_deliv[v])
   |   /* Send one broadcast packet                                        */
15 |   sendBroadcast(bestTimeslotChannel)
/* Output: all neighbors covered with the smallest number of timeslots     */
```

number of neighbors that can be covered for each timeslot for each channel (Algorithm 2, lines $6-8$). Thus, it will re-iterate by greedily choosing pairs <*timeslot, channel*> that cover the largest number of not yet covered neighbors (Algorithm 2, line 9). For each neighbor reachable through this timeslot and channel, it updates the corresponding p_{cover} (Algorithm 2, lines $10-14$), adopting the same approach as Algorithm 1 (lines $8-12$). The algorithm stops when all the neighbors are covered with a probability superior to $p_{cover_{min}}$.

In the example of Figure 4.2, a neighbor is considered covered if it received at

Parameter	**Value**
Number of nodes	200
Density (average number of neighbors)	10
Number of interfaces	3
Number of available channels (e.g., IEEE 802.11a)	12
$p_{cover_{min}}$	0.95
$p_{p_{max}}$	0.5

Table 4.2: Default Values for Broadcast Simulations.

least one copy. As explained previously, v_1 first computes timeslots (dashed lines). Then, it chooses the neighbors reachable through each interface for each timeslot and applies the greedy algorithm. For instance, node v_1 can reach node v_3 during the first timeslot through its first interface and node v_4 through the second interface of v_4. Finally, node v_1 may choose timeslot 1 via its first interface to cover v_3, v_4 and timeslot 1 via its third interface to reach node v_2.

This algorithm can be applied to the strategy that only uses dynamic interfaces: – *Dynamic Interface/Adaptive Channel Assignment*.

4.4 Performance Evaluation

We have implemented a simulator to evaluate the performance of the proposed broadcast algorithms. We generate random Unit-Disk Graphs (UDG) and plot 95% confidence intervals. Table 4.2 presents the default values used in simulations. We measure two features:

1. **Overhead**: it is defined as the average number of transmissions required by a node to cover all its neighbors;

2. **Fairness**: we measure the fairness of the load for all the channels. We use the Jain Index [127] to measure fairness. Let B_c denotes the bandwidth consumed by the broadcast on channel c. The Jain Index is calculated as follows:

$$JainIndex = \frac{\left(\sum_{c=1}^{C} B_c\right)^2}{C \cdot \sum_{c=1}^{C} B_c^2} \tag{4.7}$$

   The Jain Index ranges from $\frac{1}{C}$ (worst case) to 1 (best case), and it is maximum when all channels receive the same allocation.

We adopted the Packet Error Rate (PER) model presented by Camp *et al.* [128]. Figure 4.3 presents the model. For short radio links (first part, on the left), the

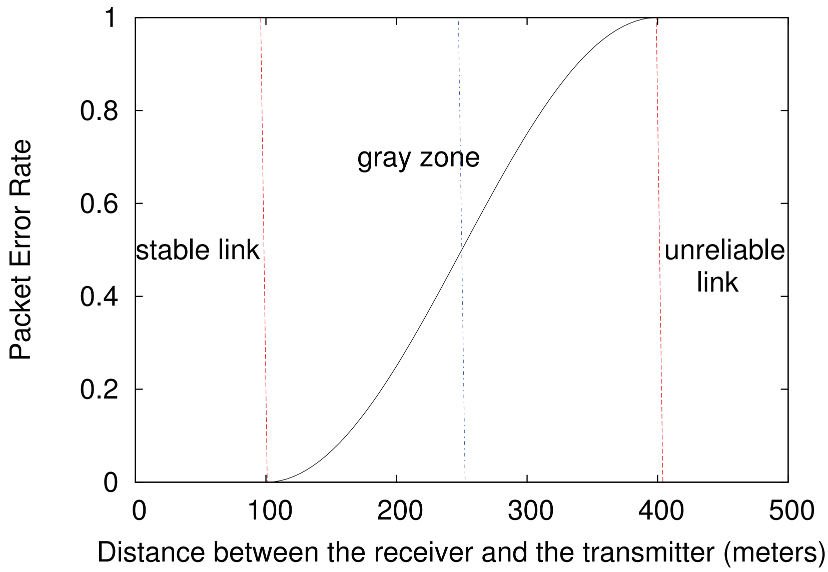


Figure 4.3: Packet Error Rate in Function of the Distance Between the Transmitter and the Receiver Nodes.

reliability is almost perfect. For the *gray zone*, the PER depends on the distance. For very large distances, the packet error rate becomes almost 1: the radio link does not exist at all. As explained in Section 4.2, the neighbors with a packet error probability (i.e., PER) superior to $p_{p_{max}}$ have not to be covered. For the numerical results, we have chosen the value of $p_{p_{max}} = 0.5$ (i.e., the gray zone up to the blue vertical line in Figure 4.3), although different values would lead to consistently the same results. A larger threshold would just increase the overhead to *cover* unreliable neighbors.

We denote each strategy as introduced in Section 3.5 and apply the broadcast algorithms defined in the previous section. In particular, the *Mixed Interfaces/Common and Adaptive Channel Assignment* strategy was implemented with a single static interface [114, 115], which is used as dedicated control channel to isolate control packets from data packets. The *Mixed Interfaces/Pseudo-Random and Adaptive Channel Assignment* was implemented with a single static interface that is used by each node to receive data from its neighbors [83, 84], such as illustrated in Figure 3.6. In relation to the *Dynamic Interfaces/Adaptive Channel Assignment* strategy, it was implemented in a way that each interface equally shares its time among all the channels following a pseudo-random sequence, as in the Slotted Seeded Channel Hopping (SSCH) solution [78] (Section 2.3.4.1). Two nodes are able to exchange packets if at least one pair of interfaces uses the same channel at the same instant.

Impact of the Number of Nodes

Figure 4.4(a) presents the overhead in function of the number of nodes when maintaining constant density (i.e., 10 neighbors on average).

The *Static/Common* and *Mixed/Common and Adaptive* strategies have the same minimal overhead. Due to the *Common Channel Assignment*, no deafness arises and all neighbors receive the broadcast transmission. Since some neighbors may present a non-null PER, several broadcasts are required before considering they are *covered*.

The *Static/Pseudo-Random* strategy requires a little less broadcast packets ($\cong 8$ transmissions) than the *Dynamic/Adaptive* strategy ($\cong 11$ transmissions). Indeed, the *Static/Pseudo-Random* strategy offers a smaller connectivity. Two nodes may be in the radio range of each other, but may not share a common static channel. In this case, this "virtual neighbor" is not anymore a neighbor in the multi-channel topology and has not to be covered. This reduces mechanically the overhead. The probability of such configuration is smaller with dynamic interfaces because we increase the probability a pair of nodes has at least one channel in common at a given instant.

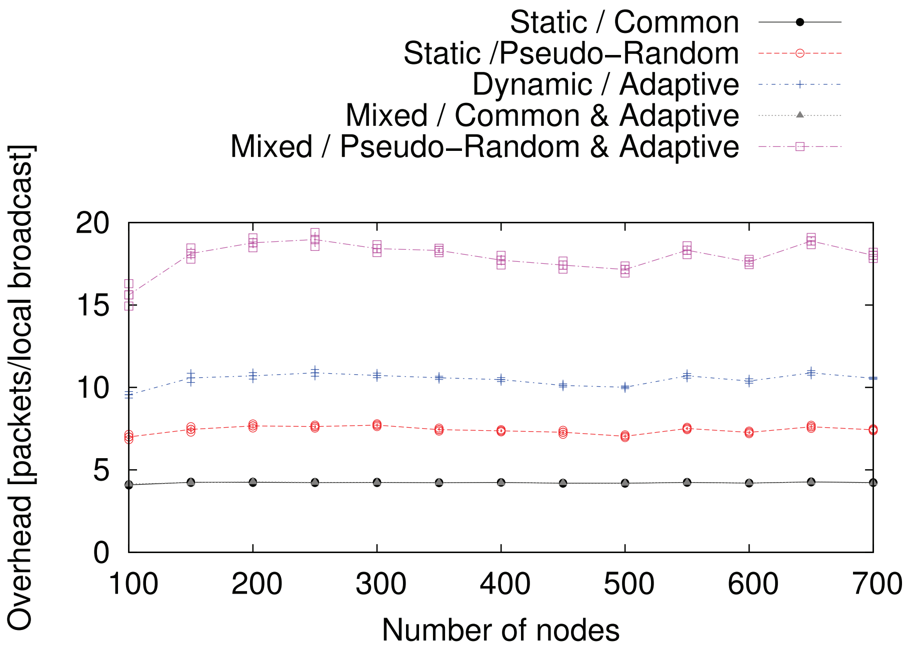
Finally, the *Mixed/Pseudo-Random and Adaptive* strategy presents the worst overhead, because it uses only one static interface, reducing the possibilities to re-use one single transmission to cover several neighbors. Thus, if we increase the number of static interfaces the overhead tends to decrease.

We have also evaluated fairness between different channels with the Jain Index. Figure 4.4(b) shows that the *Mixed/Common and Adaptive* strategy results in the Jain index of about 0.05. Indeed, only the control channel (1 of the 12 available channels) is used for broadcast [113, 114], leading to an high unfairness. Other strategies lead to almost perfect fairness, since they spread efficiently the broadcast traffic through orthogonal channels, reducing the risk of congestion.

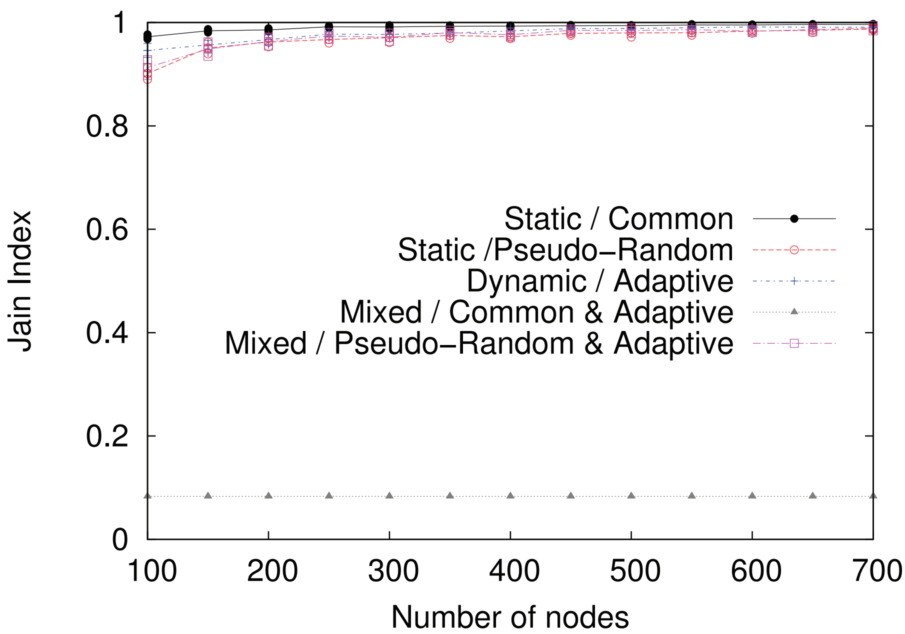
Density

Figure 4.5 shows the impact of the density on the overhead while maintaining the number of nodes constant. Only *Static/Common* and *Mixed/Common and Adaptive* strategies have the same overhead, which is perfectly scalable, because they have a *Common Channel Assignment*.

The overhead created by Algorithm 1 applied to the *Static/Pseudo-Random* slightly increases with the density: the greedy approach succeeds to schedule the transmissions. This growth is more important when we use dynamic interfaces, as more timeslots are necessary to cover the interface schedule of new neighbors.



(a) Overhead



(b) Jain Index

Figure 4.4: Impact of the Number of Nodes on the Broadcast Overhead and the Fairness.

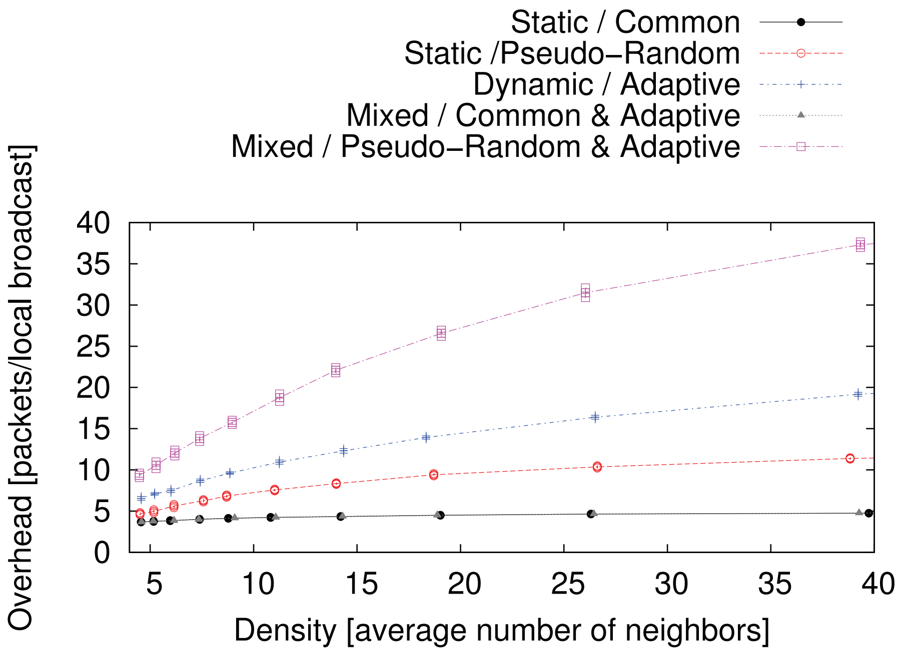


Figure 4.5: Impact of Density on the Broadcast Overhead.

The *Mixed/Pseudo-Random and Adaptive* strategy keeps on presenting the worst overhead since only one static interface is used for reception, limiting the possibilities to use one single packet to cover several neighbors.

In conclusion, our greedy strategies are particularly efficient to minimize the overhead when the density is large.

Number of Interfaces

Figure 4.6 shows the influence of the number of interfaces on the overhead.

The *Dynamic/Adaptive* and the *Static/Pseudo-Random* strategies tend to have initially a growing overhead because the number of neighbors to be covered increases since they have more chances to have a common timeslot. Then, the overhead decreases when it exceeds a threshold since the probability of having different neighbors that use the same channel increases with the number of interfaces. The *Dynamic/Adaptive* strategy begins to be more attractive when the number of interfaces is large compared to the number of channels (e.g., greater than 3 interfaces). Finally, for a very large number of interfaces (e.g., greater than 8 interfaces), these strategies tend to be similar to the common channel strategies.

The number of interfaces does not impact the strategies using a common channel for broadcast. Besides, the *Mixed/Pseudo-Random and Adaptive* strategy presents

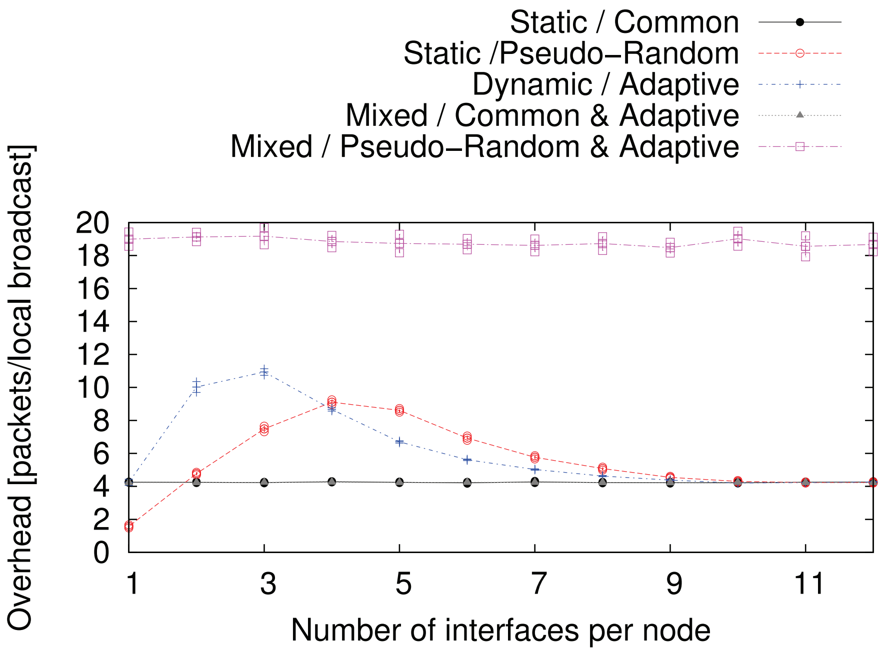


Figure 4.6: Impact of the Number of Interfaces on the Broadcast Overhead.

also a constant overhead since the unique receiving interface keeps on constituting the bottleneck.

Impact of the Reliability Threshold $p_{cover_{min}}$

Finally, Figure 4.7 presents the overhead in function of the reliability threshold $p_{cover_{min}}$. For $p_{cover_{min}} = 0.5$, note that the two strategies that adopt the *Common Channel Assignment* are able to cover all the neighbors with a single broadcast. Thus, the increase of overhead for the other strategies is not related to the quality of the link, but rather to partitions resulting from the channel assignments, that impose the broadcast replication on more than one channel to ensure that all neighbors will be covered.

However, when $p_{cover_{min}}$ increases, the overhead becomes larger for the all strategies. This is due to the fact that neighbors with a large PER may require the transmission of several copies. Nevertheless, we can remark that all the strategies follow the same tendency. The overhead becomes prohibitive when we require a very large $p_{cover_{min}}$ (e.g., ≈ 0.99). Thus, the network protocols have to cope with inconsistencies to limit the overhead. They should work in a self-stabilizing manner. In other words, even if some neighbors do not receive a particular broadcast packet, the protocol must work properly.

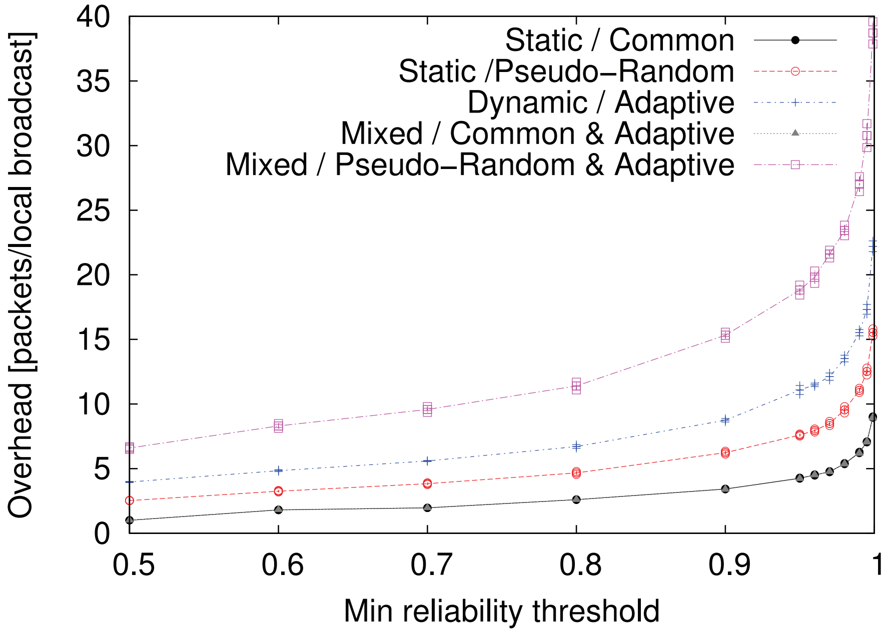


Figure 4.7: Impact of the Reliability Metric $p_{cover_{min}}$ on the Broadcast Overhead.

4.5 Conclusion

We have proposed algorithms to implement broadcast under any multi-channel multi-interface strategy. In particular, they can cope with dynamic interfaces without a common control channel. To the best of our knowledge, these algorithms are the first ones to cope with deafness in this situation. Simulations show that all the strategies have an acceptable overhead and the load is fairly distributed among channels when the *Common Channel Assignment* is not used. A greedy approach is particularly efficient to take benefit of the broadcast nature of transmissions, computing efficient schedules to cover all the neighbors.

To improve the performance of the proposed algorithms, one interesting direction of work consists in optimize the delay. When dynamic interfaces are used, a feasible solution consists in extend our algorithm in order to identify the timeslots that present the best trade-off between delay and overhead. Also, it would be interesting to develop new solutions to perform the channel and interface assignment in conjuction with the broadcast problem in order to improve the overall performance.

Chapter 5

Capacity of Multi-Channel Multi-Interface IEEE 802.11s Wireless Mesh Networks

Contents

5.1 Introduction

In Chapters 3 and 4, we have presented a comparison and performance evaluation of different Channel and Interface Assignment (CIA) strategies addressing connectivity, network topology, neighbor discovery, and broadcast issues. Our results demonstrated how to *improve* the MCMI WMN pointing out advantages and limitations of each strategy. In this chapter, our goal is to study how to *optimize* the overall network and seeks the limits of the throughput performance. In particular, we focus on the relation between the network capacity and the channel and interface assignment. The knowledge of bounds allows a better understanding of the existing solutions and thus to optimize the design of new algorithms. We consider this issue in the context of IEEE 802.11s mesh networking [19].

Gupta and Kumar [21] pioneered the analysis of capacity in wireless networks. Their main contribution consists of the observation that capacity decreases when the number of nodes increases. Their results are applicable to single-channel wireless networks, although they are extensible to multi-channel wireless networks where every node has a dedicated interface per channel [29]. Zemlianov and De Veciana [129] extended the analysis to deal with infrastructure-based wireless networks (i.e., some nodes are interconnected with each other by high capacity wired links). This type of analysis mainly considers asymptotic behavior and does not sufficiently take into account MAC and routing protocols.

Kyasanur and Vaidya [29] have theoretically shown how network capacity scales with the number of nodes, channels, and interfaces. Franklin *et al.* [130] considered Time Division Multiple Access (TDMA) based WMN. They adopted a queuing theory approach, but a comparison of different solutions is not achievable through such a formulation. CLICA [73] (Section 2.3.4.3) proposes an Integer Linear Programming (ILP) formulation to obtain a lower bound of the capacity. The authors derived greedy algorithms to assign channels to interfaces. This formulation is limited to multi-interface scenarios in which the Topology induced by the UDG must be preserved (i.e., all links in the Topology induced by the UDG must be present in the Network Topology). Kodialam and Nandagopal [69] also proposed an ILP formulation. However, they aim to obtain an upper bound of the capacity. The authors formulate a joint routing, channel assignment and scheduling problem in multi-interface mesh networks into a classic network flow problem called *multi-commodity flow problem* (i.e., multiple flow demands exist between different sources and destination nodes) [131]. This centralized approach assumes that the traffic rate of each node is known and that these traffic rates are constant.

In relation to these related work, our proposal here aims at comparing different channel and interface assignment approaches and at quantifying the impact on capacity of these design choices. Three Mixed Integer Linear Programming (MILP) formulations are proposed to extract the network capacity in MCMI WMN. In particular, our formulations model routing and bandwidth sharing constraints in presence of interference. We consider the objective function that maximizes throughput while maintaining fairness. The MILP formulations are independent on the practical assignment strategies and thus permit to compare quantitatively any assignment algorithm. Besides, we use both an ideal MAC and a TDMA solution to model the MAC layer. Another distinguishing feature of our work is the focus on the recent mesh standard IEEE 802.11s, especially on the MAC layer and routing. We propose to investigate in this chapter the choice of a suitable strategy for IEEE 802.11s.

We present numerical results that demonstrate the benefits of our three formulations in relation to a *Common Channel Assignment*. To validate the relevance of such approaches, we have extended the performance evaluation of the proposed MILP formulations to include extensive simulations. Our simulations take into account aspects neglected in the formulations such as realistic MAC layer, routing, and traffic load. In other words, even with realistic conditions and real protocols, do the results obtained numerically keep on holding?

5.2 Network Model

We consider an IEEE 802.11-based multi-hop multi-channel multi-interface Wireless Mesh Network. We model the network as a graph $G(V, E)$, in which V is the set of vertices and E is the set of edges. Each vertex $v \in V$ corresponds to a wireless node in the network. There is an edge $(v_1, v_2) \in E$ connecting vertex v_1 to v_2 if they are physically located within each other's communication range.

We assume a traffic pattern in which all packets are destined to the gateway GW (i.e., Mesh Portal). As explained in Section 2.1.3, this traffic pattern is common in WMN that provide Internet access to other nodes in the network through the Portal.

We distinguish between two types of interfaces assignment: static and dynamic. The former assigns each interface to a channel for permanent use. Dynamic assignment allows interfaces to frequently switch from one channel to another. For the dynamic assignment, we define a *timeslot* to be the time spent on a single channel.

Finally, we adopt the concept of *conflict graph* $G_C(V_C, E_C)$, described in Section 2.3.2, to represent interference in G [62]. Thus, any interfering model can be adopted, such as the Protocol Model and the Physical Model [21, 75].

5.3 MILP Formulation

In this section, we present a novel MILP formulation to evaluate the capacity of channel and interface assignment in IEEE 802.11s mesh networks. We inject the constraints specific to the assignment into the formulation to obtain the resulting capacity and the optimal assignment. Thus, we can compare different strategies with the optimal upper bound (i.e., what could we obtain with a centralized optimal formulation?). The optimal upper bound indicates the performance gap to fill with better assignment strategies to maximize the network throughput.

We focus on the traffic pattern in which the whole traffic is destined to the gateway. However, we highlight that the model can be directly applied to the bidirectional traffic, in which the traffic is either destined or generated by the gateway.

Symbol	Definition
GW	Gateway (Mesh Portal)
$N(v)$	Neighborhood of node v
Ch	Set of available orthogonal channels
BW	Bandwidth of a channel
$\mathcal{I}(v)$	Set of interfaces at node v
T	Set of timeslots
Cl	Set of all maximal cliques in the conflict graph G_C
$f(v)$	Traffic generated by node v
$f(v_1, v_2)$	Traffic link (v_1, v_2) from node v_1 to node v_2
$f(v_1, v_2, c, t)$	Traffic link (v_1, v_2) on channel c and timeslot t
$act(v_1, v_2, c, t)$	1 if link (v_1, v_2) is active on channel c during timeslot t
$ch(v, c, t)$	1 if node v uses channel c during timeslot t
$parent(v_1, v_2)$	1 if node v_2 is the parent of node v_1 in the routing tree

Table 5.1: MILP Formulation Notation.

The formulation remains unchanged.

Also, we adopt a TDMA-approach, with $|T|$ timeslots.

Table 5.1 presents the MILP formulation notation.

5.3.1 Objective Function

The capacity is often described as the maximum throughput achievable in the network [132]. In our approach, we also aim to guarantee fairness while maximizing the capacity. Thus, our objective function can be described as follows: maximize the minimum throughput associated to all the flows (i.e., *max-min-flow*). It is formulated as follows:

$$MMFlow = Max \quad (Min\,(f(v))) \tag{5.1}$$

It is subject to the constraints defined below.

5.3.2 Constraints

Flow Conservation

The transmitted traffic is equal to the sum of the forwarded traffic and the traffic generated by node v_1 itself:

$$\forall v_1 \in V - \{GW\}, \sum_{v_2 \in N(v_1)} f(v_1, v_2) = \sum_{v_2 \in N(v_1)} f(v_2, v_1) + f(v_1) \tag{5.2}$$

Gateway GW consumes all the flows:

$$\sum_{v \in N(GW)} f(v, GW) = \sum_{v \in V} f(v) \tag{5.3}$$

Multi-Channel Multi-Interface

Traffic through link (v_1, v_2) is the aggregation of the traffic transmitted over all corresponding channels:

$$\forall (v_1, v_2) \in E, \quad f(v_1, v_2) = \sum_{c \in \mathcal{C}h, t \in T} f(v_1, v_2, c, t) \tag{5.4}$$

A link can forward data only if it is active during a given timeslot on a given channel:

$$\forall (v_1, v_2) \in E, \forall c \in \mathcal{C}h, \forall t \in T, f(v_1, v_2, c, t) \leq BW \cdot act(v_1, v_2, c, t) \tag{5.5}$$

Hardware constraints are also integrated. The number of interfaces upper bounds the number of channels a node can use. Thus, if a node v has $|\mathcal{I}(v)|$ interfaces, we can then upper bound the number of channels it can use as follows:

$$\forall v \in V, \forall t \in T, \quad \sum_{c \in \mathcal{C}h} ch(v, c, t) \leq |\mathcal{I}(v)| \tag{5.6}$$

We consider that a channel is *active* for a node if one of its out/ingoing links uses it:

$$\begin{aligned} &\forall v_1 \in V, \forall v_2 \in N(v_1), \forall c \in \mathcal{C}h, \forall t \in T, \\ &\qquad \frac{1}{2 * |N(v_1)|} \sum_{v_2 \in N(v_1)} (act(v_1, v_2, c, t) + act(v_2, v_1, c, t)) \leq \\ &\qquad\qquad ch(v_1, c, t) \leq \sum_{v_2 \in N(v_1)} (act(v_1, v_2, c, t) + act(v_2, v_1, c, t)) \end{aligned} \tag{5.7}$$

In the constraint of Equation 5.7, the main idea behind the first part is to force $ch(v1, c, t)$ to be different from null if at least one act is different from null (i.e., one link is active on channel c and timeslot t). The denominator $2 * |N(v_1)|$ corresponds to the maximum number of links that can be active for a node v_1, considering both out/ingoing links. Thus, 1 is the maximum value that can be achieved by the first part of this constraint. In particular, it corresponds to the case in which all links are active. The last part is used to force $ch(v1, c, t)$ to be null if all act are null. We highlight that this type of constraint construction is usual when boolean variables are defined.

Multi/Single flows

To simplify the formulation, we may forbid multi-flow solutions (i.e. the routing structure forms a tree rooted at gateway GW). Each node has in this case a single parent:

$$\forall v_1 \in V - \{GW\}, \quad \sum_{v_2 \in N(v_1)} parent(v_1, v_2) = 1 \tag{5.8}$$

Traffic is forwarded only through tree links, in the upload or the download directions:

$$\forall (v_1, v_2) \in E, f(v_1, v_2) \leq BW \cdot |\mathcal{C}h| \cdot (parent(v_1, v_2) + parent(v_2, v_1)) \tag{5.9}$$

5.3.3 Bandwidth Sharing Constraints

We have formulated in the previous subsection all hardware (multiple channels and interfaces) and multi-flow constraints. We have now to model the MAC layer to represent how the different nodes may share the bandwidth with each other.

We formulate three models to distribute bandwidth of each radio link. We first construct an upper (*Bandwidth Sharing / Upper Bound*) and lower bound (*Bandwidth Sharing / Lower Bound*) modeling an ideal MAC layer in which the MAC protocol is able to distribute the radio bandwidth to each contending transmitter with a perfect fairness. This model works with any channel assignment strategy: we just have to fix accordingly the $act(v_1, v_2, c, t)$ variables in the formulations.

We also propose a specific model for a conflict-free assignment (*Conflict-free*), where bandwidth sharing is much simpler to model because no interference arises. If we let the $act(v_1, v_2, c, t)$ unspecified, the MILP resolution will give us the optimal assignment for the given objective.

5.3.3.1 Bandwidth Sharing / Upper Bound

To obtain the upper bound, we forbid two interfering links to be active simultaneously. They may transmit packets during the same timeslot, but not at the same instant, else a collision would occur. Thus, we just have to share the radio bandwidth with fairness in a group of interfering links. The conflict graph G_C (Section 2.3.2) provides the right abstraction for formulating such a constraint.

Here, two main concepts of graph theory must be defined: *clique* and *maximal clique* [133]. A clique is an induced subgraph in G_C that is a complete graph, which means that every two vertices in this subgraph are connected by an edge. As an example, consider the conflict graph G_C illustrated in Figure 5.1(a) and its corresponding examples of cliques highlighted in Figure 5.1(b). In particular,

Figure 5.1(b) presents one example with 1-vertex clique, two examples with 2-vertex cliques, and one example with 3-vertex cliques, respectively. A maximal clique of G_C is a complete subgraph that it is not contained in any other complete subgraph. It corresponds to a clique that cannot be extended by including one more adjacent vertice. Figure 5.1(c) shows the corresponding maximal cliques of G_C (Figure 5.1(a)). Note that in none of the examples illustrated in Figure 5.1(c) the clique can be extended.

We extract all maximal cliques to get the groups of links that interfere pairwisely. Then, for each maximal clique, all the traffic transmitted by all the radio links of the clique must be inferior or equal to the radio bandwidth allocated to each timeslot:

$$\forall cl \in \mathcal{C}l, \forall c \in \mathcal{C}h, \forall t \in T, \sum_{(v_1,v_2)\in cl} f(v_1, v_2, c, t) \leq \frac{BW}{|T|} \tag{5.10}$$

We can also remark that two links sharing a node are classified as interfering links. They will share the radio bandwidth: the MAC layer will regulate their transmissions.

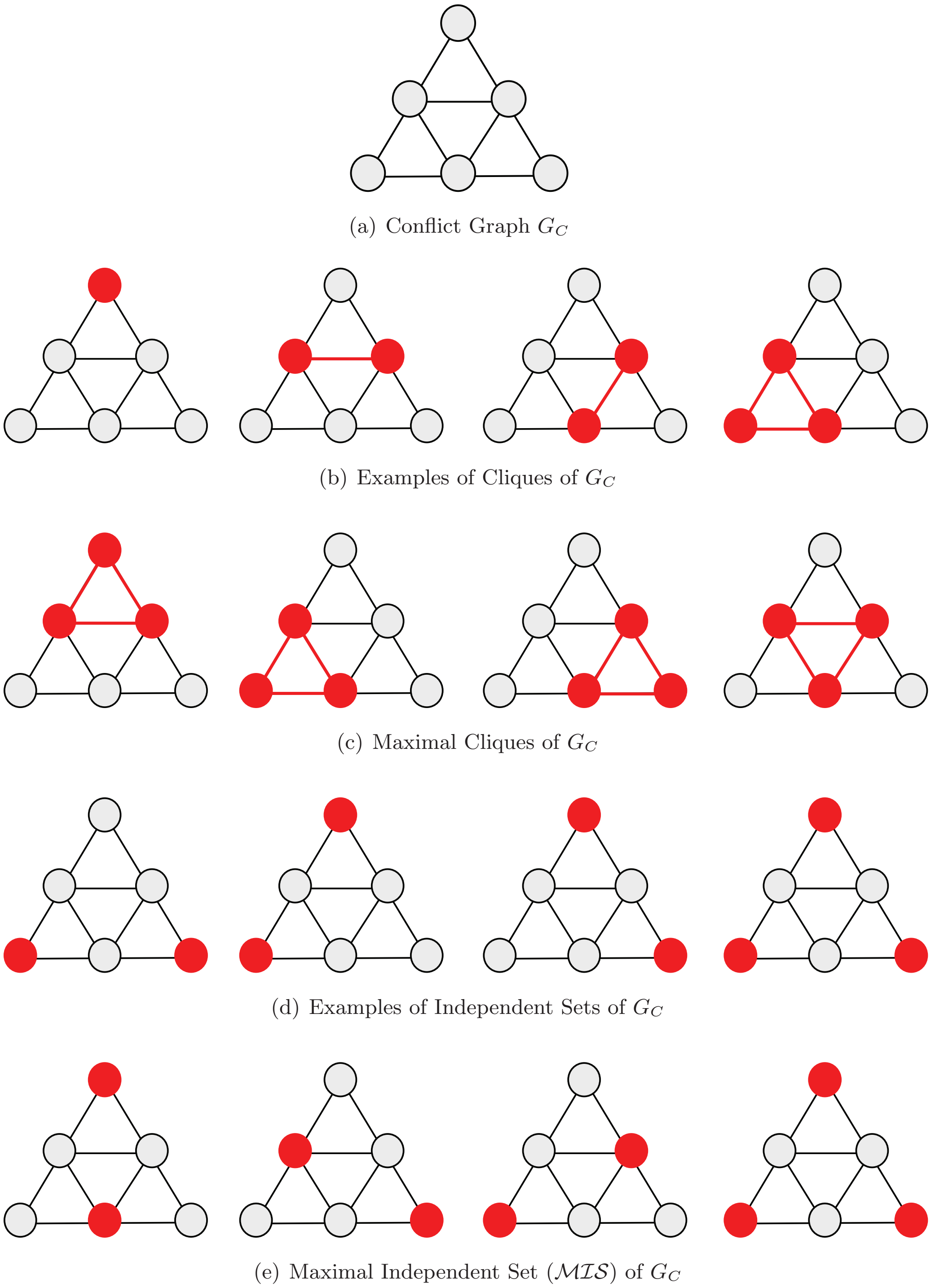
We can use the algorithm defined by Bron and Kerbosch [134] to extract all maximal cliques in the conflict graph. Although the problem is NP-complete, this algorithm leads to an acceptable computing time for small network instances (i.e. a few hundreds of nodes).

We can verify that this constraint is an upper bound: implicitly, we formulate a schedule in a group of radio links that interfere with each other. Although local scheduling is always achievable, we could obtain an infeasible global scheduling. We underestimate here the impact of interference on bandwidth sharing to construct an upper bound. In other words, although the formulation returns in which timeslot a radio link can be active, it does not specify *when* within the timeslot the radio link can be active. Consequently, interference may arise between active radio links in the same timeslot.

5.3.3.2 Bandwidth Sharing / Lower Bound

We now construct a feasible solution by creating a globally valid schedule (i.e., a set of radio links that may transmit simultaneously without creating collisions). If we assume that all packets have the same size, we can reference all these sets and give a fraction of the bandwidth to each of the sets. To formulate this constraint, we continue adopting the conflict graph G_C.

Now, two other graph theory concepts must be defined: *independent set* and *Maximal Independent Set* ($\mathcal{MIS}$) [133]. An independent set is a subset of vertices in G_C such that no two vertices in this subset represent an edge in G_C. Thus,



(a) Conflict Graph G_C

(b) Examples of Cliques of G_C

(c) Maximal Cliques of G_C

(d) Examples of Independent Sets of G_C

(e) Maximal Independent Set ($\mathcal{MIS}$) of G_C

Figure 5.1: Understanding Some Concepts of Graph Theory used in our MILP Formulations.

the size of an independent set corresponds to the number of vertices it contains. Figure 5.1(d) shows some examples of independent sets of G_C (Figure 5.1(a)). A $\mathcal{MIS}$ of G_C is an independent set that is not a proper subset of another independent set of G_C. Thus, $\mathcal{MIS}$ is an independent set such that adding any other vertex to the set forces the set to contain an edge. For example, Figure 5.1(e) shows examples of $\mathcal{MIS}$ of G_C (Figure 5.1(a)). Note that G_C may have $\mathcal{MIS}$ of varying sizes.

When a schedule is feasible, all the scheduled radio links form a $\mathcal{MIS}$ in the conflict graph. Indeed, no pair of radio links that are simultaneously activated (are parts of the $\mathcal{MIS}$) are neighbors in the conflict graph. Thus, they do not interfere with each other.

We have to reference all the $\mathcal{MIS}$. Then, we assign a fraction q_i of the bandwidth to each $\mathcal{MIS}$ i: they share the global radio bandwidth. After having referenced all the channels and timeslots, we obtain:

$$\forall c \in \mathcal{C}h, \forall t \in T, \sum_{i \in \mathcal{MIS}} q(i,c,t) \leq \frac{BW}{|T|} \tag{5.11}$$

Finally, each radio link can use at most the sum of the bandwidth assigned to each set it owns:

$$\forall (v_1, v_2) \in E, \forall c \in \mathcal{C}h, \forall t \in T, \quad f(v_1, v_2, c, t) \leq \sum_{i \ni (v_1, v_2)} q(i,c,t) \tag{5.12}$$

A radio link will benefit from the sum of bandwidth associated with each $\mathcal{MIS}$ it owns. Combined with inequality 5.5, we only count the bandwidth for the channel used by a radio link.

We use the algorithm proposed by Tsukiyama *et al.* [135] to compute the set of $\mathcal{MIS}$. The number of $\mathcal{MIS}$ can be very large leading to unsolvable problems: it is practically much larger than the number of maximal cliques. This increases the MILP resolution time significantly. Thus, we provide a lower bound by considering only the first $max_{\mathcal{MIS}}$ sets.

To obtain a feasible solution in which all radio links can be scheduled, each radio link must be present in at least one set of $\mathcal{MIS}$. We proceed in the following way. First, we extract the first $max_{\mathcal{MIS}}$ groups of radio links ($max_{\mathcal{MIS}}$ is a constant). Then, for each radio link e never present in one group, we create a new set $s = \{e\}$ and greedily insert a list of non-interfering links.

We can verify we construct here a lower bound: by only considering a subset of all $\mathcal{MIS}$, we underestimate the radio spectrum reuse. Some $\mathcal{MIS}$ may have not been considered although they would have provided better performance.

5.3.3.3 Conflict-free

We consider separately an assignment strategy where we entirely forbid simultaneous communications of two interfering pairs of nodes. In this case, bandwidth sharing constraints are simpler since a radio link receive the whole radio bandwidth to use: it has no interfering transmitters.

Scheduling has to be found such that no pair of interfering links should be active during the same timeslot (i.e. it is *conflict-free*). In this case, we reference all cliques in the conflict graph. Therefore, during a given timeslot t only one link per maximal clique is active:

$$\forall cl \in \mathcal{C}l, \forall c \in \mathcal{C}h, \forall t \in T, \sum_{(v_1,v_2)\in cl} act(v_1, v_2, c, t) \leq 1 \tag{5.13}$$

and a radio link can use the full radio bandwidth during t:

$$\forall (v_1, v_2) \in E, \forall c \in \mathcal{C}h, \forall t \in T, \quad f(v_1, v_2, c, t) \leq act(v_1, v_2, c, t) \cdot \frac{BW}{|T|} \tag{5.14}$$

5.4 Performance Evaluation

In this section, we evaluate the capacity of IEEE 802.11s through numerical analysis and simulation. We focus on the most interesting results that can guide the design of an efficient protocol in MCMI WMN. We present the results with a confidence level of 95%.

5.4.1 Numerical Results

We first analyze the results of the MILP formulation for the following strategies. We use the notation *Channel Assignment Strategy / MILP bandwidth sharing constraints* to designate each approach:

- **Common Channel / Bandwidth Sharing UP and LOW** (upper and lower bounds): the objective function is calculated, but the MILP does not decide the channel assignment. We adopt a *Common Channel Assignment* (Section 3.4): channel C_k is assigned to interface k. The MILP upper and lower bounds (Sections 5.3.3.1 and 5.3.3.2) extract the associated capacity since some interfering interfaces may use the same channel;

- **MMFlow / Bandwidth Sharing UP and LOW** (upper and lower bounds): we let the MILP decide by itself the channel assignment while maximizing the network capacity (Equation 5.1). The MILP upper and lower bounds (Sections 5.3.3.1 and 5.3.3.2) formulate bandwidth sharing constraints between possibly interfering interfaces;

- **MMFlow / Conflict-free** (Section 5.3.3.3): we use the MILP formulation which forbids interfering links to be active in the same timeslot. The MILP assigns during the resolution one channel per interface and timeslot.

A custom simulator generates Unit Disk Graphs and we consider the interference range is twice the radio range [75]. Nodes are placed randomly on a disk while maintaining an average degree of 7 (number of neighbors). We use 10 channels in the 5 GHz band to stress the channel assignment (more IEEE 802.11a channels means a larger capacity). Numerical simulations measure the MILP objective function $MMFlow$ (c.f., Equation 5.1).

Impact of the Number of Interfaces

Figure 5.2(a) shows the throughput of each flow (obtained through the objective function in Equation 5.1) as the number of interfaces increases. The capacity almost linearly increases with the number of interfaces for all the strategies. Kyasanur and Vaidya [29] obtained similar results with an asymptotic approach. Thus, even if we take into account protocol details, the result keeps on holding: using a larger number of interfaces is always beneficial.

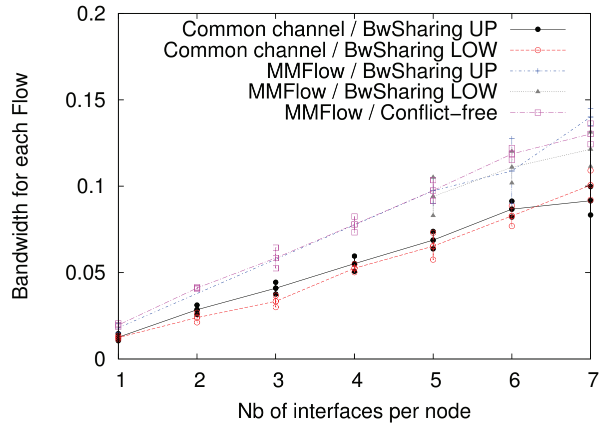
We can also remark that *Common Channel* assignment is a sub-optimal strategy. Although it is simple to implement, it negatively impacts capacity. Finally, we may also note that *MMFlow/Conflict-free* and *MMFlow/BwSharing* offer similar capacity. A simple conflict-free assignment leads to the same throughput as with an ideal MAC layer. As this fact may significantly simplify implementation, it is a promising way to the design of efficient IEEE 802.11s networks.

Dynamic Interfaces

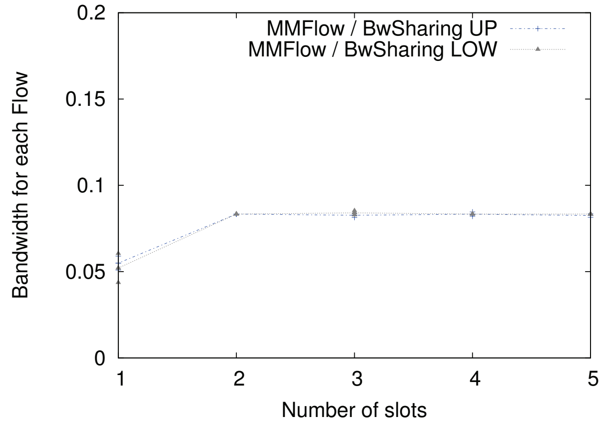
Furthermore, we study the impact of the interface assignment strategy (i.e., static versus dynamic) on the capacity. For *MMFlow/BwSharing*, we have modified the number of slots: 1 slot means that interfaces are static, while a larger number of slots increases the number of channel switching (Figure 5.2(b)). Surprisingly, we remark that network capacity does not depend on the number of slots, i.e. number of channel switching. Thus, static interfaces permit to obtain optimal capacity while limiting implementation complexity.

Conflict-Free Scheduling Complexity

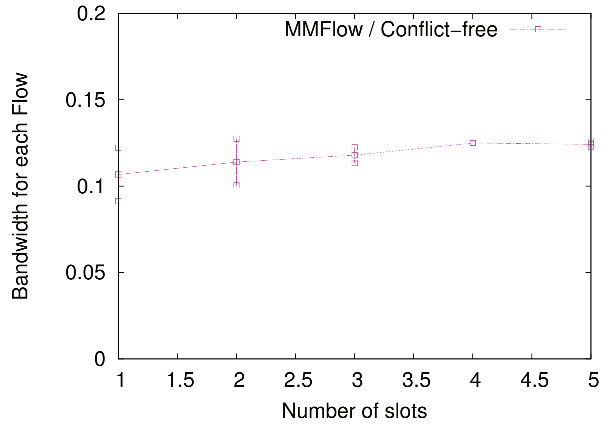
Finally, we focus on the *MMFlow/Conflict-free* scheduling approach in Figure 5.2(c). To stress IEEE 802.11, we set-up one interface per node to forward packets. Surprisingly, capacity is almost independent of the number of slots. In other words, static channel assignment is sufficient to attain near optimal capacity.



(a) 4 slots



(b) 2 interfaces



(c) 2 interfaces

Figure 5.2: Numerical results (25 nodes, 10 channels).

In summary, IEEE 802.11s can achieve optimal performance, if all interfaces use a ***static channel*** during a given timeslot, and a distributed algorithm implements a ***conflict-free channel assignment*** (i.e. during a given timeslot, no pair of interfering radio links use the same channel).

5.4.2 Simulation Results

We have validated the MILP formulation by simulations in the Network Simulator 3 version 12 (NS-3.12) [136] using the IBM ILOG CPLEX Optimizer [137] to solve linear programming presented in Section 5.3. In particular, we compare the network performance of IEEE 802.11s when different channel assignments are used: the *Common Channel* assignment (i.e., channel C_k is assigned to interface k) and the channel assignments resulting from the three bandwidth sharing modes (*MMFlow/BwSharing UP*, *MMFlow/BwSharing LOW*, and *MMFlow/Conflict-free*)

For the three bandwidth sharing modes, we have developed extensions to support the interaction between NS-3 and CPLEX, as well as to support their different channel assignments. We depict this interaction in Figure 5.3, in which a box denotes an action and an arrow represents variables passing between actions. The first step is to define the topology in NS-3. We use a single-interface network tuned to a common channel to run a neighbor discovery process (without data traffic) during 30 seconds and obtain the network topology. This topology is passed to CPLEX along with a list of parameters such as the number of channels and interfaces, the number of timeslots, and the bandwidth sharing mode (*MMFlow/BwSharing UP*, *MMFlow/BwSharing LOW*, or *MMFlow/Conflict-free*). Then, CPLEX solves the MILP problem and returns to NS-3 the list of channels to be assigned to each interface of each node. Finally, we evaluate the *channel assignment* reported by CPLEX through NS-3 simulations in a multi-channel multi-interface wireless mesh network.

Simulations validated the assumptions we used in our MILP model. In other words, even if the PHY and MAC layers are more complex, the different assignment strategies keep on achieving different network capacity. We evaluate the performance with 49 nodes arranged in a grid of 7 by 7. Each node has at most 4 neighbors in the grid. Nodes are equipped with IEEE 802.11a wireless interfaces and therefore 12 orthogonal channels are available [2]. To represent a typical WMN traffic pattern, flows are originated from the mesh routers to the gateway. Each data point in the graphs is computed as being a result of 40 different simulations. For each simulation run, a node is randomly designated as the gateway to receive the Poisson traffic generated by all other mesh routers. A new flow is started every 1 second. Thus, the variations in the obtained results mainly occur due to the randomness of the topology caused by the gateway position and flow initialization.

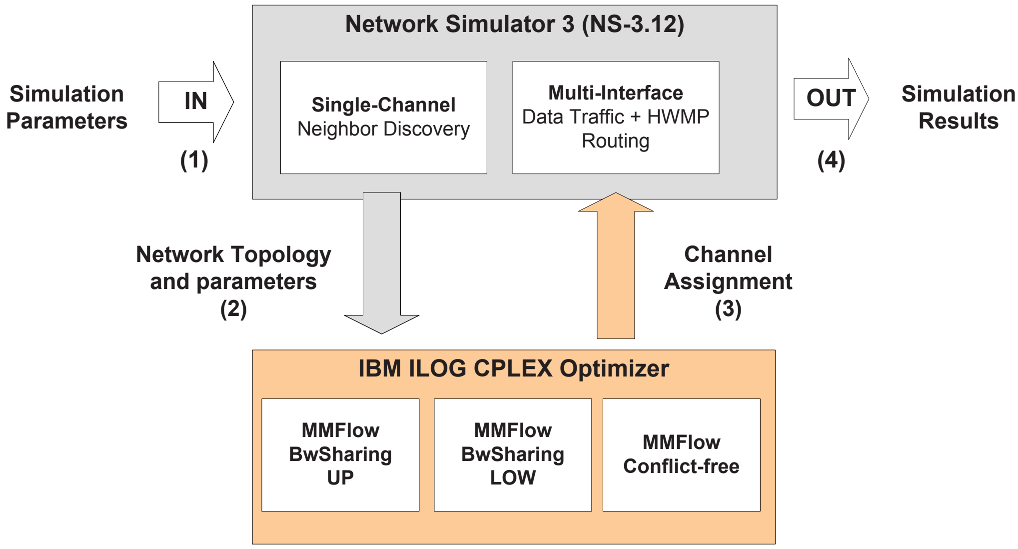


Figure 5.3: A diagram that depicts the interaction between NS-3.12 and the IBM ILOG CPLEX Optimizer.

We have considered a packet size of 512 bytes. The total simulation time is 100 seconds.

Network Capacity

We first investigate the impact of an increasing data traffic, as shown in Figures 5.4 and 5.5.

We observe that all MILP solutions present better delivery rate (Figure 5.4(a)) and aggregated throughput (Figure 5.4(b)) than the *Common Channel* assignment approach regardless of the traffic load. Moreover, as the data rate increases, the performance gap between the proposed MILP solutions and the *Common Channel* assignment also increases. As expected, the *MMFlow/Conflict-free* solution presents the highest throughput: by limiting interference and collisions, it improves the performance. Also, note that the performance of *MMFlow/BwSharing UP* is lower than *MMFlow/BwSharing LOW*. Under-estimating interference leads practically to a less efficient channel assignment.

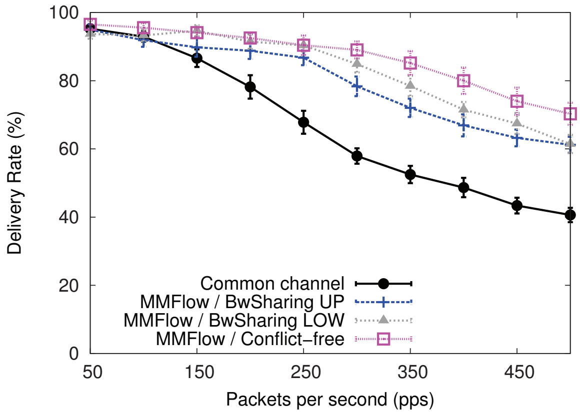
We then studied the routing behavior measuring in particular the number of transmitted Path ERRors (PERR). The PERR is used for announcing unreachable destinations. As shown in Figure 5.5(a), too many packet transmissions increase the error rate. We highlight that the high error rate increases the path cost and leads to frequent path changes. This path instability in HWMP leads to the degradation of the delivery rate (Figure 5.4(a)) and increases delay (Figure 5.4(c)), especially for the *Common Channel* assignment, because it operates only over a small portion of the available spectrum. Thus, it does not efficiently distribute the load among all available channels as in the proposed MILP formulation. Therefore, more contention and interference may occur on pre-defined channels in the *Common Channel* assignment.

In addition, the *Common Channel* assignment suffers from severe performance degradation due to the *broadcast storm problem* [105]: HWMP broadcast many control packets such as the Path Request (PREQ) on each of its interfaces, which results in serious overhead, collisions, and contention. Indeed, two neighbors in the *Common Channel* assignment always have multiple independent links to communicate with each other. Besides, the amount of transmitted and received control packets in the *Common Channel* assignment is much higher than that of MILP approaches. Therefore, while a considerable part of transmission opportunities is used by the *Common Channel* assignment to send control packets, these opportunities are used by the MILP strategies to send data packets. In particular, the *MMFlow/Conflict-free* case performs much better than the *Common Channel* assignment especially under high data traffic, because it computes a schedule for radio links so that collisions and interference are avoided.

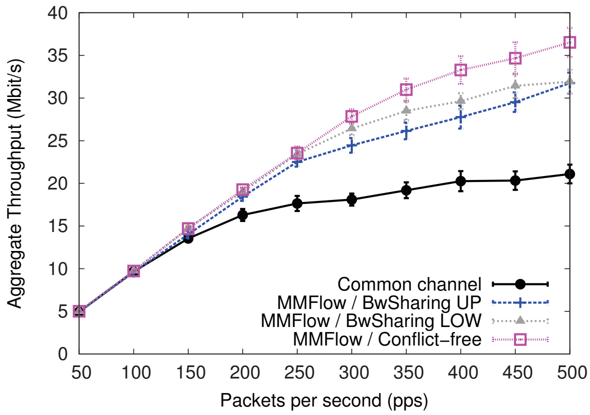
In IEEE 802.11s, a node may have several network interfaces with different MAC addresses. Thus, the standard defines a *network layer* buffer for packets waiting for routes and eventual forwarding to the next hop. Figure 5.5(b) shows the number of packet drops at the network layer (i.e. *route was not found*), while Figure 5.5(c) presents the number of frame drops at the MAC layer. We can notice that in multi-hop forwarding, the same buffer is used for both the traffic forwarded from other interfaces and traffic originated by the node itself. Thus, the nodes close to the gateway will quickly undergo buffer overflow, which creates an unfair situation for the traffic originated by the nodes close to the gateway. The fall of the curves when the data traffic increases in Figure 5.4(a) comes in part from this unfair behavior. We believe that the basic means to address this issue is to give channel assignment priority to links closer to the gateway based on the number of available channels and interfaces per node.

Impact of the Number of Interfaces

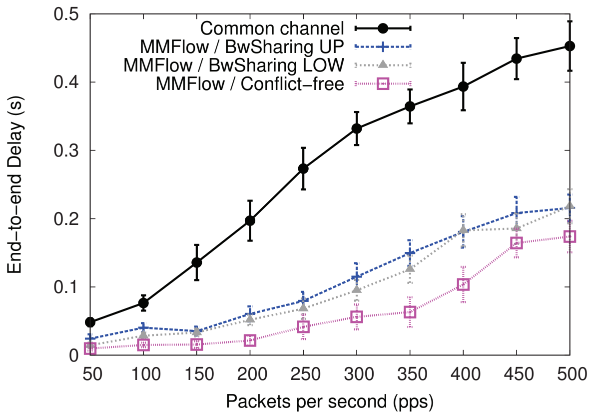
Next, we study the impact of the number of interfaces on the capacity (Figure 5.6). For all MILP solutions, delivery rate and aggregate throughput slightly increase and the end-to-end delay slightly decreases with the number of interfaces. This limitation of the increase of capacity mainly occurs because of the buffer overflow of nodes close to the gateway and the retransmission of control packets by



(a) Delivery Rate.

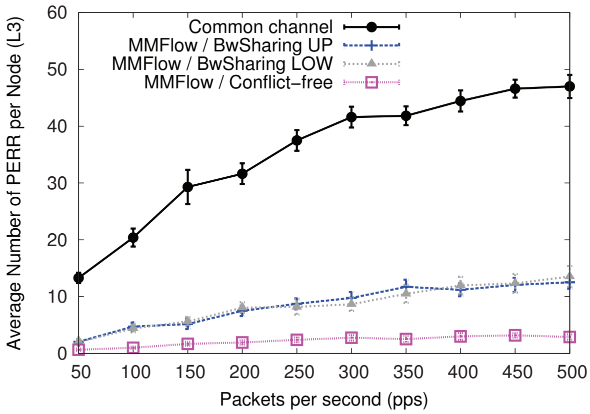


(b) Aggregate Throughput.

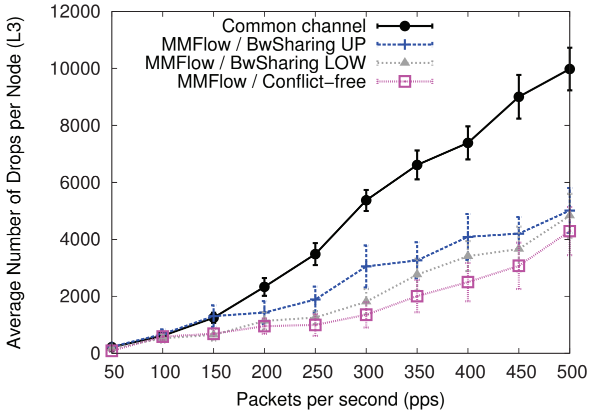


(c) End-to-end Delay.

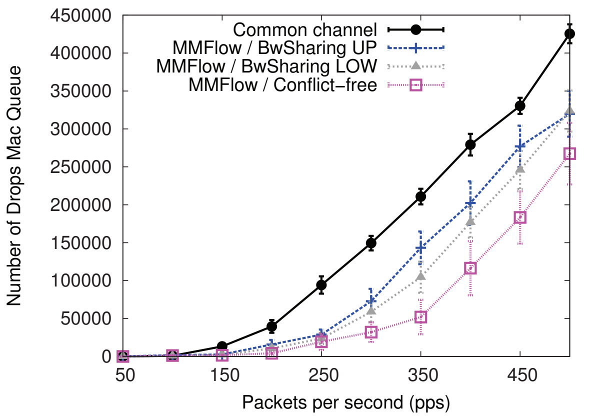
Figure 5.4: Simulation results. Impact of data traffic (3 interfaces, 12 channels) – Part 1.



(a) Path Error (PERR) L3.

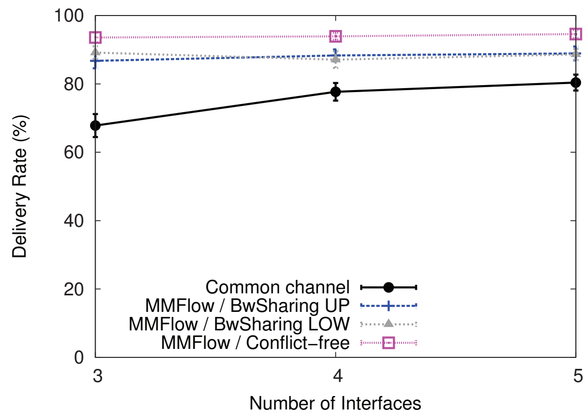


(b) Drop Network Layer.

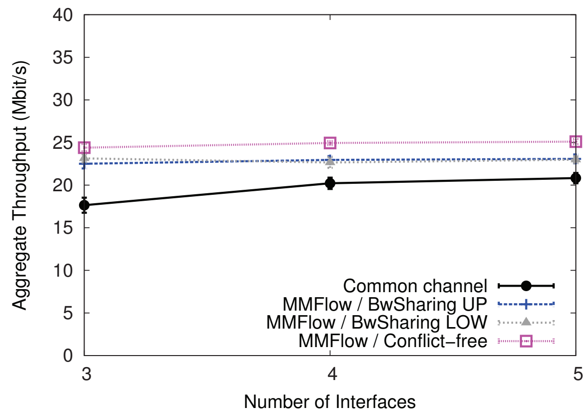


(c) Drop MAC Queue.

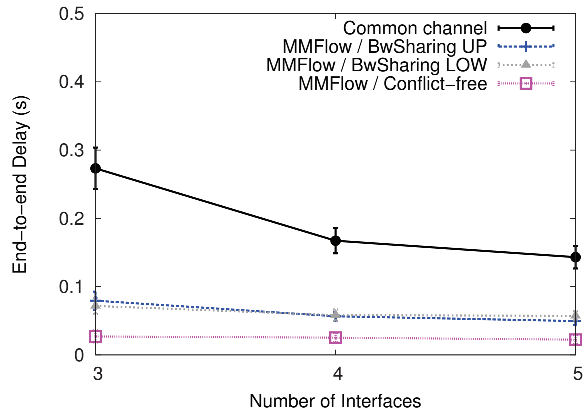
Figure 5.5: Simulation results. Impact of data traffic (3 interfaces, 12 channels) – Part 2.



(a) Delivery Rate.



(b) Aggregate Throughput.



(c) End-to-end Delay.

Figure 5.6: Simulation results. Impact of the number of interfaces (12 channels, 250 pps).

every interface. For the *Common Channel* assignment, a greater number of interfaces increases the amount of transmitted control messages. For example, PREQs are retransmitted by every interface.

It is clear that the default IEEE 802.11s routing protocol (i.e., HWMP) is not as good for multi-interface networks. Therefore, the improvement in fairness and the reduction of overhead without affecting routing protocol performance are two challenging issues to achieve better capacity in MCMI IEEE 802.11s.

5.5 Conclusion

In this chapter, we have presented MILP formulations to evaluate network capacity in multi-channel multi-interface wireless mesh networks. They allow us to quantitatively study various design choices for IEEE 802.11s and in particular, to show that static assignment is beneficial in mesh networks.

We have also run extensive simulations that take into account aspects neglected in the MILP formulations (e.g., realistic MAC layer, routing, traffic load). The simulation results extend the analysis based on the MILP formulations. First, it shows that the *Common Channel* assignment strategy is clearly sub-optimal and a large performance gap has still to be filled. Besides that, it shows that *MMFlow/BwSharing UP* and *MMFlow/BwSharing LOW* present a similar behavior and that *MMFlow/Conflict-free* presents the better performance. One direction for improvement is the use of conflict-free solutions that present the highest throughput by limiting interference and collisions.

Chapter 6

Routing in Multi-Channel Multi-Interface Wireless Mesh Networks

Contents

6.1 Introduction

Routing in Multi-Channel Multi-Interface (MCMI) Wireless Mesh Networks (WMN) is an ongoing area of research. In the beginning, much of the existing work in the area was inspired by solutions developed for single-channel single-interface wireless networks, including the routing metrics and protocols [48]. However, it has been shown that single-channel single-interface solutions do not perform as well in environments with multiple interfaces, especially because they do not profit from the diversity of channels assigned to links, such as demonstrated in [88, 104]. As typical examples, consider the Basic Routing Metrics described in Section 2.4.1.1: Hop, ETX, ETT and ALM. These metrics were designed for single-channel single-interface networks and therefore they do not discriminate between same channel paths and channel diverse paths. Although they can be used in MCMI networks, they do not account for the reduction in throughput due to interference among links that operate on the same channel. Consequently, these metrics can result in paths with poor quality.

In fact, while equipping nodes with more than one interface mitigates some concerns, other new concerns are introduced and not properly handled by single-channel solutions. A typical example was discussed in Section 5.4.2, in which we have shown the inability of conventional routing protocols to deal with the high overhead imposed by the replication of control packets on all interfaces. Furthermore, as presented in Section 2.4.1, many routing metrics are not able to depict all the factors that impact the performance of the network, be warned that it is not trivial to identify and capture all these factors. Among the well-known factors, the intra-flow and inter-flow interference, the traffic load and the link quality can be regarded as the most recent trend to depict the behavioral aspects of the network. For example, interference can be measured to favor paths with channel diversity [30], while routing solutions aware of the traffic load may prevent congested areas [63]. Also, link quality measures may be used to avoid links with high packet losses [138]. However, as summarized in Table 2.6 (Page 58), most routing metrics do not consider the combination of these factors when judging the goodness of a path. Besides, they use different information gathering methods to obtain the measurements. In consequence, each metric has an appropriate use case, which generally is not extensible to different network scenarios (e.g., mesh networks with variations of traffic and interference).

To fill these gaps, we focus here on the development of a novel cross-layer routing solution for multi-channel multi-interface wireless mesh networks. Our main goal is to obtain a precise characterization of the wireless links in order to increase the overall network performance. More specifically, we aim to explore features of different measures, and then explore the potential capabilities of their combination to capture the factors mentioned above: the intra-flow and inter-flow interference, the traffic load and the link quality.

First, we aim to estimate the available bandwidth on a path to allow nodes to perform routing preferentially through "free to use" channels. To achieve this objective, we propose a routing metric called *Path Residual Bandwidth* (PRB) to determine the residual bandwidth of a link while taking into account the traffic load and interference variations. Different from of other routing metrics, our metric calculation is based on the fact that it is inappropriate to include a link that has little residual bandwidth (i.e., a bottleneck link) into the routing path even through this link has an acceptable quality.

Unlike $iAWARE$ [68] and $MIND$ [63] routing metrics, we do not use SNR and SINR to estimates the inter-flow interference in the network. As summarized in Table 2.4, SNR and SINR are not portable measures: they are not recorded by commercial wireless cards while receiving packets and then have to be derivate from other measures [20, 65]. Thus, it is hard to accurately compute SNR and SINR in practice.

We choose to adopt more realistic measures to dynamically incorporate the inter-flow interference. Each node locally captures information from the physical layer via passive monitoring and makes this information available to the routing protocol. Also, our metric incorporates the intra-flow interference, as opposed to metrics such as ETX [90], ETT [30], ALM [19] and $RARE$ [70]. Our routing metric considers the channel diversity: the intra-flow interference is quantified in order to favor paths with links operating on different channels. In fact, we evaluate if links on a path are within the interference range of each other before assigning weights. The static nature of WMN backbone makes it possible to determine whether two nodes are in each other's interference range, and therefore the identification of interfering links. Thereby, we overcome the drawback of $WCETT$ [30] and $iAWARE$ [68], which assume that two links on a path operating on the same channel are always interfering with each other. We also overcome the drawback of MIC [97] and $MIND$ [63] that consider only one previous hop in their evaluation. In summary, our routing metric corresponds to a combination of measures, in which each measure is able to depict different aspects of the link.

Next, the metric is incorporated into a new on-demand path selection protocol that operates over the link layer protocol. In special, we propose to distribute flows among multiple paths to fully exploit the available channels and interfaces, as well as to prevent the formation of congested areas in the wireless network. We also propose two strategies to reduce the high overhead caused by broadcasting on multiple interfaces. Consequently, our protocol increases network performance.

Finally, we evaluate the performance of our cross-layer routing solution via extensive simulations. We find that in multi-channel multi-interface wireless mesh networks, our solution significantly outperforms previously proposed routing metrics and path selection protocols.

6.2 Computing the Routing Metric

Our routing metric aims at depicting the main parameters that impact the performance of a wireless mesh networks. Figure 6.1 introduces these parameters distributed into four different levels, as well as their interaction, represented by the arrows passing between levels. The objective is to properly merge all these parameters to calculate the final metric *Path Residual Bandwidth* (PRB), as presented in

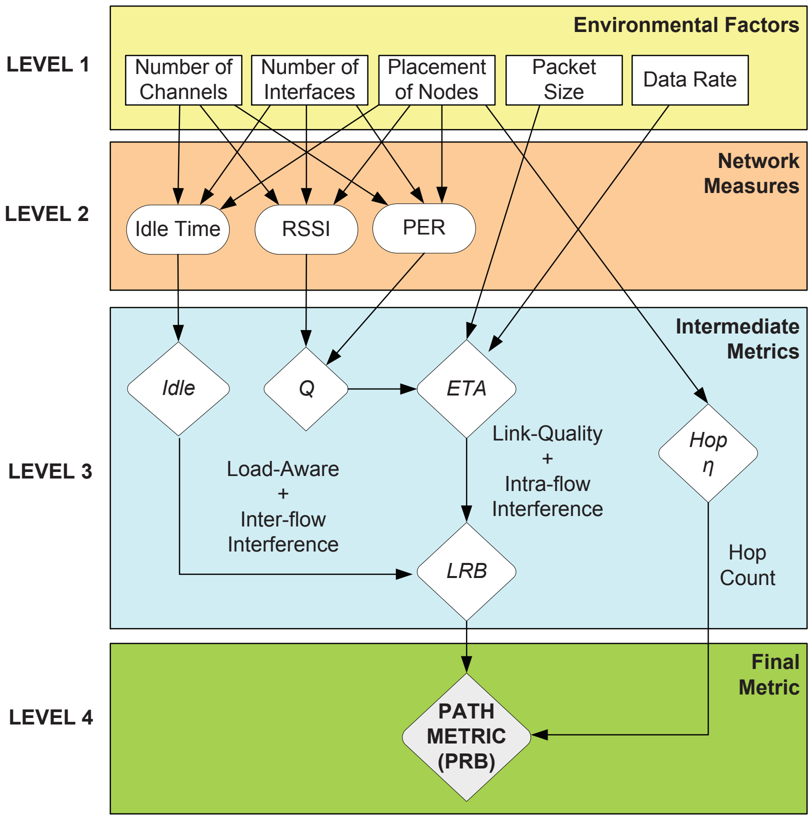


Figure 6.1: A Diagram that Depicts the Interaction of Parameters used to Calculate the Routing Path Metric.

Level 4 of Figure 6.1.

The first level of the diagram presents what we called Environmental Factors, which correspond to the parameters that have some influence in the metric calculation, but are not affected by the mesh network, such as the number of channels and interfaces, the nodes placement, packet size, and data rate[1]. In other words, Environmental Factors are not subject to feedbacks from the mesh network, but they may impact the network performance, positively or negatively.

The second level of the diagram corresponds to the Network Measures: Idle Time, Received Signal Strength Indicator (RSSI) and Packet Error Rate (PER). Unlike Environmental Factors, these measures are highly affected by the networks conditions such as the traffic load, link quality and interference. These measurements are key factors in the design of our final Path Metric. If analyzed separately, these metrics are not able to display the network behavior in a precise way. For in-

[1] Although our solution supports different data rates, we do not aim here to study the impact of data rate variations over time, such as performed by the adaptive auto rate fallback algorithm [139].

stance, although link quality measures avoid the selection of poor quality channels, they are unaware of the extent of the traffic load and thus to avoid congested areas. Therefore, how to capture and combine these network measures is an important and challenging problem in wireless mesh networks.

Finally, the last level before the Path Metric computation corresponds to what we have called the Intermediate Metrics. As observed in Figure 6.1, these metrics are calculated through Environmental Factors and/or Network Measures. At the end, all Intermediate Metrics are merged in order to calculate the path metric *Path Residual Bandwidth* (PRB). Thus, our final metric will be able to incorporate the Environmental Factors and Network Measures that affect the network performance.

In this section, we present how the measures and metrics shown in the diagram of Figure 6.1 are captured and/or calculated. At the end, we show how metrics are combined in order to perform a final path metric computation.

6.2.1 Channel Quality Estimation

We start by explaining how each node locally estimates the quality of channels available for communication. We use the *idle time* to allow nodes to be aware of the available airtime (i.e., "free to use") on each channel. The idle time measure can be regarded as a precise means of measuring the utilization of channels in wireless networks [101, 140]. The amount of idle time experienced by a node in a particular channel is related to the level of traffic load on that channel, where the greater the idle time available, the better the channel likely to be experienced. Then, the idle time allows the node to be aware of the channel contention, and thus to consider the inter-flow interference in a dynamic way. For this reason, the idle time has been widely used as an indication of local available bandwidth, not only by routing approaches [70, 84], but also by other research topics on wireless networks, such as call admission control [141, 142].

To exemplify the benefits of idle time measurement, consider the scenario illustrated in Figure 6.2, in which four flows with different data rates (i.e., packets per second - pps) are active: *Flow 1* at 50 pps with two hops ($E - A - GW$), *Flow 2* at 300 pps with one hop ($B - GW$), *Flow 3* at 100 pps with one hop ($C - GW$), and *Flow 4* at 900 pps with 2 hops ($F - D - GW$). The arrows are used to represent the active flows and their corresponding channels at each hop. Node X is equipped with three interfaces and all other nodes with two interfaces. The channel assigned to each interface is shown inside the square brackets nearby the nodes. For the sake of clarity, we only present the channel assignment of some nodes in Figure 6.2. If node X measures the idle time on each channel, it can be aware of the traffic load and channel contention in its neighborhood. For instance, node X can avoid using

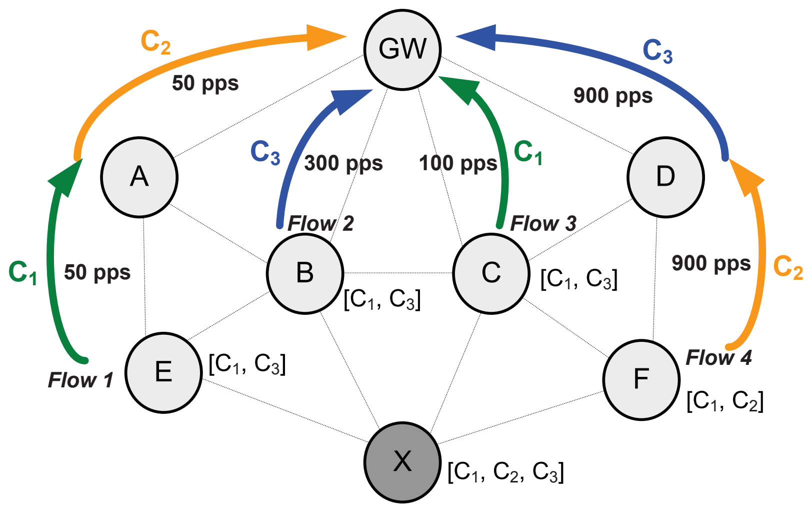


Figure 6.2: Understanding the Idle Time Estimation.

C_2 for its future flows, since this channel is the most overloaded in its neighborhood due to the active flow at 900 pps (*Flow 4*).

The idle time measure presents a set of advantages. First, idle time measure is more accurate than considering the number of active flows in the link, since it does not depend on the data rates of flows. From the point of view of node X in Figure 6.2, although channel C_1 is the channel with the greater number of active flows (i.e., *Flow 1* and *Flow 3*), C_1 actually corresponds to the channel with the lowest traffic load.

Second, the idle time is more efficient than simply estimating interference according to the number of nodes in the neighborhood tuned to the same channel, as performed by the *Metric of Interference and Channel-Switching* (MIC) [100] to estimate inter-flow interference (Section 2.4.1.2). In our example, all four neighbors of node X have an interface assigned to channel C_1, while only one and three neighbors are assigned to channels C_2 and C_3, respectively. Neverthless, C_1 remains a good channel for future communication as it corresponds to the least-used channel in terms of traffic load. Accordingly, idle time measure recognizes the level of interference on a channel independently of the position of the interfering nodes. It treats interference in a dynamic way, since it captures the level of interference over time according to the amount of traffic generated by the interfering nodes.

Finally, the idle time intervals can be measured through a passive monitoring mechanism (Section 2.4.1), and thus does not introduce measurements overhead. If a flow reduces or increases its data rate, this changing will be dynamically reflected in the idle time measures, without any change in terms of the measurement technique.

We use the idle time as one of the parameters of our routing metric. The idle time of a channel c measured by a node v corresponds to the time period in which node v considers the radio medium available for access.

We use the *Clear Channel Assessment* (CCA) operation (Section 2.1.2) to determine whether/when the wireless medium is busy or idle. Specifically, the Physical Layer Convergence Protocol (PLCP) informs the MAC layer about the status of the channel through the primitive PHY-CCA.INDICATE(STATUS) [42]. If the channel is idle, the PLCP sends a PHY-CCA.INDICATE(IDLE) primitive to the MAC layer. Otherwise, if the channel is busy, the PLCP sends a PHY-CCA.INDICATE(BUSY) primitive to the MAC layer. The PLCP within the node senses the channel continually. Thus, through the CCA operation, a node is able to constantly monitor the *channel state transitions* (i.e., from busy to idle or from idle to busy) and then record the time that the channel is in each state.

We reduce the excessive sensitivity of physical layer measurements variation adopting the statistical function Fixed History Window (FHW), in which old measurements are dropped as new measurements come available. In particular, FHW provides an average figure from a fixed number of previous measurements or from the measurements in a time interval. For our performance evaluation (Section 6.4), we calculate an average value of idle time from a fixed number of previous measurements θ. Each node locally maintains a table with the idle time values experienced by each channel assigned to its interfaces. The main idea is to give equal weights to the most recent idle time measures. For example, we set a value of $\theta = 25$ to calculate the average idle time in our simulations. Smoothing out the value of idle time intervals avoid unnecessary route oscillations.

Since each node can estimate its own local idle time, we denote by $Idle_{v,c}$ the average idle time measured by node v on channel c. Also, we use $Idle(l)$ to denote the idle time of a link l established between node v and node u through channel c. The available air time for communication between two nodes on a channel c is equal to the minimum of their idle time:

$$Idle(l) = \min(Idle_{v,c}, Idle_{u,c}), \tag{6.1}$$

In the example of Figure 6.2, the minimum idle time of link $E - X$ on channel C_1 is determined by node X. This is because node X is aware of the active flows *Flow 1* and *Flow 3* on channel C_1, while node E is aware only one of *Flow 1*. Consequently, $Idle_{(X,C_1)} < Idle_{(E,C_1)}$.

In Section 6.2.3, we describe how the $Idle(l)$ of each link l that form a path p is used to calculate the final path metric.

6.2.2 Link Quality Estimation

It is not advantageous to rely only on the quality of channels (e.g., level of congestion) to perform traffic forwarding. Besides being aware of the quality of channels, it is fundamental for a node to be aware of the quality of each connection (i.e., link) established through each channel. The main reason is due to the fact that some links tuned to unloaded channels may have poor quality (e.g., high loss rates), which may result in a high number of retransmissions and thus to an inferior performance than congested links. Thus, a node must be aware of both quality of channels and quality of links to perform routing through good paths.

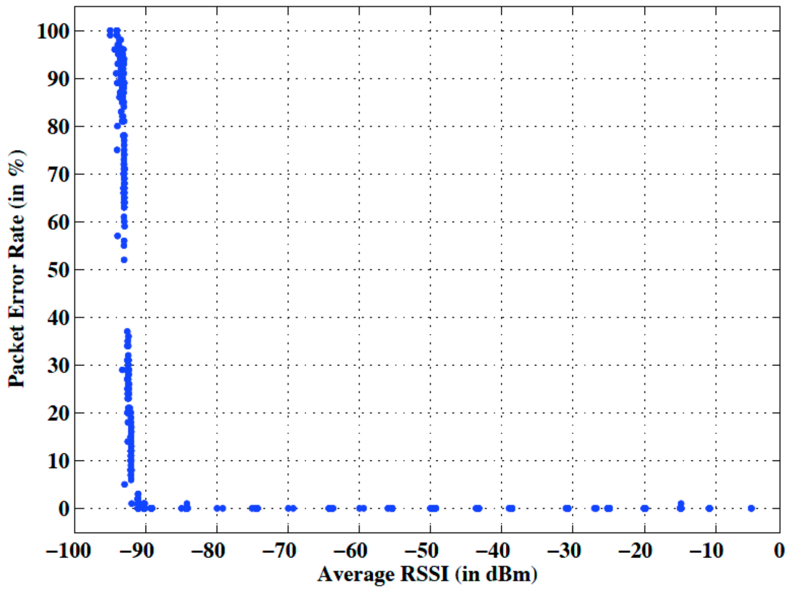
To analyze the quality of links, we focus firstly on the Received Signal Strength Indicator (RSSI), shown in Level 2 of Figure 6.1. As explained in Section 2.2.1, the RSSI is a typical measure of the PHY layer [21]. It indicates the strength of the arriving signal at a receiver.

In Table 2.4 (Page 31), we have shown that RSSI presents some drawbacks that limits its use as a good indicator of inter-flow interference. Indeed, it is not our goal here to measure RSSI as an estimator of interference. We highlight that our routing solution uses other measures to estimate inter-flow and intra-flow interference. The basic idea is to took advantage of RSSI to classify links as either *good* or *bad* basing on the RSSI variability [138, 143]. Somehow, we aim to reflect in our link metric the impact of the choice of *bad links*, since in case these links are selected to be part of a path, they would negatively impact the routing performance.

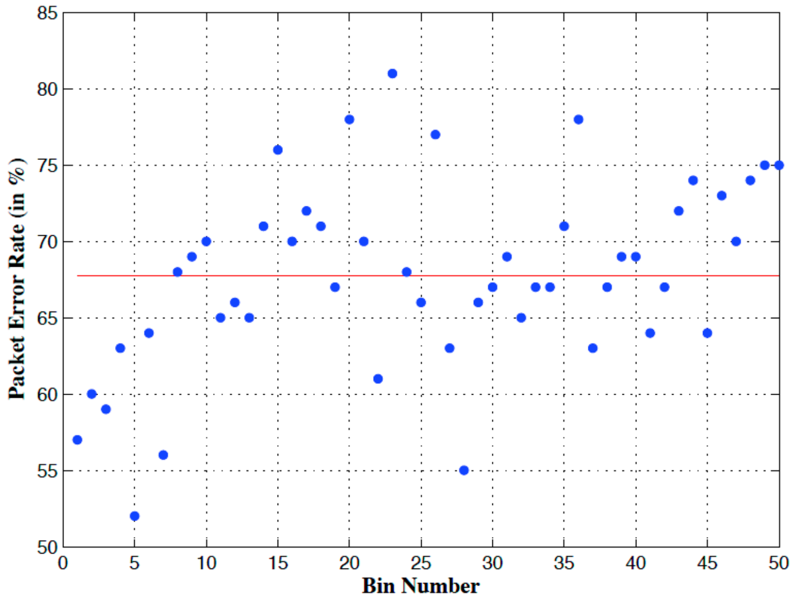
The idea to measure RSSI as an estimator of the quality of a link is supported by recent works [144, 145]. They have shown that RSSI is a promising link quality indicator ***if*** correctly analyzed in conjunction with the operating zone of the link. More specifically, links are categorized according to the observed Packet Error Rate (PER), which corresponds to the number of incorrectly received data packets divided by the total number of received packets.

Accordingly, we propose to study the characteristics of RSSI based on the observed PER, and develop a real-time method to enable nodes to categorize links into two types:
(1) **good link**: a low-loss zone where the links observe a low PER;
(2) ***gray zone* link (or bad link)**: a zone where PER fluctuates from high to low.

To illustrate this categorization of links, Figure 6.3 presents part of the results of testbed experiments conducted by Raman *et al.* in [146]. They used a setup with two nodes (i.e., one transmitter and one receiver) with different calibrations of radios. More specifically, the authors varied the attenuation from 0 dB to 93 dB in steps in order to alter the received signal strength measured at the receiver. In each instance, they conducted an experiment where the transmitter sent a series of



(a) RSSI vs. PER [146].



(b) Error Rate Variation [146].

Figure 6.3: Understanding the Categorization of Links According to the observed RSSI and PER.

5000 packets to the receiver. The packets were divided into bins of 100 transmitted packets. For each such bin, they compute the average RSSI as well as the average

PER. Figure 6.3(a) shows the scatter plot of this data. For instance, the closer the average RSSI value to 0 dBm, stronger is the signal.

As shown in Figure 6.3(a), good links are presented in zones where the RSSI is higher than a certain *sensitivity threshold* δ (around -90 dBm in the figure), which indicates that *good links* have low (i.e., almost 0%) and constant PER with a very low variance. At the edge of this threshold, the RSSI values enter into a *gray zone* where PER varies from low to high values over a small RSSI range (about 5-6 dB). To better illustrate this behavior, Figure 6.3(b) shows the observed error rate in each bin against the time (i.e., bin number in y-axis) with the attenuator setting fixed at 90 dB. We see that there is a large temporal variation in the PER, in the time-scale of 2 seconds (100 packets at 20 ms inter-packet gap). Similar behavioral aspects of good links and gray zone links were observed by other testbed experiments conducted by Srinivasan and Levis in [147], Kolar *et al.* in [138, 144], Rondinone *et al.* in [143], among others [145, 148, 149]. In conclusion, PER of gray zone links is not a good indicator of the error rate of a link due to this variation.

In this context, remember that the ETX of a link (c.f., Section 2.4.1.1) is calculated on the basis of error rate estimation of the link [90]. More specifically, the probability of error of distinct packets is assumed to be an independent and identically distributed process [30]. Thus, if successive packets were lost independently with probability equal to the average PER, ETX would be accurate [150]. However, as explained previously, PER is unstable in *bad links*. In consequence, as ETX does not consider the categorization of links, ETX is inaccurate on the estimation of PER and then in the final metric computation [151]. Also, the routing metrics derived from ETX suffer from the same drawback (e.g., ETT, $WCETT$, MIC, and $iAWARE$).

To avoid bad routing decision-making, we use measured values of RSSI and PER captured in the wireless mesh network as decisions parameters to optimize the system. Similarly to Ashraf *et al.* [152], we argue that if nodes are able to classify links as *good* or *bad*, more stability is given to the routing protocols and hence the network performance can be improved. A major advantage of our solution is that it performs this evaluation of links through the RSSI recorded by commercial wireless card [65]. The RSSI value has not to be derived from other measures, such as performed by other approaches that use SINR [63, 68, 152] (c.f., Table 2.4, Page 31).

From the results of Figure 6.3, it is clear that we cannot use a single measure of RSSI and PER as an estimate of the link quality [146]. Besides, the sensitivity threshold (i.e., link transition from the low-loss zone to the gray zone) may vary from different nodes and then cannot be assigned as a constant from one-time measurement, as demonstrated by Kolar *et al.* in [144]. Then, in our solution, each node maintains a table in which it registers the RSSI values measured for all packets correctly received. In this table, the measurements are differentiated for each neighbor, as well as for each channel in common with this neighbor (i.e., all neighboring links in the Network Topology). Through this table, a node can smooth the RSSI through its average ($RSSI_{avg}$). Then, given the $RSSI_{avg}$ and the PER variability, the node can analyze and compare the properties of both measures to categorize the link (i.e., good or bad).

In this way, we propose here a second Intermediate Metric (c.f., Level 3 in Figure 6.1) to estimate the total amount of time it would take to send a data packet along a particular link. This metric is called *Expected Transmission Airtime* (ETA). We calculate the metric of a link l as follows:

$$ETA(l) = \left(Overhead_{ch} + \frac{Size_{pck}}{Rate(l)}\right) * Q(l), \tag{6.2}$$

where $Overhead_{ch}$ is the channel access overhead (e.g., frame headers, training sequences, access protocol frames, etc), $Size_{pck}$ is the packet size in bits, $Rate_l$ is the data rate in megabits per second, and $Q(l)$ is a configurable parameter ($Q(l) > 0$) to express the impact of the quality of links in the metric.

Note that ETA approximates ETT (Equation 2.3) and ALM (Equation 2.4) metrics. The first part of Equation 6.2 reflects the expected time to transmit a packet over link l. It incorporates two important parameters into the metric: the packet size and the data rate of each link (c.f., Level 1 in Figure 6.1).

The second part of Equation 6.2 reflects the impact of the quality of a link on the expect time to transmit a packet. Basically, our new link metric differs from ETT and ALM by this second part. Our logic behind $Q(l)$ is that weak signal strength may increase the amount of airtime to transmit a packet over a link. For example, a number of retransmissions may be necessary in order to successfully send a packet over a weak signal strength link. Thus, $Q(l)$ and $ETA(l)$ must be parameters directly proportional: a higher $Q(l)$ value increases $ETA(l)$. However, since the error rate is unstable for *bad links*, quantify this time in terms of additional transmission time is a challenging task. Nevertheless, we argue in favor to represented this additional time in the *Expected Transmission Airtime* (ETA) metric, even if the weight does not correspond to the real impact in terms of airtime.

Inspired by other approaches [63, 153, 154], we simply include $Q(l)$ as a configurable parameter to aggregate an additional cost to the metric. In our case, $Q(l)$ is intended to aggregate an additional cost when links are categorized as *bad links*: bad quality transmissions use more resources than good quality ones. If a link is categorized as *good*, then $Q(l) = 1$ because the performance of a *good link* is the same irrespective of how high the RSSI is above the sensitivity threshold [138]. If a link is categorized as *bad*, then $Q(l)$ receives a value greater than one ($Q(l) > 1$) to increase the time calculated by Equation 6.2. For instance, we set $Q(l) = 1.5$ in our performance evaluation. Thus, note that here we are not evaluating the degree of "*badness*" of the link. For now, we are just weighing that a link was categorized as *bad* for communication and then that we have to punish this link in terms of additional *Expected Transmission Time* (ETA). Nevertheless, we agree that an in depth study of $Q(l)$ has to be performed if the aim is to derive the degree of "*badness*" of the link. Following a similar approach, Ashraf *et al.* [152] have performed extensive simulations to show how the network is impacted when the degree of "*badness*" of the link is varied.

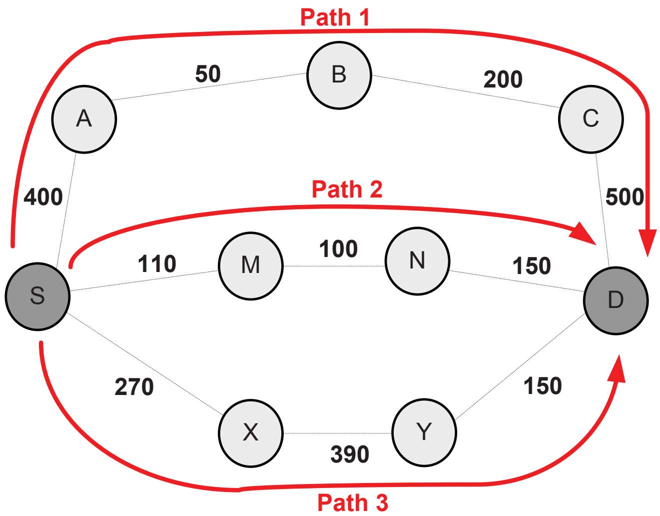
Despite the simplicity of our solution, $ETA(l)$ positively impacts the network performance, as stable links will have priority in routing.

6.2.3 Path Metric Calculation

As we have presented how the quality of channels and links are estimated, we discuss now how to merge the corresponding metrics to calculate the final path metric (i.e., Level 4 in Figure 6.1).

As previously mentioned, the main goal of our path metric is to estimate the residual bandwidth of a path (*Path Residual Bandwidth* - PRB). Following the definition of Wang and Crowcroft [155], the bandwidth of a path is defined as the minimum of the residual bandwidth of all links on the path or the *bottleneck bandwidth*. Roughly speaking, our path metric aims to avoid the selection of paths with bottleneck links (i.e., links with little residual bandwidth), even through these links have an acceptable quality.

As an example, consider the scenario shown in Figure 6.4, in which the value close to each link represents the residual bandwidth of this link. Suppose the existence of three paths between the source node S and the destination node D. First, note that in our example *Path 1* presents the links with highest residual bandwidth (link $S-A$ with 400 and link $C-D$ with 500). At the same time, this path presents the link with the lowest residual bandwidth (link $A-B$ with 50), which limits the overall performance of the other links in *Path 1*. In relation to the second path, the residual bandwidth values are more balanced than those of *Path 1*. Besides that, the link with the lowest residual bandwidth in *Path 2* (link $M-N$ with 100) has a greater value than that of *Path 1* (link $A-B$ with 50). For this reason, our path metric will consider that *Path 2* is a better path than *Path 1*. Now, if we analyse *Path 3*, we can found that the link with the lowest residual bandwidth in this path is link $Y-D$ with 150. As highlighted in the table of Figure 6.4, *Path 3* is the path



PATH	LINK MIN RESID. BANDWIDTH	VALUE
Path 1	A-B	50
Path 2	M-N	100
Path 3	Y-D	150

Figure 6.4: Understanding the Path Metric Calculation.

in which the link with the lowest residual bandwidth presents the highest value. For this reason, the path metric PRB will choose *Path 3* as the best path to forward packet between S and D.

Thus, the PRB metric for a path p can be defined as follows:

$$PRB(p) = \left(\min_{\text{link } l \in p} LRB(l) \right) * hop_p^{-\eta} \tag{6.3}$$

where $LRB(l)$ is the *Link Residual Bandwidth*, hop_p is the number of hops of path p and η is a configurable parameter subject to $0 \leq \eta \leq 1$. The higher the value of PRB, the better is a path for packet forwarding.

The parameters hop and η are introduced to balance between bandwidth and path length, since PRB is a concave metric that does not distinguish paths in relation to the number of hops [156]. Consequently, PRB may select too long paths, which may increase the probability of a packet being dropped along the path (e.g., buffer overflow and transmission failures). To limit the selection of long paths, a simply solution would be to set the maximum number of hops for a valid path [155]. However, we propose a more flexible solution with hop and η parameters, in which the higher η, the higher the weight given to the number of hops in the path metric calculation. For instance, if $\eta = 0$, the number of hops has no impact in PRB. When η increases, the PRB metric decreases. Thus, η can be seen as a flexible way to set a limit to the selection of long paths. For instance, we choose $\eta = 0.3$ in our performance evaluation scenario.

Since we know how to calculate the path metric, we focus now on the calculation of the link metric *Link Residual Bandwidth* ($LRB(l)$).

At this point, remember that we are aware of the two main information: first, the "free to use" time for each link (metric $Idle(l)$); second, the expected airtime to transmit a packet over a particular link (metric $ETA(l)$). Then, note that $\frac{Idle(l)}{ETA(l)}$ can be seen as the *expected number of packets that can be sent over a particular link* l. For instance, we could simply use this information to determine the residual bandwidth of a link. However, this solution would not be as effective as other proposed multi-interface routing metrics, since it does not incorporate, for example, the intra-flow interference and channel diversity.

Then, what we propose here is to continue to take advantage of the relation between $Idle(l)$ and $ETA(l)$, but combine these metrics in a novel way in order to calculate a path metric that incorporates intra-flow and inter-flow interference, traffic load, link-quality, packet size, data rate, among other important design issues.

To achieve this goal, we propose to calculate the residual bandwidth of a link l as follows:

$$LRB(l) = \frac{Idle(l)}{ETA(l) + \sum_{\text{conflicting links } i \text{ on } l} ETA(i)} \tag{6.4}$$

where $Idle(l)$ corresponds to the idle time calculated through Equation 6.1 and ETA is the expected transmission airtime calculated through Equation 6.2.

To address intra-flow interference, we give priority to paths with greater channel diversity. More specifically, the denominator of Equation 6.4 accumulates the ETA of link l with the ETA of all other links i in path p that are considered conflicting with link l. We consider the two following conditions to classify a link i as conflicting:

1. Following the two-hop interference model [157], the two previous links of link l on path p if they are operating on the same channel as link l.

2. The other previous links of link l on path p if they are operating on the same channel as link l **and** are interfering on link l. To be aware of the interfering links, each node locally maintains a table with the MAC address of each interface of each neighbor. This table associated with the information carried in the path discovery messages (e.g., nodes and channels of each link traversed) allows the node to know the nodes interfering in each of its links.

When the accumulated ETA increases in Equation 6.4, the LRB metric decreases representing the impact of a poor channel diversity. In this way, we improve $WCETT$ [30] and $iAWARE$ [68], since these metrics consider that two links on a path operating on the same channel are always interfering with each other no matter the distance between them. We also improve MIC [97] and $MIND$ [63], as these metrics consider only the previous hop to depict the intra-flow interference.

At the end, note that through Equations 6.3 and 6.4 nodes are able to calculate paths incorporating two important factors: first, the intersection of idle time of

Design Issues	**How it is addressed?**
Hop Count	hop and η
Packet Size	$ETA(l)$
Data Rate	$ETA(l)$
Inter-flow Interference	$Idle(l)$
Load-Aware	$Idle(l)$
Link-Quality	$Q(l)$– through $RSSI$
Intra-flow Interference	$LRB(l)$
Channel Diversity	$LRB(l)$
Stability	FHW– through θ

Table 6.1: How Design Issues are Incorporated by the Routing Metric.

interfering links along the path; second, the fact that two interfering links are not able to communicate at the same time.

We conclude this section highlighting that our path metric put together all the positive aspects captured by previously proposed routing metrics (c.f., Table 2.6), such as interference, traffic load, link quality, packet size, data rate of each link, stability, etc. Table 6.1 summarizes how our routing solution incorporates the most important design issues. In relation to previous approaches, one of the main advantages of our solution consists in the way information is gathered and then combined to evaluate a path. We take profit of measures already captured by the system to allow nodes to be aware of the quality and dynamicity of the network. So, our solution is suitable for different environments (e.g., independent of the network topology, number of available channels and interfaces). Furthermore, the fact of using passive monitoring to collect the cross-layer measures favors our solution, since it does not increases the overhead on the network. In addition, to the best of our knowledge, the way our measures and metrics are combined was not addressed by existing studies.

6.3 Path Selection Protocol

Although routing protocols play a key role in the performance of WMN, this area of research has not received the same attention as routing metrics in the context of multi-channel multi-interface networks. Most of the metrics were evaluated with conventional routing protocols such as AODV [12], DSR [14] and LQSR [108]. In consequence, the capabilities of multiples channels and interfaces are not fully explored. These protocols may even compromise the proper operation of metrics, especially with regard to the increased overhead resulted from the replication of routing control messages in multiple interfaces.

In this section, we describe the design of a new multi-channel multi-interface path selection protocol derived from AODV protocol. In the following, we will refer to our routing protocol as Multiple Interface AODV (MI-AODV). This new protocol is an on-demand distance vector solution based on Bellman-Ford algorithm. In this way, we can find efficient paths without forwarding loops even when the metric is non-isotonic [98, 99].

The main goal of our routing protocol is to select paths with the highest *Path Residual Bandwidth* (PRB).

Path Discovery Process

A node begins a path discovery process, when it has a packet to send to some destination node and does not have a path to it. During path discovery, it broadcast a Path Request (PREQ) message in each of its interfaces, which are tuned to different channels. PREQ are sent with increasing sequence numbers to ensure that nodes can distinguish current path information from stale path information at all times. Thus, sequence numbers are used to discard duplicate PREQs received from other nodes and to prevent routing loops.

In order to compute our path metric when the PREQ traverses the network, we need the link metric (based on the path metric ETA computed using Equation 6.2) for each link traversed and the channel in which they are operating. So, PREQ messages are overloaded to carry the link metric and the channel of each link traversed.

When an intermediate node receives a PREQ, it first checks the sequence number to see if it has already received the PREQ. If the node has not received this PREQ before, it creates a reverse path entry to the source of the PREQ message and associate it to the flow (i.e., pair source→destination). Then, it appends the link information (e.g., ETA metric and corresponding channel) and forwards the PREQ in each of its interfaces. This replication of PREQ in each interface is important to ensure that all neighbors with at least one channel in common will receive at least one copy of the PREQ message. In this way, our routing protocol can be applied in mesh scenarios with different *Channel and Interface Assignment*. If the PREQ has a better path metric PRB (based on the path metric computed using Equation 6.3), it updates the reverse path accordingly and then forwards the PREQ by broadcasting in each of its interface.

When the destination node receives the PREQ, it sends a Path Reply (PREP) message by unicast toward the source along the reverse path built during the path discovery. In our protocol, PREP messages are not overloaded like PREQs. As explained above, intermediate nodes are not interested in using the same forward path to the destination node as other active flows. We allow intermediate nodes to send PREP messages if they already have an entry to the current flow.

When the source node receives the PREP, it builds the path to the destination and sends out the queued data packets.

Path Maintenance Process

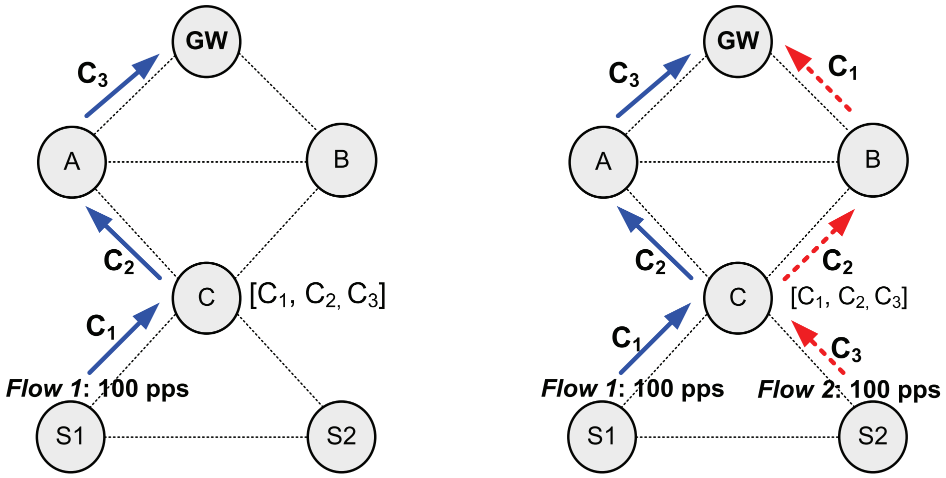
When an active path breaks, the intermediate node that detects the link break has to send a Path ERRor (PERR) message to communicate the error to the sources of affected paths. Thus, PERR is sent backwards to the precursors of all paths that are affected by the link break. The mesh nodes receiving the PERR remove the broken path(s) from their forwarding information. When the PERR reaches the corresponding sources, they will initiate a path discovery in order to find an alternative path.

Load-balancing

In our protocol, we aim to distribute the load across the channels to avoid the formation of congested areas. For example, consider Figure 6.5. First, assume that a path is already established for *Flow 1* ($S1 - C - A - GW$), as illustrated in Figure 6.5(a). After some time, the intermediate node C receives a path discovery request from another source node $S2$, that wants to communicate with the same destination GW as $S1$. Despite the fact that node C already know a valid path to this destination (i.e., being used by an active flow), our protocol aims to explore the diversity of available channels in order to reduce congestion and achieve a better throughput. In this regard, our path selection protocol gives priority to less congested links (e.g., based on the $Idle$ parameter). As illustrated in Figure 6.5(b), the traffic of *Flow 2* ($S2 - C - B - GW$) is forward through a different path to avoid interference with *Flow 1*. Accordingly, our routing protocol fully utilizes the bandwidths on channels and tries to protect existing flows.

Reducing the Overhead of Control Messages

The overhead introduced by flooding of control messages (e.g., PREQ) can seriously impair the network performance. This is a well know problem in single-interface scenarios (i.e., *broadcast storm problem* [105]) that causes severe performance degradation due to heavy contention and collisions. In multi-interface scenarios, this problem is even worse, since control messages are retransmitted by every interface. In particular, the amount of transmitted control messages exponentially increases with the number of interfaces [104].



(a) *Flow 1* is established to explore channel diversity.

(b) *Flow 2* avoids interfering with the active *Flow 1*.

Figure 6.5: Operation of our Path Selection Protocol.

To reduce the overhead of control messages, we introduce two strategies that depend upon the role of the node in the path discovery process:

- **Intermediate Node**: when an intermediate node receives a new PREQ, it waits a period of time T_i in order to receive eventual replications of this PREQ message from other neighboring nodes. In this way, intermediate nodes can evaluate the received PREQ and broadcast only one of the received replications, based on the one with the better path metric. This strategy significantly reduces the multi-interface overhead, especially when it is analyzed in a global scale.

- **Destination**: when a destination node receives a PREQ message from a new flow for the first time, it waits a period of time T_d to allow other discovered paths to reach it. This procedure improves the stability of the network, since it prevents intermediate nodes to send multiple PREPs and thus the source node to receive multiple PREPs in short time intervals.

6.4 Performance Evaluation

We have validated our approach by simulations in the NS-3.13. We have developed extensions to support the cross-layer interaction, especially regarding the information captured by the PHY layer such as the Idle Time and RSSI values.

Parameter	**Default Value**
Simulation Time	80s
Network Size	50 nodes—1 gateway
Number of Interfaces	3
Assignment Strategy	Static Interfaces/Common Channel Assignment
Topology Type	Random Disk
Radius Disk	650
Traffic Type	Poisson
Traffic Load	700 packets per second (pps)
Number of Flows	6
Packet Size	1024
PHY specification	IEEE 802.11a
Propagation Model	Rayleigh
$RSSI_{max}$	-82 dBm
$Q(l)$	1.5
θ	25
η	0.3
β ($WCETT$)	0.5

Table 6.2: Simulation Parameters.

We have evaluated the performance in different random disk scenarios with 1 gateway (i.e., Mesh Portal) and 49 static mesh routers with multi-interface capabilities. Each node has on average six neighbors in the random disk. Mesh routers are equipped with IEEE 802.11a wireless interfaces and therefore 12 orthogonal channels are available [2]. On each node, interface i is assigned to the orthogonal channel C_i (i.e., *Static Interfaces/Common Channel Assignment* defined in Section 3.5). To represent a typical WMN traffic pattern, several flows are originated from the mesh routers to the gateway.

Each data point in the graphical results is computed as being a result of 20 different simulations. For each simulation run, a node is randomly designated as the gateway to receive the Poisson traffic generated by the transmitter mesh routers. Thus, the variations in the obtained results mainly occur due to the randomness of the topology, the gateway and transmitters positions. An initial period of 10 seconds of the simulation is undertaken before the flows start (i.e., "warm-up period"). We have considered a packet size of 1024 bytes. The total simulation time is 80 seconds.

The characteristics of the scenario used for the performance evaluation are summarized in Table 6.2. Most of these configurations were defined according to previous comparative studies [20, 88, 97].

We use the notation *Routing Metric/Routing Protocol* to designate each simulated approach:

- **PRB/MI-AODV**: corresponds to our new metric and protocol approaches detailed in Sections 6.2 and 6.3, respectively;
- **ALM/HWMP**: corresponds to the metric and mesh protocol proposed by the IEEE 802.11s standard [19], as described in Sections 2.4.1.1 and 2.4.2, respectively;
- **WCETT/HWMP**: corresponds to the metric proposed by De Couto *et al.* [90] and the routing protocol of IEEE 802.11s [19]. The value of β in Equation 2.5 was set to 0.5 [88, 97].

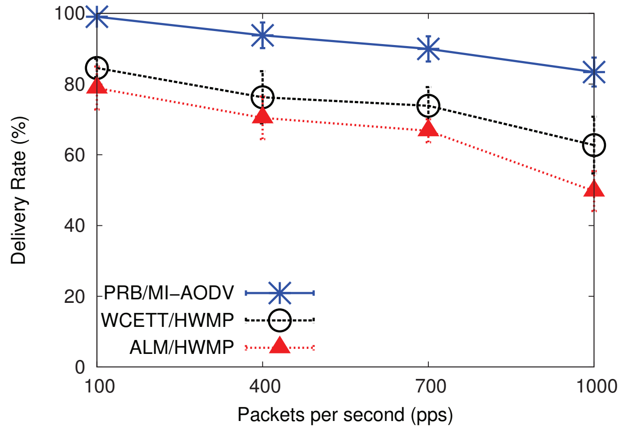
Impact of Traffic Load

We first investigate the impact of the traffic load on the performance of routing metrics and protocols.

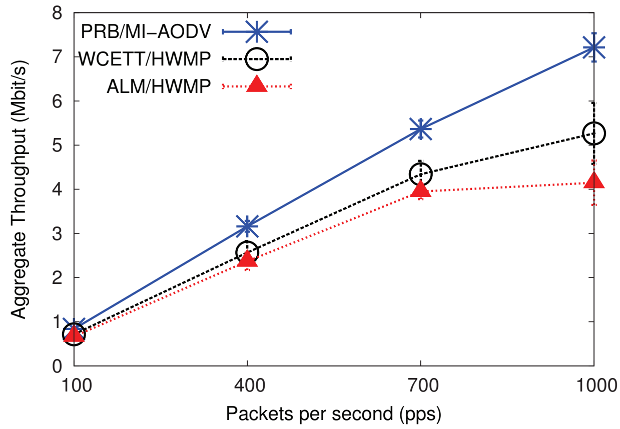
We observe that our solution *PRB/MI-AODV* presents better delivery rate (Figure 6.6(a)) and aggregated throughput (Figure 6.6(b)) than *ALM/HWMP* and *WCETT/HWMP* regardless the traffic load. Also, Figures 6.6(c) and 6.7(a) show the lowest delay and the lowest number of drops at the MAC layer, respectively, when our approach is used. Moreover, as the traffic load increases, the performance gap between our solution and the other solutions also increases.

Our solution presents the highest throughput because it considers inter-flow and intra-flow interference, traffic load and link-quality. As expected, the performance of *ALM/HWMP* is lower than *WCETT/HWMP*. The *ALM* metric does not measure interference and does not take into account the traffic load. Although *ALM* and *WCETT* metrics are based on the expected transmission time, *WCETT* results in higher throughput and lower delay than *ALM*, since it is able to pick up intra-flow interference taking into account the channel diversity.

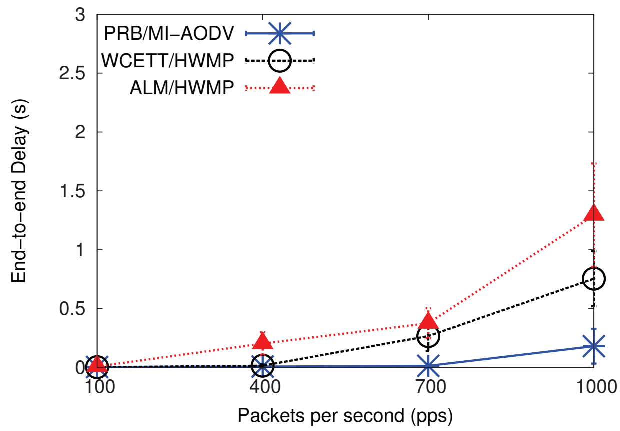
In Figure 6.6(a), *PRB/MI-AODV* achieves a delivery rate of almost 100% when packets are transmitted at 100 pps. In contrast, *ALM/HWMP* and *WCETT/HWMP* present a lower delivery rate, approximately 78.8% and 84.5%, respectively. This is mainly because *ALM* and *WCETT* do not explicitly consider the link quality when selecting a route, which may result in a number of retransmissions when *bad links* (i.e., *gray zone links*) are selected. In addition, *HWMP* does not balance the load across the available channels when new flows arise, as discussed in the example of Figure 6.5. In view of this, *ALM/HWMP* and *WCETT/HWMP* present a considerable increase in terms of the number of drops at the MAC layer when the



(a) Delivery Rate.

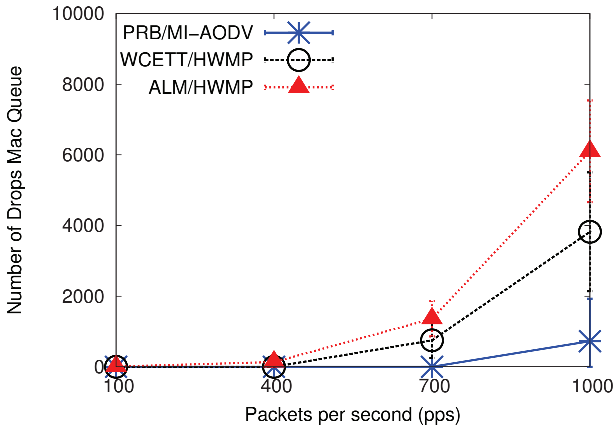


(b) Aggregate Throughput.

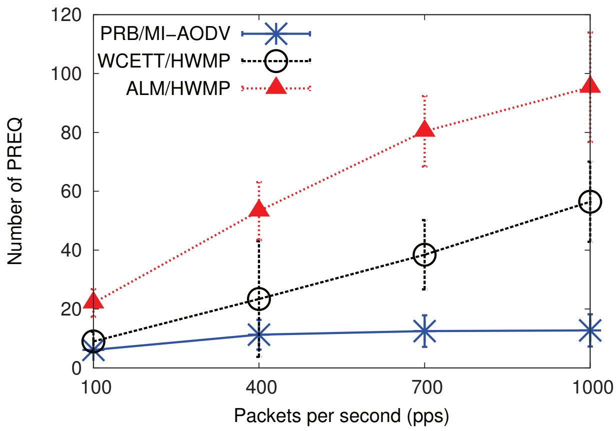


(c) End-to-end Delay.

Figure 6.6: Impact of Traffic Load – Part 1.



(a) Drop MAC Queue.



(b) PREQ Messages.

Figure 6.7: Impact of Traffic Load – Part 2.

data load increases (c.f., Figure 6.7(a)), which also impacts on the end-to-end delay (c.f., Figure 6.6(c)). Conversely, as our solution estimates the traffic load through the idle time intervals, new flows are routed in direction of less congested channels. This strategy avoids in part the accentuated fall of *PRB/MI-AODV* when data rate increases in Figure 6.6(a). For instance, while *PRB/MI-AODV* decreases approximately 16% from 100 pps to 1000 pps, this decrease is much greater for *ALM/HWMP* ($\cong$ 30%) and *WCETT/HWMP* ($\cong$ 23%). We highlight that many of the packets not delivered by our solution are due to the buffer overflow suffered by nodes close to the gateway.

We then studied the routing overhead measuring in particular the number of transmitted Path REQuest (PREQ). As shown in Figure 6.7(b), our strategies to reduce the overhead of control messages significantly reduces the number of transmitted PREQ in relation to *ALM/HWMP* and *WCETT/HWMP*. For these two approaches, the number of transmitted PREQ significantly increases with the traffic load, which results in the performance degradation due to overhead, collisions, and contention, as shown in Figure 6.6(a).

Impact of the Number of Flows

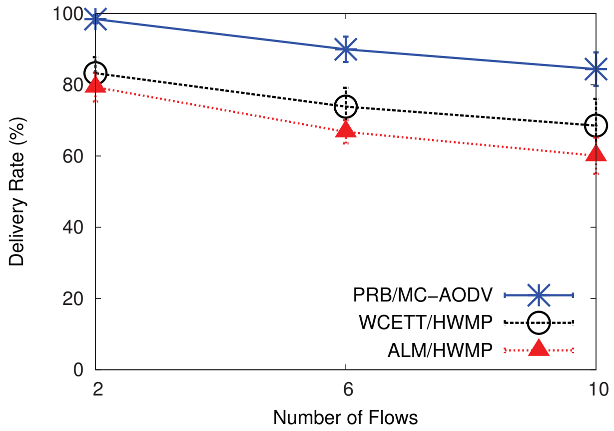
Next, we investigate the impact of the number of flows for 700 pps and 3 interfaces, as shown in Figure 6.8.

Similarly to the previous results, we observe that *PRB/MI-AODV* presents better delivery rate (Figure 6.8(a)) and aggregated throughput (Figure 6.8(b)) than the two other approaches regardless the number of active flows. Also, the end-to-end delay slightly increases with the number of flows when our solution is implemented. The main reason is that *PRB/MI-AODV* performs load-balancing and then seeks to distribute the various flows through different paths in order to prevent channel congestions (i.e., bottleneck links). In particular, the *Idle* metric is used with the purpose of measuring the residual bandwidth available at each channel. To distribute the flows, *PRB/MI-AODV* also favors paths with *good quality links*, which are identified according to the cross-over RSSI measures captured locally by each node during packet reception. The good performance of our solution is the result of the combination of these factors with other design issues such as packet size, data rate and channel diversity.

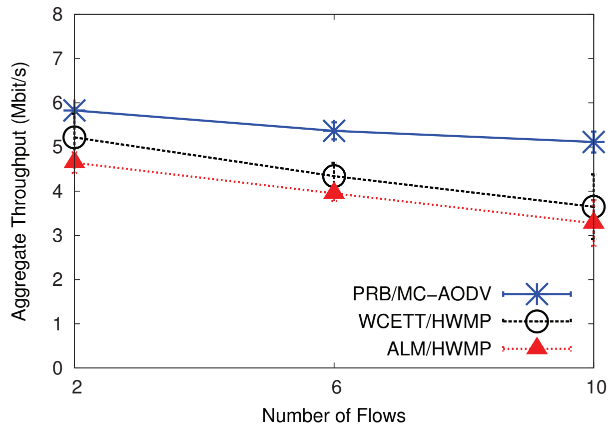
On the other hand, *ALM/HWMP* and *WCETT/HWMP* are negatively affected in terms of aggregated throughput and end-to-end delay because they do not consider dynamic characteristics of the network to perform routing, such as the traffic load and the link quality. Besides, *HWMP* fails to provide load-balancing. More specifically, the IEEE 802.11s routing protocol tends to forward new flows through already established paths, which means that all flows are led to the same congested areas. Consequently, it results in poor quality paths that spend more time to transmit data because of the high level of interference and contention. In particular, although *WCETT* considers the intra-flow interference (i.e., channel diversity), it does not fully take advantage of the available channels. As explained in Section 2.4.1.2, *WCETT* assumes that links on a path tuned to the same channel are always interfere with each other no matter the distance between them. Therefore, this assumption is somehow pessimistic, especially in mesh networks with longer paths, such as the random topologies simulated in this work.

Impact of the Number of Interfaces

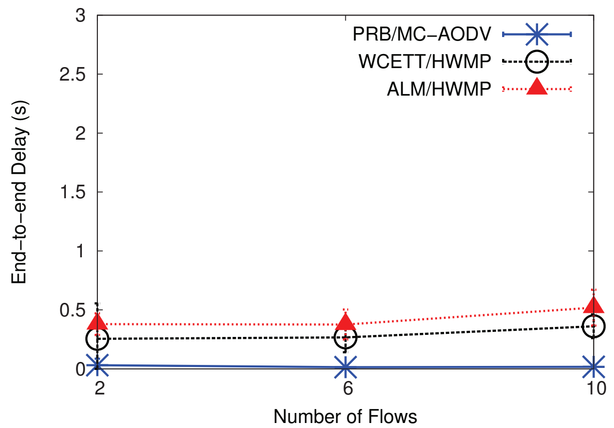
Finally, we study the impact of the number of interfaces. For all solutions, deliv-



(a) Delivery Rate.

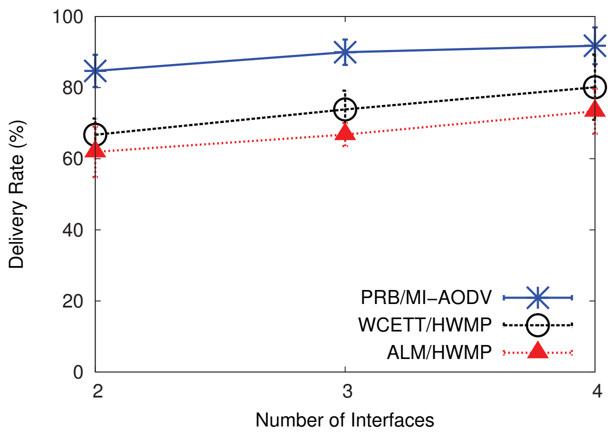


(b) Aggregate Throughput.

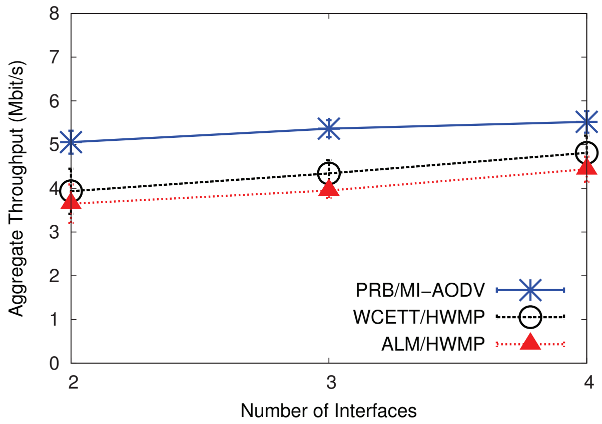


(c) End-to-end Delay.

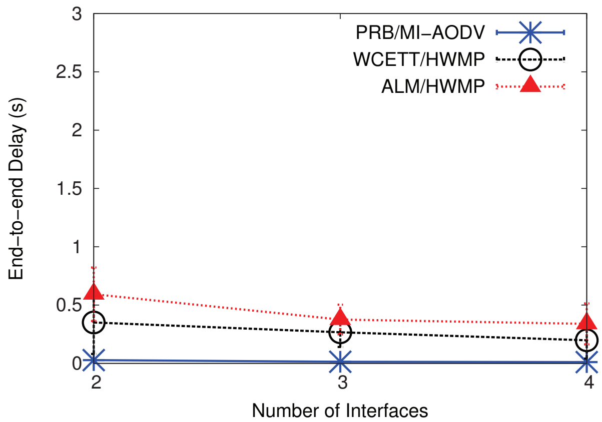
Figure 6.8: Impact of the Number of Flows.



(a) Delivery Rate.



(b) Aggregate Throughput.



(c) End-to-end Delay.

Figure 6.9: Impact of the Number of Interfaces.

ery rate and aggregate throughput increase and the end-to-end delay decreases with the number of interfaces. Note that the increase of the number of interfaces tends to approximate the approaches in terms of delivery rate and aggregate throughput. This behavior is expected as the increase of interfaces also increases the number of paths using different channels. Consequently, intra-flow and inter-flow interference can be reduced. However, the end-to-end delay does not decreases significantly for the two approaches using *HWMP*. In part, this is due to the high overhead caused by the replication of control messages in all interfaces. For all cases, we believe that the routing protocol must be improved in order to optimize the use of channels and reduce the overhead of routing control messages (i.e., PREQ messages), especially close to the gateway, where nodes may suffer from quickly buffer overflow with the increase of the traffic load.

6.5 Conclusion

We have presented a novel cross-layer routing metric, named *Path Residual Bandwidth* (PRB), to improve the performance of multi-channel multi-interface wireless mesh networks. The key properties of the metric is that it captures both intra-flow and inter-flow interference, link-quality and traffic load, while considering the diversity of channels. We incorporated this metric in a new routing protocol, called Multi-Interface AODV (MI-AODV), which is based on the well-known AODV protocol. In particular, this new routing protocol is able to find path with high residual bandwidth and efficiently distribute the load across the network. Our simulations results showed superiority of our approach; specifically, we showed that in contrast to existing link metrics (e.g., ALM) and path metrics (e.g., $WCETT$), PRB finds paths with less interference and good channel diversity.

One direction for improvement is the study of the impact of different network conditions (i.e., parameters summarized in Table 6.2) on the performance of our routing metric and protocols. In particular, we plan to quantify the influence of each parameter in the final metric calculation. Also, we aim to quantify the impact of different topologies and densities, packet sizes, and data rates in our approach, as well as in other approaches that consider traffic variations. As previously mentioned, how to represent the degree of *badness* of a link in the metric computation is particularly interesting. Besides, we believe that the routing protocol must be improved in order to reduce the control overhead and then to better exploit the diversity of channels and interfaces.

Chapter 7

Conclusions

The aim of this dissertation is to contribute to the performance improvement of multi-channel multi-interface wireless mesh networks. The dissertation contributes in the following fundamental areas for IEEE 802.11-based wireless mesh networks: conceptual channel and interface assignment framework for providing connectivity guarantees, broadcast solutions to reduce the network overhead, capacity bounds to evaluate the network performance, and routing metric and protocol to improve the overall network throughput.

The first contribution corresponds to a comparison and performance evaluation of the existing channel and interface assignment strategies addressing the connectivity issues: network topology, density of connections, and neighbor discovery. We have defined a formal framework that classifies interfaces according to their behavior in the network, as well as categorizes the channel assignment to decide which channels to assign to interfaces. Then, we have proposed a set of strategies to represent all possible combinations of interface and channel assignment. These strategies were compared using a probabilistic analysis then corroborated by simulations. An insight into the advantages and drawbacks of each strategy is provided regarding issues such as interference, routing, load balancing, and stability. Our framework and performance evaluation provide guidelines for network designers in planning multi-channel multi-interface network deployments.

The second contribution is an extension of the first contribution. It concerns the specification of broadcast algorithms to fit any of the strategies described in the channel and interface assignment framework. In particular, our proposed broadcast algorithms aim to ensure that a packet is delivered with a minimum probability to all neighbors. We have shown through simulations that the proposed broadcast algorithms efficiently limit the overhead. Our results provide important guidelines for the development of other broadcast solutions, especially because the problem has not been addressed in depth for multi-interface mesh networks.

The third contribution concerns the capacity of multi-channel multi-interface wireless mesh networks. We have proposed a set of linear programming formulations in order to seek the limits (i.e., bounds) of the throughput performance in presence of interference. In particular, our formulations estimate the network capacity we may obtain after the channel and interface assignment. Our formulations were initially validated through numerical analysis. To take into account aspects initially neglected in the formulations, we have presented extensive simulations in NS-3 using the IBM ILOG CPLEX Optimizer. The interaction between NS-3 and CPLEX was possible due to the development of a set of code extensions including the passing of parameters between them and the support of different channel and interface assignments. We have performed the simulations on the context of the mesh standard IEEE 802.11s. We have proposed a set of modifications that may improve its performance.

The final contribution proposes a novel routing metric and a routing protocol to improve the performance of multi-channel multi-interface wireless mesh networks. We have proposed a routing solution independent of the channel and interface assignment. We have adopted a cross-layer approach that benefits from PHY measures to estimate the quality of a link. Besides, it combines the most important factors in the design of efficient routing solutions: intra-flow and inter-flow interference, traffic load, channel diversity, packet size, and data rate. We have shown with extensive simulations that our approach overcomes other routing metrics such as $WCETT$ and ALM, and other routing protocols, such as HWMP.

7.1 Future Research Directions

The contributions presented in this thesis bring up interesting perspectives for the future research. We highlight three main directions:

Quality of Service (QoS)

Including QoS constraints in the channel and interface assignment formulation is still an on-going issue for multi-interface mesh networks. Instead of limiting the QoS support to layer-2 frame forwarding or layer-3 packet routing (e.g., classifying and processing differently each type of a frame or a packet), we believe that the assignment formulation should also be designed to support QoS. For instance, supporting bandwidth allocation requirements needed by the packet forwarding components in layer-2/3 protocols.

To find the best assignment, a feasible solution may be to combine efficiently different factors regarding their trade-off: intra-flow and inter-flow interference, data and loss rate of individual links, load, synchronization, channel switching delay, stability, etc. However, it is a complex issue that involves an in depth analysis of the impact of each factor.

Impact of Different Scenarios

We intend to evaluate the performance of our routing metric and protocol in different multi-interface mesh network scenarios with various applications. We aim to investigate in detail the extent of the impact of particular parameters on the network performance, especially in terms of throughput, delivery rate, and end-to-end delay. Therefore, it is possible to identify in which type of scenario our proposal is more appropriate. Besides that, finding out which parameters can be modified so that our protocol is also efficient in other scenarios. The same approach can also be applied to improve channel and interface assignment approaches.

Furthermore, we plan to investigate how to improve the stability mechanism used to estimate the average idle time used by our routing metric. We believe that the number of measurements necessary to reflect the channel status may vary according to network characteristics, such as the network topology, average number of neighbors, traffic load, etc. Thus, we aim to investigate the impact of a set of parameters on the number of measurements and then adopt a statistical function able to efficiently reflect the current state of the network. We also consider analyzing other statistical functions such as exponential weighting moving average to partially keep older measures.

Also, we plan to study solutions to statistically represent the PER variation in function of RSSI. Specifically, we want to obtain a mathematical approximation of the impact of the quality of the link on the expected transmission time. In this way, routing can be performed through more stable links and the network performance can be improved.

External Interference

Considering the external interference is an open research issue that needs to be further investigated. In particular, there is no guarantee that external wireless sources do not use the same unlicensed radio frequency bands as IEEE 802.11 standards. As the status of channels can be constantly monitored by the interfaces at each node, when external interference is detected, the channel and interface assignment algorithm can choose to switch to a new channel in order to reduce interference. We believe that exploiting the functionalities provided by the physical layer is a promising way to obtaining more information on the channel state. Also, more research effort should be devoted to considering the impact of external interference on the capacity of multi-interface networks.

Bibliography

[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, IEEE Std., 2007. 5, 6, 25, 29

[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 1: High-speed Physical Layer in the 5 GHz band, IEEE Std., 1999. 5, 25, 29, 115, 141

[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 2: Higher-speed Physical Layer (PHY) extension in the 2.4 GHz band, IEEE Std., 1999. 5, 25

[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 - Amendment 4: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band, IEEE Std., 2003. 5, 25, 29

[5] 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 5: Enhancements for Higher Throughput, IEEE Std., 2009. 5

[6] 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 6: Medium Access Control (MAC) Security Enhancements, IEEE Std., 2004. 5

[7] 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 8: Medium Access Control (MAC) Quality of Service Enhancements, IEEE Std., 2005. 5

[8] Roofnet, Massachusetts Institute of Technology (MIT). http://pdos.csail.mit.edu/roofnet/. Last Accessed: July, 2012. 5, 22, 24, 63

[9] T. He, S.-H. Chan, and C.-F. Wong. HomeMesh: a Low-Cost Indoor Wireless Mesh for Home Networking. *IEEE Communications Magazine*, 46(12):79–85, 2008. 5, 22

[10] S. Lee, S. Banerjee, and B. Bhattacharjee. The Case for a Multi-hop Wireless Local Area Network. In *Proc. IEEE INFOCOM*, 2004. 6

[11] C. E. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector routing (DSDV) for Mobile Computers. *ACM SIGCOMM Computer Communication Review*, 24(4):234–244, 1994. 6, 59

[12] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing, RFC 3561, IETF Network Working Group, 2003. 6, 48, 59, 137

[13] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626, IETF Network Working Group, 2003. 6, 59

[14] D. Johnson, Y. Hu, and D. Maltz. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4, RFC 4728 , IETF Network Working Group, 2007. 6, 48, 59, 62, 137

[15] K. Sundaresan, H.-Y. Hsieh, and R. Sivakumar. IEEE 802.11 over Multihop Wireless Networks: Problems and New Perspectives. *Ad Hoc Networks*, 2(2):109–132, 2004. 6

[16] I. Chlamtac, M. Conti, and J. J.-N. Liu. Mobile Ad Hoc Networking: Imperatives and Challenges. *Ad Hoc Networks*, 1(1):13–64, 2003. 6

[17] I. F. Akyildiz, X. Wang, and W. Wang. Wireless Mesh Networks: a Survey. *Computer Networks*, 47(4):445–487, 2005. 6, 20, 22, 23, 24, 25

[18] R. Bruno, M. Conti, and E. Gregori. Mesh Networks: Commodity Multihop Ad Hoc Networks. *IEEE Communications Magazine*, 43(3):123–131, 2005. 6, 25

[19] 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, IEEE Std., 2011. 6, 15, 16, 17, 18, 19, 20, 39, 50, 58, 59, 60, 63, 87, 103, 125, 142

[20] V. C. M. Borges, M. Curado, and E. Monteiro. Cross-layer Routing Metrics for Mesh Networks: Current Status and Research Directions. *Computer Communications*, 34(6):681–703, 2011. 7, 9, 27, 29, 30, 32, 46, 47, 49, 53, 125, 142

[21] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000. 7, 27, 104, 105, 130

[22] C. Chaudet, D. Dhoutaut, and I. G. Lassous. Performance Issues with IEEE 802.11 in Ad Hoc Networking. *IEEE Communications Magazine*, 43(7), 2005. 8

[23] Federal Communications Commission (FCC). http://www.fcc.gov/. Last Accessed: June, 2012. 8, 25, 26

[24] D. J. Leith, P. Clifford, V. Badarla, and D. Malone. WLAN Channel Selection Without Communication. *Computer Networks*, 56(4):1424–1441, 2012. 8

[25] J. Crichigno, M.-Y. Wu, and W. Shu. Protocols and Architectures for Channel Assignment in Wireless Mesh Metworks. *Ad Hoc Networks*, 6, 2008. 8, 27, 36, 37, 39, 67, 91

[26] J. So and N. H. Vaidya. Multi-Channel MAC for Ad hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver. In *ACM MobiHoc*, 2004. 8, 38, 45, 73, 76, 77

[27] J. Qadir, A. Misra, and C. T. Chou. Minimum Latency Broadcasting in Multi-Radio Multi-Channel Multi-Rate Wireless Meshes. In *Proc. IEEE SECON*, 2006. 8, 89

[28] W. Si, S. Selvakennedy, and A. Y. Zomaya. An Overview of Channel Assignment Methods for Multi-Radio Multi-Channel Wireless Mesh Networks. *Journal of Parallel and Distributed Computing*, 70(5):505–524, 2010. 8, 9, 32, 33, 36, 37, 40, 67

[29] P. Kyasanur and N. H. Vaidya. Capacity of Multi-Channel Wireless Networks: Impact of Number of Channels and Interfaces. In *Proc. ACM MobiCom*, 2005. 8, 104, 113

[30] R. Draves, J. Padhye, and B. Zill. Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks. In *Proc. ACM MobiCom*, 2004. 9, 24, 49, 50, 58, 71, 72, 76, 124, 125, 132, 136

[31] K.-H. Kim and K. G. Shin. On Accurate Measurement of Link Quality in Multi-Hop Wireless Mesh Networks. In *Proc. ACM MobiCom*, 2006. 9

[32] C. T. De Oliveira, F. Theoleyre, and A. Duda. Connectivity in Multi-Channel Multi-Interface Wireless Mesh Networks. In *Proc. IEEE IWCMC*, 2011. 11

[33] C. T. De Oliveira, F. Theoleyre, and A. Duda. Broadcast Strategies with Probabilistic Delivery Guarantee in Multi-Channel Multi-Interface Wireless Mesh Networks. In *Proc. IEEE/IFIP Med-Hoc-Net*, 2011. 11

[34] C. T. De Oliveira, F. Theoleyre, and A. Duda. Broadcast Strategies with Probabilistic Delivery Guarantee in Multi-Channel Multi-Interface Wireless Mesh Networks. Research Report RR-LIG-017, Grenoble Informatics Laboratory, 2011. 11

[35] C. T. De Oliveira, F. Theoleyre, and A. Duda. Evaluation de la Stratégie d'Assignation de Canaux pour IEEE 802.11s. In *Proc. Algotel*, 2011. 11

[36] C. T. De Oliveira, F. Theoleyre, and A. Duda. Channel Assignment Strategies for Optimal Network Capacity of IEEE 802.11s. In *Proc. PE-WASUN*, 2012. 11

[37] X. Wang and A. O. Lim. IEEE 802.11s Wireless Mesh Networks: Framework and Challenges. *Ad Hoc Networks*, 6(6):970–984, 2008. 16

[38] Lin Y.-D, S.-L. Tsao, S.-L. Chang, S.-Y. Cheng, and C.-Y. Ku. Design Issues and Experimental Studies of Wireless LAN Mesh. *IEEE Wireless Communications*, 17(2), 2010. 16, 39

[39] G. R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke. IEEE 802.11s: The WLAN Mesh Standard . *IEEE Wireless Communications*, 17(1):104–111, 2010. 16, 17, 19, 39, 71, 76

[40] R. C. Carrano, L. C. S. Magalhães, D. C. M. Saade, and C. V. N. Albuquerque. IEEE 802.11s Multihop MAC: A Tutorial. *IEEE Communications Surveys & Tutorials*, 13(1):52–67, 2011. 16

[41] G. R. Hiertz, S. Max, Y. Zang, T. Junge, and D. Denteneer. IEEE 802.11s MAC Fundamentals. In *Proc. IEEE MASS*, 2007. 17

[42] D. Valerio, F. Ricciato, and P. Fuxjaeger. On the Feasibility of IEEE 802.11 Multi-Channel Multi-Hop Mesh Networks. *Computer Communications*, 31(8):1484–1496, 2008. 17, 129

[43] F. Y. Li, A. Kristensen, and P. Engelstad. Passive and Active Hidden Terminal Detection in 802.11-based Ad Hoc Networks. In *Proc. IEEE INFOCOM*, 2006. 18

[44] K. Xu, M. Gerla, and S. Bae. How Effective is the IEEE 802.11 RTS/CTS Handshake in Ad Hoc Networks. In *Proc. IEEE GLOBECOM*, 2002. 18, 27, 28

[45] H. Wu, X. Wang, Q. Zhang, and X. Shen. IEEE 802.11e Enhanced Distributed Channel Access (EDCA) Throughput Analysis. In *Proc. IEEE ICC*, 2006. 19

[46] S. Corson and J. Macker. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, RFC 2501, IETF Network Working Group, 1999. 20, 46

[47] E. Z. Tragos, R. Bruno, E. Ancillotti, K. Grochla, and V. A. Siris. Automatically Configured, Optimised and QoS Aware Wireless Mesh Networks. In *Proc. IEEE PIMRC*, 2010. 20

[48] M. E. M. Campista, P. M. Esposito, I. M. Moraes, L. H. M. Costa, O. C. M. Duarte, D. G. Passos, C. V. N. De Albuquerque, D. C. M. Saade, and M. G. Rubinstein. Routing Metrics and Protocols for Wireless Mesh Networks. *IEEE Network*, 22(1):6–12, 2008. 20, 46, 59, 123

[49] Motorola MotoMesh. http://www.motorola.com/. Last Accessed: July, 2012. 22

[50] Microsoft Mesh Connectivity Layer. http://research.microsoft.com/en-us/projects/mesh/. Last Accessed: July, 2012. 22

[51] Intel. http://www.intel.com/. Last Accessed: July, 2012. 22

[52] Cisco Wireless Mesh Networking Solution. http://www.cisco.com/go/wirelessmesh. Last Accessed: July, 2012. 22

[53] MeshNet, University of California Santa Barbara (UCSB). http://moment.cs.ucsb.edu/meshnet/. Last Accessed: July, 2012. 22, 24

[54] Broadband and Wireless Network Laboratory, Georgia Institute of Technology (Georgia Tech). http://www.ece.gatech.edu/research/labs/bwn/mesh. Last Accessed: July, 2012. 22, 24

[55] ReMesh, Fluminense Federal University (UFF). http://mesh.ic.uff.br/. Last Accessed: July, 2012. 22, 24

[56] B. G. Mateus, C. T. De Oliveira, A. Callado, S. Fernandes, and R. M. C. Andrade. Impact of Density, Load, and Mobility on the Performance of Routing Protocols in Vehicular Networks. In *Proc. IEEE VTC-Fall*, 2012. 25

[57] Portsmouth. http:// http://www.portsmouth.gov.uk/. Last Accessed: July, 2012. 25

[58] European Telecommunications Standards Institute (ETSI). http://www.etsi.org/. Last Accessed: June, 2012. 25, 26

[59] P. Bahl, A. Adya, J. Padhye, and A. Walman. Reconsidering Wireless Systems with Multiple Radios. *ACM SIGCOMM Computer Communication Review*, 34(5):39–46, 2004. 26

[60] J. Lee, S.-J. Lee, W. Kim, D. Jo, T. Kwon, and Y. Choi. Understanding Interference and Carrier Sensing in Wireless Mesh Networks. *IEEE Communications Magazine*, 47(7):102–109, 2009. 27

[61] J. Deng, B. Liang, and P. K. Varshney. Tuning the Carrier Sensing Range of IEEE 802.11 MAC. In *Proc. IEEE GLOBECOM*, 2004. 28

[62] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu. Impact of Interference on Multi-Hop Wireless Network Performance. In *Proc. ACM MobiCom*, 2003. 28, 35, 105

[63] V. Borges, D. Pereira, M. Curado, and E. Monteiro. Routing Metric for Interference and Channel Diversity in Multi-Radio Wireless Mesh Networks.

In *Proc. ADHOC-NOW*, 2009. 28, 29, 30, 32, 47, 56, 58, 124, 125, 132, 133, 136

[64] U. Ashraf. *Quality of Service and Routing in Wireless Mesh Networks*. PhD thesis, Université de Toulouse, 2010. 29, 55, 58

[65] A. Vlavianos, L. K. Law, I. Broustis, S. V. Krishnamurthy, and M. Faloutsos. Assessing Link Quality in IEEE 802.11 Wireless Networks: Which is the Right Metric? In *Proc. IEEE PIMRC*, 2008. 29, 30, 125, 132

[66] D. M. Shila and T. Anjali. Load Aware Traffic Engineering for Mesh Networks. *Computer Communications*, 31(7):1460–1469, 2008. 30

[67] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-Based Models of Delivery and Interference in Static Wireless Networks. In *Proc. ACM SIGCOMM*, 2006. 30

[68] A. P. Subramanian, M. M. Buddhikot, and S. Miller. Interference Aware Routing in Multi-Radio Wireless Mesh Networks. In *Proc. IEEE WiMesh*, 2006. 30, 32, 53, 58, 124, 125, 132, 136

[69] M. Kodialam and T. Nandagopal. Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks. In *Proc. ACM MobiCom*, 2005. 32, 104

[70] K. Kowalik, B. Keegan, and M. Davis. RARE: Resource Aware Routing for mEsh. In *Proc. IEEE ICC*, 2007. 32, 47, 54, 58, 125, 127

[71] A. Raniwala and T.-C. Chiueh. Architecture and Algorithms for an IEEE 802.11-based Multi-Channel Wireless Mesh Network. In *Proc. IEEE INFOCOM*, 2005. 32, 36, 39, 45

[72] P. Li, N. Scalabrino, Y. Fang, E. Gregori, and I. Chlamtac. How to Effectively Use Multiple Channels in Wireless Mesh Networks. *IEEE Transactions on Parallel and Distributed Systems*, 20(11), 2009. 32, 67

[73] M. K. Marina, S. R. Das, and P. Subramanian. A Topology Control Approach for Utilizing Multiple Channels in Multi-Radio Wireless Mesh Networks. *Computer Networks*, 54(2):241–256, 2010. 35, 43, 45, 72, 76, 104

[74] 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. IEEE INFOCOM*, 2006. 35, 71, 78

[75] A. Iyer, C. Rosenberg, and A. Karnik. What is the Right Model for Wireless Channel Interference? *IEEE Transactions on Wireless Communications*, 8(5):2662–2671, 2009. 35, 105, 113

[76] L. A. DaSilva and I. Guerreiro. Sequence-Based Rendezvous for Dynamic Spectrum Access. In *Proc. IEEE DySPAN*, 2008. 37, 71, 73, 76, 81

[77] J. Mo, H.-S. W. So, and J. Walrand. Comparison of Multichannel MAC Protocols. *IEEE Transactions on Mobile Computing*, 7(1), 2008. 37, 38

[78] P. Bahl, R. Chandra, and J. Dunagan. SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-hoc Wireless Networks. In *Proc. ACM MobiCom*, 2004. 37, 45, 71, 73, 76, 81, 97

[79] H.-S. W. So, J. Walrand, and J. Mo. McMAC: A Parallel Rendezvous Multi-Channel MAC Protocol. In *Proc. IEEE WCNC*, 2007. 38, 45, 73, 76

[80] J. Chen, S.-T. Sheu, and C.-A. Yang. A New Multichannel Access Protocol for IEEE 802.11 Ad Hoc Wireless LANs. In *Proc. IEEE PIMRC*, 2003. 38, 45, 73, 76

[81] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou. A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks. In *Proc. IEEE BroadNets*, 2004. 38, 45, 71, 76, 84

[82] J. D. Camp and E. W. Knightly. The IEEE 802.11s Extended Service Set Mesh Networking Standard. *IEEE Communications Magazine*, 46(8):120–126, 2008. 39, 71, 76

[83] P. Kyasanur and N. H. Vaidya. Routing and Interface Assignment in Multi-Channel Multi-Interface Wireless Networks. In *Proc. IEEE WCNC*, 2005. 40, 71, 75, 76, 77, 97

[84] P. Kyasanur and N. 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. 40, 45, 75, 76, 97, 127

[85] Y. Ding, K. Pongaliur, and L. Xiao. Hybrid Multi-Channel Multi-Radio Wireless Mesh Networks. In *Proc. IEEE IWQoS*, 2009. 41, 76

[86] A. Dhananjay, H. Zhang, J. Li, and L. Subramanian. Practical, Distributed Channel Assignment and Routing in Dual-Radio Mesh Networks. In *Proc. ACM SIGCOMM*, 2009. 42, 45, 72, 76

[87] S. Kim, O. Lee, S. Choi, and S.-J. Lee. Comparative Analysis of Link Quality Metrics and Routing Protocols for Optimal Route Construction in Wireless Mesh Networks. *Ad Hoc Networks*, 9(7):1343–1358, 2011. 46

[88] S. Ghannay, S. N. 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 *Proc. IEEE ComNet*, 2009. 46, 123, 142

[89] R. Baumann, S. Heimlicher, M. Strasser, and A. Weibel. A Survey on Routing Metrics. Research Report, Computer Engineering and Networks Laboratory, ETH Zürich, 2006. 46, 47

[90] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A High Throughput Path Metric for MultiHop Wireless Routing. In *Proc. ACM MobiCom*, 2008. 47, 48, 58, 62, 125, 132, 142

[91] W. Zhou, D. Zhang, and D. Qiao. Comparative Study of Routing Metrics for Multi-Radio Multi-Channel Wireless Networks. In *Proc. IEEE WCNC*, 2005. 47

[92] T. Liu and W. Liao. On Routing in Multichannel Wireless Mesh Networks: Challenges and Solutions. *IEEE Network*, 22(1):13–18, 2008. 47

[93] R. Draves, J. Padhye, and B. Zill. Comparison of Routing Metrics for Static Multi-Hop Wireless Networks. *ACM SIGCOMM Computer Communication Review*, 34(4):133–144, 2004. 48, 58

[94] D. S. J. De Couto, D. Aguayo, B. A. Chambers, and R. Morris. Performance of Multihop Wireless Networks: Shortest Path is Not Enough. *ACM SIGCOMM Computer Communication Review*, 33(1):83–88, 2003. 48, 49

[95] C. E. Koksal and H. Balakrishnan. Quality-Aware Routing Metrics for Time-Varying Wireless Mesh Networks. *IEEE Journal on Selected Areas in Communication*, 24(11):1984–1994, 2006. 49, 58

[96] M. S. Islam, M. M. Alam, M. A. Hamid, C. S. Hong, and S. Lee. EFT: a High Throughput Routing Metric for IEEE 802.11s Wireless Mesh Networks. *Annals of Telecommunications*, 65(5):247–262, 2010. 50

[97] Y. Yang, J. Wang, and R. Kravets. Designing Routing Metrics for Mesh Networks. In *Proc. IEEE WiMesh*, 2005. 51, 52, 125, 136, 142

[98] J. L. Sobrinho. Algebra and Algorithms for QoS Path Computation and Hop-by-Hop Routing in the Internet. *IEEE/ACM Transactions on Networking (TON)*, 10(4):541–550, 2002. 51, 138

[99] J. L. Sobrinho. Network Routing with Path Vector Protocols: Theory and Applications. In *Proc. ACM SIGCOMM*, 2003. 51, 138

[100] Y. Yang, J. Wang, and R. Kravets. Interference-aware Load Balancing for Multihop Wireless Networks. Research Report, Department of Computer Science, University of Illinois at Urbana-Champaign, 2005. 52, 58, 128

[101] M. Davis and T. Raimondi. A Novel Framework for Radio Resource Management in IEEE 802.11 Wireless LANs. In *Proc. WiOpt*, 2005. 54, 127

[102] U. Ashraf, S. Abdellatif, and G. Juanole. An Interference and Link-Quality Aware Routing Metric for Wireless Mesh Networks. In *Proc. IEEE VTC-Fall*, 2008.

[103] S. S. N. Rao, Y. K. S. Krishna, and K. N. Rao. A Survey: Routing Protocols for Wireless Mesh Networks. *International Journal of Research and Reviews in Wireless Sensor Networks*, 1(3):43–47, 2011. 59

[104] S.-H. Lee, Y.-B. Ko, Y.-G. Hong, and H.-J. Kim. A New MIMC Routing Protocol Compatible with IEEE 802.11s based WLAN Mesh Networks. In *Proc. ICOIN*, 2011. 61, 123, 139

[105] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The Broadcast Storm Problem in a Mobile Ad Hoc Network. In *Proc. ACM MobiCom*, 1999. 61, 117, 139

[106] K. N. Ramachandran, M. M. Buddhikot, G. Chandranmenon, S. Miller, E. M. Belding-Royer, and K. C. Almeroth. On the Design and Implementation of Infrastructure Mesh Networks. In *Proc. IEEE WiMesh*, 2005. 62, 63

[107] A. A. Pirzada, M. Portmann, and J. Indulska. Evaluation of Multi-Radio Extensions to AODV for Wireless Mesh Networks. In *Proc. ACM MobiWac*, 2006. 62

[108] R. Draves, J. Padhye, and B. Zill. The Architecture of the Link Quality Source Routing Protocol. Research Report, MSR-TR-2004-57, Microsoft Research, 2004. 62, 63, 137

[109] D. Aguayo, J. Bicket, and R. Morris. SrcRR: A High Throughput Routing Protocol for 802.11 Mesh Networks. Research Report, Massachusetts Institute of Technology (MIT), 2005. 63

[110] A. A. Pirzada, M. Portmann, and J. Indulska. Performance Analysis of Multi-Radio AODV in Hybrid Wireless Mesh Networks. *Computer Communications*, 31(5):885–895, 2008. 63

[111] X.-Y. Li Xufei M. and S. Kami Makki. Static Channel Assignment For Multi-Radio Multi-Channel Multi-Hop Wireless Networks. In *Proc. WTASA*, 2007. 71, 72, 76

[112] C. Sarr, C. Chaudet, G. Chelius, and I. G. Lassous. Bandwidth Estimation for IEEE 802.11-Based Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 7(10), 2008. 71

[113] S.-L. Wu, Y.-C. Tseng, C.-Y. Lin, and J.-P. Sheu. A Multi-Channel MAC Protocol with Power Control for Multi-Hop Mobile Ad Hoc Networks. *The Computer Journal*, 45(1):101–110, 2002. 74, 76, 98

[114] N. Jain, S. R. Das, and A. Nasipuri. A Multichannel CSMA MAC Protocol with Receiver-based Channel Selection for Multihop Wireless Networks. In *Proc. ICCCN*, 2001. 74, 76, 97, 98

[115] J. Li, Z. J. Haas, M. Sheng, and Y. Chen. Performance Evaluation of Modified IEEE 802.11 MAC for Multi-Channel Multi-Hop Ad Hoc Network. In *Proc. AINA*, 2003. 74, 76, 97

[116] MATLAB. http://www.mathworks.com/products/matlab/. Last Accessed: July, 2012. 77

[117] K. S. Trivedi. *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. John Wiley and Sons Press, New York, Second edition, 2001. 78, 81

[118] Y.-D. Lin, S.-L. Chang, S.-H. Tao, and J.-H. Yeh. Realizing and Benchmarking Broadcast Algorithms in Wireless Mesh Networks. *Computer Communications*, 34(10):1169–1181, 2011. 87

[119] P. Kyasanur, J. So, and C. Chereddi N. G. Vaidya. Multichannel Mesh Networks: Challenges and Protocols. *IEEE Wireless Communications*, 13(2):30–36, 2006. 87

[120] K. Xing and X. Cheng. A Novel Channel Assignment Approach for Local Broadcast in Multi-Radio Multi-Channel Wireless Networks. In *Proc. IEEE MILCOM*, 2007. 88, 89

[121] K. Xing, X. Cheng, L. Ma, and Q. Liang. Superimposed Code Based Channel Assignment in Multi-Radio Multi-Channel Wireless Mesh Networks. In *Proc. ACM MobiCom*, 2007. 88, 89

[122] J. Qadir, C. T. Chou, A. Misra, and J. G. Lim. Localized Minimum-Latency Broadcasting in Multi-Radio Multi-Rate Wireless Mesh Networks. In *Proc. IEEE WoWMoM*, 2008. 89

[123] K. Han, Y. Li, Q. Guo, and M. Xiao. Broadcast Routing and Channel Selection in Multi-Radio Wireless Mesh Networks. In *Proc. IEEE WCNC*, 2008. 89

[124] H. S. Chiu, K. L. Yeung, and K.-S. Lui. Maximizing Broadcast Load in Multi-Channel Multi-Interface Wireless Mesh Networks. In *Proc. IEEE GLOBECOM*, 2008. 89

[125] H. S. Chiu and K. L. Yeung. A Joint Routing and Scheduling Algorithm for Efficient Broadcast in Wireless Mesh Networks. In *Proc. IEEE WCNC*, 2008. 89

[126] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized Link State Routing Protocol for Ad Hoc Networks. In *Proc. IEEE INMIC*, 2001. 92

[127] R. Jain, D. M. Chiu, and W. Hawe. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems. Research Report TR-301, Digital Equipment Corporation (DEC), 1984. 96

[128] J. Camp, J. Robinson, C. Steger, and E. Knightly. Measurement Driven Deployment of a Two-Tier Urban Mesh Access Network. In *Proc. ACM MobiSys*, 2006. 96

[129] A. Zemlianov and G. De Veciana. Capacity of Ad Hoc Wireless Networks With Infrastructure Support. *IEEE Journal on Selected Areas in Comunications*, 23(3):657–667, 2005. 104

[130] A. A. Franklin, V. Bukkapatanam, and C. S. R. Murthy. Impact of Multiple Channels and Radios on the Performance of a TDMA Based Wireless Mesh Network. *Computer Networks*, 53:2208–2224, 2009. 104

[131] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1993. 104

[132] H. Rivano, F. Theoleyre, and F. Valois. A Framework for the Capacity Evaluation of Multihops Wireless Networks. *Ad Hoc & Sensor Wireless Networks*, 9:139–162, 2010. 106

[133] R. Balakrishnan and K. Ranganathan. *A Textbook of Graph Theory*. Springer, First edition, 1999. 108, 109

[134] C. Bron and J. Kerbosch. Algorithm 457: Finding All Cliques of an Undirected Graph. *Communications of the ACM*, 16(9):575–577, 1973. 109

[135] S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa. A New Algorithm for Generating All the Maximal Independent Sets. *SIAM Journal on Computing*, 6:505–517, 1977. 111

[136] Network Simulator 3. http://www.nsnam.org/. Last Accessed: July, 2012. 115

[137] IBM ILOG CPLEX Optimizer. http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/. Last Accessed: July, 2012. 115

[138] V. Kolar, S. Razak, P. Mähönen, and N. B. Abu-Ghazaleh. Measurement and Analysis of Link Quality in Wireless Networks: An Application Perspective. In *Proc. IEEE INFOCOM Workshops*, 2010. 124, 130, 132, 134

[139] M. Lacage, M. H. Manshaei, and T. Turletti. IEEE 802.11 Rate Adaptation: A Practical Approach. In *Proc. ACM MSWIM*, 2004. 126

[140] G. Wu and T. Chiueh. Passive and Accurate Traffic Load Estimation for Infrastructure-Mode Wireless LAN. In *Proc. ACM MSWIM*, 2007. 127

[141] Y. Yang and R. Kravets. Contention-Aware Admission Control for Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 10(4), 2005. 127

[142] T. Liu, W. Liao, and J.-F. Lee. Distributed Contention-Aware Call Admission Control for IEEE 802.11 Multi-Radio Multi-Rate Multi-Channel Wireless Mesh Networks. *Mobile Networks and Applications*, 14(2):134–142, 2009. 127

[143] M. Rondinone, J. Ansari, J. Riihijärvi, and P. Mähönen. Designing a Reliable and Stable Link Quality Metric for Wireless Sensor Networks. In *Proc. REALWSN*, 2008. 130, 132

[144] V. Kolar, S. Razak, P. Mähönen, and N. B. Abu-Ghazaleh. Link Quality Analysis and Measurement in Wireless Mesh Networks. *Ad Hoc Network*, 9(8):1430–1447, 2011. 130, 132, 133

[145] B. Raman, K. Chebrolu, D. Gokhale, and S. Sen. On the Feasibility of the Link Abstraction in Wireless Mesh Networks. *IEEE/ACM Transactions on Networking*, 17(2):528–541, 2009. 130, 132

[146] B. Raman, K. Chebrolu, N. Madabhushi, D. Y. Gokhale, P. K. Valiveti, and D. Jain. Implications of Link Range and (In)stability on Sensor Network Architecture. In *Proc. WiNTECH*, 2006. 130, 131, 132

[147] K. Srinivasan and P. Levis. RSSI is Under Appreciated. In *Proc. EmNets*, 2006. 132

[148] P. Bharat U, K. Chebrolu, and P. Kulkarni. Design of RSSI based Routing Protocol for Static Wireless Sensor Networks. Research Report, Department of Computer Science and Engineering, Indian Institute of Technology, 2008. 132

[149] A. K. Mishra, K. Chebrolu, and P. Kulkarni. Design and Implementation of a RSSI based Link Quality Metric for Static Wireless Mesh Routing. Research Report, Department of Computer Science and Engineering, Indian Institute of Technology, 2008. 132

[150] C. E. Koksal. Quality-Aware Routing Metrics in Wireless Mesh Networks. In *Wireless Mesh Networks*, pages 227–243. Springer US, 2008. 132

[151] E. Amusa, O. Adjei, J. Zhang, A. Mansour, and A. Capone. An Efficient RSSI-Aware Metric for Wireless Mesh Networks. In *Proc. WiOpt*, 2011. 132

[152] U. Ashraf, S. Abdellatif, and G. Juanole. Route Maintenance in IEEE 802.11 Wireless Mesh Networks. *Computer Communications*, 34(13):1604–1621, 2011. 132, 134

[153] M. E. M. Campista, L. H. M. K. Costa, and O. C. M. B. Duarte. Privileging Long-life Packets in Multihop Wireless Networks. In *Proc. IFIP/IEEE MWCN*, 2007. 133

[154] H. Aïache, L. Lebrun, V. Conan, and S. Rousseau. A Load Dependent Metric for Balancing Internet Traffic in Wireless Mesh Networks. In *Proc. IEEE MeshTech*, 2008. 133

[155] Z. Wang and J. Crowcroft. Bandwidth-Delay Based Routing Algorithms. In *Proc. IEEE GLOBECOM*, 1995. 134, 135

[156] H. J. Chao and X. Guo. *Quality of Service Control in High-Speed Networks*. John Wiley and Sons Press, New York, 2001. 135

[157] V. S. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan. End-to-end Packet Scheduling in Wireless Ad-hoc Networks. In *Proc. ACM/SIAM SODA*, 2004. 136

Cross-Layer Design and Performance Optimization of Multi-Channel Multi-Interface Wireless Mesh Networks

Abstract:

In this PhD thesis, we focus on the design and performance optimization of multi-channel multi-interface wireless mesh networks. To take advantage of the increased capacity in such networks, a number of issues has to be handled properly. The first contribution of this thesis is a novel classification and formal evaluation of different channel and interface assignment strategies. In particular, we focus on connectivity in terms of topology formation, density of connections, and neighbor discovery. Our second contribution presents broadcast algorithms able to handle any of the multi-channel multi-interface assignment strategies. These algorithms guarantee a broadcast packet to be delivered with a minimum probability to all neighbors. The third contribution of this thesis consists in evaluating network capacity (i.e., throughput) obtained through different channel and interface assignments schemes. More specifically, we propose three mixed integer linear programming formulations to model the routing and bandwidth sharing constraints in presence of interference. We then derive upper and lower bounds for different MAC strategies. The fourth and the last contribution of this thesis is the development of a novel cross-layer routing solution for multi-channel multi-interface mesh networks. In particular, we propose a link-quality aware metric to estimate the residual bandwidth of a link. An on-demand routing protocol selects the routes offering the best throughput. All our contributions are validated through extensive simulations that demonstrate the efficiency of our solutions. In summary, this thesis provide insight into the improvement of multi-channel multi-interface wireless mesh networks, as well as guidelines for network designers in planning efficient deployments.

Keywords: Wireless Mesh Networks, Multi-Channel, Multi-Interface, Performance Optimization, Broadcast, Capacity, Routing.

Conception et Optimisation de Performance Inter-couches dans les Réseaux Maillés Radio Multi-Canal Multi-Interface

Abstract:

Dans cette thèse, nous nous concentrons sur la conception et l'optimisation de performances inter-couches dans les réseaux maillés radio multi-canal multi-interface. Afin de profiter de l'augmentation de la capacité de ces réseaux, un certain nombre de problèmes doit être résolu. La première contribution de cette thèse est une nouvelle classification et une évaluation formelle des différentes stratégies d'assignation de canaux et d'interfaces. Nous considérons en particulier la connectivité en termes de formation de topologie, densité de connexions et découverte de voisinage. La deuxième contribution présente des algorithmes de diffusion fonctionnant pour n'importe quelle stratégie d'assignation multi-canal multi-interface. Ces algorithmes garantissent qu'un paquet de diffusion est délivré avec une probabilité minimale à tous les voisins. La troisième contribution de cette thèse évalue la capacité (débit) obtenue à travers les différentes solutions d'assignation de canaux et d'interfaces. Plus précisément, nous proposons trois formulations de programmation linéaire mixte pour modéliser le routage et les contraintes de partage de bande passante en présence d'interférences. Nous dérivons ensuite des bornes supérieures et inférieures pour deux stratégies MAC différentes. Notre dernière contribution concerne le développement d'une solution de routage inter-couches pour les réseaux maillés multi-canal multi-interface. En particulier, nous proposons une métrique de qualité de lien estimant la bande passante résiduelle d'un lien. Un protocole de routage adapté permet ensuite de trouver les routes offrant le meilleur débit. Toutes nos contributions sont validées par des nombreuses simulations qui démontrent l'efficacité de nos solutions. En résumé, cette thèse fournit une analyse en profondeur des réseaux maillés radio multi-canal multi-interface, ainsi que des lignes directrices pour les concepteurs de réseaux afin de déployer des réseaux performants.

Keywords: Réseaux Maillés Radio, Multi-Canal, Multi-Interface, Optimisation de Performance, Broadcast, Capacité, Routage.