# Resource Allocation in Ad Hoc Networks with Two-Hop Interference Resolution

Stéphane Pomportes, Anthony Busson, Joanna Tomasik, Veronique Veque

# Resource Allocation in Ad Hoc Networks with Two-Hop Interference Resolution

Stéphane Pomportes*, Anthony Busson*, Joanna Tomasik† and Véronique Vèque*

*Université Paris-Sud XI

Laboratoire des signaux et systèmes (L2S) Supélec

91190 Gif-sur-Yvette, France

†SUPELEC Systems Sciences (E3S)

Computer Science Department

91192 Gif sur Yvette, France

Email: joanna.tomasik@supelec.fr

*Abstract*—The multi-user medium access mechanism OFDMA has to provide each node with a given amount of radio resources. In this paper we present a new distributed algorithm for the allocation of resource blocks in an OFDMA ad hoc network. We are principally interested in allocating resources fairly because the ad hoc networks which we work on are dedicated to be deployed in the areas of natural or man-made disasters and where the guarantee of connectivity is an important issue. Contrary to the commonly applied approach, we consider a resource allocation on the links under a two hop interference distance. The proposed allocation procedure is coupled with our other algorithm which detects and corrects two hop interferences and which has been revised and improved. The performance of our algorithm is evaluated by simulation for different topologies. We observed that simultaneous allocations in large networks allow a constant convergence time to be kept despite the networks size.

## I. INTRODUCTION

The radio medium access in ad hoc networks usually relies on a contention mechanism like CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) used for example in the standards IEEE 802.11 [1] or IEEE 802.15.4 [2]. With such a protocol, ensuring a minimum data throughput remains a very difficult task because this mecanism does not guarantee a data access rate or any service. On the other hand, the access mechanisms based on reservation like OFDMA (Orthogonal Frequency-Division Multiple Access) [3] ensures to each node a given amount of radio resources. The WiMAX uses this mechanism but only in the infrastructure mode where the allocation is centralized in Access Points. Nevertheless, the IEEE 802.16j Working Group [4] is interested in extending this technology to the ad hoc networks. In OFDMA, the smallest resource allocation unit is defined by a pair (time, frequency) called Resource Block (RB). The allocation of RBs is still an open problem. In this paper, we present a distributed algorithm for allocating RBs in an OFDMA ad hoc network.

There are two ways to allocate RBs in an OFDMA network. The first one, called the node-oriented [5] allocation, assigns the RBs to communication equipment. With this solution, RBs are assigned to each node and used to communicate with all its neighbors. The other approach, called the link-oriented [5] allocation, assigns resources to communication links. When a resource can be used for both directions of communication, a link is considered as bidirectional. Another approach consists in dedicating resources to a single direction, and links are considered as unidirectional. The choice of the approach depends essentially on the nature of communications. In the case of broadcasts, for example with a handheld transceiver, the node-oriented allocation is more appropriate. However, for point-to-point communications, such as data transfer or video transmission, the link-oriented allocation is preferable. Both approaches also have different properties. The article [6] has shown that allocating resources on the links has a better spatial reuse of resources than the node-oriented allocation and therefore a better bandwidth utilization.

Authors of [7], [8] and [9] provide a solution for the node-oriented resources allocation by considering an interference area equal to the communication area. Concerning the link oriented allocation, the authors of [10] offer a solution for bidirectional links. The unidirectional approach is discussed in papers [11], [12] and [13]. In all these works, an interference area of the same size as the transmission area is considered. In our opinion this assumption is not realistic and leads to numerous interfering allocations as the power of a signal may be too weak to be received, but still strong enough to interfere with another signal. We consider an interference area twice as large as the communication area.

To the best of our knowledge, the RBs allocation on unidirectional links, taking into consideration interference at two hops, has not been addressed beforehand. The contribution of this paper is as follows. It proposes an allocation algorithm assigning RBs to unidirectional links taking into consideration a two hop interference distance for ad hoc networks. An RB cannot be allocated if there is another transmitter for this RB less than two hops away from the receiver.

This assumption makes the allocation algorithm more complex because each node, performing an allocation, impacts its two hop neighborhood whereas it can only communicate with its direct neighbors. In Section 2, we present the model used and assumptions made in this paper. In Section 3, we present our algorithms. In Section 4, we study the performance of our algorithm by simulations on chain, grid and random

topologies.

## II. MODEL AND ASSUMPTIONS

### A. Network modeling

The network is modeled by a directed graph $G = (V, E)$ where $V$ is a set of nodes and $E$ is a set of arcs. This graph is called the *connectivity graph* in the rest of the paper. Each node is identified by a unique identifier (ID) and models a communication unit. Each arc, $(t, r)$, represents a unidirectional link between a transmitter, $t \in V$ and its associated receiver, $r \in V$. In this paper, we consider only connected graphs. We assume that there is a set $H$ of RBs, indexed from $0$ to $M$, assigned to each edge from $E$.

### B. Interference modeling

Assuming that a signal may be too weak to be received, but still strong enough to interfere with another signal, we consider that a transmitter can interfere with all its two-hop receivers. We say that two nodes are two hops away if they are not neighbors but have at least one neighbor in common. To model interference, we define an undirected graph $G_C = (V_C, E_C)$ which consists of a set of vertices $V_C$ and a set of edges $E_C$. Set $V_C$ has a one-to-one relation with the edges in set $E$ of graph $G$, i.e. for each edge $e \in E$, there is a corresponding vertex $v_c \in V_C$. We call this graph a *conflict graph*. An edge $(e_1, e_2)$ from $E_C$ exists only if the transmitter from an arc associated with one end of this edge and the receiver from the arc of the other end are nodes which are two hops away. Since each resource is supposed to be independent, we defined a set of conflict graphs $G_{Ci} = (V_{Ci}, E_{Ci})$, one graph for each channel $i$ of $H$. The set $V_{C_i}$ represents the subset of elements of $V_C$ which uses RB of index $i$. Set $E_{C_i}$ is obviously the subset of edges from $E_C$ between two elements of $V_{C_i}$. Formally, the solution proposed in the next section aims at realizing

$\forall e \in E, a(e) = 1$ with $\forall i \in H, E_{C_i} = \emptyset$

with $a(e)$ the function representing the number of resources to be allocated to the arc $e$.

## III. RESOURCES ALLOCATION AND CONFLICTS CORRECTION

In [14] we have proposed an algorithm for the correction of two hop allocation conflicts. In this section we describe firstly an algorithm we propose to allocate resources fairly. Next, we explain the algorithm to resolve interference conflicts, which is based on our previous work [14]. The modification we introduce here allows the algorithm to make a better decision on which allocation should be broken when conflict occurs.

### A. Allocation of Resource Blocks

We introduce a new algorithm to allocate a single RB on each arc of the connectivity graph. This algorithm, tries to maximize the spatial reuse of resources. Under this constraint, our algorithm can operate even with a reduced number of resources. Maximizing the spatial reuse rate means trying to use the same RB as many times as possible. Taking into account our assumption of two hop interferences, the optimal spatial reuse consists in allocating a given RB every three-hop. However, each new allocation has to take into account the already existing ones. This fact makes simultaneous allocations difficult as the optimal spatial reuse may be obtained to the price of a very long convergence time. The solution proposed in this paper offers a tradeoff between the maximization of spatial reuse and the convergence time.

This tradeoff is obtained by controlling the number of simultaneous allocations. Allocations to a link with a high level of interference are made first. We justify this choice by the fact that since a link shares the resources with its interfering links, the difficulty of assigning an available RB for a link increases with the number of interfering links. In our algorithm, this priority is introduced with a weight $W_n$ for each node $n$ which is defined by $W_n = outDegree(n) + \sum_{v \in \text{Neighbours(n)}} (outDegree(v))$, where $outDegree(n)$ is the number of outgoing links of $n$. taking into account the allocations which have already been made, each node computes a virtual weight, called $W'_n$ defined as $W'_n = W_n$ - *Number of outgoing links of v which has already received an allocation*, if all outgoing links of $n$ have not yet been allocated, and $0$ otherwise. Each node $n$ of the network which wants to realize an allocation compares its virtual weight with that of all its neighbors. If there is a neighbor with a higher virtual weight, the node does not have priority and it has to wait. In case of virtual weights equality, the differentiation is done on the weight and finally on the ID. When a node $n$ takes the allocation precedence, it allocates a resource on the outgoing link shared with its neighbor, which has a higher virtual weight. To select the resource to be allocated, the algorithm looks among all the free RBs for the one that maximizes spatial reuse. A resource which maximizes spatial reuse is identified by the states used in the algorithm of conflict correction. If such a resource does not exist, the algorithm takes an available free resource. When performing the choice between multiple resources, the algorithm always chooses the one with the smallest identifier. This property forces the algorithm to maximize use of the same resources across the network, thereby increasing the level of spatial reuse.

### B. Correction of two hops Allocation Conflicts

The algorithm for the conflict correction is distributed on each node $p$ of the network. This algorithm associates four variables for each RB indexed by $i \in H$. The variable $RBState_i(p)$ provides information on the location of the closest transmitters and receivers. This variable can be in one of the eight different states whose meaning is detailed in Table I. The second variable $PA_i(p)$, is an allocation pointer. When an RB is used to transmit (to receive, respectively), $PA_i(p)$ points to the associated receiver (the transmitter, respectively). The variables $P_{1,i}(p)$ and $P_{2,i}(p)$ represent two priorities. They are used to select more efficiently the allocation which should be retained when conflict occurs. The first priority $P_{1,i}(p)$ is based on weight $W_p$ defined in the previous paragraph. The second priority, $P_{2,i}(p)$, is used as a

TABLE I: Description of variable *RBState*

| State | Scope | Meaning |
|---|---|---|
| T | local | slot locally used for transmission |
| R | local | slot locally used for reception |
| $N_{R1}$ | one hop | at least one neighbor uses the slot for reception |
| $N_{T1}$ | one hop | at least one neighbor uses the slot for transmission |
| $N_{TR}$ | one hop | slot used by a transmitter and a receiver in neighborhood |
| $N_{T2}$ | one hop | slot used by a transmitter which is two hops away |
| $N_{R2}$ | two hops | slot used by a receiver which is two hops away |
| F | — | there is no transmitter and no receiver two hops away |

---

**Algorithm 1:** Allocation algorithm

---

CandidatesList ←Neighbors$_p$ \ {neighbors for which the outgoing link has already received an allocation.};
**if** *CandidatesList$\neq \emptyset$* **then**
    **if** $\nexists\, v\, \in\, Neighbors_p\ tq\ W'_v > W'_p$ **OR** $(\exists\, v\, \in\, Neighbors_p\ tq\ W'_v = W'_p$ **AND** $W_v < W_p)$ **OR**
    $(\exists\, v\, \in\, Neighbors_p\ tq\ W'_v = W'_p$ **AND** $W_v = W_p$ **AND** $ID_v < ID_p)$ **then**
        candidate ← node in the candidatesList with the highest virtual weight;
        **if** $\exists$ *a RB r such as r is usable and maximizes the spatial reuse* **then**
            Allocation of the RB r on the outgoing link from p to candidate;
        **else if** $\exists$ *a RB r such as r is usable* **then**
            Allocation of the RB r on the outgoing link from p to candidate;

---

Fig. 1: Example of chain topology

unique ID for each allocation which uses the same resource. $P_{2,i}(p)$ corresponds to the identifier of the transmitter. $P_{2,i}(p)$ is unique for each allocation using the same resource. Indeed, two arcs could share the same priority if their transmitters have the same identifier. However each node has a unique identifier. Therefore, it is impossible that two different arcs share the same priority $P_{2,c}$ for the same resource c. Precisely speaking, when conflict occurs, $P_{1,i}(p)$ preserves the allocation with the highest priority. When multiple allocations sharing the same value of $P_{1,i}(p)$ are in conflict, $P_{2,i}(p)$ is used to remove uncertainty by selecting the allocation which has the higher identifier.

The algorithm is composed of three steps. The first one checks that each RB already assigned is not in conflict with another of higher allocation priority. The second step updates the variables of unused RBs. When a node $p$ detects a transmitter pointing to it, the third step determines whether $p$ has to be associated with that transmitter or whether $p$ has to ignore this request.

Fig. 2: Example of grid topology

Fig. 3: Example of random topology

## IV. SIMULATIONS RESULTS

To validate our algorithm, we developed a simulator in Java. This simulator is available at [15]. We performed numerous simulations for chain, grid and random topologies. To generate random topologies, the simulator places a node inside the a simulation area of size $1000 \times 1000$, uniformly in both dimensions. When the distance between two nodes is within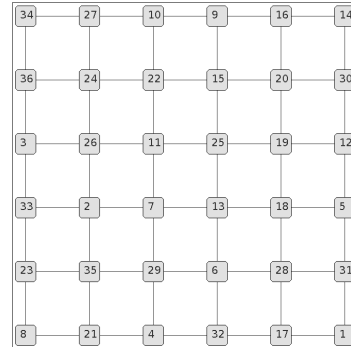 a t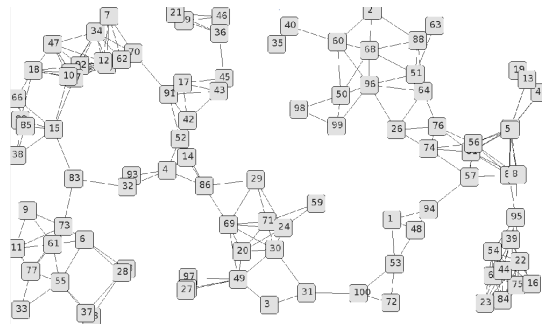hreshold called radio range, these nodes may communicate between each other. During the initialization procedure the simulator switches each node on at a time taken uniformly between 0 and 500 ms. Each node executes both algorithms of conflict correction and resources allocation. In the simulations, each node sends a message every 500 ms. Since the simulation

---

**Algorithm 2:** Conflict correction algorithm

---

**function**: $updateStates_{p,i}(newRBState, newP_{1,i}(p), newP_{2,i}(p))$ : update RBstate, $P_{1,i}(p)$ and $P_{2,i}(p)$ of i for p

**if** $RBState_i(p) = T$ **then**
  **if** $RBState_i(PA_i(p)) \notin \{R, F, N_{R2}\}$ **OR** $(RBState_i(PA_i(p)) \in \{R, N_{R2}\}$ **AND** $PA_i(PA_i(p)) \neq p)$ **then**
    $updateStates_{p,i}(F, -1, -1)$;
  **else if** $\exists v \in V(p) \setminus \{PA_i(p)\}$, $P_{1,i}(v) > P_{1,i}(p)$ **OR** $(P_{1,i}(v) = P_{1,i}(p)$ **AND** $P_{2,i}(v) > P_{2,i}(p))$ **then**
    **if** $RBState_i(v) \in \{T, R, N_{R1}, N_{TR}\}$ **then**
      $updateStates_{p,i}(F, -1, -1)$;

**if** $RBState_i(p) = R$ **then**
  **if** $RBState_i(PA_i(p)) \neq T$ **OR** $PA_i(PA_i(p)) \neq p$ **then**
    $updateStates_{p,i}(F, -1, -1)$;
  **else if** $\exists v \in V(p) \setminus \{PA_i(p)\}$, $P_{1,i}(v) > P_{1,i}(p)$ **OR** $(P_{1,i}(v) = P_{1,i}(p)$ **AND** $P_{2,i}(v) > P_{2,i}(p))$ **then**
    **if** $RBState_i(v) \in \{T, R, N_{T1}, N_{TR}\}$ **then**
      $updateStates_{p,i}(F, -1, -1)$;

**if** $RBState_i(p) \notin \{T, R\}$ **then**
  **if** $\exists\, v \in Voisins(p)$, $RBState_i(v) \in \{T, R\}\}$ **then**
    $SET_1 \quad\leftarrow\quad \{v \in Voisins(p) | RBState_i(v) \in \{T, R\}\}$;
    $V_{P_{MAX}} \quad\leftarrow\quad \{v \in Voisins_{P_{1,i}^{MAX}(SET_1)} | P_{2,i}(v) = P_{2,i}^{MAX}(Voisins_{P_{1,i}^{MAX}(SET_1)})\}$;
    **if** $\exists\, v_T \in V_{P_{MAX}}$, $RBState_i(v_T) = T$ **then**
      **if** $\nexists\, v_R \in V_{P_{MAX}}, RBState_i(v_R) = R$ **then**
        $updateStates_{p,i}(N_{T1}, P_{1,i}^{MAX}(SET_1), P_{2,i}^{MAX}(SET_1))$;
      **else**
        $updateStates_{p,i}(N_{TR}, P_{1,i}^{MAX}(SET_1), P_{2,i}^{MAX}(SET_1))$;
    **else**
      $updateStates_{p,i}(N_{R1}, P_{1,i}^{MAX}(SET_1), P_{2,i}^{MAX}(SET_1))$;
  **sinon si** $\exists\, v \in V_{P_{MAX}}$, $RBState_i(v_T) \notin \{N_{R1}, F\}$ **alors**
    $SET_2 \quad\leftarrow\quad \{v \in Voisins(p) | RBState_i(v) \in \{N_{T1}, N_{TR}\}\}$;
    $V_{P_{MAX}} \quad\leftarrow\quad \{v \in Voisins_{P_{1,i}^{MAX}(SET_2)} | P_{2,i}(v) = P_{2,i}^{MAX}(Voisins_{P_{1,i}^{MAX}(SET_2)})\}$;
    $updateStates_{p,i}(N_{T2}, P_{1,i}^{MAX}(SET_2), P_{2,i}^{MAX}(SET_2))$;
  **else**
    $SET_3 \quad\leftarrow\quad \{v \in Voisins(p) | RBState_i(v) \in \{N_{R1}\}\}$;
    $V_{P_{MAX}} \quad\leftarrow\quad \{v \in Voisins_{P_{1,i}^{MAX}(SET_3)} | P_{2,i}(v) = P_{2,i}^{MAX}(Voisins_{P_{1,i}^{MAX}(SET_3)})\}$;
    **if** $\exists\, v \in V_{P_{MAX}}$, $RBState_i(v) \in \{N_{R1}\}$ **then**
      $updateStates_{p,i}(N_{R2}, P_{1,i}^{MAX}(SET_3), P_{2,i}^{MAX}(SET_3))$;
    **else**
      $updateStates_{p,i}(F, -1, -1)$;

  $SET_4 \quad\leftarrow\quad \{v \in Voisins(p) | RBState_i(v) = T$ **AND** $PA_i(v) = p\}$;
  **if** $SET_4 \neq \emptyset$ **then**
    $v_T \quad\leftarrow\quad v \in Voisins_{P_{1,i}^{MAX}(SET_4)}, P_{2,i}(v) = P_{2,i}^{MAX}(Voisins_{P_{1,i}^{MAX}(SET_4)})$;
    **if** $RBState_i(p) \in \{F, N_{T2}\}$ **OR** $P_{1,i}(v_T) > P_{1,i}(p)$ **OR** $(P_{1,i}(v_T) = P_{1,i}(p)$ **AND** $P_{2,i}(v_T) > P_{2,i}(p))$ **then**
      $updateStates_{p,i}(R, P_{1,i}(v_T), P_{2,i}(v_T))$;
      $PA_i(p) \leftarrow v_T$;

---

time is configurable, we decided to represent the convergence time in number of rounds. One round is a time interval required for all nodes to send a message. In further figures, every point is the average of 100 simulations for chain and grid, and 1000 for random topologies. We calculated the 95% confidence interval for these averages.
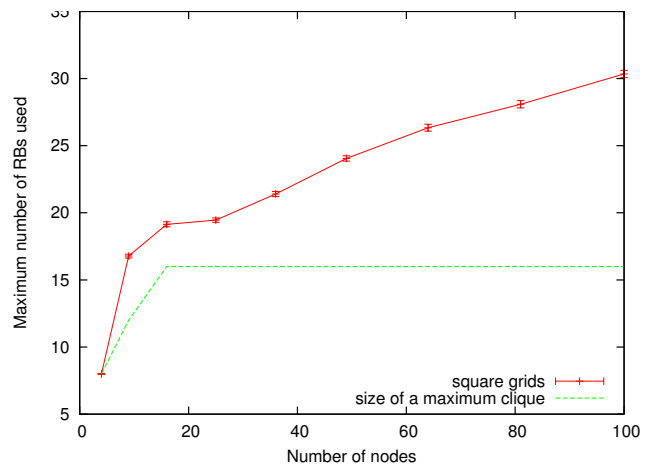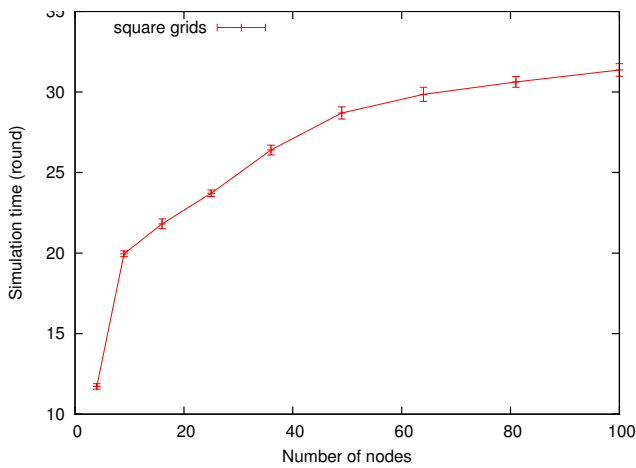
In our simulations, we are mainly interested in the evolution of the number of RBs required depending on the size of the network. Theoretically, this quantity depends on the size of
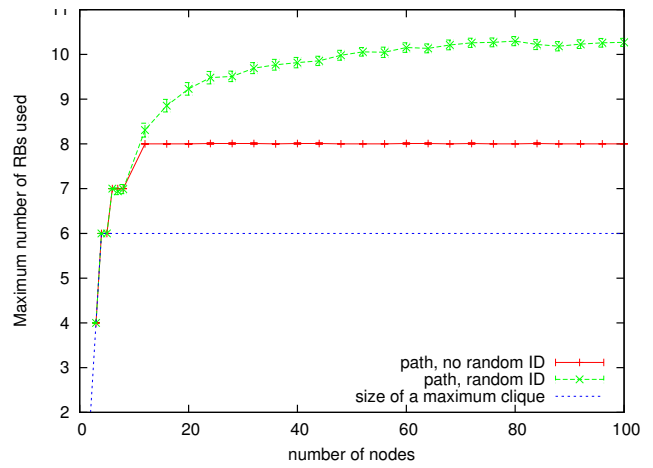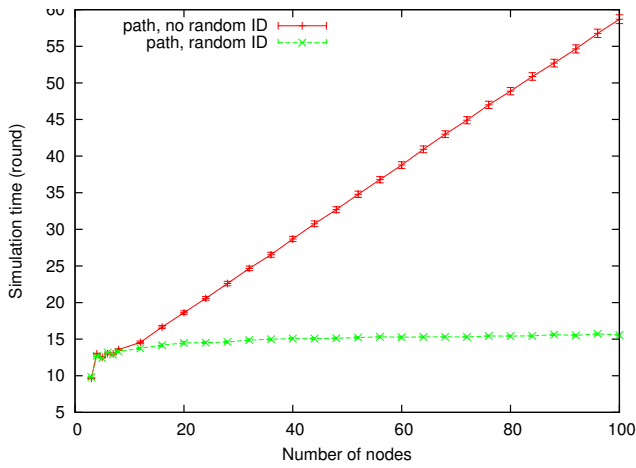
(a) Simulation time in round



(b) Maximum number of RBs used

Fig. 4: Chain simulation results
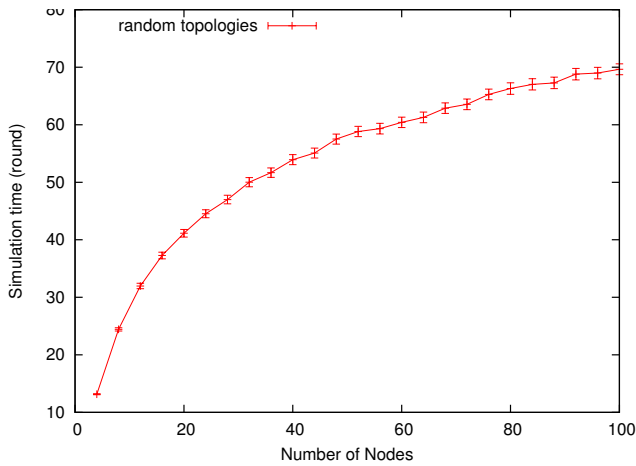


(a) Simulation time in round



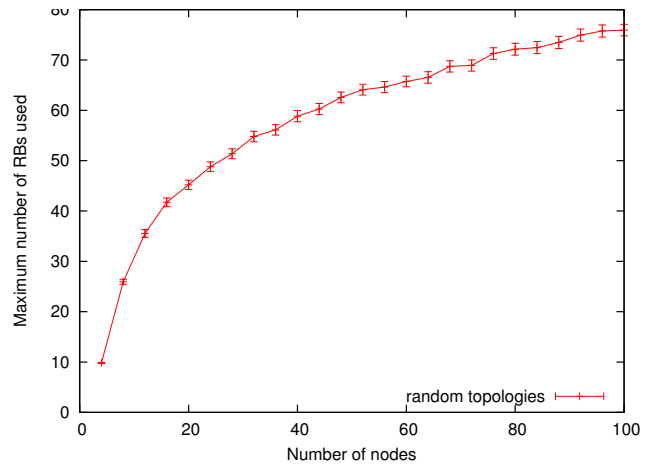(b) Maximum number of RBs used

Fig. 5: Grid simulation results

the maximal clique in the conflict graph. The maximal clique corresponds to the maximum number of mutually interfering links in the network. To check whether our solution is scalable, we should increase the size of the network without changing the size of its maximal clique. For chain and grid topologies, the size of the maximal clique reaches a constant value but this is not the case for random topologies. Moreover, it is impossible to extract the random topologies of same maximal clique size because the computation of this size is an NP-complete problem [16]. Nevertheless, the size of the network's maximal cliques partially depends on the nodes' average degree. Therefore, in the simulations, we vary the radio range to maintain a constant nodes degree.

Figures 4a, 5a and 6a show the evolution of the convergence time depending on the number of nodes for different topologies. For the chain, we choose to isolate the most unfavorable case where the ID of the nodes are ordered linearly (Fig. 4).

In this case, time increases linearly with the number of nodes. Indeed, we have seen that a node has allocation precedence if it has the highest virtual weight compared with the nodes in its neighborhood. We have also seen that when several candidates share the same virtual weight comparison is made on weight and, finally, on the IDs. In a chain, weight and virtual weight are identical for all nodes except the first two and the last two which have fewer neighbors two hops away than the other nodes. As the final comparison is made according to the ID, only the candidates with the highest IDs can perform it. If the IDs are randomly distributed, simultaneous allocations are possible dividing the convergence time by the number of simultaneous allocations in the network. In Figs. 5a and 6a we can observe a tendency of the convergence time depending on network size for the grid and random topologies. In fact, the convergence time increases with the number of links, but it decreases with the number of simultaneous allocations. Thus,

(a) Simulation time in round

(b) Maximum number of RBs used

Fig. 6: Results for random topologies with constant average node degree

when the size of a network increases, the increase of the number of links is balanced by the number of simultaneous allocations which also increases.

The simulations show that the resource demand for a random topology is greater than the demand for a grid, which is itself larger than the demand for a chain. The main reason is that the mean degree of nodes follows the same tendency as the RB requirement and indeed, the average degree is two for the chain, four for the clique and five for the random topologies. Concerning the scalable property of our solution, we can observe a convergence of the quantity of RB requirements when the network size increases. This is due to spatial reuse which allows the same resource to be used more than once in the network. As a number of simultaneous allocations is balanced by the increase in the number of links for simulation time, the increase in spatial reuse balances the increase of RB requirements.

## V. CONCLUSION

In this paper we presented our work concerning the improvement of the quality of service guaranteed in ad hoc networks. It was motivated by our previous studies of ad hoc networks deployed in disaster areas. We proposed an RB allocation and a conflict correction algorithm which are dedicated to networks with unidirectional links, assuming that the interference range is twice as large as the communication range. We developed a simulator to validate our solution and we evaluated its performance on chain, grid and random topologies. The simulation results show that the convergence time and number of RBs required converge asymptotically when the network size increases. It also shows that the value to which our solution converges depends considerably on the average degree of network nodes. The algorithms proposed here provide a solution to allocate one RB to each arc of the connectivity graph as we were principally interested in the fairness of the proposed reservation. Our future work will

focus on creating a new algorithm which allocates groups of RBs to links and which takes into account a traffic matrix.

## REFERENCES

[1] "IEEE Standard - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pp. C1 –1184, 12 2007.
[2] "IEEE Standard - Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, pp. 1–305, 2006.
[3] E. Lawrey, "Multiuser OFDM," in *in Proc. International Symposium on Signal Processing and its Applications*. N, 1999, pp. 761–764.
[4] "Ieee standard - part 16: Air interface for broadband wireless access systems amendment 1: Multiple relay specification," *IEEE Std 802.16j-2009 (Amendment to IEEE Std 802.16-2009)*, pp. c1 –290, 12 2009.
[5] P. Bjorklund and P. Varbrand, "Resource optimization of spatial TDMA in ad hoc radio networks: A column generation approach," in *INFOCOM 2003*, vol. 2, pp. 818–824.
[6] J. Gronkvist, "Assignment methods for spatial reuse TDMA," in *Proc. of the 1st ACM*, ser. MobiHoc '00, 2000, pp. 119–124.
[7] I. Rhee, A. Warrier, J. Min, and L. Xu, "Drand: Distributed randomized tdma scheduling for wireless ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 8, pp. 1384–1396, 2009.
[8] C. Zhu and M. S. Corson, "A Five-Phase Reservation Protocol (FPRP) for Mobile Ad Hoc Networks," in *Wireless Networks*, 1997, pp. 371–384.
[9] A. Kanzaki, T. Uemukai, T. Hara, and S. Nishio, "Dynamic TDMA Slot Assignment in Ad Hoc Networks," in *Proc. of AINA '03*, Washington, DC, USA, 2003, pp. 330–335.
[10] T. Salonidis and L. Tassiulas, "Distributed dynamic scheduling for end-to-end rate guarantees in wireless ad hoc networks," in *Proc. of ACM, MobiHoc '05*, 2005, pp. 145–156.
[11] P. Djukic and S. Valaee, "Link scheduling for minimum delay in spatial re-use TDMA," in *proc. of INFOCOM*, 2007, pp. 28–36.
[12] ——, "Delay aware link scheduling for multi-hop TDMA wireless networks," *IEEE/ACM Trans. Netw.*, vol. 17, pp. 870–883, June 2009.
[13] S. Gandham, M. Dawande, and R. Prakash, "Link scheduling in wireless sensor networks: Distributed edge-coloring revisited," *J. Parallel Distrib. Comput.*, vol. 68, pp. 1122–1134, August 2008.
[14] S. Pomportes, J. Tomasik, A. Busson, and V. Vèque, "Self-stabilizing algorithm of two-hop conflict resolution," in *Proc. of SSS'10*, 2010, pp. 288–302.
[15] http://www.ief.u-psud.fr/∼pomportes/netSim/networkSimulator.jar.
[16] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, first edition ed. W. H. Freeman & Co Ltd, Jan. 1979.