

# Exploiting Peer-to-Peer State Exchange for Distributed Medium Access Control

Ka Hung Hui, Tianyi Li, Dongning Guo and Randall A. Berry  
Department of Electrical Engineering and Computer Science  
Northwestern University, Evanston, IL 60208, USA

**Abstract**—Distributed medium access control (MAC) protocols are proposed for wireless networks assuming that one-hop peers can exchange a small amount of state information periodically. Each station maintains a state and makes state transitions and transmission decisions based on its state and recent state information collected from its one-hop peers. A station can adapt its packet length and the size of its state space to the amount of traffic in its neighborhood. It is shown that these protocols converge to a steady state, where stations take turns to transmit in each neighborhood without collision. An important consequence of this work is that using such protocols, an efficient time-division multiple access (TDMA) like schedule can be formed in a distributed manner, as long as the topology of the network remains static or changes slowly with respect to the execution of the protocol.

## I. INTRODUCTION

The performance of wireless networks is limited by two types of collisions: a station transmits to another transmitting station which is half-duplex; or a station receives simultaneous transmissions from multiple stations such that all transmissions cannot be decoded correctly. Medium access control (MAC) protocols are needed to eliminate or reduce both types of collisions. A typical design of MAC protocols is to let each station maintain a state variable and make transmission decisions according to the state, which is updated based on the station's observation. Besides eliminating collisions, a practical MAC protocol should be *distributed*, *i.e.*, stations make their decisions based on information available locally in space. A practical MAC protocol should also be *adaptive* to recent changes in time.

We consider static or slow-varying networks and study *self-stabilizing* MAC protocols, *i.e.*, protocols that converge to a collision-free schedule, independent of the initial state. In the steady state, these protocols behave like time-division multiple access (TDMA), in which stations take turns to transmit without collision; while in the transient state, they behave like carrier-sense multiple access (CSMA), such that stations contend with each other, trying to find a slot for transmission and avoid collisions. References [1]–[3] study self-stabilizing MAC protocols for a single collision domain, in which all stations can hear each other. By learning transmission decisions of others, stations are able to find a collision-free schedule in a decentralized manner. However, the analysis applies only to a single collision domain. It is pointed out in [1] that the

protocol therein cannot guarantee the formation of a collision-free schedule in multiple collision domains.

In practice, most networks have multiple collision domains and hence the hidden terminal and exposed terminal problems. For one- and two-dimensional *regular* networks with links between nearest neighbors only, distributed, adaptive, self-stabilizing MAC protocols are introduced in [4] and [5], respectively. The technique is to divide time into periodic cycles, where each cycle is divided into slots. A station maintains a state and transmits only over the slot corresponding to its state. Once the protocols converge, a periodic state pattern is formed throughout the regular network, and the maximum broadcast throughput is achieved. If one directly applies these ideas to networks with arbitrary topologies, sufficiently many states are needed for stations with many neighbors, but for stations with few neighbors, the wireless channel is underutilized.

In this paper, we propose multi-resolution MAC protocols for one- and two-dimensional wireless networks with arbitrary topologies, *i.e.*, a station having more neighbors uses a fine resolution (more states are used, each state corresponding to a shorter slot); while a station with fewer neighbors uses a coarse resolution (fewer states are used, each state corresponding to a longer slot). These protocols guarantee every station a chance to transmit in each cycle. In addition, they achieve approximate proportional fairness in the sense that two stations with similar number of neighbors have similar rates. For one-dimensional networks, stations can determine their resolutions in a distributed manner. The same also holds for two-dimensional networks under a mild condition. In case the condition is not met, we propose a mechanism for stations to dynamically change their resolutions until collisions do not occur in the entire network. In all cases, the convergence of such protocols to a collision-free schedule is rigorously established. We do not consider maximizing throughput or minimizing the number of states; these problems are NP-complete [6], [7].

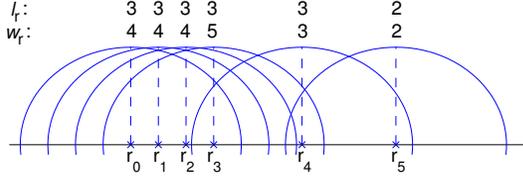
The remainder of this paper is organized as follows. The system model is described in Section II. Results for one- and two-dimensional networks are presented in Sections III and IV, respectively. We comment on the key ideas of the proof for the theorems but omit details due to space limitations. Section V concludes the paper.

## II. SYSTEM MODEL

Consider a simple model for wireless networks where two stations have a direct radio link between them if their distance

|       |     | cycle |     |     |     |     |     |     |  |
|-------|-----|-------|-----|-----|-----|-----|-----|-----|--|
|       |     | 00    |     | 01  |     | 10  |     | 11  |  |
| $r_5$ | 000 | 001   | 010 | 011 | 100 | 101 | 110 | 111 |  |
| $r_4$ | 000 | 001   | 010 | 011 | 100 | 101 | 110 | 111 |  |
| $r_3$ | 000 | 001   | 010 | 011 | 100 | 101 | 110 | 111 |  |
| $r_2$ | 000 | 001   | 010 | 011 | 100 | 101 | 110 | 111 |  |
| $r_1$ | 000 | 001   | 010 | 011 | 100 | 101 | 110 | 111 |  |
| $r_0$ | 000 | 001   | 010 | 011 | 100 | 101 | 110 | 111 |  |

(a)



(b)

Fig. 1. A multi-resolution MAC protocol in a one-dimensional network. In (b), there are 6 stations at positions  $r_0, \dots, r_5$ . The circles represent the transmission ranges of the stations. The values of  $w_r$  and  $l_r$  for different stations are shown on top of the corresponding circles. A possible schedule over a cycle is shown in (a).

is within a given range. Precisely, the network is modeled by a finite undirected *unit disk graph*  $G_R = (V, A_R)$ , where  $V = \{\mathbf{r}_i\}_{i=0}^{|V|-1}$  is the set of stations labeled by their coordinates,  $A_R = \{(\mathbf{r}_i, \mathbf{r}_j) \in V \times V : i \neq j \text{ and } d(\mathbf{r}_i, \mathbf{r}_j) \leq R\}$  is the set of undirected links,  $d(\mathbf{r}_i, \mathbf{r}_j)$  denotes the Euclidean distance between  $\mathbf{r}_i$  and  $\mathbf{r}_j$ , and  $R$  is the transmission range. Let  $V_r$  denote the set of (one-hop) peers or neighbors of station  $\mathbf{r}$ . It is assumed that every station broadcasts packets to all its one-hop peers and has saturated traffic.

Next we formalize the concept of multiple resolutions alluded to above. Consider Fig. 1(a). Station  $\mathbf{r}_5$  uses a coarse resolution consisting of four states, 00, 01, 10, 11. It transmits in state 10, *i.e.*, the third quarter of the cycle. Station  $\mathbf{r}_3$  uses a finer resolution consisting of eight states. It transmits in the sixth slot of the cycle when its state is 101. Station  $\mathbf{r}_3$ 's resolution can be seen as a refinement of that of station  $\mathbf{r}_5$ , where every slot is reduced by half. Formally, the state of station  $\mathbf{r}$  assumes values in  $\mathbb{F}_2^{l_r}$ , the set of binary  $l_r$ -tuples, where  $2^{l_r}$  is the total number of states available for station  $\mathbf{r}$  to choose and it depends on its local topology. Let  $\Omega = \prod_{\mathbf{r} \in V} \mathbb{F}_2^{l_r}$  denote the configuration space. Let  $X_{\mathbf{r}}(t) \in \mathbb{F}_2^{l_r}$  be the state of station  $\mathbf{r}$  in the  $t$ -th cycle, and  $\mathbf{X}(t) = \{X_{\mathbf{r}}(t)\}_{\mathbf{r} \in V} \in \Omega$  be the configuration in the  $t$ -th cycle.  $X_{\mathbf{r}}(t) = s \in \mathbb{F}_2^{l_r}$  means that station  $\mathbf{r}$  divides the  $t$ -th cycle into  $2^{l_r}$  slots of equal length, and it transmits in slot  $s$ . For this reason, we will use states and slots interchangeably. We assume that all stations are synchronized. Therefore, a fine resolution can be obtained by ‘splitting’ or ‘refining’ a coarse resolution, as shown in Fig. 1(a).

We assume that packets transmitted by a station fit in a slot of its own resolution. Stations using coarse resolutions can also transmit multiple packets of smaller sizes in a slot. A *collision* occurs at some receiver when two stations that are one-hop or two-hop peers transmit at the same time. Mathematically,

neighboring stations  $\mathbf{r}_i$  and  $\mathbf{r}_j$ , with  $l_{r_i} \leq l_{r_j}$ , collide in the  $t$ -th cycle when

$$\text{the binary string } X_{\mathbf{r}_i}(t) \text{ is a prefix of } X_{\mathbf{r}_j}(t). \quad (1)$$

For example, in Fig. 1(a), since 10 is a prefix of 101, stations  $\mathbf{r}_3$  and  $\mathbf{r}_5$  collide. In a *collision-free configuration*, (1) does not hold for any pair of one-hop and two-hop peers.

We assume that at the end of each cycle, each station acquires the current states of its one-hop and two-hop peers, error-free. The careful reader may object that this itself requires a collision-free schedule. However, since this control information is relatively low-rate, we assume that other techniques can be utilized for sending it. For example, the rapid on-off division duplex (RODD) scheme in [8] can enable all stations to exchange their control messages simultaneously.

Let stations choose their next states based only on the current states of their one-hop and two-hop peers and themselves. The state process of the MAC protocol can be modeled as a *Markov Chain of Markov Fields* (MCMF) [9], *i.e.*, a process for which the states  $\mathbf{X} = \{\mathbf{X}(t)\}_{t \in \mathbb{N}}$  satisfy

- $\mathbf{X}(1), \mathbf{X}(2), \dots$  is a Markov chain on  $\Omega$ , and
- for every  $t$ ,  $\mathbf{X}(t)$  is a Markov field on  $\Omega$  conditioned on  $\mathbf{X}(t-1)$ .

In fact,  $\mathbf{X}(t)$  is a set of independent random variables conditioned on  $\mathbf{X}(t-1)$  in our case. Here, we only consider protocols in which stations make identically distributed decisions conditioned on the same previous states of their one-hop and two-hop peers and themselves.<sup>1</sup>

The performance measure we use is the one-hop broadcast throughput  $\rho$ , the average proportion of time a station receives packets in each cycle. With proper scaling, this also implies a lower bound on the unicast or multicast throughput. A station receives a packet if and only if it does not transmit and only one of its peers transmits. If there is no collision,

$$\rho = \langle \sum_{\mathbf{r}' \in V_r} 2^{-l_{r'}} \rangle_{\mathbf{r}} = \langle |V_r| 2^{-l_r} \rangle_{\mathbf{r}} \quad (2)$$

where  $\langle \cdot \rangle_{\mathbf{r}}$  in (2) is the spatial average over all stations, *i.e.*,  $\langle g(\mathbf{r}) \rangle_{\mathbf{r}} = \frac{1}{|V|} \sum_{\mathbf{r} \in V} g(\mathbf{r})$  for any function  $g$ . The two expressions are obtained by counting throughput from the receiver side and the transmitter side respectively.

### III. ONE-DIMENSIONAL NETWORKS

#### A. Determining the number of states

Station  $\mathbf{r}$  determines its resolution (in bits)  $l_r$  as follows:

- 1) station  $\mathbf{r}$  computes  $w_r = 1 + |V_r|$ , the number of stations within  $\mathbf{r}$ 's one-hop neighborhood, and exchanges this with all its one-hop peers (*e.g.*, using techniques of [8]),
- 2) station  $\mathbf{r}$  sets  $l_r = \lceil \log_2(\max_{\mathbf{r}' \in V_r \cup \{\mathbf{r}\}} w_{\mathbf{r}'}) \rceil$ .

The intuition is that the size of the state space of a station should be at least equal to the size of the largest one-hop neighborhood to which the station belongs. Fig. 1(b) illustrates the procedure.

<sup>1</sup>This rules out protocols in which stations, for example, are simply assigned to transmit or not based on their location (*e.g.*, in [10]).

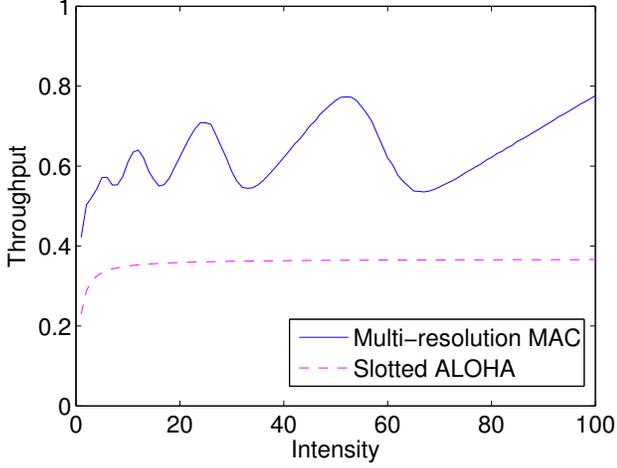


Fig. 2. Throughput of one-dimensional networks versus node intensity.

*Theorem 1:* For any one-dimensional network, if station  $\mathbf{r}$  uses  $l_{\mathbf{r}} = \lceil \log_2(\max_{\mathbf{r}' \in V_{\mathbf{r}} \cup \{\mathbf{r}\}} w_{\mathbf{r}'}) \rceil$ , where  $w_{\mathbf{r}} = 1 + |V_{\mathbf{r}}|$ , then it is possible for each station to choose a state such that collision-free configurations exist. The resulting one-hop broadcast throughput is

$$\rho = \langle |V_{\mathbf{r}}| 2^{-\lceil \log_2(\max_{\mathbf{r}' \in V_{\mathbf{r}} \cup \{\mathbf{r}\}} (1 + |V_{\mathbf{r}'|}) \rceil} \rangle_{\mathbf{r}}. \quad (3)$$

Consider finite segments of one-dimensional networks where stations are distributed following a Poisson point process with intensity  $\lambda$ , and the transmission range is  $R = 1$ . We evaluate (3) by averaging over different realizations of the networks. How the throughput  $\rho$  varies with the intensity  $\lambda$  is shown in Fig. 2. The throughput oscillation is due to the fact that the size of any state space is a power of 2. In a worst-case scenario, if a station determines that it needs  $2^n + 1$  states, it actually has to use a resolution of  $2^{n+1}$  states, meaning that potentially almost half of the cycle will be left idle, hence the throughput can be very close to 0.5. After forming a collision-free configuration, there may still be many idle slots in certain neighborhoods. How stations reclaim these idle slots is left to future work.

For comparison, we also compute the throughput for slotted ALOHA. Consider a segment of a one-dimensional network of length  $2R$  with a station at the center. This station has  $k$  peers with probability  $\exp(-\lambda 2R) \frac{(\lambda 2R)^k}{k!}$ . Assume stations transmit with a fixed probability  $p$ . This station receives a packet successfully with probability  $kp(1-p)^k$ . Then,

$$\begin{aligned} \rho(p) &= \sum_{k=1}^{\infty} \exp(-\lambda 2R) \frac{(\lambda 2R)^k}{k!} kp(1-p)^k \\ &= \lambda 2R p(1-p) \exp(-\lambda 2R p). \end{aligned}$$

Optimizing over all  $p$ , the throughput is

$$\rho = \frac{\lambda 2R}{2 + \sqrt{4 + (\lambda 2R)^2}} \exp\left(-\frac{2\lambda 2R}{2 + \lambda 2R + \sqrt{4 + (\lambda 2R)^2}}\right).$$

---

### Protocol 1 Multi-Resolution MAC Protocol

---

- 1:  $\mathbf{r}$  sets the votes on all states to zero.
  - 2: **for**  $\mathbf{r}' \in V_{\mathbf{r}} \cup \{\mathbf{r}\}$  **do**
  - 3:     **if**  $\mathbf{r}$  is the only station occupying its current state in station  $\mathbf{r}'$ 's one-hop neighborhood **then**
  - 4:          $\mathbf{r}$ 's current state is assigned a single vote of weight one.
  - 5:     **else**
  - 6:          $\mathbf{r}$  determines which states (according to  $\mathbf{r}$ 's resolution) are idle or have collisions in  $\mathbf{r}'$ 's one-hop neighborhood.
  - 7:         A vote of weight  $\frac{1}{n}$  is given to each such state, where  $n$  is the number of such states.
  - 8:     **end if**
  - 9: **end for**
  - 10:  $\mathbf{r}$  selects state  $s$  with a probability proportional to  $f(n_s)$ , where  $n_s$  is the total weight state  $s$  receives.
- 

This optimized throughput, with  $R = 1$ , is plotted in Fig. 2. The multi-resolution MAC protocol provides 46.7% to 112.2% improvement in terms of throughput over slotted ALOHA.

#### B. A Multi-Resolution MAC Protocol

In the following we propose a *multi-resolution protocol* that leads to a collision-free configuration *starting from an arbitrary initial configuration*. Stations can learn two-hop state information in each cycle as follows. In the  $t$ -th cycle, station  $\mathbf{r}$  collects  $\{(X_{\mathbf{r}'}(t))\}_{\mathbf{r}' \in V_{\mathbf{r}}}$ , and then broadcasts  $\{(X_{\mathbf{r}'}(t))\}_{\mathbf{r}' \in V_{\mathbf{r}} \cup \{\mathbf{r}\}}$ . Hence, station  $\mathbf{r}$  knows  $X_{\mathbf{r}'}(t)$  for all one-hop and two-hop peers  $\mathbf{r}'$ , and selects its state at the  $(t+1)$ -st cycle following Protocol 1.

In Protocol 1,  $f: \mathbb{R} \mapsto \mathbb{R}$  is an increasing function with  $f(0) = 0$ , e.g.,  $f(n_s) = \exp(Jn_s)\mathbf{1}_{\{n_s > 0\}}$ , where  $\mathbf{1}_{\{\cdot\}}$  is the indicator function and  $J > 0$  is the *strength of interaction* (discussed later). The idea of Protocol 1 is that a station ‘reserves’ a slot for a peer if it knows that this peer does not collide with other peers, and notifies any peer experiencing collisions to stay away from these ‘reserved’ slots. We have the following convergence result for this protocol.

*Theorem 2:* If each station in a one-dimensional network chooses its resolution following Theorem 1 and executes Protocol 1, then a collision-free configuration will be formed after a sufficiently long time, regardless of the initial state.

This is proved by showing that with a nonzero probability each station arrives at an optimal state and remains there since it is an absorption state.

#### C. Simulations: Convergence Speed-up by Annealing

Simulations of the proposed protocol show that it may take a long time for a collision-free configuration to appear. Here we propose speeding up the convergence by *annealing*, i.e., we consider the multi-resolution protocol with  $f(n_s) = \exp(J(t)n_s)\mathbf{1}_{\{n_s > 0\}}$ , where  $J(t) = \gamma J(t-1)$ ,  $\gamma > 1$  controls the increase in the strength of interaction, and  $J(0) = 1$ . Define the convergence time to be the first time that a certain

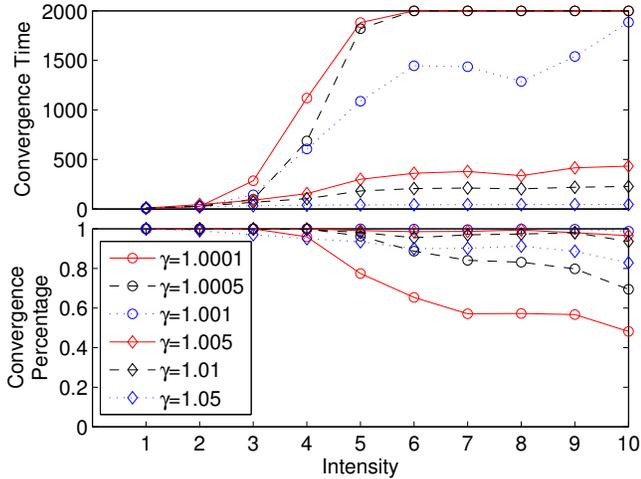


Fig. 3. Simulations of the multi-resolution protocol with annealing on one-dimensional networks.

configuration is observed and remains unchanged till the end of the simulation, and the convergence percentage to be the proportion of stations that do not collide in that configuration. *This configuration may not be collision-free, hence, there is a nonzero probability that the network transits to another configuration, but this probability is so small (as  $J(t)$  is very large, resulting in every station staying in the state with maximum vote) that this transition is practically impossible.* We consider a line segment of length 50 on which stations are distributed following a Poisson point process with intensity  $\lambda$ . The transmission range is  $R = 1$ . The convergence time and percentage are plotted in Fig. 3. When  $\gamma$  is too small, the effect of annealing is not significant. When  $\gamma$  is too large, the convergence time is reduced drastically, but the convergence percentage is also reduced.

#### IV. TWO-DIMENSIONAL NETWORKS

##### A. Determining the number of states

Unlike in one-dimensional networks, the resolution  $l_r$  cannot be completely determined by two-hop topology information in two-dimensional networks. An example is shown in the left part of Fig. 4(a). If the rule in Theorem 1 is used here, all stations should use a resolution of four states. But since every station is within two hops of every other station, at least five states are needed to resolve any collision. This situation can be remedied if the network  $G_R$  is ‘well-connected’ and the protocol is modified as follows:  $l_r$  is determined by the size of the maximum clique in  $G_R^2$  containing  $r$ , where  $G_R^2 = (V, A_R^2)$  is the square of  $G_R$ , i.e.,  $(r_i, r_j) \in A_R^2$  if  $r_i$  and  $r_j$  are one-hop or two-hop peers in  $G_R$ .

*Theorem 3:* Suppose a two-dimensional network  $G_R$  has a chordal square, i.e.,  $G_R^2$  is chordal. If station  $r$  uses resolution  $l_r = \lceil \log_2 |C_r| \rceil$ , where  $C_r$  is the maximum clique in  $G_R^2$  containing  $r$ , then it is possible for each station to choose a state such that collision-free configurations exist.

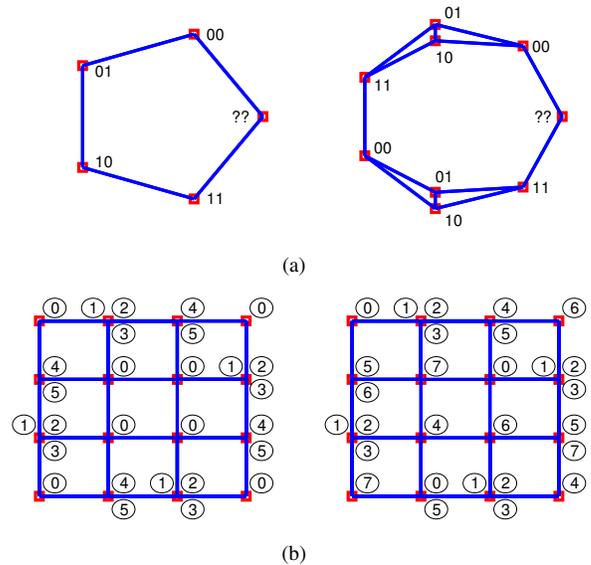


Fig. 4. Intricacies for two-dimensional networks: (a) determining the number of states (‘?’ labels stations that are unable to pick a state without collision), (b) converging to a collision-free configuration.

A graph is *chordal* if in any cycle of at least four vertices, there must exist an edge between some two nonadjacent vertices. Furthermore, a graph is chordal if and only if it has a perfect elimination ordering of vertices [11]. A *perfect elimination ordering* is constructed by repeatedly finding a vertex such that all its neighbors form a clique, and then removing it along with all incident edges. The order that the vertices are removed is a perfect elimination ordering. Theorem 3 is true because stations can pick their states following the reverse of a perfect elimination ordering, and so guarantee no collisions occur.

The condition in Theorem 3 is sufficient but not necessary. For example, the right part of Fig. 4(a) shows that a collision-free configuration cannot be found using the resolutions predicted in Theorem 3. Consider also the right part of Fig. 4(b), which is a  $4 \times 4$  square lattice where multiple stations are collocated on some lattice points. Theorem 3 predicts that every station uses a resolution of eight states, and a collision-free configuration exists, as shown in the figure. For illustrative purposes, we use octal representation of the states, e.g., state 101 is denoted as 5. In both cases,  $G_R^2$  is not chordal.

Since the sizes of all state spaces are powers of 2, additional states are provisioned in many cases. Therefore, a collision-free configuration can be formed using local information exchange even for many networks without chordal squares.

##### B. A Modified Multi-Resolution MAC Protocol

The multi-resolution protocol for one-dimensional networks may not work for all two-dimensional networks. An example is illustrated in Fig. 4(b). Every station uses a resolution of eight states. The right part of Fig. 4(b) shows that a collision-free configuration exists. But, if the initial configuration is the one shown in the left part of Fig. 4(b), and the protocol in Section III is used, then the following occurs:

- 1) All stations in initial states 1, 2, 3, 4, 5 remain in their current states with probability one, since they do not collide. All stations in initial state 0 can only choose 0, 6, 7 as their next states, because all other states are not available. This repeats for all subsequent iterations.
- 2) Consider the four stations in the middle of the network, which have initial states 0. They are within two hops of each other, so they must use different states. However, only states 0, 6, 7 are available to them in any cycle. Hence, collision-free configurations cannot be reached.

A simple modification prevents the preceding deadlock. For any station, if the votes received do not all point to a single state, then the station increases the total weight of the votes received for any state by a nonzero constant. A station in this situation will have nonzero probability of choosing any state to be its next state. This randomization does not affect any absorption configuration, and is necessary to establish the counterpart of Theorem 2 for two-dimensional networks.

*Theorem 4:* For a two-dimensional network, suppose each station uses a sufficiently fine resolution so that the existence of collision-free configurations is guaranteed. Then, starting from an arbitrary initial configuration, the modified multi-resolution protocol will, after a sufficiently long time, result in a collision-free configuration.

Theorem 4 is established by showing that there is a nonzero probability to reach the all-zero configuration and then reach a collision-free configuration in the next cycle.

### C. Simulations: Dynamically Adjusting the Number of States

For a general two-dimensional network, it is difficult for stations to predict the resolutions they need. It may still be difficult even for networks with chordal squares, since it is not known whether it is possible to find the maximum clique in the square of a unit disk graph efficiently. Therefore, we propose the following dynamic algorithm. Initially, every station follows the rule proposed in Theorem 1 and executes the modified multi-resolution protocol with annealing. If a station knows that the local configuration within its two-hop neighborhood remains the same for a number of iterations (10 in our simulations), but it still experiences collisions, then it checks if there are any idle states within its two-hop neighborhood. If such states exist, it selects one of these states; otherwise, it doubles the size of its state space (*i.e.*, it ‘refines’ its resolution), picks its state randomly, resets the strength of interaction it uses and continues executing the protocol.

For simulation, we consider  $10 \times 10$  square area of two-dimensional networks where stations are distributed following a Poisson point process with intensity  $\lambda$ . All other simulation settings are the same as those for one-dimensional networks. The convergence time and percentage are plotted in Fig. 5. The convergence time is longer compared to one-dimensional networks, since stations may need to adjust their resolutions. When  $\gamma$  is too large, the convergence time increases drastically. For large  $\gamma$ , the interaction between stations is so large that the protocol behaves like *majority vote* shortly after the

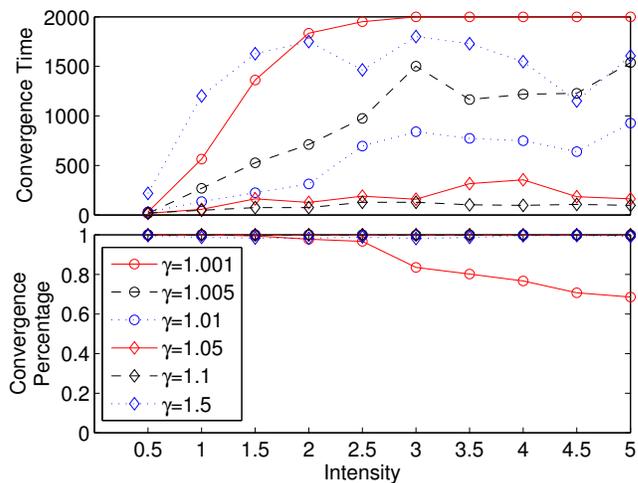


Fig. 5. Simulations of the modified multi-resolution protocol with annealing on two-dimensional networks.

protocol is executed. This makes the randomization of states after each refinement not effective in resolving collisions.

## V. CONCLUSION

In this paper, we have proposed multi-resolution MAC protocols for wireless networks with arbitrary topologies. We have shown that collisions can be eliminated in a distributed manner, by letting stations make their transmission decisions based on recent transmission decisions of their peers. These protocols do not require all stations to use the same resolution, *i.e.*, the same state space or the same length of each slot.

## REFERENCES

- [1] J. Lee and J. Walrand, “Design and analysis of an asynchronous zero collision MAC protocol,” 2008, arXiv:0806.3542v1 [cs.NI].
- [2] J. Barcelo, B. Bellalta, C. Cano, and M. Oliver, “Learning BEB: Avoiding collisions in WLAN,” *Eunice Summer School*, 2008.
- [3] M. Fang, D. Malone, K. R. Duffy, and D. J. Leith, “Decentralised learning MACs for collision-free access in WLANs,” 2010, arXiv:1009.4386v1 [cs.NI].
- [4] K. H. Hui, D. Guo, R. A. Berry, and M. Haenggi, “Performance analysis of MAC protocols in wireless line networks using statistical mechanics,” in *Allerton Conference on Communication, Control and Computing*, Monticello, IL, USA, Sep. 2009.
- [5] K. H. Hui, D. Guo, and R. A. Berry, “Medium access control via nearest-neighbor interactions for regular wireless networks,” in *International Symposium on Information Theory*, Austin, TX, USA, Jun. 2010.
- [6] A. Ephremides and T. V. Truong, “Scheduling broadcasts in multihop radio networks,” *IEEE Trans. Commun.*, vol. 38, no. 4, pp. 456–460, Apr. 1990.
- [7] R. Ramaswami and K. K. Parhi, “Distributed scheduling of broadcasts in a radio network,” in *INFOCOM ’89*, Ottawa, Ontario, Canada, Apr. 1989, pp. 497–504.
- [8] D. Guo and L. Zhang, “Virtual full-duplex wireless communication via rapid on-off-division duplex,” in *Allerton Conference on Communication, Control and Computing*, Monticello, IL, USA, Sep. 2010.
- [9] X. Guyon and C. Hardouin, “Markov chain markov field dynamics: Models and statistics,” *Statistics*, vol. 36, no. 4, pp. 339–363, 2002.
- [10] N. Wen and R. A. Berry, “Location-based MAC protocols for mobile wireless networks,” in *2007 Information Theory and Applications Workshop (ITA)*, San Diego, CA, USA, 2007.
- [11] D. R. Fulkerson and O. A. Gross, “Incidence matrices and interval graphs,” *Pacific Journal of Mathematics*, vol. 15, no. 3, pp. 835–855, 1965.