A Scalable Decentralized MAC Scheduling for Cognitive Wireless Mesh Network

Mehdi Msakni, Luong Dinh Dung, Mohamad Haidar and Zbigniew Dziong
Electrical Engineering Department
Ecole de Technologie Supérieure (ETS)
Montréal, Quebec, Canada
e-mail : {mehdi.msakni.1, luong.dinhdung.1, mohamad.haidar.1, zbigniew.dziong}@ens.etsmtl.ca

Abstract— **Cognitive Wireless Mesh Networks (CWMN) is a promising technology that combines the advantages of Wireless Mesh Networks (WMN) with the capacity enhancement feature due to the use of available channels discovered with cognitive radio technology. In CWMN, Medium Access Control (MAC) layer has to schedule data communications in a dynamic environment in which available channels change in space and time. Therefore, scheduling in a CWMN is more difficult than scheduling in multi-channel IEEE 802.11 since each node in a CWMN can support different set of channels for data transmission where as in IEEE 802.11 all nodes share same set of channels. In this paper, we propose an efficient link scheduling algorithm in a distributed architecture in CWMN. The solution utilizes 2-distance vertex coloring scheme at the node level which increases the processing speed of the scheduling algorithm and lessens the overhead control data. Simulation results show that the proposed algorithm improves the scalability, the speed, and the amount of control data exchange when compared with existing algorithms.**

Keywords-Wireless Mesh Networks; Cognitive Radio; Medium Access Control layer scheduling; link scheduling; vertex colouring

I. INTRODUCTION

Wireless Mesh Networks (WMNs) have a growing popularity due to its simple architecture, ease of installation, and low costs of maintenance [1]. Because WMNs utilize the air medium, this makes it susceptible to radio frequency interference that limits the network capacity. In fact, the radio frequency interference limits the number of simultaneous transmissions in a single channel causing the network capacity to drop [2].

One approach to enhance the network capacity is to utilize the multiple available channels that are discovered by means of Cognitive Radio (CR) [3]. CR is a promising technology that allows communication on channels without acquiring a license. The nodes, in reference to the WMN nodes in a CR system, also referred to as secondary users (SUs), sense the spectrum periodically to find unutilized channels. Then, the SUs communicate among themselves using the discovered unutilized channels with the condition that they do not interfere with channels' owners, also known as the primary users (PUs). Therefore, a Cognitive Wireless Mesh Networks (CWMN) requires a new MAC layer enhancement that can meet the challenges of this new environment. Unlike multi-channel networks, such as IEEE 802.11, where the set of channels is shared among all nodes, this type of network faces continuous changes in availability of channels in space and time. In fact, each node in the network can have different set of channels. Consequently, any pair of nodes that wish to communicate has to establish a common communication channel and time to exchange packets without causing interference with other existing transmissions. Therefore, CWMN requires an efficient scheduling algorithm that can tackle the above challenges.

In this paper, we consider Time Division Multiple Access (TDMA) as our channel access protocol. Since it avoids collision and provides efficient channel utilization. The goal of TDMA MAC scheduling in a CWMN is to minimize the number of timeslots needed for data transmission by assigning a channel and a timeslot to each link without interfering with already scheduled links. Each TDMA slot is assigned to at least one link which represents the transmitter-receiver pair that has data to exchange. To achieve this objective, we propose an algorithm to schedule the links in a distributed manner.

The rest of the paper is organized as follows. Section II defines the problem at hand in a set of equations. Section III describes the related works to the domain. Section IV reveals our approach to the link scheduling algorithm. Section V presented the numerical results of our approach. Finally, Section VI concludes the paper.

II. PROBLEM FORMULATION

Our motivation is to improve the scalability of the network, reduce the amount of control overhead, and minimize the overall scheduling time. We consider a CWMN in which the nodes are equipped with an omni-directional radio antenna and use one common control channel. We assume that all the radios have a channel switching delay equal to zero and their interference range is equal to their communication range. Each node may have different set of data channels available.

A link l is defined by:

- A transmitter $t(l)$ and a receiver $r(l)$ which are in the communication range of each others and have a packet to exchange.

ISBN: 978-1-61208-123-6

- *$LinkChSet_l$* is the set of common available channels between nodes *t(l)* and *r(l)* and is not empty.
- *$InterferenceSet_l$* is the set of links that can interfere with link *l*. Link *n* belongs to *$InterferenceSet_l$* if *r(l)* is in the communication range of *t(n)* or if *r(n)* is in the communication range of *t(l)*.
- *$ExclusionSet_l$* is the set of links that have at least a node in common with link *l*. Link *n* belong to *$ExclusionSet_l$* if the transmitter *t(n)* or the receiver *r(n)* is also the transmitter *t(l)* or the receiver *r(l)*.
- X_{lcs} is a variable where

$$X_{lcs} = \begin{cases} 1\, if\ \ link\ l\ is\ active\ on\ channel\ c\ and\ timeslot\ s. \\ 0\ otherwise. \end{cases}$$

In CWMN, which have a set of links *LinkSet* and a set of channels *ChSet*, the scheduling algorithm should satisfy:

$$\forall l \in LinkSet, \sum_{c \in LinkChSet_l} \sum_{s=1}^{k} X_{lcs} = 1 \qquad (1)$$

$$\forall l \in LinkSet, \forall c \in LinkChSet_l, \forall s\, 1 \le s \le k, \\ X_{lcs} + \sum_{n \in InterfercenceSet_l} X_{ncs} \le 1 \qquad (2)$$

$$\forall l \in LinkSet, \forall s\, 1 \le s \le k, \\ \sum_{c \in ChSet} X_{lcs} + \sum_{c \in ChSet} \sum_{n \in ExclusionSet_l} X_{ncs} \le 1 \qquad (3)$$

$$\forall l \in LinkSet, \forall c \notin LinkChSet_l, \forall s\, 1 \le s \le k, X_{lcs} = 0 \qquad (4)$$

Equation (1) states that each link is assigned to one and only one channel and one timeslot. Equation (2) verifies that the interfering links are not assigned to the same channel and timeslot. Equation (3) ensures that a node does not perform two operations simultaneously (i.e. transmit two packets or receive two packets or transmit one packet and receive one packet). Equation (4) guarantees that link *l* can only be scheduled on a channel that is common to *t(l)* and *r(l)*. Our objective is to propose a scalable and fast scheduling algorithm that satisfies the four equations while minimizing the number of timeslots, *k*, used to schedule all the links in the network.

III. RELATED WORK

In multi-channel single-radio 802.11 networks, different approaches to scheduling at the MAC layer have been proposed. In [4], [5] and [6] a MAC layer scheduling based on reservation of channel by RTS/CTS mechanism on the control channel has been presented. In [7], the proposed approach is divided into two steps: The first step is a control one during which the nodes use the control channel to select the channels to be used, and, in the second step the nodes send their data on the selected channel using the RTS/CTS mechanism. In [8] and [9], nodes synchronously execute a common sequence of hops across all channels. A pair of nodes stops performing the channel hopping sequence in order to make data transmission in which they reserve the channel by RTS/CTS mechanism. Once data transmission ends, they rejoin the common hopping sequence. In [10] and [11], each node carries out a different sequence of hops generated from a random number. A pair of nodes wishing to communicate must meet on a particular channel, stops the sequence of hops to carry out the data transmission, and rejoins their respective sequence of hops at the end of the transmission.

For a single-radio Cognitive network, different MAC layer scheduling algorithms have been proposed. A MAC layer scheduling based on reservation of channel by RTS/CTS mechanism on the control channel was presented in [12], [13], and [14]. The objective of [15] is to achieve efficient channel and timeslot assignments to the links in a distributed way which is compared to the optimal scheduling solution found by an Integer-Linear Programming (ILP) formulation. In the considered cognitive radio network, the nodes are equipped with a radio and a control channel. The CSMA/CA scheme is used to access the control channel and the TDMA scheme is used to access the data channels. Each node has a rank which depends on the number of active links and the number of channels. The node with the highest rank in its two-hop neighborhood processes the algorithm of assignment of timeslots. This algorithm assigns the first available timeslot to the link. Nodes that finish the execution of the algorithm are marked as covered. Then, the uncovered nodes with the highest rank in its two-hop neighborhood execute the algorithm. As soon as all the nodes are covered, each node with the lowest rank in its two-hop neighborhood starts to examine if it has the highest schedule length (the number of timeslots needed to schedule all the links) in the network. Finally, the node that has the highest schedule length broadcasts a message to all the nodes to indicate the highest schedule length and the start time of communication phase. In case of tie among two or more nodes, the highest ID node broadcasts its message to all of the remaining nodes.

From the results shown in [15], we can see that this approach, referred to as ranking approach, demonstrates a schedule length near to the optimal. However, the ranking approach presents a problem of scalability, high overhead, and a long execution time. The channel and timeslot assignment algorithm in [15] is carried out according to an order established by the ranks of the nodes and has a low simultaneous execution number. In a chained wireless network topology whose nodes have ranks which increase gradually starting from the beginning of the chain, only one node execute the algorithm at a time. Therefore, the total time of scheduling will increase exponentially with the number of nodes in the network. Moreover, the use of CSMA/CA scheme forces to send a copy of the control packet to each of the neighbors. Each sent control packet requires RTS, CTS and an ACK packet which causes a significant increase in the amount of control data exchanged. Finally, the use of the backoff mechanism also adds an additional delay to access the control channel.

IV. SCHEDULING ALGORITHM

In this section, we present our link scheduling solution in the network. To achieve a scalable solution, each node is assigned a color which is unique in its two-hop neighborhood (vertex coloring) [16] and [17]. The order of execution of the channel and data timeslot assignment algorithm is determined by the color of the node. So the nodes which have the same color carry out the algorithm at the same time. Therefore, the total time of scheduling is defined by the number of colors which depends on the network's node density.

To avoid any inconvenience due to the use of CSMA/CA, the nodes utilize the TDMA scheme to send control packets on the control channel. Therefore, there will be two sub-frames in the TDMA frame as shown in Figure 1. The control sub-frame is divided into mini-timeslots to send control packets and the data sub-frame is divided into timeslots to send data packets. By avoiding contention to access the control channel, a node can broadcast a control packet in the control mini-timeslots to its neighbors which leads to reduce the amount of data control packets. In a control sub-frame, each control mini-timeslot corresponds to a color in which the nodes with this color can transmit. It is of importance to note that TDMA synchronization in the network is achieved by using a GPS device and that the nodes sense the spectrum and exchange control information about available channels between each frame.

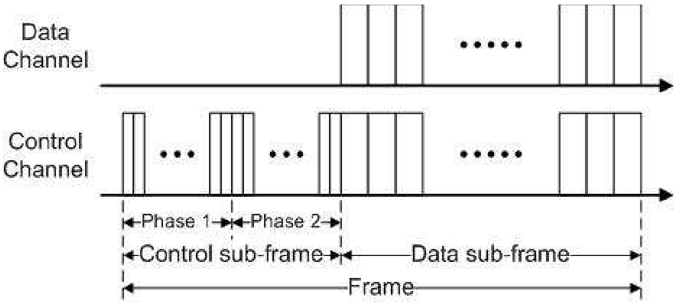

Figure 1. Structure of frame which is composed of a control phase and a data communication phase in the CWMN

In our coloring approach, the execution of the channel and data timeslot assignment algorithm needs two phases. In the first phase, referred to as the scheduling link phase, nodes schedule their outgoing links where as in the second phase, referred to as the establishing data sub-frame size, nodes determine the maximum data sub-frame size in the network. Here are more details about each phase:

A. Phase 1- Scheduling Link Phase

In the first phase, the execution order of the channel and data timeslot assignment algorithm is done according to node colors. Nodes who have the lowest color index execute the algorithm first. In a network with *M* colors, the first phase will need *M* step as described in the Figure 2. Each step has *M* control mini-timeslots. At the beginning of a step *i*, all nodes of color *i* assign channels and data timeslots to their outgoing links and broadcast the scheduling information in the message *Schedule_Information* to their neighbors in the first control mini-timeslot (which have the color *i*). In particular, a node selects the outgoing links with the lowest number of available channel. In case of a tie, the source node selects the outgoing links based on the lowest ID of the destination node and so forth. Then, this node assigns to each outgoing link a data timeslot without causing interference with the already scheduled links by starting with the first available data timeslot of each channel. Step *i* have *M* control mini-timeslots to allow the neighbors of different colors to forward the information in the two-hop neighborhood. Control mini-timeslots are colored from *i* to *M* then from 1 to *i*-1. At the end of step *i*, all nodes of color *i* are marked covered. Thus, the total number of control mini-timeslots needed to schedule all the links in the network is M^2.

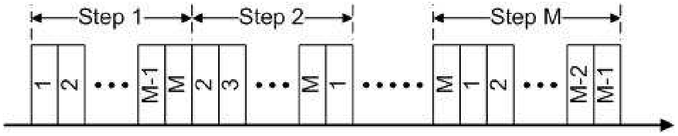

Figure 2. Timeslot colouring in the first phase of the scheduling algorithm

B. Phase 2 – Establishing the Data Sub-Frame Size

In the second phase, nodes determine the maximum data sub-frame size in the network. The control mini-timeslots are colored cyclically from *M* to 1 as presented in Figure 3. Nodes that have the highest color in its two-hop neighborhood are termed as root nodes and start to establish the maximum data sub-frame size in a breadth-first fashion. The breadth-first fashion is described as follows. A root node sends an *Attempt_size* message which includes the following information: its actual maximum data sub-frame size, identity of the root node, and identity of parent node which is equal to its identity. The neighbors of a root node become its children in the logical tree.

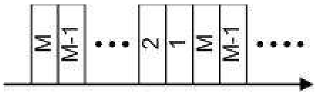

Figure 3. Timeslot colouring in the second phase of the scheduling algorithm

When a node *x* receives an *Attempt_size* message of the root node *z* from node *y* does one of the following:

- If the maximum data sub-frame size contained in the message is bigger than the actual maximum data sub-frame size of node *x*, then node *x* becomes a child of node *y* in the logical tree of root node *z* and updates its maximum data sub-frame size. In case of a tie, for the maximum data sub-frame size, node *x* does one of the following: (i) if node *x* is not part of any logical tree, then, node *x* becomes a child in the logical tree of root node *z*, (ii) else node *x* is already a part of a logical tree of root node *v*. Therefore, node *x* becomes a child in the logical tree of root node *z* if the ID of node *z* is higher than the ID of node *v*. After becoming a child in the logical tree of root node *z*, node *x* writes in the *Time_Recv_Attempt* message the reception time of the message *Attempt_size*. It broadcasts the information about the maximum data sub-frame size to all of its neighbors in the message *Attempt_size* which includes the maximum data sub-frame size, the identity of the root node, and the identity of the parent node. All of the neighbors except its parent become a potential child of node *x*. By receiving the message *Attempt_size*, node *y* has a confirmation that node *x* is its child in the logical tree.

- If the maximum data sub-frame size contained in the message is lower than the actual maximum data sub-frame size of node *x*, then node *x* discards the message.
- If the message indicates that node *x* is the parent of node *y*, then node *x* changes the status of node *y* from a potential child to child.
- If node *x* is already part of the logical tree of the root node *z* and the message indicates that node *w* is the parent of node *y*, then node *x* removes node *y* from its list of potential children. When a node has empty list of potential children and has no children, it sends an *Acknowledgement* message which contains *Time_Recv_Attempt* to its parent. When a node has *Acknowledgement* messages from all of its children and has an empty list of potential children, it sends an *Acknowledgement* message to its parent. The *Acknowledgement* message contains the highest value of *Time_Recv_Attempt* message received in the *Acknowledgement* messages of its children.

The root node who has received an *Acknowledgement* message from all of its children deduce that its data sub-frame size is the highest and that all nodes in the network are part of its logical tree. So, it has spread successfully the maximum data sub-frame size to all the nodes in the network. Then, it computes the delay to reach the furthest node from the highest value of *Time_Recv_Attempt* and the time of transmission of the message *Attempt_size*. With this delay known, the node knows the time it will take for its message to reach all the nodes in the network and, then, can establish the start time of the communication phase. Then, the node sends a *Communication_Phase_Information* message which contains the start time and the number of timeslots of the data communication phase and indicates the end of the second phase. The *Communication_Phase_Information* message travel through the logical tree.

V. PERFORMANCE ANALYSIS

In this section, we compare the ranking approach and the coloring approach in two sets of simulations to determine scalability improvement and the reduction of the execution time of the channels and data timeslots assignments. To achieve this objective, we analyze the time spent for transmitting control packets and data packets, the data sub-frame size, and the goodput of a link. The results shown in the figures below present the mean value of 50 simulations of 60 seconds each. The simulation platform used in our simulations for the two approaches is NS-2 [18]. Nodes have a communication range of 250m and always have a packet of 1500 bytes to send to each neighbor. The number of channels available at each node is randomly chosen between 1 and 25. The channel identities are chosen between 1 and 50. The data timeslot is 0,6675ms which is the time necessary to transmit a packet of 1500 bytes. In the first set of simulation scenarios, nodes are placed to form a grid topology and separated by a distance of 176m. The number of nodes in the grid is 16, 25, 36, 49, 64, 81, 100, 121, and 144. In the second set of simulation scenarios, nodes are placed randomly in a square area. In this set, the area of the network is 467m x 467m, 738m x 738m, 1044m x 1044m, 1279m x 1279m, 1477m x 1477m, 1651m x 1651m, and 1809m x 1809m for 10, 25, 50, 75, 100, 125, and 150 nodes, respectively, ensuring the same density.

Figure 4 and Figure 5 show the time spent for transmitting control data for the first set of simulations and the second set of simulations. We observe an important difference between the two approaches. The time spent for transmitting control data for the ranking approach vary from 1,074s to 3,046s for the first set of simulation scenarios and from 0,416s to 2,533s for the second set of simulation scenarios. The time spent for transmitting control data for the coloring approach ranges from 4,288ms to 10,075ms for the first set of simulation scenarios and from 2,846ms to 11,211ms for the second set of simulation scenarios. The difference can be justified due to the fact that the coloring approach is using 2-distance vertex coloring to allow simultaneous execution of the scheduling algorithm where as the coloring approach is using TDMA to avoid contention to access the control channel. We notice that the curve of the time spent for the first phase of the coloring approach have similar behavior to that of the number of colors needed for the network (Figure 8 and Figure 9) because the duration of the first phase depends on the number of colors.

Figure 6 and Figure 7 show the schedule length for the first set of simulation scenarios and for the second set of simulation scenarios. We observe that the ranking approach achieve a better schedule length than the coloring approach. In the scenario of 144 nodes (12 x 12 grid) for the first set of simulations, we notice a difference of 5,58 timeslots between the two approaches which represents 3,724ms. While in the scenario of 150 nodes for the second set of simulations, we observe a difference of 4,46 timeslots between the two approaches which represents 2,977ms. The ranking approach presents a better schedule length because it gave priority to nodes that have a low ratio between the number of channels and the number of links to schedule their links. However, this difference is relatively small compared to time spent to send control data for the ranking approach. We see a difference in the schedule length between the first and the second set of simulations since in the first set the nodes are placed deterministically in the network to cover the network area and to maximize the number of neighbors for each node. Therefore, this difference can be explained by the fact that in the grid topology the mean number of neighbors and outgoing links to be scheduled is higher than in the mesh topology.

Figure 8 and Figure 9 show the mean number of colors needed in the network for the coloring approach in the first set of simulation scenarios and in the second set of simulation scenarios. In the first set, we observe that the number of colors become constant for more than 100 nodes in the network. In the second set, we notice that the mean number of colors converge to 11,93 for more than 125 nodes in the network. The difference in the behavior of the curve of the number of colors between the first and the second set of simulations is because in the first set of simulations the nodes are placed deterministically in the network while in the second set of simulations the nodes are placed randomly in the network.

Figure 10 and Figure 11 show the mean goodput of a link for the first set of simulation scenarios and for the second set of simulation scenarios. As expected, the coloring approach achieves a better goodput than the ranking approach. In fact, the coloring approach accomplishes a goodput for a link of 375,048 to 672,694 Kbits/s for the grid topology and 394,102 to 987,515 Kbits/s for the mesh topology and a goodput for a node of 2,635 to 3,531 Mbits/s for the grid topology and 2,488 to 3,717 Mbits/s for the mesh topology. On the other hand, the ranking approach accomplishes a goodput for a link of 3,915 to 11,037 Kbits/s for the grid topology and 4,705 to 28,2 Kbits/s for the mesh topology and a goodput for a node of 27,519 to 57,945 Kbits/s for the grid topology and 29,713 to 106,146 Kbits/s for the mesh topology. The major difference in goodput between the two approaches can be explained by the fact that the ranking approach spends more time to exchange control information.

It is also noticed that the coloring approach uses the channels more efficiently than the ranking approach. In the coloring approach, the time spent for data transmission represents %68,51 to %75,96 for the grid topology and %63,17 to %76,57 for the mesh topology. On the other hand, in the ranking approach, the time spent for data transmission represents %0,59 to %1,13 for the grid topology and %0,63 to %2,05 for the mesh topology. Therefore, the coloring approach provides a better efficient use of the channels than the ranking approach.

The results at hand show that the coloring approach improves the scalability of the scheduling in the networks. Indeed, the 2-distance vertex coloring of the node allows achieving a scalable solution because the execution of the scheduling algorithm depends on the number of colors in the network. This permits the nodes which have the same color to schedule their outgoing links at the same time. Also, it utilizes the TDMA scheme to send control packets on the control channel which reduces the amount of data control packets sent and the time of execution of the scheduling algorithm. Nonetheless, in the coloring approach, the links need less delay to transmit a packet than in the ranking approach. As a result, the coloring approach presents an advantage for real-time application.

The ranking approach presents a better scheduling because the scheduling algorithm gives priority to nodes that present a low ratio between the number of channels and the number of links to be scheduled while the ranking approach needs a large amount of control packets exchanged leading to poor efficiency utilization of the channels.

VI. CONCLUSIONS

In this paper, we investigated the MAC-scheduling in a CWMN. We proposed an approach to improve the scalability, decrease the amount of control data sent, and diminish the time of execution of the scheduling while decreasing the schedule length. The approach utilizes 2-distance vertex coloring approach of the node to increase the simultaneous activity in the execution of scheduling algorithm in the network. It also uses the TDMA scheme to avoid contention to access the control channel. We compared our approach to another approach from the existing literature which is referred as the ranking approach. The results show a significant improvement in term of scalability, goodput, efficient utilization of channels, amount of control data sent, and the time of execution of scheduling. The ranking approach achieves a better schedule length but the gap is smaller compared to the advantages shown by the coloring approach. In future works, we plan to extend the proposed work to take the quality of service into consideration.

REFERENCES

[1] I. F. Akyildiz, X. Wang, and W. Wang, “Wireless mesh networks: a survey,” *Computer Networks*, Elsevier, Netherlands, vol. 47, pp. 445-487, Mar. 2005.

[2] P. Gupta and P. R. Kumar, “The capacity of wireless networks,” *IEEE Transactions on Information Theory*, IEEE, United States of America, vol. 46, pp. 388-404, Mar. 2000.

[3] J. Mitola Iii and G. Q. Maguire Jr, “Cognitive radio: making software radios more personal,” *IEEE Personal Communications*, IEEE, United States of America, vol. 6, pp. 13-18, Aug. 1999.

[4] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu. “A New Multi-Channel MAC protocol with on demand channel assignment for mobile ad-hoc networks,” In *Proc. International Symposium on Parallel Architectures, Algorithms and Networks*, Dallas/Richardson, Texas, United States of America, Dec. 2000, pp. 232-237.

[5] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, C.-Y. Lin, and J.-P. Sheu. “A Multi-Channel MAC protocol with Power Control for Multihop Mobile Ad-hoc Networks,” *The Computer Journal*, Oxford University Press, United Kingdom, Vol. 45, pp. 101-110, 2002.

[6] W.-C. Hung, K. L. Eddie Law, and A. Leon-Garcia, “A Dynamic Multi-Channel MAC for Ad-Hoc LAN,” In *Proc. Biennial Symposium on Communications*, Kingston, Canada, Jun. 2002, pp. 31-35.

[7] J. Chen, S. Sheu, and C. Yang, “A new multichannel access protocol for IEEE 802.11 ad hoc wireless LANs,” In *Proc. International Symposium on Personal, Indoor and Mobile Radio Communications*, Vol. 3, Beijing, China, Sep. 2003, pp. 2291-2296.

[8] A. Tzamaloukas and J.J. Garcia-Luna-Aceves, “Channel-Hopping Multiple Access,” In *Proc. IEEE International Conference on Communications*, New Orleans, Louisiana, United States of America Jun. 2000, pp. 415-419.

[9] A. Tzamaloukas and J.J. Garcia-Luna-Aceves, “Channel-hopping multiple access with packet trains for ad hoc networks,” In *Proc. IEEE Mobile Multimedia Communications*, Tokyo, Oct. 2000.

[10] 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. Annual International Conference on Mobile*

Computing and Networking, Philadelphia, Pennsylvania, United States of America, Sep. 2004, pp. 216-230.

[11] H. S. So, W. Walrand, and J. J. Mo, “McMAC: A Parallel Rendezvous Multi-Channel MAC Protocol,” In *Proc. IEEE Wireless Communications and Networking Conference*, Hongkong, China, Mar. 2007, pp. 334-339.

[12] A. Chia-Chun Hsu, D.S.L. Weit, and C.-C.J. Kuo, “A cognitive MAC protocol using statistical channel allocation for wireless ad-hoc networks,” In *Proc. IEEE Wireless Communications and Networking Conference*, Hongkong, China, Mar. 2007, pp. 105-110.

[13] J. Jia, Q. Zhang, and X. Shen, “HC-MAC: A hardware-constrained cognitive MAC for efficient spectrum management,” *IEEE Journal on Selected Areas in Communications*, Vol. 26, pp 106-117, Jan. 2008.

[14] L.-C. Wang, A. Chen, and D. S. L. Wei, “A cognitive MAC protocol for QoS provisioning in overlaying ad hoc networks,” In *Proc. Annual IEEE Consumer Communications and Networking Conference*, Las Vegas, Nevada, United States, Jan. 2007, pp. 1139-1143.

[15] M. Thoppian, S. Venkatesan, R. Prakash, and R. Chandrasekaran, “MAC-layer scheduling in cognitive radio based multi-hop wireless networks,” In *Proc. IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Buffalo-Niagara Falls, New York, United states, Jun. 2006, pp. 191-200.

[16] Z. Xizheng, “Efficient broadcast scheduling using a fuzzy-hopfield-network based mixed algorithm for ad hoc network,” In *Proc. International Conference on Advanced Language Processing and Web Information Technology*, Luoyang, Henan, China, Aug. 2007, pp. 405-408.

[17] S. Haixiang and W. Lipo, “Broadcast scheduling in wireless multihop networks using a neural-network based hybrid algorithm,” *Neural Networks*, Elsevier, United Kingdom, vol. 18, pp. 765-71, Aug. 2005.

[18] Network Simulator NS-2, http://www.isi.edu/nsnam/ns/, [15 Jun. 2010].

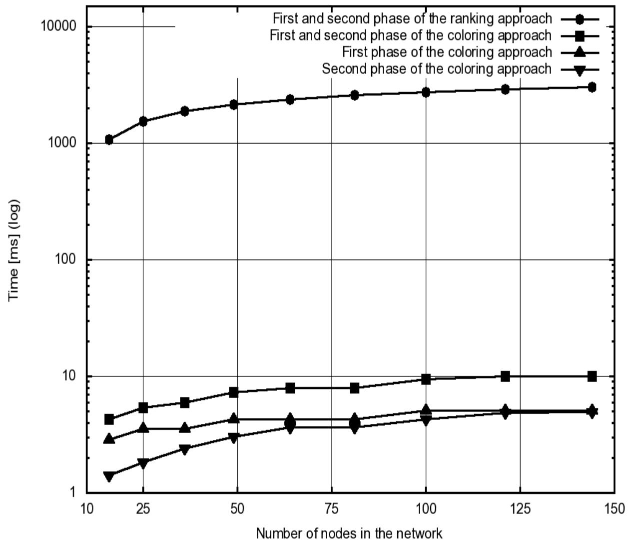

Figure 4. Time spent in a frame for scheduling links (first phase) and establishing the data sub-frame size (second phase) for the coloring and the ranking approach in the grid network

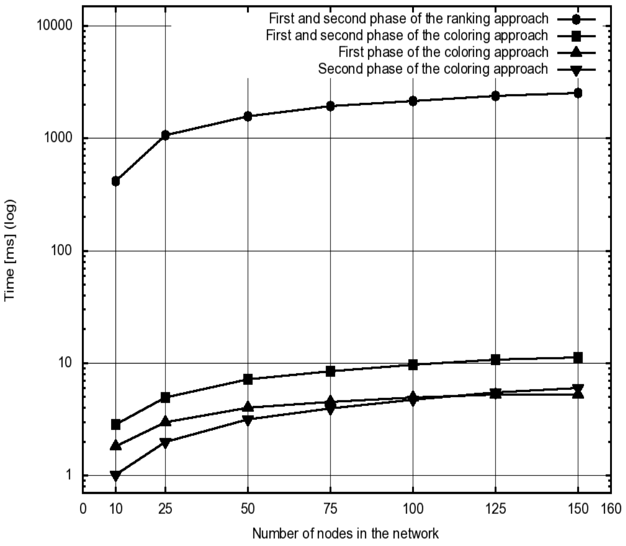

Figure 5. Time spent in a frame for scheduling links (first phase) and establishing the data sub-frame size (second phase) for the coloring and the ranking approach in the mesh network

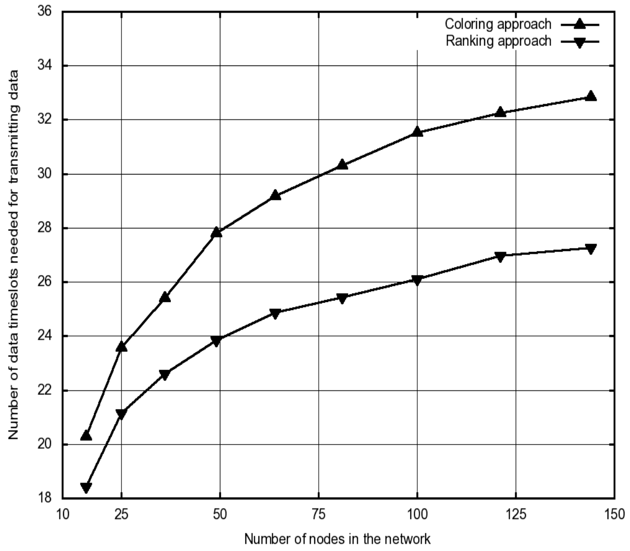

Figure 6. Schedule length of the coloring and the ranking approach in the grid network

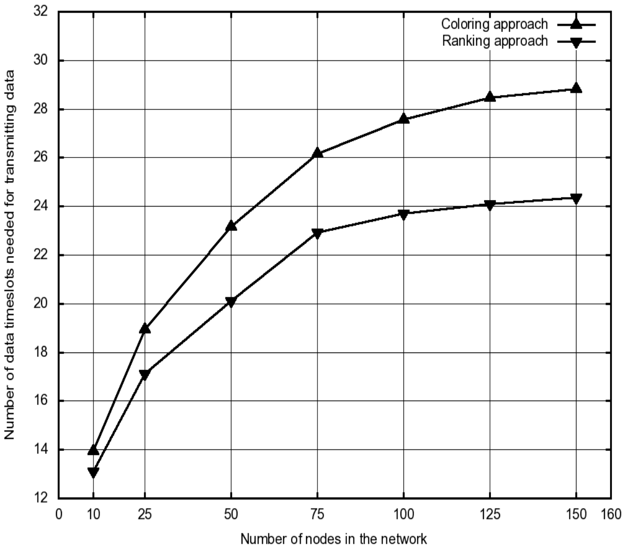

Figure 7. Schedule length of the coloring and the ranking approach in the mesh network

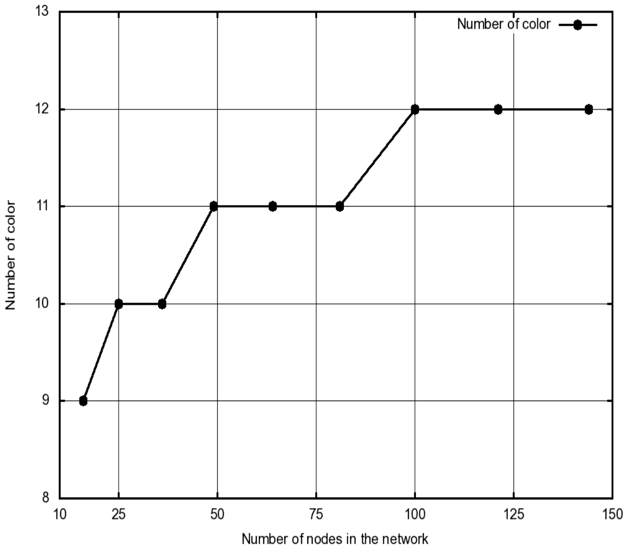

Figure 8. Number of colors needed in the coloring approach in the grid network

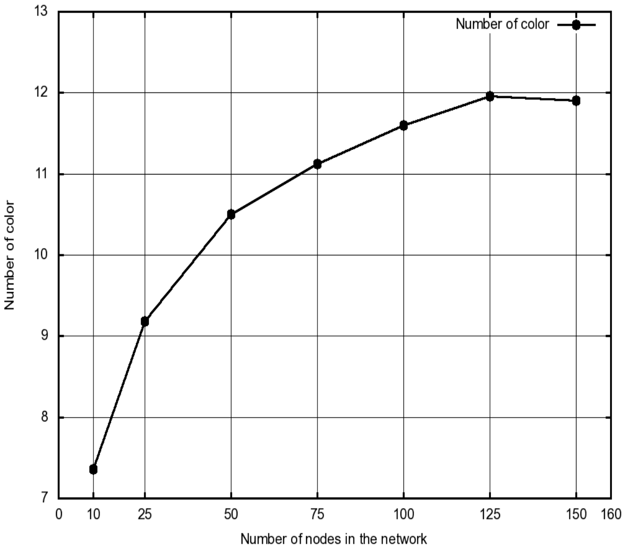

Figure 9. Number of colors needed in the coloring approach in the mesh network

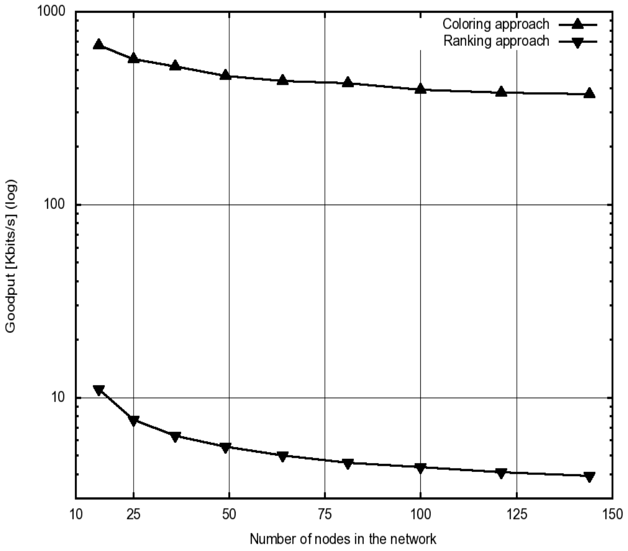

Figure 10. Mean goodput of a link for the coloring and the ranking approach in the grid network

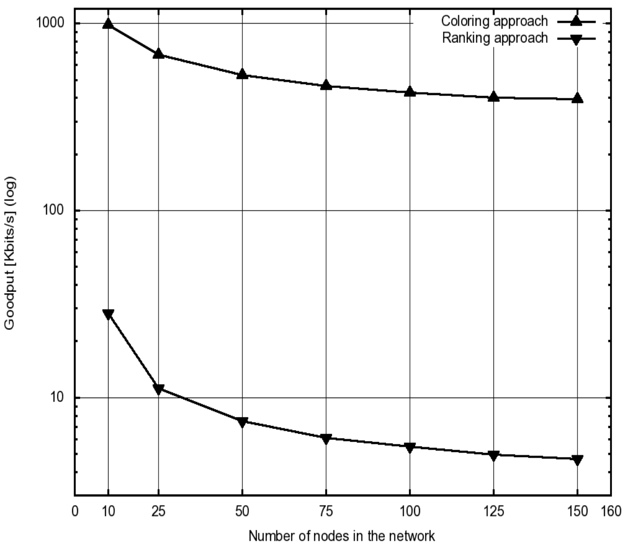

Figure 11. Mean goodput of a link for the coloring and the ranking approach in the mesh network