Broadcast Strategies with Probabilistic Delivery Guarantee in Multi-Channel Multi-Interface Wireless Mesh Networks

Carina Teixeira de Oliveira

Grenoble Informatics Laboratory UMR 5217
UJF-Grenoble 1-CNRS
681 Rue de la Passerelle, BP72
38402 Saint Martin d'Heres, France
Email: oliveira@imag.fr

Fabrice Theoleyre

CNRS – LSIIT – UMR 7005
University of Strasbourg
Boulevard Sebastien Brant
67412 Illkirch, France
Email: theoleyre@unistra.fr

Andrzej Duda

Grenoble Informatics Laboratory UMR 5217
Grenoble Institute of Technology-CNRS
681 Rue de la Passerelle, BP72
38402 Saint Martin d'Heres, France
Email: duda@imag.fr

Abstract

Multi-channel multi-interface Wireless Mesh Networks permit to spread the load across orthogonal channels to improve network capacity. Although broadcast is vital for many layer-3 protocols, proposals for taking advantage of multiple channels mostly focus on unicast transmissions. In this paper, we propose broadcast algorithms that fit any channel and interface assignment strategy. They guarantee that a broadcast packet is delivered with a minimum probability to all neighbors. Our simulations show that the proposed algorithms efficiently limit the overhead.

1 Introduction

Wireless Mesh Networks (WMN) have attracted increasing attention in recent years because of their low-cost and ease of deployment. These multi-hop networks are self-organized and self-configured without any centralized control. They are composed of static wireless routers and some of them act as gateways toward the Internet. In this paper, we consider WMN with routers based on the IEEE 802.11 technology.

When mesh routers use a single interface (i.e. wireless network card) tuned to a single channel, the network capacity degrades with the increase of the network size due to channel contention and spatial problems such as exposed and

hidden nodes [1]. One way of improving the performance of WMN is to use multiple non-overlapping channels (free of inter-channel interference) so that mesh routers can transmit in parallel and without collisions [2]. To take advantage of multiple channels, nodes may have multiple interfaces to simultaneously transmit/receive packets. These networks are called Multi-Channel Multi-Interface (MCMI) WMN.

It has been shown that one can significantly improve the network capacity by carefully choosing the set of channels a mesh router may use for each of its interfaces and which of them it will use to communicate with its neighbors. A pair of nodes should have sufficient channels in common with each other while minimizing interference with other active pairs.

Recently, there has been an increasing interest in developing new solutions for MCMI WMN. However, few studies considered efficient broadcast in such networks—the work done has only focused on static interface assignment in which all nodes are tuned to a common set of channels [3, 4]. While it is true that static approaches provide suitable stability for routing protocols without path changes, re-ordering, channel switches etc., on the other hand, these approaches do not efficiently distribute the load among all available channels. Consequently, the inability to adapt interfaces under heavy load and interference variations can drastically reduce the overall network performance. Therefore, the channel and interface assignments play a key role on MCMI WMN performance, which can not be neglected by broadcast solutions.

We propose to focus here on the broadcast problem in multi-channel multi-interface wireless mesh networks. Our contribution is threefold:

1. we introduce a classification of Channel and Interface Assignment (CIA) strategies in MCMI WMN;

2. we propose broadcast algorithms that fit any CIA strategy guaranteeing a packet is delivered with a minimum probability to all neighbors;

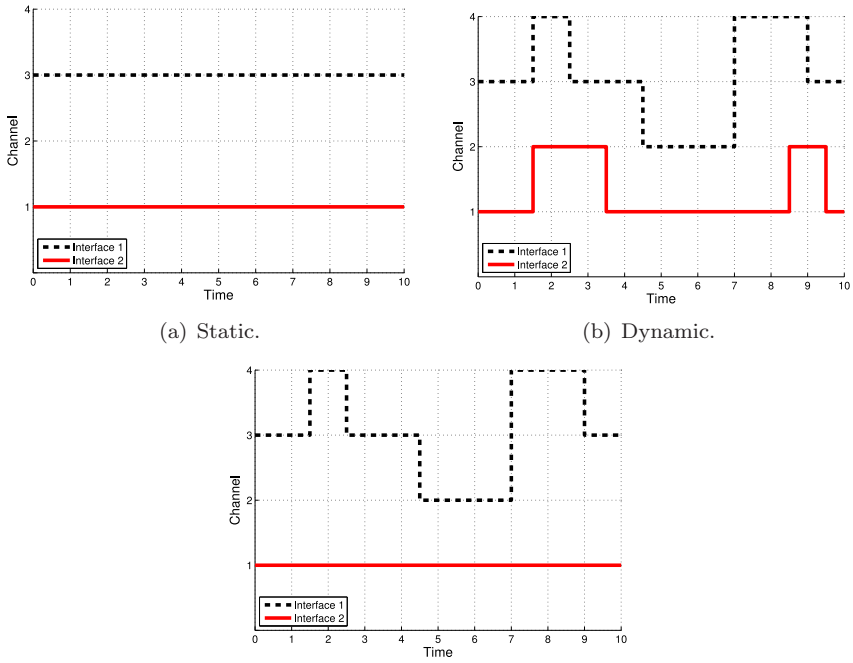
3. we provide simulation results to compare different strategies and choose the most suitable one for a given situation.

The rest of this paper is organized as follows. We first introduce the system model in the next section. Section 3 presents a review of relevant related work. In Section 4, we introduce the probabilistic delivery guarantee. Section 5 presents the broadcast algorithms with the probabilistic guarantee. Simulation results are presented in Section 6. Finally, we conclude the paper in Section 7.

2 Model and Assumptions

We model a wireless mesh network as an undirected graph $G = (V, E)$, where V is the set of nodes (mesh routers) and E the set of edges corresponding to two nodes able to directly communicate.

We adopt the following notation:



(a) Static.

(b) Dynamic.

(c) Mixed.

Figure 1: Examples of interface assignment with 4 channels and 2 interfaces.

- C: the number of non-overlapping (orthogonal) channels that can be used by all nodes $v \in V$. We assume that a node can tune each of its interfaces to a set of channels $c \subseteq C$;
- $Intf(v)$: the set of interfaces of node $v \in V$;
- $\mathcal{S}(i)$: the schedule of interface i—a list of tuples $\{(channels, timeStart, timeStop)\}$. The schedule may be periodical or anarchical;
- $\mathcal{I}_v$: the number of interfaces of node v, where $\mathcal{I}_v = \mathcal{I}_v^S + \mathcal{I}_v^D$ with $\mathcal{I}_v^S$ denoting the number of static interfaces and $\mathcal{I}_v^D$ the number of dynamic interfaces of node $v \in V$.

Two mesh routers in the transmission range of each other are able to communicate if at least one of their interfaces uses the same channel at the same instant. More formally, $(u, v) \in V^2, (u, v) \in E$ if:

$$\exists i \in Intf(u), \exists j \in Intf(v) \; such \; that \; \mathcal{S}(i) \cap \mathcal{S}(j) \neq \emptyset \tag{1}$$

2.1 Interface Assignment

We distinguish between three types of *interface assignment*:

2.1.1 Static Interfaces

In this case, all interfaces are static and remain on a channel for a long period of time:

$$\forall v \in V,\ \mathcal{I}_v^D = 0 \rightarrow \mathcal{I}_v = \mathcal{I}_v^S. \tag{2}$$

Figure 1(a) shows an example of static interface assignment with four channels and two interfaces. The x-axis is the time and the y-axis is the channel id. Note that each interface remains on the same channel regardless of time.

2.1.2 Dynamic Interfaces

In this case, all interfaces are dynamic and frequently switch from one channel to another:

$$\forall v \in V,\ \mathcal{I}_v^S = 0 \rightarrow \mathcal{I}_v = \mathcal{I}_v^D. \tag{3}$$

Figure 1(b) presents an example of dynamic interface assignment. Note that two interfaces of a node should not use the same channel at the same instant. Otherwise, we would waste bandwidth since the pair of interfaces will interfere.

2.1.3 Mixed Interfaces

In this case, $\mathcal{I}_v^S$ static interfaces permanently stay on a channel and $\mathcal{I}_v^D$ dynamic interfaces frequently switch from one channel to another:

$$\forall v \in V,\ \mathcal{I}_v^S \geq 1,\ \mathcal{I}_v^D \geq 1. \tag{4}$$

Figure 1(c) shows an example of mixed interface assignment in which *Interface 1* switches from one channel to another and *Interface 2* remains tuned to channel 1.

2.2 Channel Assignment

Furthermore, we classify *channel assignment* according to one of the following approaches:

2.2.1 Common Channel Set

Static interfaces correspond to a *Common Channel Set* (CCS)—once a channel is assigned to an interface, it does not change.

One of the simplest set works as follows: on every node, interface 1 is assigned to channel 1, interface 2 is assigned to channel 2, and so on. CCS can also be built upon the concept of non-overlapping channels in order to reduce interference. Adya *et al.* simulated scenarios in which each node has two static interfaces tuned to orthogonal channels: one interface is assigned to channel 1 and the other one to channel 11 [5].

2.2.2 Pseudo-Random

Static interfaces at different nodes may be assigned to a different set of channels. Each node locally decides which channel to allocate to static interfaces.

A simple solution consists in choosing channels at random [6]. Another solution is to use some well-known function f of the node identifier to select the channels to assign to the static interfaces [7]. Neighbors of a node $v \in V$ can use the same function f to compute the channel used by v. An alternative solution consists in explicitly exchanging *hello* packets that contain information about the channels used by static interfaces. Based on the received *hello* packets, nodes may choose to set their channels to an unused or a lightly loaded channel.

2.2.3 Adaptive

It uses some information or criteria to dynamically assign channels based on for instance time instants (defined or random), a pre-defined channel visiting order, the interference level, or the available bandwidth.

Finally, we distinguish between the following types of broadcast communications:

- Discovery Broadcast: a node has to discover neighbors;
- Local Broadcast: all neighbors must receive a packet;
- Flooding: a packet has to be received by all nodes in the network.

We will focus here on Local Broadcast. We considered Discovery Broadcast elsewhere [8].

3 Related Work

3.1 Multi-channel Multi-interface Strategies

Table 1: Channel and Interface Assignment (CIA) Strategies

Strategies	**Interface**			**Channel**			**Ref**
	S	D	M	C	P-R	A	
Static/Common	X			X			[5, 9]
Static/Pseudo-Random	X				X		[6, 10]
Dynamic/Adaptive		X				X	[11, 12]
Mixed/Common & Adaptive			X	X		X	[13, 14]
Mixed/Pseudo-Random & Adaptive			X		X	X	[7, 15]

Interface Assignment – S: Static, D: Dynamic, M: Mixed
Channel Assignment – C: Common, P-R: Pseudo-Random, A: Adaptive

A multi-channel multi-interface strategy consists of a combination of *interface* and *channel* assignments [16]. We can identify the following strategies:

1. *Static Interfaces/Common Channel Assignment:* all interfaces are static and use the CCS approach (all nodes use channel i on the i^{th} interface) [5, 9].

   We can note that the strategy is optimal when the number of interfaces equals the number of available channels, which is seldom the case;

2. *Static Interfaces/Pseudo-Random Channel Assignment:* similar to the previous strategy, it assigns a channel to each interface for permanent use. However, static interfaces at different nodes may be assigned to a different set of channels [6, 10].

   This approach does not guarantee connectivity since two nodes may choose different channels for their interfaces. Therefore, deafness may arise;

3. *Dynamic Interfaces/Adaptive Channel Assignment:* all interfaces are dynamic. Thus, the topology is also dynamic: two nodes may temporarily be able to communicate until one interface switches its channel. Often, a rendezvous mechanism has to be used to avoid deafness [11, 12]. In other words, nodes can agree on the channels they will use in the next time interval. Then, channel re-assignment ensures that statistically, a node can reserve a common channel for each of its neighbors after a sufficiently long time.

4. *Mixed Interfaces/Common and Adaptive Channel Assignment:* static interfaces use a common channel set (CCS) while dynamic interfaces act in an on-demand manner [13, 14].

   A pair of nodes may use its static interfaces to negotiate a channel for data exchange. Negotiation takes place on a dedicated *control channel* to isolate control RTS/CTS traffic from data.

   The strategy uses all available channels with a trade-off between connectivity (static interfaces) and capacity optimization (dynamic interfaces);

5. *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. Static interfaces are used in reception while dynamic interfaces are used for transmissions. A node has just to know the list of channels used by the static interfaces of its neighbors to communicate with them: no deafness appears.

   This strategy often uses one single static interface ($\mathcal{I}_v^S = 1$) [7].

Table 1 presents an overview of all Channel and Interface Assignment (CIA) strategies.

3.2 Broadcast

Many network protocols use local broadcast for various purposes: routing, coordination, synchronization, etc. The functionality of reaching all neighbors

in a local broadcast is useful in flooding and efficient flooding protocols aim at limiting the number of retransmissions. In a single channel network, one single packet is sufficient to *cover* all the neighbors (all neighbors receive it), because of the broadcast nature of radio transmissions.

Several papers tried to tackle the broadcast problem in multi-channel multi-interface wireless mesh networks. Qadir *et al.* [3] proposed to optimize the delay for multi-rate mesh networks. However, they focus on the global flooding problem, i.e. how each node in the network receives the flooded packets. Each node selects a subset of neighbors to forward the packet, pruning redundant transmissions. They assume a static interface assignment (Static/Common) in which all nodes are tuned to the same channels. Song *et al.* [17] presented also independently a broadcast protocol to achieve a 100% reliability with minimum latency. They constructed a broadcasting tree using a link quality metric, but focused also on a network wide flooding. Yang *et al.* [4] additionally introduced network coding to reduce the associated overhead in mesh networks.

Xing *et al.* [18] proposed superimposed codes to tackle both the unicast and broadcast problems in multichannel multiradio mesh networks.

In conclusion, no proposal is sufficiently generic to deal with any CIA strategy.

4 Probabilistic delivery guarantee

Transmission in wireless networks may suffer from errors due to various effects at PHY and MAC layers: attenuation, interference, fading, multipath propagation, synchronization errors, or collisions. Our goal is to design broadcast protocols that guarantee the reception of a broadcast packet by each neighbor with a certain probability.

4.1 Packet Error Estimation

We denote by p_e bit error probability and by p_p packet error probability. They are related by the following relation:

$$p_p = 1 - (1 - p_e)^{size} \tag{5}$$

where $size$ denotes the size in bits of a packet. p_{deliv} is the probability of successful packet delivery $p_{deliv} = 1 - p_p$. Since it depends on a given radio link, we use the notation $p_{deliv}(u, v)$ for the transmission from u to v.

4.2 Probabilistic guarantee

We propose a *probabilistic guarantee* of local broadcasts.

We consider that a particular neighbor is *covered* by a broadcast if it receives at least one copy of the corresponding packet with a probability superior or equal to $p_{cover_{min}}$, a parameter of the protocol. Higher layers may specify its value

when they want to transmit a broadcast packet. A local broadcast is *successful* if all neighbors are *covered*.

Let $N(v)$ represent the neighbors of v. We denote by $p_{cover}(u \rightarrow v)$ the probability that node v correctly receives the broadcast of node u, i.e., v is *covered*. Our protocol will imply that

$$\forall v \in N(u), p_{cover}(u \rightarrow v) \geq p_{cover_{min}}. \tag{6}$$

To provide guarantees, we limit the links to those with packet error probability of at least $p_{p_{max}}$, i.e. a node does not maintain radio links of low quality.

5 Broadcast Algorithms

In this section, we introduce the broadcast algorithms based on the classification proposed in Section 3.1.

5.1 Static Interfaces with Common Channel Assignment

With the common channel assignment, the i^{th} channel is assigned to the i^{th} static interface, so there is no deafness. Thus, broadcast is simple: a node has just to broadcast a packet through any of its static interfaces and all its neighbors will receive it.

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 receives at least one of the k copies from u is:

$$p_{cover}(u \rightarrow v) = 1 - (1 - p_{deliv}(u, v))^k \tag{7}$$

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

$$K = \left\lceil \frac{log(1 - p_{cover_{min}})}{log(1 - p_{deliv}(u, v))} \right\rceil \tag{8}$$

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

When only one static interface is tuned to the control channel, we can use this interface 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.

We can apply this approach to Strategies 1 and 4 (those that use a Common Channel Set) in Section 3.1.

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

```
/* initialization                                                    */
/* neighs ≡ {(neigh, channel)}                                       */
neighs [] ←getListNeighStaticIntf() ;
/* initially, no neighbor is covered                                 */
for i ∈ [0..|neighs|] do
    Pcover [i] ←0 ;
/* if at least one uncovered neighbor exists                         */
while (∃n ∈ neighs such that Pcover[n] < p_cover_min ) do
    /* count the nb of neighbors covered for each channel.           */
    for c ∈ [0..|channels|] do
        nbCovered [c] ←nbNeighsCovered(c) ;
    /* select one of the best channels                               */
    bestChannel ←rand(nbCovered) ;
    /* update the Pcover proba for each newly covered neighbor
       */
    for n ∈ neighUsing(bestChannel) do
        /* this neighbor was not yet covered at all                  */
        if Pcover [n] = 0 then
            Pcover [n] ← Pdeliv [n] ;
        else
            Pcover [n] ← 1 − (1 − Pcover[n]) · (1 − Pdeliv[n]) ;
    /* send one broadcast packet                                     */
    sendBroadcast(bestChannel) ;
```

5.2 Static Interfaces with Pseudo-Random Channel Assignment

With this kind of assignment, a single transmission is not sufficient for local broadcast, because not all neighbors use the same channel. 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 (this is a feature of the unicast protocol): a node will also use this information for its broadcast transmissions.

We propose a greedy approach inspired by MultiPoint-Relays [19]: a node chooses the minimum number of channels that cover the largest number of neighbors (cf. Algorithm 1). More precisely, 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). It initially considers that all its neighbors are

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

```
1  /* schedule ≡ {(T_start, T_end, neigh, channel)}:  the channel used
       by each neighbor during each timeslot                          */
2  schedule [] ←constructSchedule(neighs);
3  /* greedy steps                                                    */
4  while (∃n ∈ neighs such that Pcover[n] < p_cover_min) do
5      /* sort the slots according to the number of new neighbors
          they cover                                                  */
6      slots ←sort (schedule, Pcover) ;
7      /* select one of the slots that covers most neighbors         */
8      bestSlot ←randAmongMAX(slots) ;
9      /* update the Pcover probability                              */
10     for n ∈ neighDuring(bestSlot) do
11         Pcover [n] ← 1 − (1 − Pcover[n]) · (1 − Pdeliv[n]) ;
12     /* send one broadcast packet                                  */
13     sendBroadcast(bestSlot) ;
```

uncovered (Algorithm 1, lines $1-6$).

- while at least one neighbor is covered with a probability inferior to $p_{cover_{min}}$, the node searches for the channel with the best quality:
  - it counts the number of newly covered neighbors for each channel (their covering probability is inferior to $p_{cover_{min}}$);
  - it randomly chooses one of the best channels (to balance the load among channels);
  - for each neighbor reachable through this channel, it updates the probability of reception. It corresponds to the delivery probability for the link (u, v) if u did not yet schedule a packet for v. Otherwise, it recursively applies Equation 9.

$$p_{cover}(u,v) = 1 - (1 - p_{cover}(u,v))\,(1 - p_{deliv}(u,v)) \tag{9}$$

We can apply this approach to Strategies 2 and 5 in Section 3.1 that use static interfaces to receive packets.

5.3 Dynamic Interfaces with Adaptive Channel Assignment

When a node only uses dynamic interfaces, it needs to avoid deafness by correctly choosing both an interface and a schedule. We propose the Algorithm 2.

A node first creates the schedule of its interfaces and thus of its neighbors: a node knows the channel switching instants of all its neighbors for all their

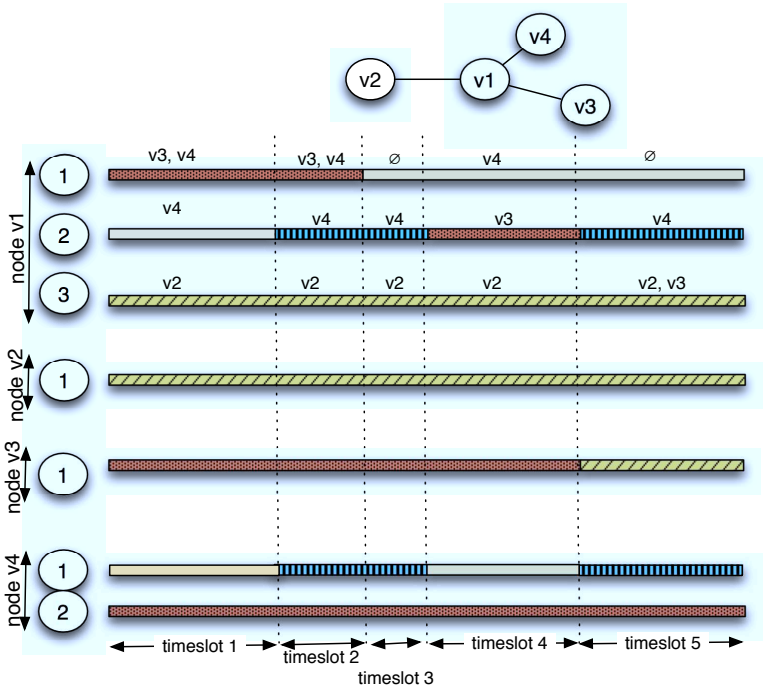


Figure 2: Local broadcast with mixed interfaces—*each color of the bars represents a different channel and we report the list of neighbors reachable through each of v_1 interfaces at any instant (v_1 has 3 interfaces, v_2 and v_3 one interface, and v_4 2 interfaces). We consider in this example a neighbor covered if it received at least one copy*

interfaces (otherwise transmissions are impossible due to deafness). It constructs timeslots so that itself and all its neighbors stay tuned to the same channel during one timeslot. We do not require all the nodes to switch their channels at the same time. Let us consider the example in Figure 2 in which timeslots are delimited by dashed lines: the first interface of node v_1 stays tuned to the same channel during timeslots 1 and 2 while the second interface switches between both timeslots. The schedule consists of a kind of the lowest common denominator between the different channel switching instants for all neighbors.

After having constructed this schedule, the transmitter is able to compute the number of neighbors that can be covered for each interface for each timeslot (i.e. when the channels match). Thus, it will re-iterate by greedily choosing pairs <`timeslot,interface`> that cover the largest number of not yet covered neighbors. When a node sends a copy of a broadcast packet, it updates the probability of delivery for each neighbor, adopting the same approach as Algorithm 1 (lines $15-20$). The algorithm stops when all the neighbors are covered

Table 2: Default parameter values

Parameter	**Default value**
Number of nodes	200
Density (avg. number of neighbors)	10
Number of interfaces	3
Number of channels	12
$p_{cover_{min}}$ (probability above which we consider the node is covered)	0.95

with a probability superior to $p_{cover_{min}}$.

Let us consider the example in Figure 2. We consider in this example a neighbor covered if it received at 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 apply to the strategy that only uses dynamic interfaces (Strategy 3 in Section 3.1).

6 Performance Evaluation

We have implemented a simulator to evaluate the broadcast performance* [20]. We assume the ideal MAC layer to focus on broadcast performance with no collision. We generate random positions of nodes and plot 95% confidence intervals. We use the standard values depicted in Table 2. The simulation measures:

1. the overhead defined as the average number of transmissions required by a node to cover all its neighbors;

2. the Jain Index of the load for all the channels to measure fairness. Let B_c denotes the bandwidth consumed by the broadcast on channel c:

$$JainIndex = \frac{\left(\sum_{c=1}^{C} B_c\right)^2}{C \cdot \sum_{c=1}^{C} B_c^2} \tag{10}$$

We denote each strategy as introduced in Section 3.1 and apply the broadcast algorithms defined in the previous section. In particular, we have implemented the Dynamic/Adaptive strategy in a way that each interface equally shares its

*the simulator is freely available at `http://forge.imag.fr/projects/graphsim` in the subversion repository

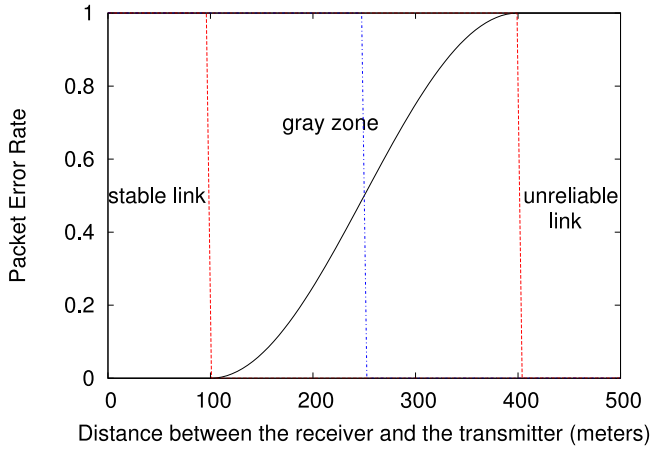


Figure 3: Impact of the distance between the receiver and the transmitter on the Packet Error Rate (PER).

time among all the channels following a pseudo-random sequence [12]. Two nodes are able to exchange packets if at least one pair of interfaces uses the same channel at the same instant.

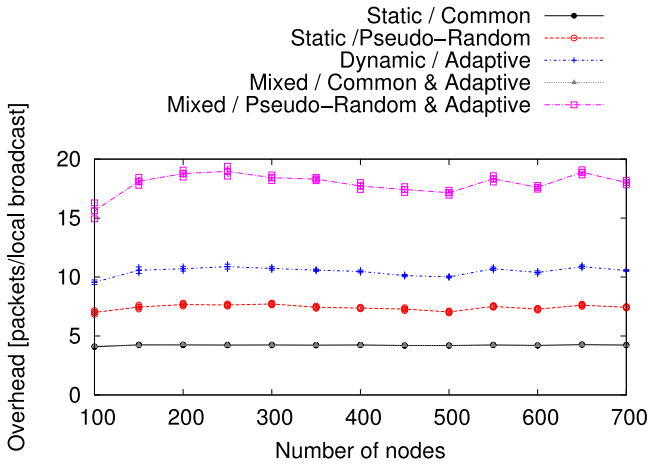
6.1 Packet Error Rate

Simulation takes into account packet error probability through the Packet Error Rate (PER) model represented in Figure 3 [21] that shows the relation between the PER and the distance. As explained above, the neighbors with a PER superior to $p_{p_{max}}$ have not to be covered. For the numerical results, we have chosen the value of $p_{p_{max}} = 0.5$ although different values would lead to consistently the same results.

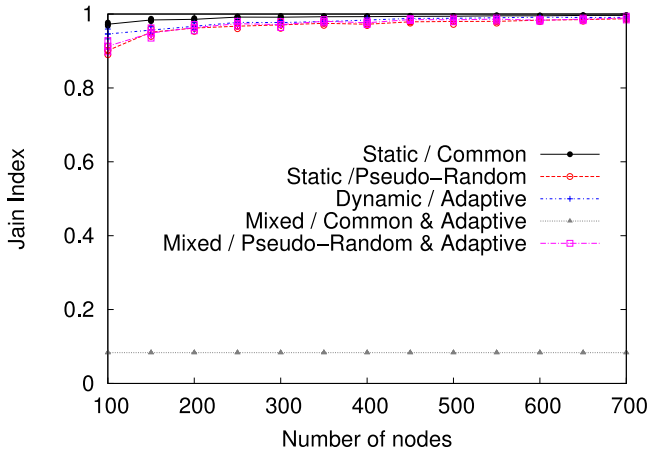
6.2 Network cardinality

Figure 4(a) presents the overhead in function of the number of nodes when maintaining constant density. The Static/Common and Mixed/Common & Adaptive strategies have the same minimal overhead: no deafness arises and all neighbors receive a broadcast transmission. Since some neighbors may present a non-null Packet Error Rate, several non acknowledged broadcasts are required before considering they are *covered*.

The Dynamic/Adaptive strategy shows a limited overhead by greedily choosing the most suitable channels. The Static/Pseudo-Random strategy requires a little less broadcast packets (7 transmissions). Indeed, this strategy results in lower connectivity: two nodes may be in the radio range with each other, but may not share a common static channel. In this case, this kind of a *virtual neighbor* is not anymore a *neighbor* in the multi-channel topology and has



(a) Overhead



(b) Jain Index

Figure 4: Impact of the number of nodes, 10 avg. neighbors, 3 interfaces, 12 channels

not to be covered. This mechanically reduces the overhead. The probability of such configuration is smaller with dynamic interfaces since we increase the probability that a pair of nodes has at least one channel in common at a given instant.

Finally, the Mixed/Pseudo-Random & Adaptive strategy presents the worst overhead, because it uses only one static interface, which reduces the possibilities to re-use one single transmission to cover several neighbors.

We have also evaluated fairness between different channels with the Jain Index (Figure 4(b)). Mixed Interfaces with Common/Adaptive Channel assign-

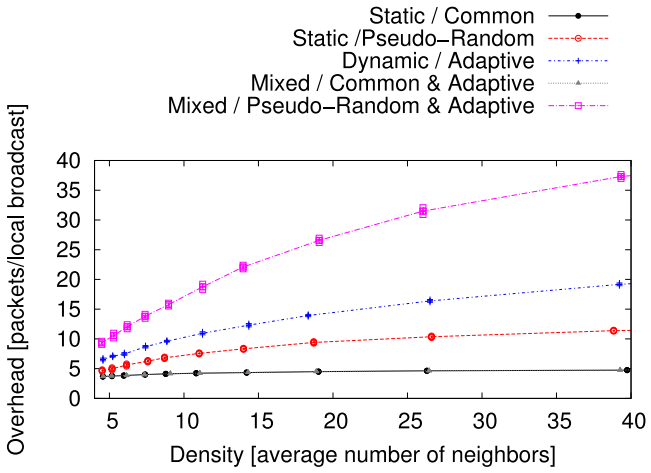


Figure 5: Impact of the density on the overhead, 200 nodes, 3 interfaces, 12 channels

ment result in the Jain index of about 0.08. Indeed, only the control channel (1 of the 12 channels) is used for broadcast leading to high unfairness.

Other strategies lead to almost perfect fairness: they efficiently spread the broadcast traffic through orthogonal channels reducing the risk of congestion. In particular, the Static/Common strategy does not use a single control channel, leading to a good fairness.

6.3 Density

We have also evaluated the impact of the density on the overhead while maintaining the number of nodes constant (cf. Figure 5). Only Static/Common and Mixed/Common & Adaptive strategies have the same overhead, which is perfectly scalable, because they have a common channel set.

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

The Mixed/Pseudo-Random & 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 in minimizing the overhead when the density is large.

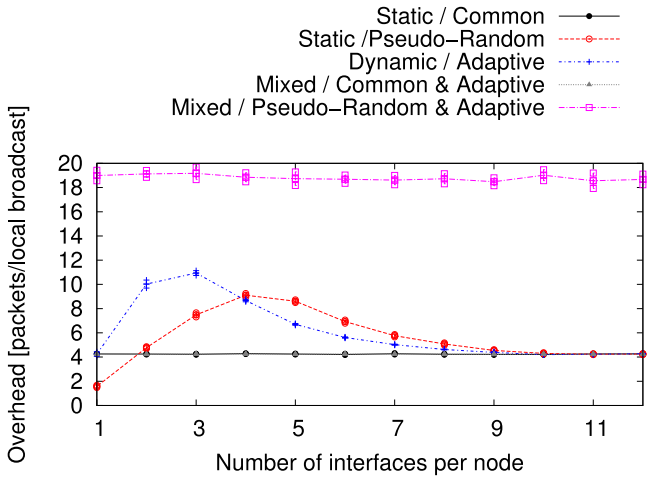


Figure 6: Impact of the number of interfaces on the overhead, 200 nodes, 10 avg. neighbors, 12 channels

6.4 Number of interfaces

We have also considered the influence of the number of interfaces (cf. Figure 6) on the overhead.

The Dynamic/Adaptive and the Static/Pseudo-Random strategies tend to have initially a growing overhead: the number of neighbors to be covered increases since they have more chance 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 begins to be more attractive when the number of interfaces is large compared to the number of channels (> 3 interfaces). Finally, for a very large number of interfaces (> 8), these strategies tend to be similar to the common channel strategies.

The strategies using a common channel for broadcast are not impacted by the number of interfaces. Besides, the Mixed/Pseudo-Random & Adaptive strategy presents also a constant overhead since the unique receiving interface keeps on being the bottleneck.

6.5 Impact of threshold $p_{cover_{min}}$

Finally, we have measured the impact of threshold $p_{cover_{min}}$ on the overhead in Figure 7. When $p_{cover_{min}} = 0.5$, each neighbor is covered when the transmitter sends one single copy: we discard radio links with a larger PER, considering them unreliable. Thus, strategies with a common channel set required only one broadcast transmissions to cover all the neighbors.

When $p_{cover_{min}}$ increases, the overhead becomes larger: neighbors with a

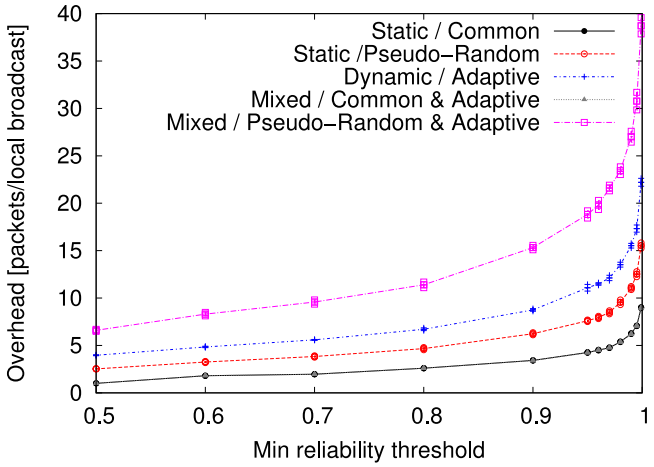


Figure 7: Impact of threshold $p_{cover_{min}}$ on the overhead, 200 nodes, 10 avg. neighbors, 3 interfaces, 12 channels

large packet error probability may require the transmission of several copies. However, we can note that all the strategies follow the same trend. The overhead becomes prohibitive when we require very large $p_{cover_{min}}$ (e.g. ≈ 0.99). Thus, the network protocols have to cope with broadcast unreliability. In particular, they should work in a self-stabilizing manner: even if some neighbors do not receive a particular broadcast packet, the protocol must work properly.

7 Conclusion and Future Work

We have proposed algorithms for local broadcast in multi-channel multi-interface wireless mesh networks. 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 Set strategy is not used. A greedy approach is particularly efficient to take advantage of the broadcast nature of transmissions.

We plan to study how we can deal with multiple rates: different bit rates may cover a different set of neighbors with different PER. We also plan to adapt the proposed strategies to dynamic conditions adopting an opportunistic approach. Besides, we aim at optimizing the delay, e.g. consider the question of which timeslot would present the best trade-off between the delay and the overhead when we use dynamic interfaces.

Acknowledgments

This work was supported in part by the French Government (MESR) and the Competitive Clusters Minalogic and System@tic under contracts FUI SensCity as well as the French National Research Agency (ANR) project ARESA2 ANR-09-VERS-017.

References

[1] C. Chaudet, D. Dhoutaut, and I. G. Lassous. Performance issues with IEEE 802.11 in ad hoc networking. *IEEE Communications Magazine*, 43(7), 2005.

[2] P. Li, N. Scalabrino, Y. M. 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), 2008.

[3] J. Qadir, C. T. Chou, A. Misra, and J. G. Lim. Localized minimum-latency broadcasting in multi-rate wireless mesh networks. In *IEEE WoWMoM*, 2007.

[4] Z. Yang, M. Li, and W. Lou. R-Code: Network coding based reliable broadcast in wireless mesh networks with unreliable links. In *IEEE GLOBECOM*, 2009.

[5] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou. A multi-radio unification protocol for IEEE 802.11 wireless networks. In *IEEE BroadNets*, 2004.

[6] X.-Y. Li Xufei M. and S. Kami Makki. Static channel assignment for multi-radio multi-channel multi-hop wireless networks. In *WTASA*, 2007.

[7] P. Kyasanur and N. H. Vaidya. Routing and interface assignment in multi-channel multi-interface wireless networks. In *IEEE WCNC*, 2005.

[8] D. Abdelali, F. Theoleyre, A. Bachir, and A. Duda. Neighborhood discovery and activity monitoring in multichannel mesh networks. In *IEEE WCNC*, 2010.

[9] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *ACM MobiCom*, 2004.

[10] 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), 2009.

[11] L. A. DaSilva and I. Guerreiro. Sequence-based rendezvous for dynamic spectrum access. In *IEEE DySPAN*, 2008.

[12] P. Bahl, R. Chandra, and J. Dunagan. SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In *ACM MobiCom*, 2004.

[13] 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.

[14] Y.-C. Tseng, S.-L. Wu, C.-Y. Lin, and J.-P. Sheu. A multi-channel mac protocol with power control for multi-hop mobile ad hoc networks. In *International Workshop on Wireless Networks and Mobile Computing*, 2001.

[15] Y. Ding, K. Pongaliur, and L. Xiao. Hybrid multi-channel multi-radio wireless mesh networks. In *IEEE IWQoS*, 2009.

[16] 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.

[17] Min Song, Jun Wang, and Qun Hao. Broadcasting protocols for multi-radio multi-channel and multi-rate mesh networks. In *International Conference on Communications (ICC)*, pages 3604 –3609. IEEE, june 2007.

[18] Kai Xing, Xiuzhen Cheng, Liran Ma, and Qilian Liang. Superimposed code based channel assignment in multi-radio multi-channel wireless mesh networks. In *international conference on Mobile computing and networking (MOBICOM)*, pages 15–26, Montreal, Canada, 2007. ACM.

[19] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *IEEE INMIC*, pages 62–68, 2001.

[20] graphsim. `http://forge.imag.fr/projects/graphsim`, broadcastWMN module.

[21] J. Camp, J. Robinson, C. Steger, and E. Knightly. Measurement driven deployment of a two-tier urban mesh access network. In *ACM MobiSys*, pages 96–109, 2006.