Avoiding redundant channel blocking in cooperative multi-channel MAC protocols through virtual topology inferencing

Stepan Ivanov, Dmitri Botvich, Sasitharan Balasubramaniam
Telecommunication Software and Systems Group
Waterford Institute of Technology
Carriganore campus, Waterford, Ireland
{sivanov, dbotvich, sasib}@tssg.org

Nina Popova
Faculty of Computational Mathematics and Cybernetics
Moscow State University
Vorobievi Gori 1, Moscow, Russia
popova@cs.msu.su

Abstract—**The mechanism of channel allocation used in the Medium Access Control (MAC) sub-layer for wireless devices in IEEE 802.11 standard is through a single MAC channel. The drawback with the single channel solution is when the number of nodes increases, the throughput is also reduced. However, using multiple channels solves this problem and increases channel utilisation efficiency, where channel allocations can be managed through distributed cooperative negotiations. Cooperation techniques of channel allocations have been investigated by T. Luo et al [6] (Asynchronous Cooperative Multi-channel MAC protocol (CAM-MAC)), which uses cooperative neighbour updating and vetoing techniques for channel selection. The technique allows nodes to efficiently select channels which in turn leads to higher network performance. However, using such cooperative technique can lead to "Redundant channel blocking", which blocks certain nodes from using valid channels that are not in range with nodes using the same channel. In this paper, we propose a solution to mitigate this problem using virtual topology inferencing, where simulations have also been presented to validate the proposed solution.**

I. INTRODUCTION

The IEEE 802.11 standard supports wireless data transmissions [9], where it uses single data channel for MAC layer and supports multiple channels in physical layer. Due to the single channel in the MAC layer, each node has to share this channel with all other nodes located in its transmission range and effectively use the same channel for control as well as data transmission. When the number of nodes increases, this leads to noticeable performance degradation. One solution to this problem is to increase the number of channels in order to increase the overall throughput. However, this technique poses certain challenges for large scale distributed ad hoc networks. One of these challenges stems from the limitations of IEEE 802.11 device, where each device is equipped with only a single half-duplex transceiver. Therefore, each device is only allowed to transmit or listen to only one channel at a time. In the event the device is transmitting data through a channel or listening to a channel, it doesn't have the ability to listen or communicate through other channels.

One approach towards supporting distributed channel allocation for single transceiver devices is through cooperation. However, most solutions have limitations such as tight clock synchronization or require modifications to standard IEEE 802.11, which make their implementation quite difficult. However, the CAM-MAC solution in [6] does not require time synchronization or changes to standard equipment. One drawback of the solution is when information of selected channels are propagated to distant nodes, this can prevent nodes from using the specific channel even if the node's using that channel is not in range. We refer to this as the "Redundant Channel Blocking" (RCB) problem. In this paper we investigate the RCB problem, and propose a solution towards mitigating this problem through virtual topology inferencing, where nodes will observe negotiations performed by other nodes and determine the topology of their neighbouring nodes. Based on the virtual topology, nodes will be able to accurately cooperate with other nodes during the channel allocation process.

The rest of the paper is organized as follows: Section II presents the Related Works. Section III describes the cooperative negotiation of existing work, while section IV describes RCB problem. Section V presents the virtual topology inferencing technique, and Section VI presents the simulation of the proposed approach. Finally section VII presents the conclusion.

II. RELATED WORK

In this section, Split phase [8] solutions for multi-channel MAC protocols are presented. As mentioned before, all the solutions use standard IEEE 802.11 equipment which contains a single half duplex tuneable transceiver [9]. Each device contains a "control channel", and the remaining channels that are used for data transmission are known as "data channels". The control channel is used for negotiating the data channels with the neighbouring nodes.

The MMAC protocol, proposed by So and Vaidya [1] allows transmitting data through both control channel and data channels. The control channel is predefined for all nodes and is divided into Beacon intervals of fixed size where each interval contains an ATIM window, and the remaining interval is for data transmission. In each ATIM window, nodes negotiate in order to schedule data transmissions. All nodes are tuned to the control channel in each ATIM window which requires time synchronization between the nodes. At the same time, fixed size Beacon interval results in time wastage in cases of different packet sizes. Maheshwari et al. [2] proposed a similar protocol with variable ATIM window size. Cordiero and Challapali [3] proposed the C-MAC protocol where nodes negotiate in predefined slots as it is done in MMAC. The distinguishing feature of C-MAC is using dynamically defined and distributed control channel. A disadvantage of this protocol is that the negotiation duration must include all neighbouring nodes. This leads to a restriction on the amount of neighbours, which can be large in real deployment situations. Extending the negotiation phase will make the protocols inefficient for topologies with small node density.

The AMCP protocol, proposed by Shi et al [4] allows nodes to negotiate through the control channel to select a suitable data channel for transmissions. During negotiation, each node can get information of utilized channels and avoid using these channels. However, during data transmission the sender and the receiver don't listen to the control channel (since they have switched to the data channel). Hence they are not fully aware of new channels that are selected by the neighbours, and in the events the node selects the same channel this will lead to data collision. To solve this problem the sender and the receiver refrain from using those channels for duration of one data packet transmission. This leads to increasing packet transmission delay and data channels under utilization. A similar problem is also encountered in the Bi-McMac protocol, proposed by Kuang and Williamson [5]. However, the CAM-MAC protocol proposed by T. Luo [6], counters the problems above by allowing the neighbours to further propagate information to other neighbours during negotiation, to update the latest channels being used by surrounding nodes. The next section will describe the CAM-MAC [6] protocol. Results from [6] show that CAM-MAC protocol is more efficient than IEEE 802.11. However, CAM-MAC has certain drawbacks that affects the performance. In this paper we present a solution towards mitigating this problem.

III. COOPERATIVE NEGOTIATION

Our solution is based on the extension of the CAM-MAC protocol. In the solution we consider pre-defined routing mechanisms between the nodes. The bandwidth is divided into multiple orthogonal channels, and each node has a single IEEE 802.11 transceiver. The transceiver can transmit or listen to only one channel at a time. A node is able to receive data from a neighbour which is within a certain transmission range and using the same channel.

Similar to the AMCP and the Bi-McMac protocol, the CAM-MAC negotiates through a fixed control channel, and transmits data through the data channel. However, the CAM-MAC protocol has a technique of updating its neighbours of the current channels being used by the surrounding nodes. The next section will briefly describe the CAM-MAC negotiation process. For the process of negotiation we use the terms "control session" for control negotiation, and "data session" for data transmission. During a data session, a sender transmits only one fixed size data packet. To ensure reliability of packet delivery, an acknowledgment is sent back through the selected data channel to confirm the receipt of the data packet.

A. *Negotiation without cooperation*

In this section we illustrate and describe the CAM-MAC negotiation protocol. Each time a node is neither receiving nor sending data, the node will tune to the control channel (idle state) and collect information about data channels that are negotiated by the neighbours. However, the nodes can't collect this information when they're taking part in data transmission because they are equipped with only one half-duplex transceiver. Therefore, during data transmission, nodes will not be aware of the latest channels negotiated and used by neighbouring nodes. Fig. 1 (a) – (c) illustrate a case when the nodes negotiate without full knowledge.

In the example presented in Fig. 1 (a), initially a data transmission from node 1 to node 2 through Channel 1 was established. During the negotiation between nodes 1 and 2, the surrounding nodes were updated that channel 1 is used.

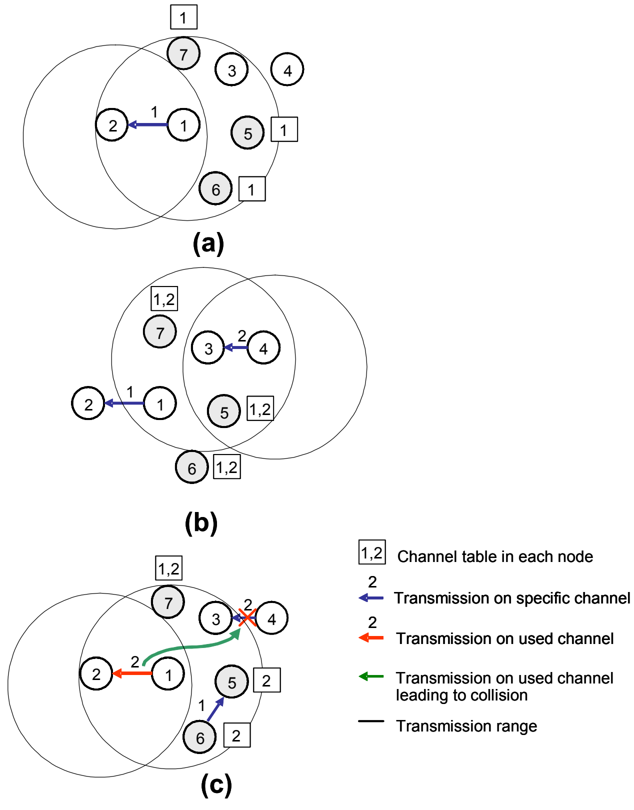


Figure 1. Drawback of a negotiation with limited information

Immediately after that node 3 and 4 began negotiating for the use of channel 2, and again this information is updated to surrounding nodes (Fig. 1 (b)). However, this occurred when nodes 1 and 2 were transmitting through channel 1. Therefore, nodes 1 and 2 will not have new updated information about channel 2 being used by nodes 3 and 4. Now let's consider nodes 5 and 6, which began negotiating and transmitting through channel 1 after nodes 1 and 2 finished. The negotiation of nodes 5 and 6 will be heard by nodes 1 and 2, leading nodes 1 and 2 to negotiate for the use of channel 2. However, this will lead to collision with nodes 3 and 4 using channel 2. This is due to the fact that the negotiation between nodes 3 and 4 occurred when nodes 1 and 2 were transmitting data and didn't have an opportunity to update the latest channels used by its neighbours.

B. *Negotiation through CAM-MAC*

In order to mitigate the problems of channel collision, the CAM-MAC [6] protocol incorporates vetoing technique through cooperation of neighbouring nodes. In the event that nodes 1 and 2 begins negotiations for channel 2, the request will be heard by node 7, which will then veto the request, leading nodes 1 and 2 to back-off and renegotiate for a different channel. Therefore, the benefits of the CAM-MAC protocol is to update nodes with the latest knowledge of used channels in order to minimize any possible collision with nearby nodes that are using the same channels.

IV. Redundant Channel Blocking

Although the CAM-MAC protocol ensures the nodes are fully updated with the latest information, there is a major drawback with the protocol. This drawback is the RCB problem which is illustrated in Fig. 2 (Fig. 2 is the example from Fig.1). As described earlier, the information of channel 2 being used by nodes 3 and 4 is propagated to nodes 5 and 7, which are in the ranges of nodes 3 and 4.

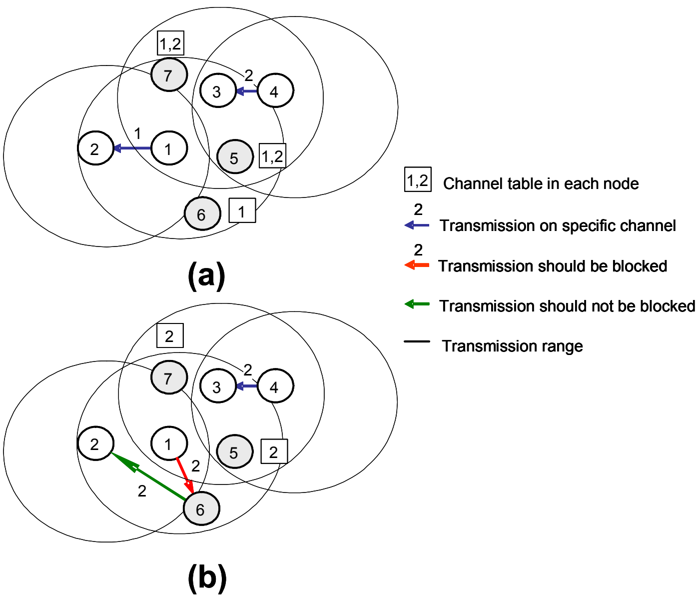


Figure 2. Redundant channel blocking

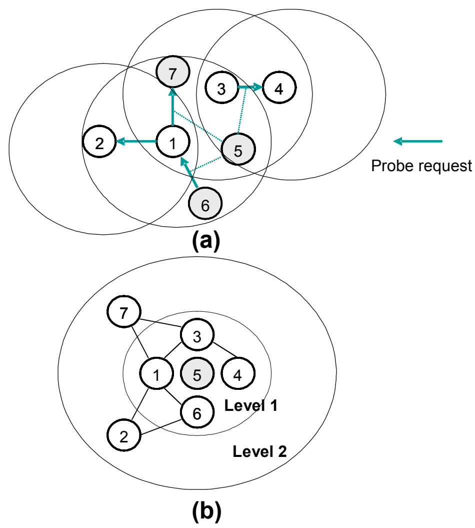
The drawback with this is that, in the event that node 6 would like to transmit data and begin negotiation, node 6 will not be able to use channel 2 even though it is out of range from nodes 3 and 4. This will occur because when node 6 begins negotiation, node 5 will immediately veto and prevent node 6 from using channel 2 even if nodes 6 and 2 are out of range from nodes 3 and 4. This will in turn prevent nodes from transmitting at higher throughput, if the knowledge of used channels in the neighbourhood is not clearly propagated.

To address the RCB problem a node should not only determine the channels that are used in the neighbourhood, but also determine the surrounding topology. The mechanism of topology estimation is described in the next section.

V. Virtual Topology Inferencing

As we mentioned above using information of only data channel states leads to the RCB problem. In the example shown on Fig. 2(b) if node 5 knew that nodes 6 and 2 are not in range of nodes 3 and 4, it should not prevent node 6 from negotiating the use of channel 2. In order to avoid this problem, we propose the virtual topology inferencing technique that each node performs by listening to neighbours negotiations. Through the inferencing process, each node can create a virtual topology knowledge of surrounding nodes.

Fig. 3(a) illustrates the mechanism, where node 5 will listen to the neighbours' negotiation process. During the negotiation process, each node transmits a probe request which contains the source and destination address, and the channel being requested.



Level 1	Level 2
3 (1, 4)	7
4 (3)	
6 (1)	2
1 (3, 6)	2, 7
⋮	⋮
⋮	⋮

(c)

Source	Destination	Channel
3	4	1
1	2	2
⋮	⋮	⋮
⋮	⋮	⋮

(d)

Figure 3. (a) Node 5 sensing neighbours probe request, (b) virtual topology, (c) virtual topology table by node 5, and (d) channel table of node 5

Node 5 will then categorise the different source and destination addresses and their corresponding nodes to either level 1 or level 2 neighbours in its virtual topology. Level 1 indicates that the node is an immediate neighbour, while level 2 indicates the node is a neighbour of the neighbouring node

(Fig. 3 (b)). This will lead to each node containing two tables, which is the table of virtual topology (Fig. 3 (c)), and the table of channels used by the neighbouring nodes (Fig. 3 (d)).

Fig 4 illustrates (based on the scenario of Fig. 2 (b)) the negotiation process for with and without virtual topology knowledge. The negotiation process is with respect to node 5's vetoing activity with neighbouring nodes. As shown in Fig. 2, in the event that node 4 and 3 start transmitting on channel 2, this information will be recorded by node 5.

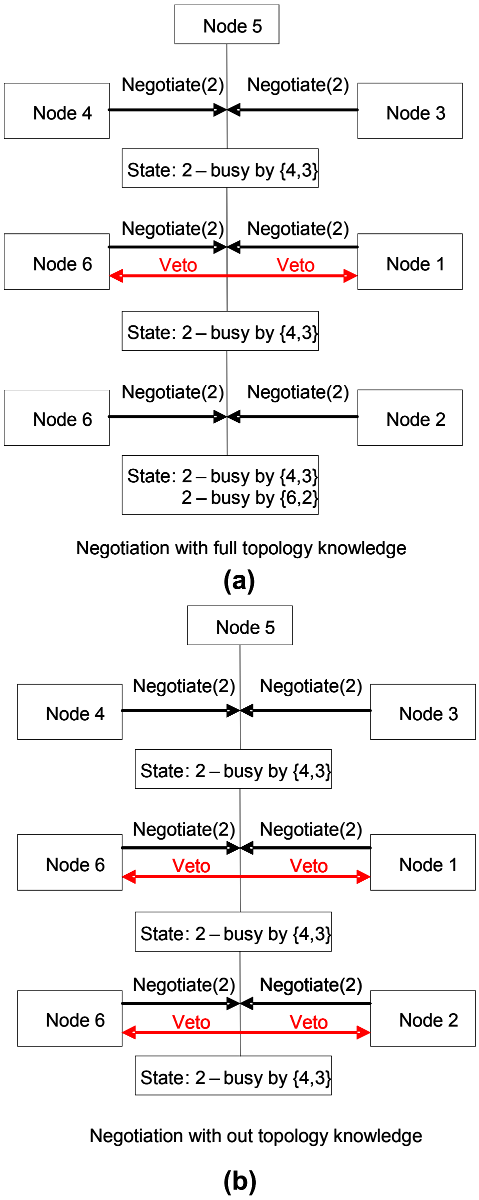


Figure 4. (a) Cooperative negotiation with virtual topology knowledge, (b) cooperation without virtual topology knowledge

Fig. 4 (b), shows that when both node 6 and 1 or node 6 and 2 negotiate to use channel 2, this will be blocked by node 5. However, in the case of cooperation with virtual topology knowledge in Fig. 4 (b), only the node 6 and 1 will be blocked by node 5 from using channel 2. Our technique allows nodes to continually infer and learn of new nodes in their neighbourhood which is an advantage for supporting networks with node mobility.

VI. SIMULATION

In order to validate our virtual topology inferencing and its benefits, we have performed simulations and compared to the CAM-MAC protocol [6] (our solution avoiding RCB problem is referred to as CAM-MAC ARCB). Our simulation is divided into two sections, which includes tests for 40 nodes with different loads, and for different number of nodes with 60% load.

A. 40 node topology with varying load

Fig. 5 – 7 presents the results of our simulations. We use pair-wise networks where nodes form sets of pairs and each node sends and receives data from neighbouring nodes. In our simulation, the nodes form a grid topology. We assume that upper layer functionalities (e.g. routing) have been set and rely on the underlying MAC layer to allocate the channel for transmission. Data packets are generated at each sender according to a Poisson process.

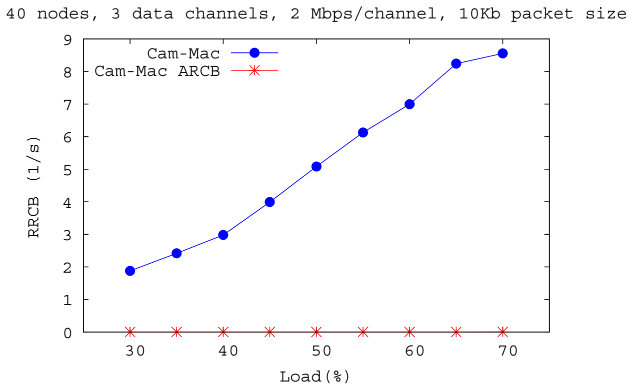


Figure 5. Rate of channel blocking

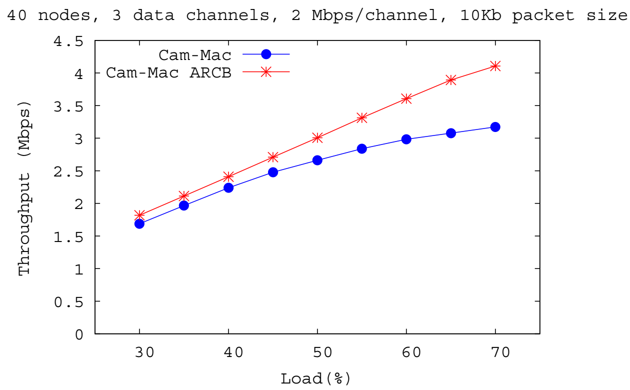


Figure 6. Average throughput

Each node is equipped with a transceiver which contains one control and 3 data channels, each with 2 Mbps capacity. All statistics are averaged over 10 simulation runs. Each simulation is performed on a static ad-hoc network of 40 nodes with average neighbouring node density of 15. In the simulations, we consider each MAC layer packet delivery timeout to be 100 ms which indicates maximum time the packet can reside in the queue. To evaluate the performance comparison with CAM-MAC, we use the following metrics: Average throughput – average amount of data successfully transmitted; Average delay overhead – average time that a successfully delivered packet has spent in the buffer while waiting for a free channel (ms); Rate of RCB cases (RRCB) - average number of times when negotiation of eligible channel was vetoed (1/s).

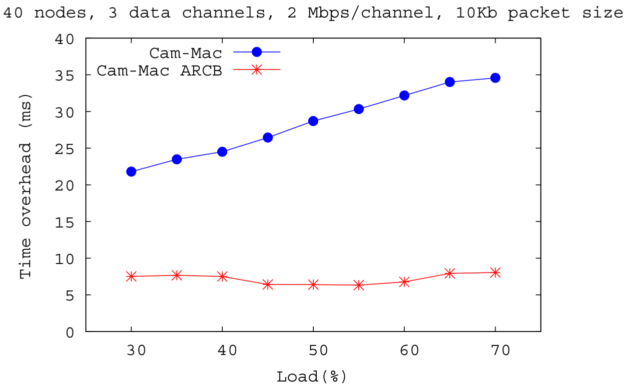


Figure 7. Delay Overhead

Fig. 5 illustrates the results for RRCB metrics and shows that the RCB worsens for CAM-MAC when the network load increases compared to CAM-MAC ARCB. Fig. 6 illustrates the average throughput and shows that as the network load increases, the virtual topology benefit increases (there is a very little difference when the load is low, but CAM-MAC ARCB is still slightly better than CAM-MAC). Fig. 7 illustrates the delay overhead that packets wait in the buffer before transmission, and as shown in the result, the delay is reasonably static for all loads in CAM-MAC ARCB case, but worsens for CAM-MAC as the network load increases.

B. Varying number of node with 60% load

The results for varying number of nodes with 60% load are illustrated in Table 1 and 2.

TABLE I. PERFORMANCE FOR CAM-MAC

No. of nodes	Average Throughput (Mbps)	Average Delay overhead (ms)	RRCB
50	2.93	31.84	6.59
60	2.79	31.48	6.87
70	2.75	32.05	6.87

The results illustrate that as the topology size increases, the average throughput, average delay overhead, and RRCB for CAM-MAC ARCB outperforms the CAM-MAC. At the same time, we can see more stable results for CAM-MAC ARCB as the number of nodes increases for all metrics.

TABLE II. PERFORMANCE FOR CAM-MAC ARCB

No. of nodes	Average Throughput (Mbps)	Average Delay overhead (ms)	RRCB
50	3.59	7.50	0.003
60	3.61	7.09	0.003
70	3.59	8.57	0.0031

VII. CONCLUSION

The use of multiple channels on single transceiver devices ensures higher throughput capability for wireless networks with large number of nodes. However, a distributed channel allocation mechanism is required to ensure efficient use of channel resources. Cooperation between the nodes can help ensure that channels are efficiently allocated between the nodes. In this paper, we have identified the Redundant Channel Blocking problem for the CAM-MAC negotiation protocol and extended this through virtual topology inferencing to improve its performance. Simulation comparison has also shown the improvements for varying network loads.

ACKNOWLEDGEMENT

This work has received support from the Higher Education Authority, Ireland under the PRTLI Cycle 4 programme, in the project - Serving Society: Management of Future Communications Networks and Services.

REFERENCES

[1] J. So and N. Vaidya, “Multi-channel mac for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver,” in MobiHoc, 2004.

[2] R. Maheshwari, H. Gupta, and S. R. Das, “Multichannel MAC protocols for wireless networks,” in IEEE SECON, 2006.

[3] C. Cordeiro, K. Challapali “C-MAC: A Cognitive MAC Protocol for Multi-Channel Wireless Networks,” in DySPAN 2007.

[4] J. Shi, T. Salonidis, and E. W. Knightly, “Starvation mitigation through multi-channel coordination in CSMA multi-hop wireless networks,” in ACM MobiHoc, 2006.

[5] T. Kuang, C. Williamson, “A Bidirectional Multi-Channel MAC Protocol for improving TCP performance on Multihop Wireless Ad Hoc Networks”, in MSWiM, 2004.

[6] Tie Luo, Mehul Motani, Vikram Srinivasan, “CAM-MAC: A Cooperative Asynchronous Multi-Channel MAC Protocol for Ad Hoc Networks”, Proceedings of BroadNets 2006, San Jose, CA, USA.

[7] Tie Luo, Mehul Motani, Vikram Srinivasan, "Cooperative Asynchronous Multichannel MAC: Design, Analysis, and Implementation," IEEE Transactions on Mobile Computing, vol. 8, no. 3, pp. 338-352, Mar. 2009

[8] J. Mo, H. W. So, and J. Walrand, “Comparison of multichannel mac protocols,” in MSWiM, 2005.

[9] IEEE 802.11 Working Group, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 2007.