Evaluation of Broadcast Scheduling Algorithms for Ad-hoc TDMA Networks

Dimitrios J. Vergados #1, Maria-Yvonn Manolaraki #2, Dimitrios D. Vergados *3,**4

# National Technical University of Athens
Heroon Polytechniou 9, Zografou, Athens, GR-157 73, Greece
1 djvergad@telecom.ntua.gr
2 ge01054@mail.ntua.gr

* Department of Informatics, University of Piraeus
80, Karaoli & Dimitriou St., GR-185 34, Piraeus, Greece
3 vergados@unipi.gr

** Department of Information and Communication Systems Engineering, University of the Aegean
Karlovassi, Samos, GR-832 00, Greece
4 vergados@aegean.gr

Abstract—Ad-hoc networks rely on multihop transmission among the nodes on the same channel. Possible simultaneous transmissions may cause collisions, whenever transmitting nodes have a common destination node in their interference range. To avoid these collisions while minimizing the frame length, the NP-complete Broadcast Scheduling Problem (BSP) should be approximated. This is usually done by interpreting the BSP into a corresponding Graph Coloring Problem. This paper proposes an algorithm that tries to approximate the BSP, using an interference vector. Additionally, the node ordering policies used for scheduling are evaluated in terms of frame length and execution time. Simulation results show that the proposed algorithm has smaller execution time than the ones using graph coloring, and decreasing degree ordering results to the best frame length.

I. INTRODUCTION

Ad-hoc networks are used for many applications because of their simplicity, their ability to include new nodes at any time and their random topology. Their wireless nature and temporary topology require algorithms and protocols that can adjust to these special properties. In ad-hoc networks, there are no base stations to act as routers and routing is performed by the nodes themselves. Therefore, data should be delivered from source to destination through multiple hops. Transmissions must deal with the possibility of collisions, in order to successfully deliver the data. Collisions happen when the successful reception of a transmission is obstructed by the interference caused by other neighboring transmissions.

These collisions may be avoided through the application of Media Access Control (MAC) protocols, which are intended to coordinate the nodes’ transmissions accordingly. Contention based MAC protocols, like CSMA (Carrier Sense Multiple Access) use random backoff timers, and RTS/CTS packets for coordinating channel access separately for every packet, whereas reservation based protocols like Time Division Multiple Access (TDMA), Frequency Division Multiple Access (FDMA) and Code Division Multiple Access (CDMA) use scheduling algorithms for assigning transmission opportunities a-priory in a way that collisions are avoided.

In this paper, we focus on TDMA which is used in various technologies like the Digital-Advanced Mobile System (D-AMPS), the Global System for Mobile Communications (GSM), the Personal Digital Cellular (PDC), and the Bluetooth. TDMA can easily organize transmission of data, while avoiding simultaneous transmissions and allowing users to share the same frequency channel. Nodes have to transmit or receive in a predefined time slot and they can turn off their radio parts for the rest of the frame length.

The most common representation of ad-hoc networks is that of graphs with nodes connected with edges, where the graph’s vertices are the network’s nodes. There are three conditions that define a collision: a node may not transmit two packets simultaneously, a node may not receive and transmit a packet simultaneously (primary conflict), and a node should not receive two transmissions at the same time (secondary conflict). These conditions are ensured as long as the transmitting nodes are at least two hops apart in the above representation. The broadcast scheduling problem requires a schedule that eliminates the possibility of conflicts. At the same time the schedule should satisfy two constraints, i) each node must transmit at least once in every TDMA frame and ii) every slot must be filled with as many nodes as possible avoiding collisions.

Apart from broadcast scheduling, there is the other option of scheduling links, where instead of assigning a time slot to a node, pairs of connected nodes are assigned to time slots. Link scheduling avoids conflicts only on the intended recipient for every transmission, whereas conflicts on other nodes are allowed. This approach increases the number of possible concurrent transmissions, but also increases the number of slots required for every node.

There are several performance metrics to be taken into account, when evaluating scheduling algorithms. The most important include frame length, fairness, capacity, complexity,
robustness, time delay, throughput, overhead, packet loss rate and QoS capabilities.

In this paper, we present a new algorithm for approximating the broadcast scheduling problem, aims not only at minimizing the frame length (i.e. the number of slots required for every node to transmit), but also to reduce the required execution time. Different ordering criteria are evaluated in terms of frame length and execution time, and the results are compared to legacy scheduling algorithms.

The rest of the paper is organized as follows: section II discusses the related work, whereas section III describes the proposed scheduling algorithm. Section IV presents the performance analysis results and finally, section V concludes the paper.

II. RELATED WORK

The approximations of the Broadcast Scheduling Problem (BSP) are usually implemented using a graph representation of the network [1]. The vertices of the graph represent the nodes in the network, and the edges of the graph represent a neighboring relation among the nodes. Nodes are considered as neighbors or not depending on the distance among them. The transmission range refers to the maximum distance that a transmission may successfully received by another node, assuming no interference. On the contrary the interference range refers to the maximum distance where a node’s transmission affects other nodes even though it is not addressed to it [2]. The broadcast scheduling problem requires solutions where nodes aren’t scheduled in the same timeslot if they are one or two-hop neighbors in the above graph. In [3], the transformation of the broadcast scheduling problem to graph coloring problem is proposed, where nodes that can be assigned to the same slot are colored with the same color.

A centralized option of coloring which requires knowledge over the whole network topology is proposed in [4], where clustering is chosen as an effective solution partitioning the network to smaller ones. On the other side distributed coloring algorithms don’t require knowledge of the entire network topology. An example of a distributed algorithm is proposed in [5], where nodes have information only of their two hops away neighbors and exchange messages containing this information with neighbor nodes so that finally all nodes are acquainted with the network topology.

As numerous different kinds of graphs exist with each kind having special properties, the proper graph representation may be selected depending on the network model. Thus, other algorithms [6]-[10] take advantage of the properties that describe different kind of graphs such as trees, planar graphs, point graphs or random graphs. In [6] coloring is applied on planar point graphs, while the authors of [7]-[8] use factor graphs which contain two types of nodes, the variables nodes which are the network’s nodes and the agent nodes which represent the constraints that the broadcast schedule must satisfy. Finally, unit disk graphs [9]-[10] are a special kind of graphs whose properties resemble the ones of ad-hoc networks.

Two stage algorithms approximate the BSP in two steps. In the first step the lower bound of the frame length (M) is found usually with graph coloring. Then M is considered known and equal to the value found above and the capacity optimization stage takes place. Taking into consideration the problem’s constraints and metrics, the optimal solutions are the ones that achieve the maximization of the network’s throughput. After reaching a maximum throughput, the frame length increases so that all nodes are added. The final frame length is as close to the lower bound as it can get. Neural networks are used in [11] and [12], which prepare the schedule based on the network information they take as input. Their output gives the network schedule. Mean Field Annealing (MFA) [13]-[14], follows the same steps of optimization, but assuming a different lower bound for the frame length. Another two stage approach uses Genetic Algorithms [15]-[17] where the ‘strongest’ solution survives and it is chosen by the algorithm in the second step.

There are several algorithms that combine the above methods during the two steps. In [18], a hybrid of a neural network and a genetic algorithm is proposed. The authors in [19] use a neural network to minimize delays, but only after the lower frame length is found by sequential coloring. Algorithm FC-HNN [20] combines a different approach of graph coloring with a neural network called Hopfield Neural Network (HNN). The optimization part provides throughput maximization. Latin squares [21] offer a multichannel solution in case the nodes have multiple receivers to receive many packets at the same time. The rows and columns of the Latin squares give the timeslots of each channel. In [22], the authors describe two different options whether we are interested in a better performance in terms of frame length or network throughput. Hamming codes and TSAF [23] also use existing types. More recent studies propose the division of the network to geographic cells depending on the nodes’ location and then apply the coloring problem to these cells [24]-[25]. Such an algorithm is characterized as position-based one.

Based on previous proposals the authors in [26] suggested an algorithm which allows the entrance of new nodes in the network. New topology information is flooded in the entire network via negotiation messages. This is the beginning of a new category of algorithms for the broadcast scheduling problem, topology independent or topology transparent algorithms, which can handle topologies that change and networks with mobile nodes. USAP [27] is a distributed algorithm which allows the entrance of new nodes and permits spatial and temporal frame length adjustment, as long as the frame length is power of two. Distributed algorithms of [28]-[29] have the additional advantage of minimizing time delay by combining the scheduling problem with that of routing data in the network.

III. SCHEDULING ALGORITHMS

The BSP has been proven to be equivalent to the graph coloring [2]. Given that the network is represented by its connectivity matrix A, which is a NxN matrix, if N is the number of nodes in the network, where
\[ A_{i,j} = \begin{cases} 1 & \text{if nodes } i, j \text{ are connected, or } i = j \\ 0 & \text{if nodes } i, j \text{ are not neighbors} \end{cases} \]  

(1)

it must be transformed into the two-hop connectivity matrix

\[ A_{2i,j} = \begin{cases} 1 & \text{if nodes } i, j \text{ are one ore two hops away, or } i = j \\ 0 & \text{if nodes } i, j \text{ are more than two hops away} \end{cases} \]  

(2)

before executing a coloring algorithm. As we have already discussed, two nodes can be assigned to the same slot if they are more than two hops away in the graph representation. By applying a greedy graph coloring algorithm on the matrix \( A_{2} \), a broadcast schedule may be created, with each color being equivalent to a timeslot. The number of colors it produces is equal to the frame length. This algorithm is denoted as the “coloring broadcast scheduling” algorithm. However, since this is a slow process, a new broadcast scheduling algorithm is proposed, that does not require this transformation.

A more effective solution for approximating the broadcast scheduling problem is the interference vector algorithm [16]. The basic idea behind this algorithm is the introduction of the interference vector, which helps in rapidly identifying whether a node will interfere with the existing nodes in a slot. We represent the set of nodes transmitting during timeslot \( k \) as \( S_k \).

Also, we consider the collision avoidance vector \( \vec{F} \), which helps to determine if a node may collide with node \( j \) that is already scheduled to transmit in slot \( k \), where

\[ F_i = \sum_{j \in S_k} A_{i,j} \]  

(3)

If node \( k \not\in S_k \) and there is at least one node \( l \in S_k \), where \( k \) and \( l \) have \( z \) as a common neighbor, then \( A_{z,l} = A_{z,k} = 1 \).

Therefore, \( F_z \geq 1 \) and \( \vec{A}_k \cdot \vec{F} > 0 \). On the contrary, if

\[ \vec{A}_k \cdot \vec{F} = 0 \]  

(4)

there is no \( l \in S_k \) that has a common neighbor with node \( k \) and consequently node \( k \) can be added to the slot. This technique is an easy test to determine if a node collides with a node in the TDMA frame, and it reduces the complexity of the algorithm. The algorithm uses this technique for determining if a node may be scheduled to a slot containing other nodes. Each node is added to the first possible time slot and during the first step the first time slot is filled with all the possible nodes, then the second one and so on until at least one timeslot is assigned to each node. Thus, the interference vector algorithm differs from using coloring algorithms in that matrix \( A_{2} \) does not need to be formed, to check two-hop connectivity. As will be shown in the next section, this strategy exhibits improved performance in terms of execution time.

Regardless of the scheduling algorithm used, the order of examining the nodes affects the produced frame length. Nodes are ordered and examined in order to produce the first time slot. Afterwards, the same procedure is repeated for the second slot and so on until all nodes have been assigned to at least one time slot. There are many options regarding the selection order of the nodes. This is important because proper ordering may lead to better distribution of nodes to the time slots of the frame and smaller frame length. Our purpose is to add as many nodes as possible to every slot. This paper suggests that ordering nodes in a descending series of their degree, where degree means the number of neighbors, gives better results than other ordering criteria i.e. random selection, position ordering. In every step the node assigned to fewer time slots is chosen to be examined first. Among two nodes with the same number of slots, the one that has more neighbors is chosen, and this is the node with the highest degree. The reason is that nodes with many neighbors have more chances to collide when transmitting and therefore, less potential slots during scheduling so they should be examined first to have more options. Other possibilities include random selection of nodes, descending x-coordinate ordering selection (checking the most distant node first) and ordering nodes with the node with most two hop away neighbors on top, which is similar to descending degree ordering, but considers as neighbors both one and two hop nodes.

Nodes are ordered in a descending series of degree and the algorithm is named ‘\( \text{maxdeg} \)’. In the same way ‘random selection’ performs collision checks with \( \vec{F} \), but there is no specific ordering of nodes. As we expect random selection consumes less time but the frame length is bigger and in some cases the difference is quite annoying. The frame length cannot be bounded. ‘Max_x’ selection performs the act of scheduling by checking all nodes and adding them to the first available for them timeslot while using a geographic parameter; it examines more distant nodes first which are nodes with a bigger x-coordinate. In some occasions the results are satisfying, but depending on the topology we can expect some rather disappointing results. Finally, ‘maxdeg2’ uses \( A_{2} \) and orders nodes from the one with the most one and two hop away neighbors to the node with the least number of neighbors. \( A_{2} \) is formed using the connectivity matrix \( A \). This is a logical idea since collisions involve nodes with a distance of two hops.

IV. PERFORMANCE ANALYSIS

In order to produce the TDMA schedule, according to the proposed algorithm, all nodes need to be tested up to once for every time slot. In addition, for a network consisting of \( N \) nodes, \( N \) comparisons are needed to check for collisions. Thus, in the worst case, the complexity is \( O(N^2) \). However, in most cases the complexity is significantly smaller.

In our simulation scenario, we consider random topologies where the number of nodes varies from 200 to 1500 nodes. The nodes are uniformly distributed in a square topology of dimension either 200 or 800. The transmission range is assumed to be equal to the interference range, and its value is set to 50. For every configuration, 20 iterations were performed, each with a new random topology. Figures 1 and 2 depict the average frame lengths as a function of the numbers of nodes, for topologies with dimension 200 and 800 respectively.
Fig. 1. The average frame length for each node ordering (random, ‘max_x’, ‘maxdeg’, ‘maxdeg2’), for networks with various number of nodes. The dimension of the networks is 200 units.

Fig. 2. The average frame length for each node ordering (‘random’, ‘max_x’, ‘maxdeg’, ‘maxdeg2’), for networks with various number of nodes. The dimension of the networks is 800 units.

Fig. 3. The average execution time for the interference vector, and coloring broadcast scheduling algorithms (using ‘maxdeg’ ordering), for networks with various number of nodes. The dimension of the networks is 800 units.

Fig. 4. The average execution time for the node ordering algorithms (‘random’, ‘max_x’, ‘maxdeg’, ‘maxdeg2’), for networks with various number of nodes. The dimension of the networks is 800 and the interference vector scheduling is used.

Fig. 5. The average execution time for the node ordering algorithms (‘random’, ‘max_x’, ‘maxdeg’, ‘maxdeg2’), for networks with various number of nodes, using interference vector scheduling. The dimension of the networks is 800. Non-logarithmic vertical axis illustrates the differences between the three fastest algorithms.

Figures 3, 4 and 5 depict the average execution time, for a network with dimension 800. More specifically, Figure 3 shows the affect of the scheduling algorithm on the execution time, for the ‘maxdeg’ ordering. The maxdeg_F denotes the execution time when the interference vector scheduler is used. The coloring scheduler execution time consists of the A2 matrix construction time (maxdeg_A2) and the scheduling time (maxdeg_selftime). From Figure 3, we can see that the interference vector scheduler requires much less time than the coloring scheduler, mainly due to the time-consuming construction of the A2 matrix. It should be noted, that the scheduling time itself is also usually faster for the interference vector scheduler.

Figure 4 depicts the average execution time for the interference vector scheduler, when the 4 ordering criteria (‘random’, ‘max_x’, ‘maxdeg’ and ‘maxdeg2’) are applied, whereas Figure 5 shows the same results, omitting the time-consuming ‘maxdeg2’ scheduler for better resolution. The most time-consuming ordering criterion is the ‘maxdeg2’, due to the construction of matrix A2. The ‘max_x’ and the ‘maxdeg’ orderings require significantly less execution time, since the A2 matrix doesn’t need to be constructed. These algorithms require similar time for sorting the nodes; however, the ‘maxdeg’ exhibits shorter execution times probably due to
the decreased frame length. Finally, the fastest execution time is achieved from the random ordering, since no sorting is required. However, this comes at the cost of the frame length.

V. CONCLUSIONS

Broadcast scheduling is an important issue when setting ad-hoc networks. Many broadcast scheduling algorithms may be found in the literature. This paper presents a different way of dividing networks. Many broadcast scheduling algorithms may be seen asLegacy Coloring algorithms. In addition, the algorithm provides a descending series of a node’s degree ordering provides usually the least time consuming. In addition, the algorithm provides significant advantages in terms of execution time, over the legacy coloring algorithms.

ACKNOWLEDGMENT

This paper is part of the 03ED485 - “Design and Development Models for QoS Provisioning in Wireless Broadband Networks” research project, implemented within the framework of the “Reinforcement Programme of Human Research Manpower” (PENED) and co-financed by National and Community Funds (20% from the Greek Ministry of Development-General Secretariat of Research and Technology and 80% from the EU-European Social Fund).

REFERENCES