

Channel Access Competition in Linear Multihop Device-to-Device Networks

Vaggelis G. Douros, Stavros Toumpis, George C. Polyzos
Mobile Multimedia Laboratory, Department of Informatics
Athens University of Economics and Business
Patision 76, 104 34, Athens, Greece
E-mail:{douros, toumpis, polyzos}@aueb.gr

Abstract—We study a linear multihop network that is formed by wireless devices that can directly communicate pairwise whenever two devices are within range of each other. This Device-to-Device communication model is expected to play a significant role in future 5G wireless networks due to its advantages (e.g., cellular offloading, increased throughput and low cost/energy communication). In such networks, devices are typically selfish and compete for channel access aiming at maximizing their own throughput while at the same time avoiding packet collisions.

In this setup, we study how an efficient coexistence of these devices may be achieved, using a game-theoretic approach. First, we model the contention for the channel as a game and study the structural properties of the resulting Nash Equilibria (NE). Then, we design a distributed, round-based scheme that is guaranteed to converge to a NE. We compare quantitatively and qualitatively this scheme with previous work. We show that this scheme converges faster to a NE, in a number of rounds that is proportional to the logarithm of the number of nodes of the network. Moreover, the convergence is monotonic, meaning that the percentage of nodes that finalize their strategy is increasing in each round.

Keywords—Distributed algorithms, maximal scheduling, Nash Equilibrium.

I. INTRODUCTION

In future 5G wireless networks, devices that use the same channel and are close to each other are expected to be able to communicate directly, without needing to use a Base Station or Access Point. This concept, called Device-to-Device (D2D) communication [1], is receiving increasing attention since it can facilitate various applications: peer-to-peer file sharing, video dissemination, cellular offloading, etc. In such networks, devices will be forming multihop wireless networks. Typically, since these devices belong to different users, they will be selfishly competing for channel access, meaning that each one will be interested in sending its own data, without regard for the interference it is causing to other users.

Following up on our previous work [2], we model this scenario as a non-cooperative game, focusing on the Nash Equilibrium (NE) concept, *i.e.*, a state where no device has motivation to unilaterally change its strategy. The goal of this work is two-fold. Firstly, we are interested in studying theoretically the properties of Nash Equilibria and, secondly, exploit these properties to design a practical, efficient and distributed scheme that converges fast to a NE. The main extensions from [2] are the following: (i) We analyze the structural properties of

a strategy vector that is a NE. (ii) We propose a sophisticated scheme that is guaranteed to monotonically converge to a NE using these properties. (iii) We study this scheme under both the unicast and the multicast communication model (in [2], we studied only unicast). (iv) For each particular network, there are different NE where nodes either transmit their data or wait and receive data. We show that, under this scheme, all nodes have a significant probability to end up at a NE as transmitters.

On the other hand, in this work, we consider a linear multihop device-to-device network, whereas in [2] we have studied tree topologies, which are more general. This is due to the fact that the analysis of our scheme assuming a tree topology is quite complex and also because the one-dimensional model is more suitable for emphasizing the key features of our scheme. Furthermore, there have been proposed important real-life applications where nodes naturally form a linear multihop network: For example, monitoring some critical infrastructures and geographic areas by using wireless sensor networks [3] as well as vehicle-to-vehicle networks for road safety communications [4].

Our work is different from classic scheduling approaches that aim at finding maximal scheduling vectors [5]. A maximal scheduling vector corresponds to a feasible schedule such that no more devices can be activated without violating the interference constraints. As we have shown in [2], a NE strategy vector in the setup that we study is a maximal scheduling vector but a maximal scheduling vector is not necessarily a NE strategy vector. A maximal scheduling vector that is not a NE will never be chosen by autonomous devices, since at least one of them will be able to improve its payoff by simply switching to another strategy. Therefore, in our setting it makes sense to focus on NE instead of maximal scheduling vectors.

II. SYSTEM MODEL

We consider a scenario where N nodes form a linear network $\{1 - 2 - \dots - N\}$ and each node i can communicate with either its left-neighboring node $i - 1$ (L transmission) or its right-neighboring node $i + 1$ (R transmission). Time is slotted: transmissions start only at the beginning of a slot and last for the complete duration of the slot. Nodes always have packets to send to their neighbors. An R transmission is successful iff nodes $i + 1$ and $i + 2$ have chosen to wait, whereas a L transmission is successful iff nodes $i - 1$ and $i - 2$ have chosen to wait. As in [2], a successful transmission offers a payoff of $1 - c$, where $c \in (0, 1)$, a failed transmission offers a payoff of $-c$, and choosing to wait offers a zero payoff.

Nodes exchange messages either in a 2-hop or in a 3-hop neighborhood.

This setup can be easily modeled as a non-cooperative game with the players being the nodes and the strategy s_i of a player i being one of the following: $\{R, L, W\}$. In fact, it is a special type of non-cooperative game, called graphical game [6], since the payoff of a node is determined only by a subset of the players (those that are two hops away) instead of all the players. For this setup, we have shown in [2] that: (i) a NE exists, (ii) each NE is Pareto optimal, (iii) the best response dynamics scheme [6] does not always converge to a NE and (iv) there is a naive scheme that can lead to a NE. In this work, we will analyze the structural properties of a strategy vector that is a NE and, based on these properties, we propose a more sophisticated scheme that leads faster to a NE.

III. NE PROPERTIES

In this section, we state two theorems that specify useful properties of strategy (sub)vectors at a NE. From now on, we use the pronoun “he” when we refer to a node, in line with the standard convention of game theory for the players.

Theorem 1: Let $s = (s_1, s_2, \dots, s_N)$ be a strategy vector that corresponds to a NE with $s_i = R$. Then:

- 1) If $i = N - 1$, then the subvector (s_{N-1}, s_N) is equal to (R, W) .
- 2) If $i = N - 3$, then the subvector $(s_{N-3}, s_{N-2}, s_{N-1}, s_N)$ is equal to (R, W, W, L) .
- 3) If $i \leq N - 4$, then the subvector $(s_i, s_{i+1}, s_{i+2}, s_{i+3})$ is equal to (R, W, W, L) or (R, W, W, R) , where, in the second case, the R transmission of node $i + 3$ satisfies this theorem as well.

Proof: For each case, it is enough to show that the following two conditions hold:

- Condition A: Nodes whose strategy appears in one of the above subvectors do not have motivation to change unilaterally their strategies.
 - Condition B: There is no other strategy subvector that fulfills condition A with $s_i = R$.
- 1) As for condition A, it is straightforward to check it. As for condition B, the only other possible subvector is (R, L) which does not fulfill condition A since it leads to a collision.
 - 2) Concerning condition A, indeed, no node can improve his payoff by changing his strategy on his own. Concerning condition B, if node $N - 2$ or node $N - 1$ chooses to transmit, condition A cannot be satisfied since there will be a collision with the R transmission of node $N - 3$ and he would be motivated to refrain from transmitting. If node N chooses W , condition A is not satisfied as node $N - 1$ has motivation to choose R .
 - 3) As for condition A, we have already discussed the case (R, W, W, L) . As for the case (R, W, W, R) , this subvector fulfills condition A only if the subvector that starts with node $i + 3$ fulfills condition A as well. As for condition B, we have argued on why no other subvectors may arise in the previous paragraph.

As a final comment, it is worth mentioning that if $i = N - 2$, there is no NE where node i transmits R . This is due to the fact that the subvector $(s_{N-2}, s_{N-1}, s_N) = (R, W, W)$, that is the only subvector that corresponds to a successful R transmission, is collision-free but cannot be part of a NE, since node $N - 1$ is motivated to choose R . Therefore, node i cannot choose R at a NE. ■

Theorem 2: Let $s = (s_1, s_2, \dots, s_N)$ be a strategy vector that corresponds to a NE with $s_i = L$. Then:

- 1) If $i = 2$, then the subvector (s_1, s_2) is equal to (W, L) .
- 2) If $i = 4$, then the subvector (s_1, s_2, s_3, s_4) is equal to (R, W, W, L) .
- 3) If $i \geq 5$, then the subvector $(s_{i-3}, s_{i-2}, s_{i-1}, s_i)$ is equal to (R, W, W, L) or (L, W, W, L) , where, in the second case, the L transmission of node $i - 3$ satisfies this theorem as well.

Proof: The proof is similar to the proof of Theorem 1, so we omit it. ■

IV. FINDING A NE

In [2], we discuss a naive algorithm under which nodes exchange messages with nodes that are up to two hops away. The idea is the following: Initially, each node chooses randomly his strategy. If he has a successful transmission, he transmits at the following round too. This imposes some limitations on the strategies of some of his one-hop and two-hop neighbors. More specifically, if a node i has a successful R transmission, then nodes $i - 1$ and $i - 2$ (if they exist) will not be able to have a successful R transmission. So, when they update their strategy, they should exclude this option. Similarly, if a node i has a successful L transmission, then nodes $i + 1$ and $i + 2$ (if they exist) should exclude L from further updates of their strategies.

Each node that does not have a successful transmission examines these limitations to decide whether he can have a successful transmission (e.g., he may have a successful L transmission but not a successful R transmission) or he should wait for the next round. If so, he flips a coin and if the coin allows it, he transmits, otherwise he stays quiet. The algorithm ends when each node has either a successful transmission or waits and each node that waits cannot have a successful transmission.

In this naive scheme, a node is interested in learning only whether he has a successful transmission or not, using exchanges of messages with up to his two-hop neighbors. This information is not sufficient to guarantee that a node that has a successful transmission will not need to change his strategy at a following iteration of the algorithm. For example, consider 3 nodes with the strategy vector $(s_1, s_2, s_3) = (R, W, W)$. Node 1 will choose R in the next iteration, even if no NE with an R transmission for node 1 can arise. This has two undesirable effects for the nodes: It is a waste of resources and delays the convergence to a NE.

We propose a more sophisticated scheme where nodes have motivation to exchange messages with up to their three-hop neighbors to alleviate the shortcomings of our algorithm in [2]. The algorithm is presented in pseudocode. The core of

the algorithm is based on Propositions 1 and 2 that are closely related to Theorems 1 and 2.

Proposition 1: Let $s = (s_1, s_2, \dots, s_N)$ be a strategy vector with $s_i = R$. Then:

- 1) If $i = N - 1$ and $(s_{N-1}, s_N) = (R, W)$, the algorithm will end up at a NE where this equality holds.
- 2) If $i = N - 3$ and $(s_{N-3}, s_{N-2}, s_{N-1}, s_N)$ is equal to (R, W, W, L) , the algorithm will end up at a NE where this equality holds.
- 3) If $i \leq N - 4$ and $(s_i, s_{i+1}, s_{i+2}, s_{i+3}) = (R, W, W, L)$, the algorithm will end up at a NE where this equality holds.
- 4) If $i \leq N - 4$ and $(s_i, s_{i+1}, s_{i+2}, s_{i+3}) = (R, W, W, R)$, the algorithm will end up at a NE where the strategy vector includes either this subvector or the subvector (R, W, W, L) .

Proposition 2: Let $s = (s_1, s_2, \dots, s_N)$ be a strategy vector with $s_i = L$. Then:

- 1) If $i = 2$ and $(s_1, s_2) = (W, L)$, the algorithm will end up at a NE where this equality holds.
- 2) If $i = 4$ and $(s_1, s_2, s_3, s_4) = (R, W, W, L)$, the algorithm will end up at a NE where this equality holds.
- 3) If $i \geq 5$ and $(s_{i-3}, s_{i-2}, s_{i-1}, s_i) = (R, W, W, L)$, the algorithm will end up at a NE where this equality holds.
- 4) If $i \geq 5$ and $(s_{i-3}, s_{i-2}, s_{i-1}, s_i) = (L, W, W, L)$, the algorithm will end up at a NE where the strategy vector includes either this subvector or the subvector (R, W, W, L) .

Due to space constraints, we omit the proof of Propositions 1 and 2. The algorithm aims at identifying strategy subvectors that are guaranteed to be part of the eventual NE. Nodes that belong to these subvectors do not change any more their strategies, having completed their statuses. The rest of them go on updating their strategies by taking into account the limitations due to successful transmissions that we discussed in the naive scheme. Clearly, if a node has a unique strategy left as an option, then he completes his status as well. So, when a node has a successful transmission, he transmits at the same direction at the next transmission round only if his strategy is part of a strategy subvector mentioned in either Theorems 1 or 2. Otherwise, he flips a coin to decide upon his strategy. When all nodes complete their statuses, a NE has arisen and the algorithm ends.

After the initial random choice of the strategies (lines 2-5), there are two big for-loops that are executed in each transmission round. In the first for-loop (lines 7-31), the algorithm examines cases 1-3 of Propositions 1 and 2. In the second for-loop (lines 32-43), it examines case 4 of Propositions 1 and 2. In the last two lines, it examines whether all nodes have completed their statuses or another round starts.

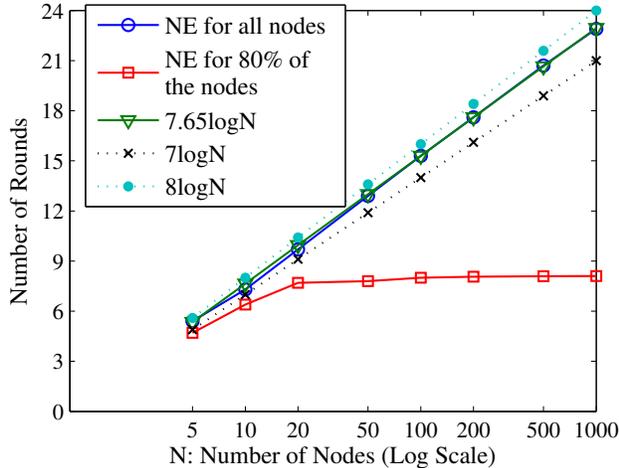
As a final comment, the algorithm uses the best response concept: (i) when it identifies strategy subvectors that are guaranteed to be part of a NE (*i.e.*, cases where nodes complete their statuses), (ii) in lines 36 and 41 of the pseudocode. In both cases, no oscillations may arise and the adoption of the best response scheme will definitely lead to strategies that will be part of a NE.

Algorithm 1 Finding a NE through a distributed iterative scheme: The Unicast Case

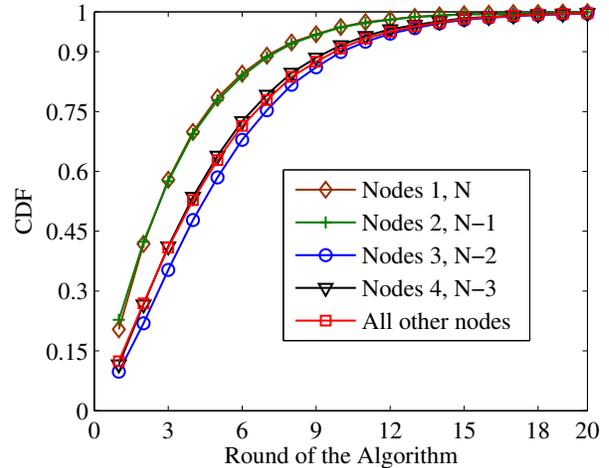
```

1: Notation:  $C$ : Completed status,  $P$ : Pending status,  $N$ :
   Number of Nodes,  $S$ : Successful Transmission
2: for  $i = 1 \rightarrow N$  do
3:    $i.status = P$ 
4:   each node  $i$  chooses randomly his strategy.
5: end for
6: for  $k = 1 \rightarrow \text{MAX\_NUMBER\_OF\_ITERATIONS}$  do
7:   for  $i = 1 \rightarrow N$  do
8:     If node  $i$  has chosen  $R$  or  $L$ , node  $i \pm 1$  informs him
       whether the transmission was successful or not. Each
       node  $i$  computes his payoff.
9:     if  $i.transmission == S$  and  $i.strategy == R$  then
10:      if  $i \leq N - 3$  and  $(i + 3).strategy != W$  then
11:         $i.status = C, (i + 1).status = C, (i + 2).status = C$ 
12:        if  $(i + 3).strategy == L$  then
13:           $(i + 3).status = C$ 
14:        end if
15:      end if
16:      if  $i == N - 1$  then
17:         $i.status = C, (i + 1).status = C$ 
18:      end if
19:    end if
20:    if  $i.transmission == S$  and  $i.strategy == L$  then
21:      if  $i \geq 4$  and  $(i - 3).strategy != W$  then
22:         $i.status = C, (i - 1).status = C, (i - 2).status = C$ 
23:        if  $(i - 3).strategy == R$  then
24:           $(i - 3).status = C$ 
25:        end if
26:      end if
27:      if  $i == 2$  then
28:         $i.status = C, (i - 1).status = C$ 
29:      end if
30:    end if
31:  end for
32:  for  $i = 1 \rightarrow N$  do
33:    if  $i \geq 4$  and  $i.status == P$  and  $i.transmission == S$ 
      and  $i.strategy == R$  and  $(i - 3).status == C$  then
34:       $i.status = C$ 
35:    else
36:       $i.next\_strategy = L$ 
37:    end if
38:    if  $i \leq N - 3$  and  $i.status == P$  and
       $i.transmission == S$  and  $i.strategy == L$  and
       $(i + 3).status == C$  then
39:       $i.status = C$ 
40:    else
41:       $i.next\_strategy = R$ 
42:    end if
43:  end for
44:  Nodes that have completed their statuses send a local
    broadcast message to their neighbors along with their
    strategy.
45:  if all nodes have completed their statuses, the algorithm
    ends at a NE. Else, the nodes that have incomplete
    status, update randomly their strategy by taking into
    account any limitations that are imposed by the strategy
    of the nodes that have completed their statuses (as
    discussed in the text).
46: end for

```



(a) Average round for convergence to a NE.



(b) CDF of round at which a node completes his status.

Fig. 1: Performance evaluation for the unicast case for the sophisticated algorithm: Expected value and the Cumulative Distribution Function (CDF) of the convergence time.

V. PERFORMANCE EVALUATION FOR THE UNICAST CASE

We have simulated our schemes to evaluate their performances under topologies of various sizes (from 5 nodes up to 1000 nodes). For each topology size, we have executed 10,000 simulations. In [2], we studied the efficiency of a NE in terms of the min./max./average number of successful transmissions at a NE. In this work, we focus on the convergence time of the proposed schemes to lead to a NE. The first set of simulations is used to evaluate the average number of iterations so that the algorithm converges to a NE versus the size of the topology. We compare the sophisticated scheme with two variations of the naive scheme of [2]: (i) A scheme that uses an unbiased coin when a node needs to decide whether to transmit or not and (ii) a scheme that uses a biased coin giving higher probability to transmit. We experimented with different values of the probability to transmit and we present the results for $2/3$, which is a representative value for the trends that we notice. The motivation for this biased version is that, in principle, a node would prefer to transmit than to wait.

Fig. 2 presents the results. As expected, the number of rounds increases with the size of the topology. The sophisticated scheme presents the best performance demanding, on the average, from 5 rounds (for 5 nodes) to 23 rounds (for 1000 nodes). However, this increase is quite slow, e.g., augmenting the nodes from 200 to 500 demands only 3 more rounds on the average to find a NE. This means that even for topologies that consist of many nodes, the algorithm converges fast. Actually, the increase is proportional to the logarithm of the number of nodes N of the topology. Experimentally, we find that the average number of steps for the convergence to a NE is $\cong 7.65 \log_{10}(N)$ (see Fig. 1a). We also note that the algorithm converged to a NE without exceeding the maximum number of iterations (which was set to 50) with probability > 0.999 for all studied topologies.

The unbiased version performs quite well, demanding a small number of extra rounds with respect to the sophisticated

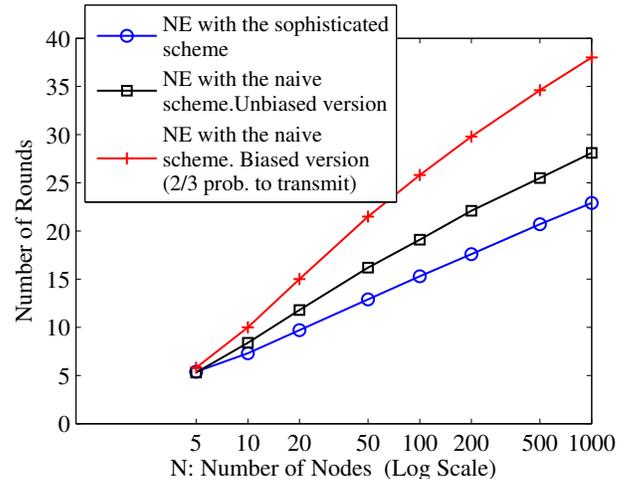


Fig. 2: Comparison of the proposed algorithms. Average round for convergence to a NE under unicast.

scheme to find a NE. The number of rounds is proportional to the logarithm of the nodes of the topology, however the constant multiplier is bigger than the algorithm that we already proposed. On the other hand, the performance of the biased version is worse and deteriorates as the number of nodes increases. Moreover, about 5% of the simulations of big topologies exceed the maximum number of iterations without converging to a NE. These undesirable features of the biased version are due to the fact that favoring nodes' probability to choose to transmit (even though, in principle, a node would prefer to transmit than to wait) increases also the probability for collisions and delays the convergence to a NE. This is the reason why the unbiased version performs better.

Next, we examine in which round, on average, 80% of the nodes have completed their statuses. We focus only on the sophisticated algorithm, since the convergence to a NE for a node that uses the naive algorithm is not monotonic, meaning that he may change his status from complete to incomplete and vice versa. As Fig. 1a shows, for all studied topologies that

consist of at least 20 nodes, 80% of nodes will have converged to their final strategies in just 8 rounds. This means that, on average, 800 out of 1000 nodes will have converged to their final strategies in up to round #8 and only 200 of them will go on updating their strategies in up to round #23.

To further explore that issue for the sophisticated algorithm, we distinguish the nodes in five categories: (i) Node 1/node N that have no left/right neighbor. (ii) Node 2/node $N - 1$ that have one left/right neighbor. (iii) Node 3/node $N - 2$ that have two left/right neighbors. (iv) Node 4/node $N - 3$ that have three left/right neighbors. (v) Every other node that has at least four left/right neighbors. We use this grouping based on Theorems 1 and 2, as the nodes that belong to the same category are expected to have similar probabilities to participate in a strategy subvector that is guaranteed to be part of a NE. This is due to the fact that this probability depends on the number of left/right neighbors, so nodes that have the same number of left/right neighbors (0, 1, 2, 3, 4+ neighbors respectively) should be studied together.

Fig. 1b presents the results after 10,000 experiments on a topology that consists of 10 nodes. The results are very similar for any bigger topology too. The horizontal axis corresponds to the round of the algorithm and the vertical axis to the probability that a node of each category will have completed his status up to that particular round. The fast convergence for the vast majority of the nodes is verified by these results. As we can see, at round #8, each node has a probability of more than 0.8 to have converged to his final strategy. Moreover, it is interesting to note that any node can complete his status up to round #12 with probability > 0.95 and also any node can complete his status up to round #17 with probability > 0.99 . Further analysis of this diagram leads to the conclusion that all nodes have significant probability to end up at a NE as transmitters, which is, in principle, desirable for them.

Finally, as the convergence to a NE for a node that uses the naive algorithm is not monotonic, the percentage of nodes that have completed their status in round $k + 1$ can be smaller than in round k . The monotonic convergence to a NE is a great advantage of the sophisticated scheme.

VI. ON THE NE UNDER MULTICAST TRAFFIC

In this section, we study the multicast transmission scheme, where each node aims at sending his packet to all neighbors that are one hop away from him. Clearly, each node can choose between two strategies: to transmit (T) or to wait (W). Concerning the payoff, for each intermediate player i , there are some differences from the unicast case due to the fact that the transmission cost is equally divided to the number of nodes to whom the packet is sent. Therefore, if a node waits, his payoff is again 0; If he transmits and the transmission is successful for both neighbors (fully-successful transmission), then his payoff is $2(1 - c/2) = 2 - c$; If one transmission is successful and the other fails (semi-successful transmission), then his payoff is $1 - c/2 - c/2 = 1 - c$; If both transmissions fail, then his payoff is $-c/2 - c/2 = -c$.

Analyzing the conditions for a successful transmission and using similar arguments with the unicast case, we find that a strategy subvector of the form $(s_{i-2}, s_{i-1}, s_i, s_{i+1}, s_{i+2}) = (W, W, T, W, W)$ should exist so that node i has a fully-successful transmission. If either $(s_i, s_{i+1}, s_{i+2}) = (T, W, W)$

Algorithm 2 Finding a NE through a distributed iterative scheme: The Multicast Case

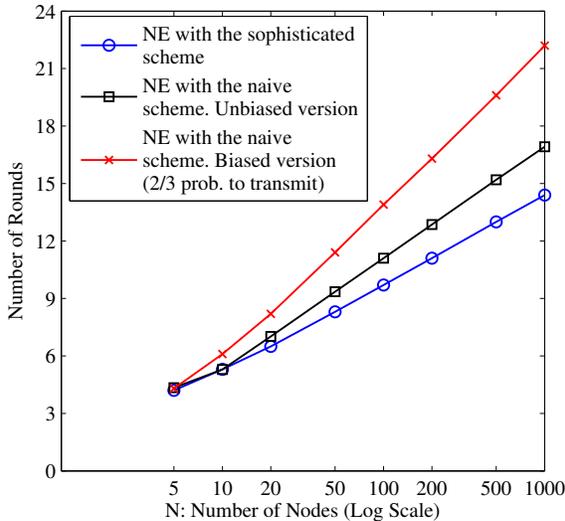
```

1: Notation:  $C$ : Completed status,  $P$ : Pending status,  $N$ :
   Number of Nodes,  $S$ : Successful Transmission
2: for  $i = 1 \rightarrow N$  do
3:    $i$ .status= $P$ 
4:   each node  $i$  chooses randomly his strategy.
5: end for
6: for  $k = 1 \rightarrow \text{MAX\_NUMBER\_OF\_ITERATIONS}$  do
7:   for  $i = 1 \rightarrow N$  do
8:     If node  $i$  has chosen  $T$ , nodes  $i \pm 1$  inform him
       whether the transmission was successful or not. Each
       node  $i$  computes his payoff.
9:     if  $i$ .transmission== $S$  then
10:      if  $i \leq N - 3$  and  $(i + 3)$ .strategy!= $W$  then
11:         $i$ .status= $C$ ,  $(i + 1)$ .status= $C$ ,  $(i + 2)$ .status= $C$ ,
           $(i + 3)$ .status= $C$ 
12:      end if
13:      if  $i == N - 1$  then
14:         $i$ .status= $C$ ,  $(i + 1)$ .status= $C$ 
15:      end if
16:      if  $i \geq 4$  and  $(i - 3)$ .strategy!= $W$  then
17:         $i$ .status= $C$ ,  $(i - 1)$ .status= $C$ ,  $(i - 2)$ .status= $C$ ,
           $(i - 3)$ .status= $C$ 
18:      end if
19:      if  $i == 2$  then
20:         $i$ .status= $C$ ,  $(i - 1)$ .status= $C$ 
21:      end if
22:    end if
23:  end for
24:  Nodes that have completed their statuses send a local
    broadcast message to their neighbors along with their
    strategy.
25:  if all nodes have completed their statuses, the algorithm
    ends and this strategy vector is a NE for the topology.
    Else, the nodes that have incomplete status, update
    randomly their strategy by taking into account any
    limitations that are imposed by the strategy of the nodes
    that have completed their statuses.
26: end for

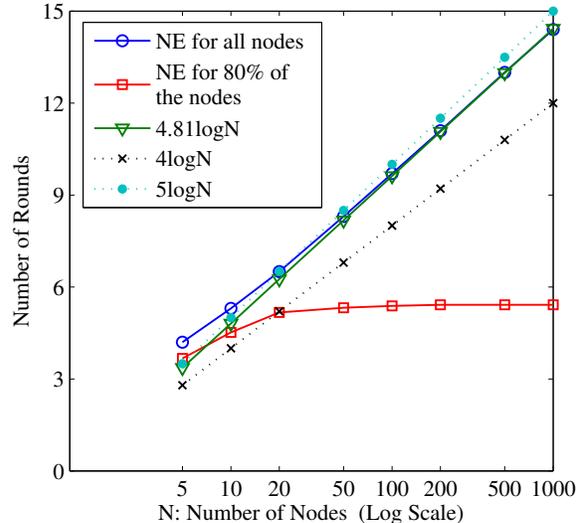
```

or $(s_{i-2}, s_{i-1}, s_i) = (W, W, T)$ hold, then node i has a semi-successful transmission. Cases where a node has less than 2 left/right neighbors are treated similarly with the unicast case. At a NE, each node should either wait, or have a fully-successful transmission, or have a semi-successful transmission as, even in that case, he has no motivation to change his strategy to wait, as his payoff will be decreased from $1 - c$ to 0 (note that $c \in (0, 1)$).

We then focus on how to find a NE under multicast traffic. As the naive scheme can be applied directly without further changes, we highlight the changes that should be adopted for the sophisticated scheme. A strategy subvector $(s_i, s_{i+1}, s_{i+2}, s_{i+3})$ or $(s_{i-3}, s_{i-2}, s_{i-1}, s_i)$ is guaranteed to be part of a NE of the topology if it is of the form (T, W, W, T) . This is true since these transmissions will be (at least) semi-successful and the intermediate nodes that wait cannot have a (semi-)successful transmission. The only difference from the unicast case is that the part of the Algorithm 1 in lines 32-43 is not needed any more, as node $i \pm 3$ should



(a) Comparison of the proposed algorithms. Average round for convergence to a NE under multicast for the proposed schemes.



(b) Analysis of the convergence time of the sophisticated scheme.

Fig. 3: Performance evaluation for the multicast case: Comparison of the proposed schemes and analysis of the average convergence time of the sophisticated scheme.

not need to examine further his strategy and (probably) adopt a best response scheme in the next round. Corner cases with less than 3 left/right neighbors are treated similarly with the unicast case. So, using the above rules and making straightforward changes to the pseudocode of the sophisticated scheme for the unicast case, we present in Algorithm 2 a distributed iterative scheme that converges to a NE from every initial state.

We finally evaluate the performance of the algorithms using the same metrics with the unicast case. Again, for the naive scheme, we present both an unbiased and a biased version. We notice from Fig. 3a that the sophisticated scheme converges very fast to a NE. The convergence is proportional to $k \log_{10}(N)$, where k is a coefficient and N is the number of the nodes of the topology. Our simulations show that $k = 4.81$ approximates closely the results from the simulations for various sizes of the topology (Fig. 3b). The unbiased version of the naive scheme works quite well but this is not the case for the biased version. Our comments on Fig. 2 hold for Fig. 3a as well. Concerning the convergence of the 80% of the nodes of the topology to a NE under the sophisticated scheme, this is done in up to 5.4 iterations for any type of the topology that consists of at least 20 nodes (Fig. 3b).

VII. CONCLUSIONS

We focus on linear multihop device-to-device networks where devices decide autonomously their strategy (either to transmit or to wait and receive data). In such networks, we show that the analysis of the structure of a strategy vector at a NE is not only useful from a theoretical perspective; it can be the key factor for a practical scheme that has appealing properties. We propose a sophisticated scheme where devices communicate in a 3-hop neighborhood that clearly outperforms our naive scheme in [2] where devices exchange information in a 2-hop neighborhood. Devices that apply the sophisticated scheme converge to a NE in a number of rounds that is proportional to the logarithm of the network size. Moreover, when

devices in the neighborhood end up in a strategy subvector that is a local NE, it is guaranteed that this will be part of the global NE of the network. This both reduces the waste of resources and contributes to the faster convergence to a NE.

ACKNOWLEDGMENT

Vaggelis G. Douros is supported by the HERAKLEITOS II Programme which is co-financed by the European Social Fund and National Funds through the Greek Ministry of Education.

The research of Stavros Toumpis has been co-financed by the European Union (European Social Fund ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF) Research Funding Program: THALES. Investing in knowledge society through the European Social Fund.

REFERENCES

- [1] K. Doppler, M. Rinne, C. Wijting, C. Ribeiro, and K. Hugl, “Device-to-device communication as an underlay to LTE-advanced networks,” *IEEE Communications Magazine*, vol. 47, no. 12, pp. 42–49, 2009.
- [2] V. G. Douros, S. Toumpis, and G. C. Polyzos, “On the Nash Equilibria of graphical games for channel access in multihop wireless networks,” in *Proc. Wireless Evolution Beyond 2020 Workshop, in conjunction with IEEE Wireless Communications and Networking Conference (WCNC)*, 2014.
- [3] I. Jawhar, N. Mohamed, K. Shuaib, and N. Kesserwan, “An efficient framework and networking protocol for linear wireless sensor networks,” *Ad Hoc & Sensor Wireless Networks*, vol. 7, no. 1-2, pp. 3–21, 2009.
- [4] F. El Ali, B. Ducourthial, and S.-M. Senouci, “On the capacity of a linear vehicular network,” in *Proc. IEEE Vehicular Technology Conference (VTC-Spring)*, 2011.
- [5] C. Zhu and M. S. Corson, “A five-phase reservation protocol (FPRP) for mobile ad hoc networks,” *Springer Wireless Networks*, vol. 7, no. 4, pp. 371–384, 2001.
- [6] M. Kearns, *Graphical Games, chapter 7 of Algorithmic Game Theory*. Cambridge University Press, edited by N. Nisan et al., 2007.