

Self-Stabilizing TDMA Algorithms for Dynamic Wireless Ad-hoc Networks ^{*}

Pierre Leone [†] Elad M. Schiller ^{‡§}

June 11, 2013

Abstract

In dynamic wireless ad-hoc networks (DynWANs), autonomous computing devices set up a network for the communication needs of the moment. These networks require the implementation of a medium access control (MAC) layer. We consider MAC protocols for DynWANs that need to be autonomous and robust as well as have high bandwidth utilization, high predictability degree of bandwidth allocation, and low communication delay in the presence of frequent topological changes to the communication network. Recent studies have shown that existing implementations cannot guarantee the necessary satisfaction of these timing requirements. We propose a self-stabilizing MAC algorithm for DynWANs that guarantees a short convergence period, and by that, it can facilitate the satisfaction of severe timing requirements, such as the above. Besides the contribution in the algorithmic front of research, we expect that our proposal can enable quicker adoption by practitioners and faster deployment of DynWANs that are subject changes in the network topology.

^{*}The proceeding and technical versions of this work appears in [38], and respectively [39].

[†]Computer Science Department, University of Geneva, Centre Universitaire d'Informatique, Battelle bâtiment A, route de Drize 7, 1227 Carouge, Geneva, Switzerland. Email: pierre.leone@unige.ch

[‡]Chalmers University of Technology, Rännvägen 6B, S-412 96 Göteborg Sweden. Email: elad.schiller@chalmers.se

[§]The work of this author was partially supported by the EC, through project FP7-STREP-288195, KARYON (Kernel-based ARchitecture for safetY-critical cONtrol).

1 Introduction

Dynamic wireless ad-hoc networks (DynWANs) are autonomous and self-organizing systems where computing devices require networking applications when a fixed network infrastructure is not available or not preferred to be used. In these cases, computing devices may set up a short-lived network for the communication needs of the moment, also known as, an ad-hoc network. Ad-hoc networks are based on wireless communications that require implementation of a *Medium Access Control* (MAC) layer. We consider MAC protocols for DynWANs that need to be autonomous, robust, and have high bandwidth utilization, a high predictability degree of bandwidth allocation, and low communication delay [23] in the presence of frequent changes to the communication network topology. Existing implementations cannot guarantee the necessary satisfaction of timing requirements [6, 7]. This work proposes an algorithmic design for self-stabilizing MAC protocols that guarantees a short convergence period, and by that, can facilitate the satisfaction of severe timing requirements. The proposed algorithm possesses a greater degree of predictability, while maintaining low communication delays and high throughput.

The dynamic and difficult-to-predict nature of wireless ad-hoc networks gives rise to many fault-tolerance issues and requires efficient solutions. DynWANs, for example, are subject to transient faults due to hardware/software temporal malfunctions or short-lived violations of the assumed settings for modeling the location of the mobile nodes. Fault tolerant systems that are *self-stabilizing* [16] can recover after the occurrence of transient faults, which can cause an arbitrary corruption of the system state (so long as the program's code is still intact), or the model of dynamic networks in which communication links and nodes may fail and recover during normal operation [17]. The proof of self-stabilization requires convergence from an arbitrary starting system state. Moreover, once the system has converged and followed its specifications, it is required to do so forever. The self-stabilization design criteria liberate the application designer from dealing with low-level complications, such as bandwidth allocation in the presence of topology changes, and provide an important level of abstraction. Consequently, the application design can easily focus on its task – and knowledge-driven aspects.

The IEEE 802.11 standard is widely used for wireless communications. Nonetheless, the research field of MAC protocols is very active and requires further investigation. In fact, the IEEE 802.11 amendment, IEEE 802.11p, for wireless access in vehicular environments (WAVE), has just been published. It was shown that the standard's existing implementations cannot guarantee channel access before a finite deadline [6, 7]. Therefore, applications with severe timing requirements cannot predictably meet their deadlines, e.g., safety-critical applications for vehicular systems.

ALOHAnet and its synchronized version Slotted ALOHA [1] are pioneering wireless systems that employ a strategy of “random access”. Time division multiple access (TDMA) [44] is another early approach, where nodes transmit one after the other, each using its own timeslot, say, according to a defined schedule. Radio transmission analysis in ad-hoc networks [21] and relocation analysis of mobile nodes [35] show that there are scenarios in which MAC algorithms that employ a scheduled access strategy have lower throughput than algorithms that follow the random access strategy. How-

ever, the scheduled approach offers greater predictability of bandwidth allocation and communication delay, which can facilitate fairness [24] and energy conservation [53].

Our design choices have basic radio technology in mind, whilst aiming at satisfying applications that have severe timing requirements. We consider TDMA frames with fixed number of fixed length timeslots. The design choice of TDMA frames with fixed-length radio time fits well applications that have severe delay requirements. By avoiding the division of fixed length frames into timeslots of non-equal length, as in [10, 24], we take into consideration the specifications of basic radio technology.

In the context of the above design choices, there are two well-known approaches for dealing with contention (timeslot exhaustion): (1) employing policies for administering message priority (for meeting timing requirements while maintaining high bandwidth utilization, such as [42]), or (2) adjusting the nodes' individual transmission signal strength or carrier sense threshold [46]. The former approach is widely accepted and adopted by the IEEE 802.11p standard, whereas the latter has only been evaluated via computer simulations. The proposed algorithm facilitates the implementation of both of the above approaches. We consider implementation details of the standard approach in Section 7.

For the sake of presentation simplicity, we start by considering a single priority MAC protocol and base the timeslot allocation on vertex-coloring, before considering multi-priority implementation in Section 7. The proposed algorithm allocates timeslots to a number of nearby transmitters, i.e., a number that is bounded by the TDMA frame size, whereas non-allocated transmitters receive busy channel indications. The analysis considers saturated situations in which the node degree in the message collision graph is smaller than the TDMA frame size. As explained above, this analysis assumption does not restrict the number of concurrent transmitters when implementing the proposed MAC algorithm.

1.1 Related work

We are not the first to propose a MAC algorithm for DynWANs that follows the TDMA's scheduled approach. STDMA [54] and Viqar and Welch [51] consider GNSS-based scheduling (Global Navigation Satellite System [47]) according to the nodes' geographical position and their trajectories. Autonomous systems cannot depend on GNSS services, because they are not always available, or preferred not to be used, due to their cost. Arbitrarily long failure of signal loss can occur in underground parking lots and road tunnels. We propose a self-stabilizing TDMA algorithm that does not require GNSS accessibility or knowledge about the node trajectories. Rather it considers an underlying self-stabilizing local pulse synchronization, such as [14, 41], which can be used for TDMA alignment, details appear in [41].

When using collision-detection at the receiving-side [11, 32, 46, 50, 54], it is up to the receiving-side to notify the sender about collisions via another round of collision-prone transmissions, say, by using FI (frame information) payload fields that includes T entries, where T is the TDMA frame size. Thus far, the study of FI-based protocols has considered stochastic resolution of message collision via computer network simulation [2, 12, 32, 48, 50, 54]. Simulations are also used for evaluating the heuristics of MS-ALOHA [46] for dealing with contention (timeslot exhaustion) by adjusting

the nodes' individual transmission signal strength and / or carrier sense threshold. We do not consider lengthy frame information (FI) fields, which significantly increase the control information overhead, and yet we provide provable guarantee regarding the convergence time. Further analysis validation of the proposed algorithm via simulations and test-bed implementation can be found in Section 8, and respectively, in [41].

The proposed algorithm does *not* consider collision-detection mechanisms that are based on signal processing or hardware support, as in [15]. Rather, it employs a variation on a well-known strategy for eventually avoiding concurrent transmissions among neighbors. This strategy allows the sending-side to eventually observe the existence of interfering transmissions. Before sending, the sender waits for a random duration while performing a clear channel assessment using basic radio technology (details appear in Section 3).

There are several MAC algorithms that are based on clear channel assessment. A recent example [10] focuses on fair bandwidth allocation for single-hop-distance broadcasting while basing the interference model on discrete graphs. The authors do not consider self-stabilization. This work also uses consider clear channel assessment. However, we employ a strategy of random transmission delay in a way that allows the recipients to notice, in a probabilistic manner, prospective transmissions. We show that after a small number of rounds, the system is able to use the above strategy for allocating the network bandwidth for single-hop-distance broadcasting when basing the interference model on discrete graphs. Further mitigation efforts of transmission pathologies, such as hidden terminal phenomena when unicast are considered, can be taken, e.g., self-stabilizing two-hop-distance vertex coloring [8], equalizing transmission power, coding-based methods [19], to name a few.

An abstract MAC layer was specified for DynWANs in [29]. The authors mention algorithms that can satisfy their specifications. However, they do not consider predictability.

Local algorithms [20, 22] considers both theoretical and practical aspects of MAC algorithms [52, and references therein] and the related problem of clock synchronization, see [33] and references therein. For example, the first partly-asynchronous self-organizing local algorithm for vertex-coloring in wireless ad-hoc networks is presented in [45]. However, this line currently does not consider dynamic networks and predictable bandwidth allocation.

Two examples of self-stabilizing TDMA algorithms are presented in [24, 28]. The algorithms are based on vertex-coloring and the authors consider (non-dynamic) ad-hoc networks. Recomputation and floating output techniques ([16], Section 2.8) are used for converting deterministic local algorithms to self-stabilization in [34]. The authors focus on problems that are related to MAC algorithms. However, deterministic MAC algorithms are known to be inefficient in their bandwidth allocation when the topology of the communication network can change frequently [35]. There are several other proposals related to self-stabilizing MAC algorithms for sensor networks, e.g., [4, 5, 30, 31]; however, none of them consider dynamic networks and their frame control information is quite extensive.

The MAC algorithms in [35–37, 41] *have no proof* that they are self-stabilizing. The authors of [35] present a MAC algorithm that uses convergence from a random starting state (inspired by self-stabilization). In [36, 37, 41], the authors use computer

network simulators for evaluating self- \star MAC algorithms.

1.2 Our contribution

This work proposes a self-stabilizing MAC algorithm that demonstrates rapid convergence without the extensive use of frame control information. Our analysis shows that the algorithm facilitates the satisfaction of severe timing requirements for DynWANs.

We start by considering transient faults and topological changes to the communication network, i.e., demonstrating self-stabilization in Theorem 4.2. We then turn to focus on bounding the algorithm’s convergence time after an arbitrary and unbounded finite sequence of transient faults and changes to the network topology. Theorem 5.1 shows that the expected local convergence time is brief, and bounds it in equation (7). Theorem 6.2 formulates the expected global convergence time in equation (21). Moreover, for a given probability, the global convergence time is calculated in equation (22).

For discussion (Section 8), we point out the algorithm’s ability to facilitate the satisfaction of severe timing requirements for DynWANs. Moreover, the analysis conclusions explain that when allowing merely a small fraction of the bandwidth to be spent on frame control information and when considering any given probability to converge within a bounded time, the proposed algorithm demonstrates a low dependency degree on the number of nodes in the network (as depicted by Fig. 4 and Fig. 6).

We note that some of the proof details appear in the Appendix for the sake of presentation simplicity.

2 Preliminaries

The system consists of a set, P , of N anonymous communicating entities, which we call *nodes*. Denote every node $p_i \in P$ with a unique index, i .

2.1 Synchronization

Each node has fine-grained, real-time clock hardware. We assume that the MAC protocol is invoked periodically by synchronous (*common*) *pulse* that aligns the starting time of the TDMA frame. This can be based, for example, on TDMA alignment algorithms [41], GPS [25] or a distributed pulse synchronization algorithm [14]. The term (*broadcasting*) *timeslot* refers to the period between two consecutive common pulses, t_x and t_{x+1} , such that $t_{x+1} = (t_x \bmod T) + 1$, where T is a predefined constant named the *frame size*. Throughout the paper, we assume that $T \geq 2$. In our pseudo-code, we use the event *timeslot*(t) that is triggered by the common pulse. We assume that the timeslots are aligned as well.

2.2 Communications and interferences

At any instance of time, the ability of any pair of nodes to communicate is defined by the set, $N_i \subseteq P$, of (*direct*) *neighbors* that node $p_i \in P$ can communicate with directly.

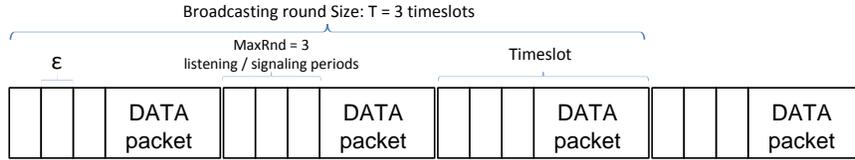


Figure 1: An example of TDMA frame, with three timeslots and three listening/signaling periods of size ϵ (signal exposure time). Each timeslot has a constant number, $MaxRnd = 4$, of *listening/signaling periods* in which beacons can be transmitted. The duration of each listening/signaling period is ϵ (signal exposure time); the period during which a beacon that is sent by node $p_i \in P$ is transmitted and raises the ca received by all neighbors $p_j \in \mathcal{N}_i$. Namely, the period between p_i 's transmission and p_j 's rise of the `carrier_sense()` event.

Wireless transmissions are subject to interferences (collisions). We consider the potential of nodes to interfere with each other's communications. The interference model in this paper is based on discrete graphs.

The set $\mathcal{N}_i \supseteq N_i$ is the set of nodes that may interfere with p_i 's communications when any nonempty subset of them, $I \subseteq \mathcal{N}_i : I \neq \emptyset$, transmit concurrently with p_i . We call \mathcal{N}_i the (*extended*) *neighborhood* of node $p_i \in P$ and $d_i = |\mathcal{N}_i|$ is named the (*extended*) *degree* of node p_i . We assume that at any time, for any pair of nodes, $p_i, p_j \in P$ it holds that $p_j \in \mathcal{N}_i$ implies that $p_i \in \mathcal{N}_j$. Given a particular instance of time, we define the (*interference*) *graph* as $G := (P, E)$, where $E := \cup_{i \in P} \{(p_i, p_j) : p_j \in \mathcal{N}_i\}$ represents the interference relationships among nodes.

2.3 Communication schemes

We consider (basic technology of) radio units that raise the event `carrier_sense()` when they detect that the received energy levels have reached a threshold in which the radio unit is expected to succeed in carrier locking, see [26]. Timeslots allow the transmission of DATA packets using the primitives of `transmit()` and `receive()` after fetching (`MAC_fetch()`) a new packet from the upper layer, and respectively, before delivering (`MAC_deliver()`) the packet to the upper layer. A *beacon* is a short packet that includes no data load, rather the timing of the event `carrier_sense()` is the delivered information [10]. We assume that every node $p_i \in P$ that invokes the operation `transmit()` causes the event `carrier_sense()` to be raised by its neighbors, $p_j \in \mathcal{N}_i$, within the exposure time, ϵ . Before the transmission of the DATA packet in timeslot t , our communication scheme uses beacons for signaling the node's intention to transmit a DATA packet within t , see Fig. 1.

2.4 System settings

We consider the interleaving model [16]. Every node, $p_i \in P$, executes a program that is a sequence of *atomic steps*. The *state* st_i of a node p_i consists of the value of all the variables of the node (including messages in transit for p_i). Variables are associated with

individual node states by using the subscript notation, i.e., var_i is the value of variable var in p_i 's state. The term *configuration* is used for a tuple of the form $(G, \{st_i\}_{i=1}^N)$, where G is the (interference) graph, and $\{st_i\}_{i=1}^N$ are the nodes' states (including the set of all incoming communications). An *execution* (run) $R := (c(0), c(1), \dots)$ is an unbounded sequence of system configurations $c(x)$, such that each configuration $c(x+1)$ (except the initial configuration $c(0)$) is obtained from the preceding configuration $c(x)$ by the execution of steps, $\{a_i(x)\}_{p_i \in P}$, taken by all nodes.

Let τ (task) be a specification (predicate) set and LE a set of all executions that satisfy task τ . Let us consider TDMA-based MAC protocols for which the task τ_{TDMA} requires that every node has its own broadcasting timeslot that is unique within its neighborhood. We note that τ_{TDMA} 's requirements are obviously satisfiable when the ratio between the extended degree and the frame size is less than one, i.e., there is no timeslot exhaustion when $\forall p_i \in P : 1 \lesssim T/d_i$. Therefore, the studied task also deals with timeslot exhaustion by delivering busy channel indications, \perp , to the nodes for which there was no timeslot left. We define LE_{TDMA} to be the set of legal executions, R , for which $\forall p_i \in P : ((s_i \in [0, T-1]) \wedge (p_j \in \mathcal{N}_i) \Rightarrow s_i \neq s_j) \vee (s_i = \perp \Rightarrow \forall t \in [0, T-1] \exists p_j \in \mathcal{N}_i : s_j = t)$ holds in all of R 's configurations.

We say that configuration c_{safe} is *safe* if there is an execution $R \in LE$, such that c_{safe} is R 's starting configuration. Let R be an execution and $c \in R$ its arbitrary starting configuration. We say that R *converges* with respect to τ if within a bounded number of steps from c , the system reaches a *safe* configuration c_{safe} . The *closure* property requires that for any execution, R , that starts from c_{safe} implies that $R \in LE$. An algorithm is said to be *self-stabilizing* if it satisfies both the convergence and the closure properties.

We describe execution R as an unbounded number of concatenated finite sequences of configurations. The finite sequences, $R(x) = (c_0(x), \dots, c_{T-1}(x))$, $x > 0$, is a *broadcasting round* if (1) configuration $c_0(x)$ has a clock value, t , of 0 and immediately follows a configuration in which the clock value is $T-1$, and (2) configuration $c_{T-1}(x)$ has a clock value of $T-1$ and immediately precedes a configuration in which the clock value is 0.

3 Algorithm Description

The proposed MAC algorithm periodically performs clear channel assessments. It uses these assessments when informing each node about the nearby unused timeslots. The nodes use this information for selecting their broadcasting timeslots, assess the success of their broadcasts and reselecting timeslots when needed.

The MAC algorithm in Fig. 2 satisfies the τ_{TDMA} task. During the convergence period several nodes can be assigned to the same timeslot. Namely, we may have $p_i \in P : p_j \in \mathcal{N}_i \wedge s_i = s_j$. The algorithm solves such timeslot allocation conflicts by letting the node p_i and p_j to go through a (listening/signaling) competition before transmitting in its broadcasting timeslot. The competition rules require each node to choose one out of *MaxRnd* listening/signaling period for its broadcasting timeslot, see Fig. 1. This implies that among all the nodes that attempt to broadcast in the same timeslot, the ones that select the earliest listening/signaling period win this broadcasting timeslot and

```

Constants, variables, macros and external functions
2 MaxRnd (n in the proofs) : integer = bound on round number
   s : [0, T-1] ∪ {⊥} = next timeslot to broadcast or null, ⊥
4 signal : boolean = trying to acquire the channel
   unused[0,T-1] : boolean = marking unused timeslots
6 unused_set = { k : unused[k] = true } : unused timeslots, macro
   MAC_fetch()/MAC_deliver() : MAC layer interface
8 transmit/receive/carrier_sense : communication primitives

10 Upon timeslot(t)
   if t = 0 ∧ s = ⊥ then s := select_unused(unused_set)
12 (unused[t], signal) := (true, false) (* remove stale info. *)
   if s ≠ ⊥ ∧ t = s then send(MAC_fetch())
14
   Upon receive(< DATA, m >) do MAC_deliver(< m >)
16
   Function send(m) (* send message m to pi's neighbors *)
18 for ((signal, k) := (true, 0); k := k + 1; k ≤ MaxRnd) do
   if signal then with probability  $\rho(k) = 1 / (\text{MaxRnd} - k)$  do
20   signal := false (* quit the competition *)
   transmit(< BEACON >) (* try acquiring the channel *)
22   wait until the end of competition round (* exposure period alignment *)
   if s ≠ ⊥ then transmit(< DATA, m >) (* send the data packet *)
24
   Upon carrier_sense(t) (* defer transmission during t *)
26 if s = t ∧ signal then s := ⊥ (* mark that the timeslot is not unique *)
   (signal, unused[t]) := (false, false) (* quit the competition *)
28
   Function select_unused(set) (* select an empty timeslot *)
30 if set = ∅ then return ⊥ else return uniform_select(set)

```

Figure 2: Self-stabilizing TDMA-based MAC algorithm, code of node p_i .

access the communication media. Before the winners access their timeslots, they signal to their neighbors that they won via beacon transmission. The signal is sent during their choice of listening/signaling periods, see Fig. 1. When a node receives a beacon, it does not transmit during that timeslot, because it lost this (listening/signaling) competition. Instead, it randomly selects another broadcasting timeslot and competes for it on the next broadcasting round.

In detail, the MAC algorithm in Fig. 2 is invoked at the start of every timeslot, t . When t is the first timeslot, the algorithm tries to allocate the broadcasting timeslot, s_i , to p_i (line 11) by randomly selecting a timeslot for which there is no indication to be used by its neighbors. Later, when the timeslot t becomes p_i 's broadcasting timeslot, s_i , the node attempts to broadcast (by calling the function $\text{send}()$ in line 13). We note that the start of timeslot t also requires the marking of t as an unused timeslot and the removal of stale information (line 12). This indication is changed when the $\text{carrier_sense}(t)$ event is raised (line 27) due to a neighbor transmission. Namely, when the detected energy levels reach a threshold in which the radio unit is expected to succeed in carrier locking, see [26].

When a node attempts to broadcast it uses the (listening/signaling) competition mechanism for deciding when to signal to its neighbors that it is about to transmit a DATA packet. The competition has MaxRnd rounds and it stops as soon as the node transmits a beacon or a neighbor succeeds in signaling earlier (lines 18 to 23). We note that this signaling is handled by the $\text{carrier_sense}(t)$ event (line 27). Moreover, beacons

are not required to carry payloads or any other information that is normally stored in packet headers. They are rather used to invoke the carrier sense event in \mathcal{N}_i .

The carrier sense in timeslot t indicates to each node that it needs to defer from transmission during t (line 25). In particular, it should stop using timeslot t for broadcasting, stop competing and mark t as a used timeslot. Lastly, arriving DATA packets are delivered to the upper layer (line 15).

4 Correctness Proof: Outline and Notation

The proof starts by considering networks that do not change their topology and for which the ratio between the extended node degree and the frame size is less than one, i.e., $\forall p_i \in P : 1 \lesssim T/d_i$.¹ For these settings, we show that the MAC algorithm in Fig. 2 is self-stabilizing with respect to task τ_{TDMA} (sections 9 to 10 of the Appendix), before considering the converge time within a single neighborhood (Section 5) and the entire neighborhood (Section 6). These convergence estimations facilitate the exploration of important properties, such as predictability, and dealing with changes in the network topology of DynWANs (Section 8).

4.1 Proof outline

The exposition of the proof outline refers to Definition 4.1, which delineates the different states at which a node can be in relation to its neighbors. Definition 4.1 groups these states into three categories of *relative states*: (1) Ready to be allocated, when the node state depicts correctly its neighbor states, (2) Obtaining a timeslot, when the node is competing for one, but there is no agreement with its neighbor states, and (3) Allocated to a timeslot, when the node is the only one to be allocated to a particular timeslot in its neighborhood. The correctness proof shows that the MAC algorithm in Fig. 2 implements τ_{TDMA} in a self-stabilizing manner by showing that eventually all nodes are allocated with timeslots, i.e., all nodes are in the relative state Allocated, see Definition 4.1.

Let R be an execution of the MAC algorithm in Fig. 2 and $R(x)$ is the x -th complete broadcasting round of R , where $x > 0$ is an integer. We simplify the presentation by using uppercase notation for the configurations, $c_t^{\text{name}}(x)$, where $t \in [0, T - 1]$ is a timeslot. This notation includes the *name* of the first event to be triggered immediately after configuration c , i.e., $R(x) = (c_0^{\text{timeslot}}(x), \dots, c_{T-1}^{\text{carrier.sense/receive}}(x))$.

Definition 4.1. *We say that node $p_i \in P$ is Ready (to be allocated) to a timeslot in configuration $c_0^{\text{timeslot}}(x)$, if properties (1), (2) and (3) hold for node p_i but Property (4) does not. We say that p_i is Obtaining timeslot s_i in configuration $c_0^{\text{timeslot}}(x)$, if properties (1) to (4) hold for node p_i , but Property (5) does not. We say that node $p_i \in P$ is in Allocated state, with respect to timeslot s_i in configuration $c_0^{\text{timeslot}}(x)$, if properties (1) to (5) hold for node p_i .*

$$\text{signal}_i = \text{false} \tag{1}$$

¹We deal with cases in which $\forall p_i \in P : 1 \lesssim T/d_i$ does not hold in Section 8.

$$(t \in \text{unused}_i \wedge t \neq s_i) \leftrightarrow (\forall p_k \in \mathcal{N}_i : s_k \neq t) \quad (2)$$

$$s_i \neq \perp \vee \text{unused_set}_i \setminus \{s_i\} \neq \emptyset \quad (3)$$

$$s_i \neq \perp \quad (4)$$

$$\forall p_j \in \mathcal{N}_i : ((s_i \neq s_j) \wedge (\text{unused}_j[s_i] = \text{false})) \quad (5)$$

Property (1) implies that node p_i finishes any broadcast attempts within a timeslot. Properties (2) to (3) consider the case in which p_i 's internal state represents correctly the timeslot allocation in its neighborhood. In particular, property (2) means that processor p_i views timeslot t as an unused one if, and only if, it is indeed unused. Property (3) implies that when node p_i is not using any timeslot, there is an unused timeslot at its disposal. Property (4) says that node p_i is using timeslot s_i . Property (5) refers to situations in which p_i 's neighbors are not using p_i 's timeslot during the next broadcasting round.

Starting from an arbitrary configuration, we show that node p_i becomes Ready within two broadcasting rounds (or one complete broadcasting round), see Section 9 of the Appendix. Then, we consider the probability, $\text{OnlyOne}_i(x)$, that a node enters the relative state Allocated from either Ready or Obtaining, see equation (6) (and sections 10 and 12 of the Appendix). Namely, equation (6) considers the probability that node p_i is the *only one* to use its broadcasting timeslot in its neighborhood, where $\rho_k = 1/\text{MaxRnd} = 1/n$ is p_i 's probability to select the k -th listening/signaling period for transmitting its beacon.

$$\text{OnlyOne}_i(x) \geq \sum_{k=1}^n \rho_k \left(1 - \sum_{\ell=1}^k \rho_\ell \right)^{\frac{d_i}{T}} \quad (6)$$

Theorem 4.2 demonstrates self-stabilization.

Theorem 4.2 (Self-Stabilization, the proof appears in Section 11 of the Appendix). *The MAC algorithm in Fig. 2 is self-stabilizing with respect to the task τ_{TDMA} .*

Bounding the convergence time We bound the time it takes the MAC algorithm in Fig. 2 to converge by considering the relative states, Ready, Obtaining, and Allocated, and describe a state machine of a Markovian process. This process is used for bounding the convergence time of a single node (Section 5), and the entire network (Section 6).

In detail, give node $p_i \in P$, its neighborhood, \mathcal{N}_i , we define a random environment of a Markov chain, see Fig. 3. By looking at this random environment, we can focus our analysis on p_i 's relative states while avoiding probability dependencies and considering average probabilities [9]. Suppose that p_i 's environment, e , is known. Theorem 5.1 estimates two bounds on the expectation of probability, $q_i | e$, which is literally the probability q_i given that the environment is e .

In order to do that, we consider a set, \mathcal{R} , of executions of the MAC algorithm, such that each execution $R \in \mathcal{R}$ starts in a configuration, $c \in R$, in which: (I) for any node $p_j \in P$, properties (1), (2) and (3) holds, and (II) node p_i is in the relative state Ready, which implies that (III) eventually, node p_i arrives to the relative state Allocated.

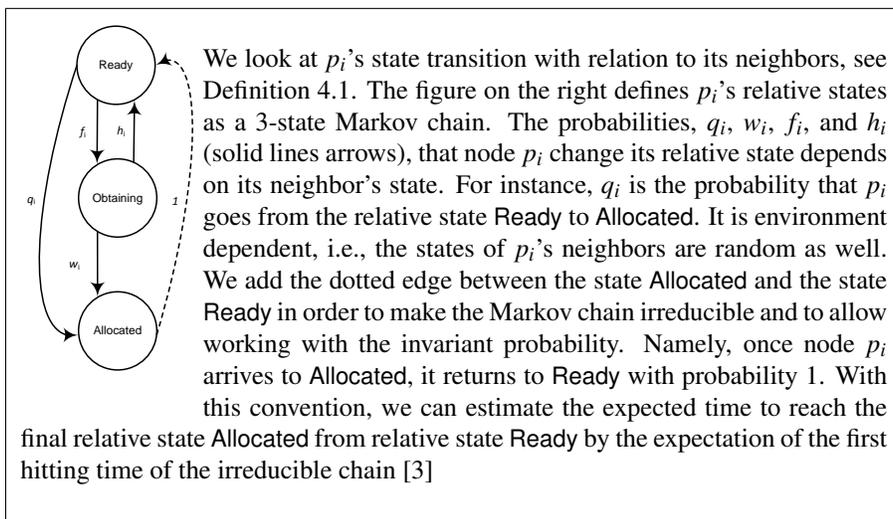


Figure 3: Markov chain describing p_i 's relative state transitions.

With this convention, we can add a probability 1 to transit from the relative state Allocated to Ready, see the dashed line in the state-machine diagram of Fig. 3. This allows us to estimate the expected time to reach the final relative state Allocated from relative state Ready by the expectation of the first hitting time of the irreducible Markov chain [3].

When computing the expected time for node p_i to reach state Allocated within its neighborhood, we see that it is sufficient to consider the lower bound of the probability $OnlyOne_i(x)$ to obtain an upper bound on the expected time to converge, see section 5. Moreover, when considering the network convergence time, i.e., the expected convergence time of all nodes in the network, we see that the most dominant parameter is the mean neighborhood size. We do that by applying the AM-GM (Arithmetic Mean vs Geometric Mean) inequality and bounding the expected network convergence time, see Section 6.

4.2 Notation

Throughout the paper, we denote the states of the Markov chain by $\{X_t\}_{t \geq 0}$, $T_i^+ = \min\{t > 0 \text{ such that } X_t = i\}$ and $E_i(\cdot)$ is the expectation given that we start in relative state i , $E_i(T_i^+) = E(T_i^+ | X_0 = i)$. In this paper, the states 1, 2, and 3 of the Markovian process correspond respectively to states Ready, Obtaining and Allocated, and the time $t = 0, 1, \dots$ corresponds to configuration $c_0^{timeslot}(x+t) \in R(x+t)$, where $R(x)$ is the first complete broadcasting round in R that starts in a configuration, $c_0^{timeslot}(x)$, in which all nodes are in the relative state Ready. For example, $E_3(T_3^+)$ is the expected time to

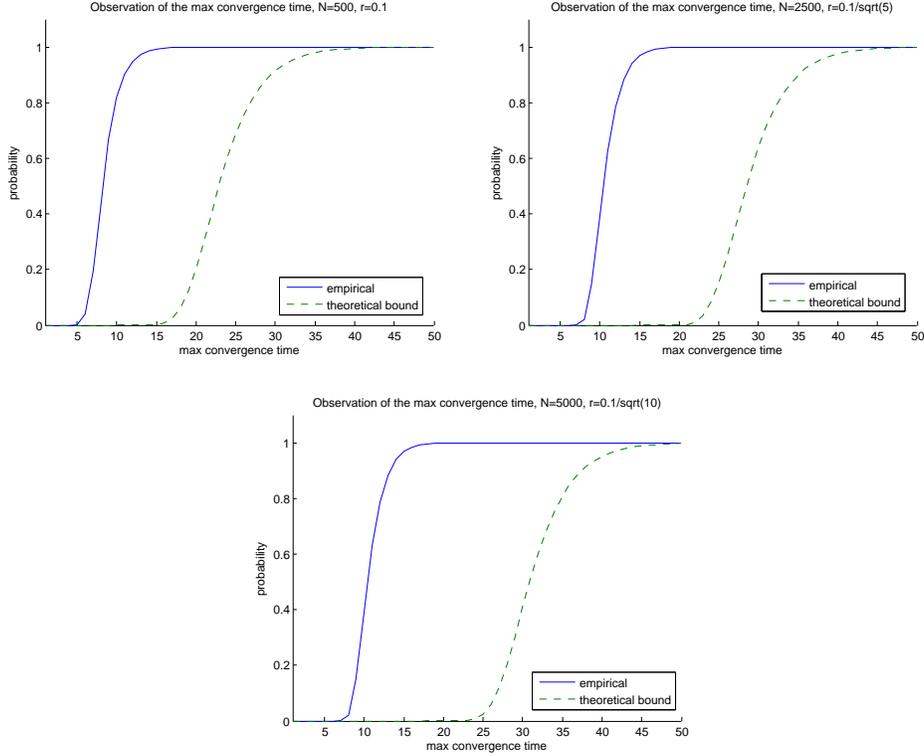


Figure 4: Numerical validation of Theorem 6.2’s bound on the network-wise convergence time. We compare the bound, $P(t_{\max} < k) = (1 - (1 - q)^k)^N$, with the numerical results, which consider random geometric graphs in which the nodes are randomly placed on the unit square. The charts considers $N \in \{500, 2500, 5000\}$ nodes (from left to right). All experiments considered 2 listening/signaling periods, interference range of $0.1/\sqrt{(\frac{N}{500})}$, which result in an average extended degree of 15, $d_i/T = 1$ on average, and $q_i = 1/4$.

reach the Allocated state.

Let $p_i \in P$ be a node for which $s_i \neq \perp \wedge \exists p_j \in \mathcal{N}_i : s_j = s_i$ in configuration $c_0^{\text{timeslot}}(x)$. We define $M_i(x) = \{p_j \in \mathcal{N}_i : s_i = s_j\}$ to be the set of p_i ’s (broadcasting timeslot) *matching* neighbors, which includes all of p_i ’s neighbors that, during broadcasting round $R(x)$, are attempting to broadcast in p_i ’s timeslot. In our proofs, we use n as the number of *listening/signaling periods*, $MaxRnd$.

5 Convergence within a Neighborhood

Theorem 5.1 bounds the expected time, \mathcal{S}_i , for a node to reach the relative state Allocated, and follows from Proposition 5.3 and equation (12). Note that $\mathcal{S}_i \leq 4$ when the

number of listening/signaling periods is $n \geq 2$, and considering saturated situations in which the extended node degree $d_i < T$ is smaller than the TDMA frame size. Namely, the proposed algorithm convergence with a neighborhood is brief.

Theorem 5.1 (Local Convergence). *The expected time, \mathcal{S}_i , for node $p_i \in P$ to reach the relative state Allocated satisfies equation (7), where n is the number of listening/signaling periods, T the TDMA frame size, and d_i is p_i 's extended degree.*

$$\boxed{\mathcal{S}_i \leq \min\left\{\left(\frac{2n}{n-1}\right)^{\frac{d_i}{T}}, \frac{d_i+1}{n} \left(\frac{n}{n-1}\right)^{\frac{d_i}{T}+1}\right\}} \quad (7)$$

We look into the transition probability among relative states by depicting the diagram of Fig. 3 as an homogeneous Markov chain. We estimate the diagram transition probabilities in a way that maximizes the expected time for reaching the diagram's final state, Allocated. It is known that the first hitting time is given by $E_i(T_i^+) = \frac{1}{\pi_i}$, where $\pi = (\pi_1, \pi_2, \pi_3)$ is the invariant probability vector [3]. Let \mathcal{S}_i be the expected time it takes node p_i that starts at the relative state Ready to reach Allocated. It is clear that $\mathcal{S}_i = T_3^+ - 1$, because $T_3^+ - 1$ is the return time of the relative state Allocated. In our case, the transition matrix P is given by equation (8).

$$P = \begin{pmatrix} 1 - f_i - q_i & f_i & q_i \\ h_i & 1 - h_i - w_i & w_i \\ 1 & 0 & 0 \end{pmatrix} \quad (8)$$

The invariant probability vector π satisfying $\pi P = \pi$ is given by equation (9).

$$\pi = \frac{(h_i + w_i, f_i, q_i h_i + q_i w_i + f_i w_i)}{h_i + w_i + f_i + h_i q_i + q_i w_i + f_i w_i} \quad (9)$$

The estimation of the maximal expected time necessary to assign the node p_i to a timeslot requires to compute bounds on the probabilities f_i , h_i , q_i and w_i that maximize equation (10).

$$E_3(T_3^+) = \frac{1}{\pi_3} = \frac{h_i + w_i + f_i + h_i q_i + q_i w_i + f_i w_i}{q_i h_i + q_i w_i + f_i w_i} \quad (10)$$

The expected time for p_i to reach the relative state Allocated is bounded in equation (11).

$$\boxed{\mathcal{S}_i = E_3(T_3^+) - 1 = \frac{h_i + w_i + f_i}{q_i h_i + q_i w_i + f_i w_i}} \quad (11)$$

Equation (7) has a compact and meaningful bound for equation (11). We achieve that by studying the impact of the parameters T and n on the MAC algorithm in Fig. 2. Lemma 5.2 and equation (11) imply equation (12).

$$\boxed{\mathcal{S}_i \leq \frac{h_i + w_i + f_i}{q_i h_i + q_i w_i + f_i q_i} = \frac{1}{q_i}} \quad (12)$$

Lemma 5.2. *Suppose that $n \geq 2$ is the number of listening/signaling periods, see line 2 of the code in Fig. 2. Then $w_i \geq q_i$.*

Proof. Let us consider node $p_i \in P$ that is in relative state Ready. Given that p_i has v_i neighbors that compete for the same timeslot, the probability that p_i gets allocated, $q_i | v_i$, is given by equation (13).

$$q_i | v_i = \sum_{k=1}^{n-1} \rho_k (1 - \rho_1 - \dots - \rho_k)^{v_i} \quad (13)$$

Consider next that p_i is in relative state Obtaining, and thus we know that p_i transmitted during the preceding broadcasting round and transited from relative state Ready to Obtaining. Moreover, p_i is using the same timeslot for the current broadcasting round. The only neighbors of p_i that are using the same timeslot are the neighbors that are also in relative state Obtaining and, have chosen the same listening/signaling period as p_i during the preceding broadcasting round. Let us denote by ℓ_i the number of such neighbors. Given ℓ_i the probability $w_i | \ell_i$ that p_i is allocated to the timeslot is given by equation (14).

$$w_i | \ell_i = \sum_{k=1}^{n-1} \rho_k (1 - \rho_1 - \dots - \rho_k)^{\ell_i} \quad (14)$$

We have that ℓ_i is stochastically dominated by v_i [43], i.e., $E(\ell_i) \leq E(v_i)$. Indeed, v_i is a random variable that counts the number of neighbors that choose the same timeslot as p_i while ℓ_i counts the number of neighbors that choose the same timeslot *and* listening/signaling period as p_i . For $n \geq 2$, ℓ_i 's expected value is smaller than v_i 's expected value. To conclude, we remark that expressions (13) and (14) are the same decreasing function, $f_i \rightarrow \sum_{k=1}^{n-1} \rho_k (1 - \rho_1 - \dots - \rho_k)^{f_i}$, that is evaluated at two different point, v_i and ℓ_i respectively. Moreover, since ℓ_i is stochastically dominated by v_i , equation (15) holds.

$$w_i = E(w_i | \ell_i) \geq E(q_i | v_i) = q_i \quad (15)$$

□

Proposition 5.3 demonstrates equation (16) and leads us toward the proof of Theorem 5.1.

Proposition 5.3. *Let $\rho_i = 1/\text{MaxRnd}$. Equation (16) bounds from below the probability q_i , see Section 12 of the Appendix.*

$$q_i \geq \max \left\{ \left(\frac{n-1}{2n} \right)^{\frac{d_i}{T}}, \frac{1}{\frac{d_i}{T} + 1} \left(1 - \frac{1}{n} \right)^{\frac{d_i}{T} + 1} \right\} \quad (16)$$

The first bound, $\frac{1}{q_i} \leq \left(\frac{2n}{n-1} \right)^{\frac{d_i}{T}}$ (equation (7)), has a simple intuitive interpretation. Let us consider first that two nodes compete for a same timeslot. The two nodes choose independently any of the n listening/signaling periods and there are n^2 different possible outcomes. Among these outcomes n correspond to the situation where the two

nodes choose the same listening/signaling period and there is no winner. We then have $n^2 - n = n(n - 1)$ outcomes that lead to a winner. There is then a probability of $n(n - 1)/n^2 = (n - 1)/n$ that one of the node wins the (listening/signaling) competition. Since the game is symmetric, the probability that p_i wins is $(n - 1)/(2n)$. The fact that we have T timeslots divides the number of competing nodes, d_i , and imply that there are d_i/T competing nodes to the same timeslot. If we interpret the game as a collection of d_i/T independent games, where for each game p_i wins with probability $(n - 1)/(2n)$. Thus, the probability q_i that p_i wins is $(\frac{n-1}{2n})^{\frac{d_i}{T}}$. The inverse of this expression gives the average time for the event to occur and is the bound by equation (7).

6 Network Convergence

We estimate the expected time for the entire network to reach a safe configuration in which all nodes are allocated with timeslots. The estimation is based on the number of nodes that are the earliest to signal in their broadcasting timeslot. These nodes are winners of the (listening/signaling) competition and are allocated to their chosen timeslots. However, counting only these nodes leads to under-estimate the number of allocated nodes, which then results in an over-estimation of the convergence time. Indeed, node $p_i \in P$ might have a neighbor $p_j \in \mathcal{N}_i$ that selects the earliest listening/signaling period in \mathcal{N}_i , but p_j does not transmit because one of its neighbors, $p_k \in \mathcal{N}_j \setminus \mathcal{N}_i$, had transmitted in an earlier listening/signaling period. Our bound consider only p_k while both p_i and p_k transmit, became p_j is inhibited by p_k 's beacon.

Lemma 6.1 shows that the assumption that the nodes are allocated independently of each other's is suitable for bounding the network convergence time, \mathcal{S} . Theorem 6.2 uses Lemma 6.1 for bounding the network convergence time, \mathcal{S} .

In Section 5, we prove a bound on the expected time, \mathcal{S}_i , for a *single* node to be allocated to a timeslot. We observe that the bound depends uniquely on the number of listening/signaling periods, n , as well as the ratio between the extended degree and the frame size, d_i/T . In order to obtain a bound valid for all nodes, we bound this ratio with x/T where x is as defined in Lemma 6.1. We note that the time needed for the allocation of timeslots to *all* the nodes depends on N , the total number of nodes.

In detail, the convergence time estimation considers the (fixed and independent) bound, q_i , for the probability that a node reach the relative state Allocated within a broadcasting round. Then, the convergence time, t , is a random variable with geometric probability, i.e., $P(t = k) = (1 - q)^{k-1}q$. Let us denote t_1, \dots, t_N the time it takes for the nodes p_1, \dots, p_N to respectively reach the relative state Allocated. The convergence time, \mathcal{S} , for all the nodes is given by $\max(\{t_1, \dots, t_N\})$, which depends on N .

Lemma 6.1. *The expected number of nodes, $E(W)$, that win the (listening/signaling) competition after one broadcasting round satisfies equation (17), where $x = \frac{2A}{N}$, T is the number of timeslots, A the number of edges in the interference graph, G , and $N = |P|$ the number of nodes that attempt to access the communication media.*

$$E(W) \geq N \sum_{j=1}^n \rho_j (1 - (\rho_1 + \dots + \rho_j))^{\frac{x}{T}} \quad (17)$$

Proof. The nodes that are allocated to a timeslot can previously be on relative state Ready or Obtaining. The probability of a transition from relative state Obtaining to Allocated is w_i , and, a transition from relative state Ready to Allocated is q_i . As proved in Lemma 5.2, we always have $w_i \geq q_i$. To bound the number of nodes that get allocated during a broadcasting round, we use the lower bound on the probability q_i that a node gets allocated to a timeslot. Moreover, in the computations, we use the AM-GM bound [49], which says that if $\sum b_k = 1$ then $\prod a_k^{b_k} \leq \sum b_k a_k$ and, denote d_i the number of neighbors of node p_i . As proved in Proposition 12.1, since there are T timeslots the number of neighbors of i that choose the same timeslot as i and compete for it is bounded by d_i/T . This lemma is proved by equation (18), where the last line of the expression holds because $\sum_i d_i = 2A$.

$$\begin{aligned}
E(W) &\geq \tag{18} \\
E\left(\sum_{i=1}^N 1_{|p_i \text{ selects the earliest signaling period}}\right) &= \\
\sum_{i=1}^N \left(\rho_1 (1-\rho_1)^{\frac{d_i}{T}} + \dots + \rho_{n-1} \left(1 - \sum_{k=1}^{n-1} \rho_k\right)^{\frac{d_i}{T}}\right) &= \\
\sum_{j=1}^n N \sum_{i=1}^N \frac{1}{N} \rho_j \left(1 - \sum_{k=1}^j \rho_k\right)^{\frac{d_i}{T}} &\geq
\end{aligned}$$

$$\begin{aligned}
N \sum_{j=1}^n \prod_{i=1}^N \rho_j^{\frac{1}{N}} \left(1 - \sum_{k=1}^j \rho_k\right)^{\frac{d_i}{NT}} &= \tag{19} \\
N \sum_{j=1}^n \rho_j \left(1 - \sum_{k=1}^j \rho_k\right)^{\frac{1}{TN} \sum d_i} &= \\
N \sum_{j=1}^n \rho_j \left(1 - \sum_{k=1}^j \rho_k\right)^{\frac{x}{T}} &
\end{aligned}$$

We note that we use the AM-GM bound to reach the 4-th row of equation (18). \square

By arguments similar to the ones used in the proof of Proposition 5.3, we deduce that if N nodes compete, the expected number $E(W)$ of nodes that get allocated to a timeslot is lower bounded in equation (20).

$$E(W) \geq N \max \left\{ \left(\frac{n-1}{2n}\right)^{\frac{x}{T}}, \frac{\left(\frac{n-1}{n}\right)^{\frac{x}{T}+1}}{\frac{x}{T}+1} \right\} \tag{20}$$

Theorem 6.2 bounds the system convergence time. We numerically validate Theorem 6.2, see Fig. 4. Moreover, our experiments showed that the average convergence time of the network is below the upper bound of equation (21).

Theorem 6.2 (Global Convergence). *The expected number of retransmissions is smaller than $\left(\frac{2n}{n-1}\right)^{d/T} - 1$, where $d = \max(\{d_i : p_i \in P\})$. Hence, we have that the expected number of broadcasting rounds, S , that guarantee that all nodes to reach the relative state Allocated satisfies equation (21).*

$$S \leq \left(\frac{2n}{n-1}\right)^{d/T} \quad (21)$$

Moreover, given that there are N nodes in the network and $\alpha \in (0, 1)$, the network convergence time is bounded by equation (22) with probability $1 - \alpha$.

$$k = 1 + \frac{\log(1 - \sqrt[T]{1 - \alpha})}{\log\left(1 - \left(\frac{n-1}{2n}\right)^{\frac{d}{T}}\right)} \quad (22)$$

This means that with probability α all nodes are allocated with timeslots in maximum k broadcasting rounds, see Fig. (6).

Proof. Theorem 5.1 bounds the convergence time of a particular processor, see equation (7). Lemma 6.1, see equation (20) $E(W) \geq N\left(\frac{n-1}{2n}\right)^{x/T}$, proves that this bound is still valid if we replace the term d_i/T with x/T , i.e., we consider the average degree instead of the particular degree of a node. If we replace x/T by $\max\{d_i\}/T$ in expression (20) we obtain a larger bound because $x/T \leq \max\{d_i\}/T$, i.e. $E(W) \geq N\left(\frac{n-1}{2n}\right)^{x/T} \geq N\left(\frac{n-1}{2n}\right)^{\max\{d_i\}/T}$.

The bound $E(W) \geq N\left(\frac{n-1}{2n}\right)^{\max\{d_i\}/T}$ and the discussion in the 1st paragraph of section 6, show that the number of processors that are allocated during a broadcasting round is bounded by the random variable $\sum_{i=1}^N z_i$, where z_i are identically and independently distributed random variables that are 1 with probability $\left(\frac{n-1}{2n}\right)^{\max\{d_i\}/T}$ and 0 with probability $1 - \left(\frac{n-1}{2n}\right)^{\max\{d_i\}/T}$ (the second random variable dominate the first one, see [40]). This means that we lower bound the number of processors that are allocated if we consider that they are allocated independently with probability $\left(\frac{n-1}{2n}\right)^{\max\{d_i\}/T}$.

While the processors get allocated to a timeslot, the parameters d_i and T change because some timeslots are no longer available (T decreases and some nodes are allocated d_i decreases). Actually the ratio becomes $\frac{\max\{d_i\} - h_i}{T - f_i}$, where $h_i \geq f_i$ because if a timeslot is allocated or sensed used by processor p_i then T , the number of available timeslots decreases by 1 and d_i , the number of competing nodes, must decrease at least by one since there must be at least one processor that uses the busy timeslot (there may be multiple that are in state Obtaining). Under these circumstances we always have $\frac{\max\{d_i\}}{T} \geq \frac{\max\{d_i\} - h_i}{T - f_i}$. Thus, we can obtain a lower bound for the expected time to reach the relative state Allocated by assuming that all nodes are allocated independently with probability $x = \left(\frac{n-1}{2n}\right)^{\max\{d_i\}/T}$. We simplify the following arguments by using this definition of x .

To bound the number of broadcasting rounds we consider the following game. The bank pays 1 unit to the nodes that get in state *Allocated* (get allocated to a timeslot), and receives $x/(1-x)$ units per nodes that fails to get in state *Allocated*. The game is fair because in each round the expected gain is $1 \times x - x/(1-x) \times (1-x) = 0$. If we denote by W_i the number of processors that get in state *Allocated* during the i -th broadcasting round and by L_i the number of processors that fail we have that the gain is given by equation 23, where t denotes the total number of rounds.

$$\text{gain} = \sum_{i=1}^t \left(\frac{x}{1-x} L_i - W_i \right) \quad (23)$$

The expected gain is 0 because the game is fair ($E(\text{gain}) = 0$) and $\sum_{i=1}^t W_i = N$ because eventually all the nodes get in state *Allocated* and the bank pays 1 unit for each such processors. If we compute the expectation on both sides of equation (23), we then obtain equation 24.

$$N = \frac{x}{1-x} E \left(\sum_{i=1}^t L_i \right) \quad (24)$$

We observe that $E(\sum_{i=1}^t L_i)$ is the expected total number of retransmissions and $E(\sum_{i=1}^t L_i)/N$ is the average expected number of retransmissions whose value is $(1-x)/x$. replacing x with its expression, we obtain that the average number of retransmission is bounded by $(2n/(n-1))^{\max\{d_i\}/T} - 1$ and, this leads to the bound equation (21).

To prove the second assertion, let t_1, \dots, t_N be the convergence time of nodes $1, \dots, N$, respectively. The random variables, t_i , are bound by random variables with geometric random distribution with expectation of $(2n/(n-1))^{d/T}$, with $d = \max\{d_i : d_i \in P\}$. We require that $t_{\max} = \max\{t_1, \dots, t_N\}$ in order to ensure that all nodes are allocated with timeslots. The fact that the random variables, t_i , are independent and identically distributed, implies equation (25), where t is a random geometrical random variable, i.e., $\Pr(t = k') = (1-q)^{k'-1}q$ and $\Pr(t \geq k') = (1-q)^{k'-1}$.

$$\begin{aligned} \Pr(t_{\max} \leq k') &= P(t_1 \leq k', \dots, t_N \leq k') = \\ &\Pr(t_1 \leq k') \cdot \dots \cdot P(t_N \leq k') = P(t \leq k')^N \end{aligned} \quad (25)$$

Which $t_{\max} \leq k'$ satisfies equation (26) with probability α ?

$$\begin{aligned} \Pr(t_{\max} < k') &= \Pr(t < k')^N = \\ &\left(1 - (1-q)^{k'-1} \right)^N \geq 1 - \alpha \end{aligned} \quad (26)$$

By solving equation (26), we observe that equation (26) is satisfied for any $k' \geq k$, where k satisfies equation (22). This proves that, with probability $1 - \alpha$, the network convergence time is bounded by equation (22). \square

```

Additional constants and variables
2  $CW_{start}$  and  $CW_{end}$  : backoff parameters
    $count$  : statically allocated variable that counts the backoff steps
4
Function select_unused( $set$ )
6 let  $rtn\_val = \perp$  // indicate busy channel (default return value)
   if  $count \leq 0$  then  $count \leftarrow uniform\_select([CW_{start}, CW_{end}])$ 
8    $count \leftarrow count - |set|$ 
   if  $count \leq 0$  then ( $count, rtn\_val$ )  $\leftarrow (0, uniform\_select(set))$ 
10 return  $rtn\_val$ 

```

Figure 5: select_unused() with TDMA-based back-off

7 Implementation

Existing MAC protocols offer mechanisms for dealing with contention (timeslot exhaustion) via policies for administering message priority, such as [42]. In particular, the IEEE 802.11p standard considers four priorities and techniques for facilitating their policy implementation. We explain similar techniques that can facilitate the needed mechanisms.

7.1 Prioritized listening/signaling periods

We partition the sequence of listening periods, $[0, MaxRnd)$, into $MaxPrt$ subsequences, $[0, MaxRnd_0), \dots, [MaxRnd_{MaxPrt-2}, MaxRnd_{MaxPrt-1})$, where $[MaxRnd_{k-1}, MaxRnd_k)$ is used only for the k -th priority. E.g., suppose that there are six listening/signaling periods, and that nodes with the highest priority may use the first three listening/signaling periods, $[0, 2]$, and nodes with the lowest priority may use the last three, $[3, 5]$. In the case of two neighbors with different listening period parameters, say $[0, 2]$ and $[3, 5]$, that attempt to acquire the same broadcasting timeslot, the highest priority node always attempts to broadcast before the lowest priority one.

7.2 TDMA-based back-off

Let us consider two back-off parameters, CW_{start} and CW_{end} , that refer to the maximal and minimal values of the contention window. Before selecting an unused timeslot, the procedure counts a random number of unused ones. Fig. 5 presents an implementation of the select_unused() function that facilitates back-off strategies as an alternative to the implementation presented in line 29 of Fig. 2.

The statically allocated variable $count$ records the number of backoff steps that node p_i takes until it reaches the zero value. Whenever the function select_unused() is invoked with $count_i = 0$, node p_i assigns to $count_i$ a random integer from $[CW_{start}, CW_{end}]$ (cf. line 7). Whenever the value of $count_i$ is not greater than the number of unused timeslots, the returned timeslot is selected uniformly at random (cf. lines 8 to 9). Otherwise, a \perp -value is returned after deducting the number of unused timeslots during the previous broadcasting round (cf. lines 6 and 10).

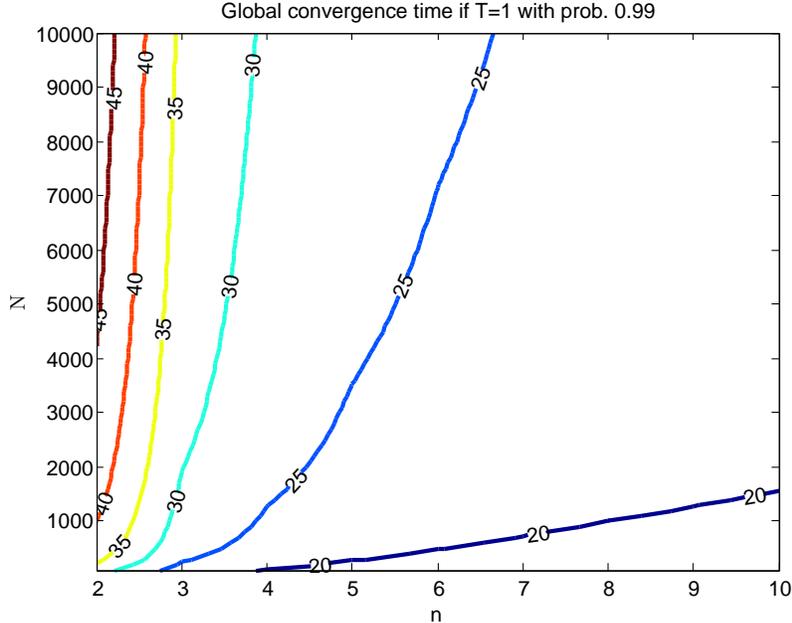


Figure 6: Contour plot of equation (22) for $s = d/T = 1$. *Contour charts* [13] present two parameter functions, e.g., the convergence time function, $k(n, N)$ presented in equation (22). Contour lines in Fig. 6 connect values of $k(n, N)$ that are the same (see the text tags along the line). When N nodes attempt to access the medium, the convergence time, S (cf. the contour lines), is stable in the presence of a growing number, n , of listening/signaling periods.

8 Discussion

Thus far, both schedule-based and non-schedule-based MAC algorithms could not consider timing requirements within a provably short recovery period that follows (arbitrary) transient faults and network topology changes. This work proposes the first self-stabilizing TDMA algorithm for DynWANs that has a provably short convergence period. Thus, the proposed algorithm possesses a greater predictability degree, whilst maintaining low communication delays and high throughput.

In this discussion, we would like to point out the algorithm's ability to facilitate the satisfaction of severe timing requirements for DynWANs by numerically validating Theorem 6.2. As a case study, we show that, for the considered settings of Fig. 4, the global convergence time is brief and definitive. Fig. 6 shows that when allowing merely a small fraction of the bandwidth to be spent on frame control information, say three listening/signaling periods, and when considering 99% probability to converge within a couple of dozen TDMA frames, the proposed algorithm demonstrates a

low dependency degree on the number of nodes in the network even when considering 10,000 nodes.

We have implemented the proposed algorithm, extensively validated our analysis via computer simulation, and tested it on a platform with more than two dozen nodes [41]. These results indeed validate that the proposed algorithm can indeed facilitate the implementation of MAC protocols that guarantee satisfying these severe timing requirements.

The costs associated with predictable communications, say, using cellular base-stations, motivate the adoption of new networking technologies, such as MANETs and VANETs. In the context of these technologies, we expect that the proposed algorithm will contribute to the development of MAC protocols with a higher predictability degree.

Acknowledgments We thank Thomas Petig for helping with improving the presentation.

References

- [1] Norman Abramson. Development of the ALOHANET. *Info. Theory, IEEE Trans. on*, 31(2):119–123, 1985.
- [2] Fabrizio Abrate, Andrea Vesco, and Riccardo Scopigno. An analytical packet error rate model for wave receivers. In *VTC Fall*, pages 1–5. IEEE, 2011.
- [3] David Aldous and Jim Fill. Reversible markov chain and random walks on graph. unpublished, available at <http://www.stat.berkeley.edu/~aldous/RWG/book.html>, 1999.
- [4] M. Arumugam and S.S. Kulkarni. Self-stabilizing deterministic time division multiple access for sensor networks. *AIAA Journal of Aerospace Computing, Info., and Comm. (JACIC)*, 3:403–419, 2006.
- [5] Mahesh Arumugam and Sandeep S. Kulkarni. Self-stabilizing deterministic TDMA for sensor networks. In Goutam Chakraborty, editor, *2nd Inter. Conf. Distributed Computing and Internet Technology (ICDCIT)*, volume 3816 of *LNCS*, pages 69–81. Springer, 2005.
- [6] Katrin Bilstrup, Elisabeth Uhlemann, Erik G. Ström, and Urban Bilstrup. Evaluation of the IEEE 802.11p MAC method for vehicle-to-vehicle communication. In *VTC Fall*, pages 1–5. IEEE, 2008.
- [7] Katrin Bilstrup, Elisabeth Uhlemann, Erik G. Ström, and Urban Bilstrup. On the ability of the 802.11p MAC method and STDMA to support real-time vehicle-to-vehicle communication. *EURASIP Journal on Wireless Comm. & Net.*, 2009:1–13, 2009.

- [8] Jean R. S. Blair and Fredrik Manne. An efficient self-stabilizing distance-2 coloring algorithm. In Shay Kutten and Janez Zerovnik, editors, *SIROCCO*, volume 5869 of *Lecture Notes in Computer Science*, pages 237–251. Springer, 2009.
- [9] Robert Cogburn. The ergodic theory of markov chains in random environments. *Probability Theory and Related Fields (Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete)*, 66(1):109–128, 1984.
- [10] Alejandro Cornejo and Fabian Kuhn. Deploying wireless networks with beeps. In Nancy A. Lynch and Alexander A. Shvartsman, editors, *24th Inter. Symp. on Distributed Computing (DISC'10)*, volume 6343 of *LNCS*, pages 148–162. Springer, 2010.
- [11] Hector Agustin Cozzetti and Riccardo Scopigno. Rr-aloah+: A slotted and distributed mac protocol for vehicular communications. In *Vehicular Networking Conference (VNC), 2009 IEEE*, pages 1–8, Oct. 2009.
- [12] Hector Agustin Cozzetti, Riccardo Scopigno, Luca Casone, and Giuseppe Barba. Comparative analysis of ieee 802.11p and ms-aloah in vanet scenarios. In Markus Kirchberg, Patrick C. K. Hung, Barbara Carminati, Chi-Hung Chi, Rajaraman Kanagasabai, Emanuele Della Valle, Kun-Chan Lan, and Ling-Jyh Chen, editors, *APSCC*, pages 64–69. IEEE, 2009.
- [13] AM Crocker, WL Godson, and CM Penner. Frontal contour charts. *Journal of the Atmospheric Sciences*, 4(3), 1947.
- [14] Ariel Daliot, Danny Dolev, and Hanna Parnas. Self-stabilizing pulse synchronization inspired by biological pacemaker networks. In Shing-Tsaan Huang and Ted Herman, editors, *6th Inter. Symp. on Self-Stabilizing Systems (SSS)*, volume 2704 of *LNCS*, pages 32–48. Springer, 2003.
- [15] Murat Demirbas and Muzammil Hussain. A MAC layer protocol for priority-based reliable multicast in wireless ad hoc networks. In *3rd Inter. Conf. on Broadband Comm., Net., and Systems (BROADNETS)*. IEEE, 2006.
- [16] Shlomi Dolev. *Self-Stabilization*. MIT Press, 2000.
- [17] Shlomi Dolev and Ted Herman. Superstabilizing protocols for dynamic distributed systems. *Chicago J. Theor. Comput. Sci.*, 1997.
- [18] Shlomi Dolev, Amos Israeli, and Shlomo Moran. Analyzing expected time by scheduler-luck games. *IEEE Trans. Software Eng.*, 21(5):429–439, 1995.
- [19] Shyamnath Gollakota and Dina Katabi. Zigzag decoding: combating hidden terminals in wireless networks. In Victor Bahl, David Wetherall, Stefan Savage, and Ion Stoica, editors, *SIGCOMM*, pages 159–170. ACM, 2008.
- [20] Olga Goussevskaia, Roger Wattenhofer, Magnús M. Halldórsson, and Emo Welzl. Capacity of arbitrary wireless networks. In *INFOCOM*, pages 1872–1880. IEEE, 2009.

- [21] Martin Haenggi. Outage, local throughput, and capacity of random wireless networks. *Trans. Wireless. Comm.*, 8(8):4350–4359, 2009.
- [22] Magnús M. Halldórsson and Roger Wattenhofer. Wireless comm. is in apx. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikolettseas, and Wolfgang Thomas, editors, *ICALP (1)*, volume 5555 of *LNCS*, pages 525–536. Springer, 2009.
- [23] H. Hartenstein and K. Laberteaux. *VANET: Vehicular Applications and Inter-Networking Technologies*. Wiley, 2010.
- [24] Ted Herman and Sébastien Tixeuil. A distributed TDMA slot assignment algorithm for wireless sensor networks. In *5th Inter. Workshop on Algo. Aspects of Wireless Sensor Net. (ALGOSENSORS)*, volume 3121 of *LNCS*, pages 45–58. Springer, 2004.
- [25] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global positioning System. Theory and Practice*. Springer-Verlag, 1993.
- [26] Kyle Jamieson, Bret Hull, Allen K. Miu, and Hari Balakrishnan. Understanding the real-world performance of carrier sense. In *ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis (E-WIND)*, Philadelphia, PA, August 2005.
- [27] Johan Ludwig William Valdemar Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1):175–193, 1906.
- [28] Arshad Jhumka and Sandeep S. Kulkarni. On the design of mobility-tolerant TDMA-based media access control (MAC) protocol for mobile sensor networks. In Tomasz Janowski and Hrushikesh Mohanty, editors, *ICDCIT*, volume 4882 of *LNCS*, pages 42–53. Springer, 2007.
- [29] Fabian Kuhn, Nancy A. Lynch, and Calvin C. Newport. The abstract MAC layer. In Idit Keidar, editor, *23rd Inter. Symp. on Distributed Computing (DISC)*, volume 5805 of *LNCS*, pages 48–62. Springer, 2009.
- [30] Sandeep S. Kulkarni and Mahesh Umamaheswaran Arumugam. *Sensor Network Operations*, chapter SS-TDMA: A self-stabilizing MAC for sensor networks. IEEE Press, 2006.
- [31] Andreas Lagemann, Jörg Nolte, Christoph Weyer, and Volker Turau. Mission statement: Applying self-stabilization to wireless sensor networks. In *8th GI/ITG KuVS Fachgespräch “Drahtlose Sensornetze” (FGSN)*, pages 47–49, August 2009.
- [32] M. Lenoble, K. Ito, Y. Tadokoro, M. Takanashi, and K. Sanda. Header reduction to increase the throughput in decentralized TDMA-based vehicular networks. In *Vehicular Networking Conference (VNC), 2009 IEEE*, pages 1–4. IEEE, 2009.

- [33] Christoph Lenzen, Thomas Locher, Philipp Sommer, and Roger Wattenhofer. Clock Synchronization: Open Problems in Theory and Practice. In *36th Inter. Conf. on Current Trends in Theory and Practice of Computer Science (SOFSEM), Āpindleruv Mlŷn, Czech Republic*, January 2010.
- [34] Christoph Lenzen, Jukka Suomela, and Roger Wattenhofer. Local algorithms: Self-stabilization on speed. In Rachid Guerraoui and Franck Petit, editors, *11th Inter. Symp. on Stabilization, Safety, and Security of Distributed Systems (SSS)*, volume 5873 of *LNCS*, pages 17–34. Springer, 2009.
- [35] Pierre Leone, Marina Papatriantafilou, and Elad Michael Schiller. Relocation analysis of stabilizing MAC algorithms for large-scale mobile ad hoc networks. In *5th Inter. Workshop on Algo. Aspects of Wireless Sensor Net. (ALGOSENSORS)*, pages 203–217, 2009.
- [36] Pierre Leone, Marina Papatriantafilou, Elad Michael Schiller, and Gongxi Zhu. Analyzing protocols for media access control in large-scale mobile ad hoc networks. In *Workshop on Self-Organising Wireless Sensor and Comm. Net. (Somsed)*, October 2009. Hamburg, Germany.
- [37] Pierre Leone, Marina Papatriantafilou, Elad Michael Schiller, and Gongxi Zhu. Chameleon-mac: Adaptive and self- \star algorithms for media access control in mobile ad hoc networks. In Shlomi Dolev, Jorge Arturo Cobb, Michael J. Fischer, and Moti Yung, editors, *12th Inter. Symp. on Stabilization, Safety, and Security of Distributed Systems (SSS'10)*, volume 6366 of *LNCS*, pages 468–488. Springer, 2010.
- [38] Pierre Leone and Elad Michael Schiller. Self-stabilizing tdma algorithms for dynamic wireless ad-hoc networks. In Amotz Bar-Noy and Magnŷs M. Halldŷrsson, editors, *ALGOSENSORS*, Lecture Notes in Computer Science. Springer, 2012. (to appear).
- [39] Pierre Leone and Elad Michael Schiller. Self-stabilizing tdma algorithms for dynamic wireless ad-hoc networks. *CoRR*, abs/1210.3061, 2012.
- [40] Torgny Lindvall. *Lectures on the Coupling Method*. Dover Publications, Inc., 1992.
- [41] Mohamed Mustafa, Marina Papatriantafilou, Elad Michael Schiller, Amir Tohidi, and Philippas Tsigas. Autonomous TDMA alignment for VANETs. In *IEEE 76th Vehicular Technology Conference (VTC'12-Fall)*, 2012.
- [42] Raphael Rom and Fouad A. Tobagi. Message-based priority functions in local multiaccess communication systems. *Computer Networks*, 5:273–286, 1981.
- [43] Sheldon M. Ross. *Stochastic Processes*. John Wiley & Sons, Inc., 1996.
- [44] William G. Schmidt. Satellite time-division multiple access systems: Past, present and future. *TeleComm.*, 7:21–24, 1974.

- [45] Johannes Schneider and Roger Wattenhofer. Coloring unstructured wireless multi-hop networks. In Srikanta Tirthapura and Lorenzo Alvisi, editors, *28th Annual ACM Symp. on Principles of Distributed Computing (PODC)*, pages 210–219. ACM, 2009.
- [46] Riccardo Scopigno and Hector Agustin Cozzetti. Mobile slotted aloha for VANETs. In *VTC Fall*, pages 1 – 5. IEEE, 2009.
- [47] Riccardo Scopigno and Hector Agustin Cozzetti. GNSS synchronization in VANETs. In Khalidoun Al Agha, Mohamad Badra, and Gregory B. Newby, editors, *NTMS*, pages 1–5. IEEE, 2009.
- [48] Riccardo Scopigno and Hector Agustin Cozzetti. Evaluation of time-space efficiency in CSMA/CA and slotted VANETs. In *VTC Fall*, pages 1–5. IEEE, 2010.
- [49] J.M. Steele. *The Cauchy-Schwartz Master Class*. Cambridge University Press, 2004.
- [50] Yukihiro Tadokoro, Kenji Ito, Junji Imai, Noriyoshi Suzuki, and Nobuo Itoh. Advanced transmission cycle control scheme for autonomous decentralized TDMA protocol in safe driving support systems. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 1062 –1067, june 2008.
- [51] Saira Viqar and Jennifer L. Welch. Deterministic collision free communication despite continuous motion. In *5th Inter. Workshop on Algo. Aspects of Wireless Sensor Net. (ALGOSENSORS)*, pages 218–229, 2009.
- [52] Roger Wattenhofer. Theory Meets Practice, It’s about Time! In *36th Inter. Conf. on Current Trends in Theory and Practice of Computer Science (SOFSEM), Špindleruv Mlýn, Czech Republic*, January 2010.
- [53] Wei Ye, John S. Heidemann, and Deborah Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *INFOCOM*, 2002.
- [54] Fan Yu and Subir Biswas. Self-configuring TDMA protocols for enhancing vehicle safety with dsrc based vehicle-to-vehicle communications. *Selected Areas in Communications, IEEE Journal on*, 25(8):1526 –1537, oct. 2007.

Appendix

The proof of Theorem 4.2 uses the propositions in sections 9 and 10.

9 Properties (1) to (3)

Propositions 9.1, 9.2 and 9.3 imply that properties (1), (2), and respectively, (3) hold within two broadcasting rounds (or one complete broadcasting round). Let R be an execution of the MAC algorithm in Fig. 2, $x > 0$ an integer, and $c_0^{\text{timeslot}}(x)$ the first configuration in a complete broadcasting round $R(x) = (c_0^{\text{timeslot}}(x), \dots, c_{T-1}^{\text{carrier_sense/receive}}(x))$. We note that $c_0^{\text{timeslot}}(x)$ follows an arbitrary starting configuration.

Proposition 9.1 shows that, within a broadcasting round from $c_0^{\text{timeslot}}(x)$, Property (1) holds.

Proposition 9.1. *In $c_0^{\text{timeslot}}(x+1)$, it holds that $\text{signal}_i = \text{false}$.*

Proof. The value of signal_i is updated in line 18 (assigned to true) and in lines 12, 20, and 27 (assigned to false). Let us look into these assignments.

In every timeslot, the value false is assigned to signal_i (cf. line 12). Suppose that the function $\text{send}()$ is called, and thus, true is assigned to signal_i (line 18). We proposition that before returning from the function $\text{send}()$ and after true is assigned to signal_i (line 18), node p_i must assign false to signal_i either in line 20 or 27. To see that, let us look at lines 18 and 19. Eventually either $\text{signal}_i = \text{false}$ (because of an assignment in line 27) or $\rho(k) = \text{true}$ (line 19) holds (note the condition when $k = \text{MaxRnd}$). The latter case implies the execution of line 20. \square

Proposition 9.2 shows that, within a broadcasting round from $c_0^{\text{timeslot}}(x)$, Property (2) holds.

Proposition 9.2. $(\exists t \in \text{unused_set}_i \setminus \{s_i\}) \leftrightarrow (\nexists p_k \in \mathcal{N}_i : s_k = t)$ in $c_0^{\text{timeslot}}(x+1)$.

Proof. Recall that $\text{unused_set}_i = \{k : \text{unused}_i[k] = \text{true}\}$ (see line 6) and that the proposition statement does not consider the cases in which: (1) $s_i = s_k$ (because $t \neq s_i$) in $c_0^{\text{timeslot}}(x+1)$, or (2) There exists a configuration $c \in R(x)$, such that $s_k \neq \perp$ in c and $s_k = \perp$ in $c_0^{\text{timeslot}}(x+1)$ (because by unused_set 's definition, \perp is never in unused_set_i).

We note that in every broadcasting round, node $p_k \in P$ at most once: (1) Allocates the broadcasting timeslot s_k (when $t_k = 0$, see line 11), (2) Transmits a packet (when $t_k = s_k$, see line 13), and (3) Deallocates the broadcasting timeslot s_k (by assigning \perp to s_k when $t_k = s_k$ and the carrier_sense(t) event is raised, see line 26). Moreover, node p_i updates $\text{unused}_i[t]$ only in lines 12 (true) and 27 (false), when p_i removes stale information just before timeslot t , and respectively, when the event carrier_sense(t) is raised.

Line 12 is executed at the start of every timeslot, whereas line 27 is executed after, and only when the event carrier_sense(t) is raised. The event carrier_sense(t) is raised after, and only when the node $p_k \in \mathcal{N}_i$ transmits in timeslot t . In other words, none of p_i 's neighbors, $p_k \in \mathcal{N}_i$, that transmits in timeslot $s_k = t$, can avoid causing the event carrier_sense(t) to be raised, and timeslot t to be included in $\text{unused_set}_i \setminus \{s_i\}$. \square

Proposition 9.3 shows that, within a broadcasting round from $c_0^{\text{timeslot}}(x)$, Property (3) holds.

Proposition 9.3. $(s_i \neq \perp) \vee (\text{unused_set}_i \setminus \{s_i\} \neq \emptyset)$ holds in $c_0^{\text{timeslot}}(x+1)$.

Proof. If $s_i \neq \perp$ in $c_0^{\text{timeslot}}(x+1)$, we are done. Let us suppose that $s_i = \perp$ in $c_0^{\text{timeslot}}(x+1)$ and show that $\text{unused_set}_i \setminus \{s_i\} \neq \emptyset$ in $c_0^{\text{timeslot}}(x+1)$.

Let us assume, in the way of proof by contradiction that, $\text{unused_set}_i \setminus \{s_i\} = \emptyset$ and show that $d_i/T > 1$, i.e., a contradiction with the assumption that $\forall p_i \in P : d_i/T \lesssim 1$.

Recall that $\text{unused_set}_i = \{k : \text{unused}_i[k] = \text{true}\} \subseteq [0, T-1]$ (see line 6). Therefore, the assumption that $s_i = \perp$ implies that $\text{unused_set}_i = \text{unused_set}_i \setminus \{s_i\} \subseteq [0, T-1]$, because by unused_set 's definition, \perp is never in unused_set_i .

By Proposition 9.2, we can say that $\forall t \in [0, T-1] : (\nexists t \in \text{unused_set}_i) \leftrightarrow (\exists p_k \in \mathcal{N}_i : s_k = t)$. Since $\text{unused_set}_i \subseteq [0, T-1]$, we can write $[0, T-1] \setminus \text{unused_set}_i \subseteq \{s_k \in [0, T-1] : p_k \in \mathcal{N}_i\}$. By the fact that $\text{unused_set}_i = \emptyset$, we have that $T \leq |\{s_k \in [0, T-1] : p_k \in \mathcal{N}_i\}|$. Since $d_i = |\mathcal{N}_i|$ (by definition), we have that $|\{s_k \in [0, T-1] : p_k \in \mathcal{N}_i\}| \leq d_i$, which implies $T \leq d_i$: a contradiction with the assumption that $d_i/T \lesssim 1$. \square

10 Properties (4) to (5)

Section 9 of this Appendix shows that, starting from an arbitrary configuration, node $p_i \in P$ enters the relative state Ready within two broadcasting rounds. This section shows considers the probability for p_i to enter the relative states Obtaining and Allocated.

Let $x > 0$ and R be an execution of the MAC algorithm in Fig. 2. Suppose that $c_0^{\text{timeslot}}(x)$ is the first configuration in a complete broadcasting round $R(x)$ for which properties (1) to (3) hold in configuration $c_0^{\text{timeslot}}(x)$ with respect to node $p_i \in P$, i.e., p_i is in relative state Ready, Obtaining or Allocated. Propositions 10.1, 10.2 and 10.3 show that there is a nonzero probability that node p_i enters the relative state Allocated from either Ready or Obtaining in configuration $c_0^{\text{timeslot}}(x+1)$.

Proposition 10.1 shows that p_i attempts to broadcast once in every round.

Proposition 10.1. During broadcasting round $R(x)$, p_i executes line 13 and calls the function $\text{send}()$.

Proof. If $s_i \neq \perp$ in $c_0^{\text{timeslot}}(x)$, we are done by lines 11 and 13. Let us consider the case of $s_i = \perp$ in $c_0^{\text{timeslot}}(x)$. By Property (4), $\text{unused_set}_i \neq \emptyset$ and thus when line 11 is executed, the function $\text{select_unused}()$ returns a non- \perp element from unused_set_i and $s_i \neq \perp$ when executing line 13. \square

Propositions 10.2 and 10.3 consider the set $M_i(x+1) = \{p_k \in \mathcal{N}_i : s_k = t'\}$ and the number $m_i = |M_i(x+1)|$ of p_i 's neighbors that attempt to broadcast during p_i 's timeslot, t' , of broadcasting round $R(x)$.

Let ρ_j be the probability for p_i to transmit in the j -th listening/signaling period of timeslot t' (cf. line 19). This paper considers the concrete transmission probability $\rho_i = 1/\text{MaxRnd}$. We motivate our implementation choice of the transmission probability, ρ_i ,

Defining optimal transmission probabilities for any choices of T, n, d_i is not possible. We choose to consider and look for optimal choices when $d_i \simeq T$ (the 'hard' case) and make a case for a uniform probability $\rho_i = \frac{1}{n} : i \in [1, n]$. Let us consider node $p_i \in P$ that competes, together with $k - 1$ other neighbors, for the same unique timeslot. The probability that node p_i wins the (listening/signaling) competition is $\rho_1(1 - \rho_1)^{k-1}$, where ρ_1 is the probability of choosing the first listening/signaling period. The value $\rho_1 = \frac{1}{k}$ maximizes this probability. In the more general case where there is more than one timeslot, we consider a strategy that aims at guessing the number, k , of competing neighbors, which the optimal probability of transmission depends on. During the first listening/signaling period, the strategy considers the case in which there are $n = \text{MaxRnd}$ signaling nodes, and thus, the transmission probability is $1/\text{MaxRnd}$, where $\text{MaxRnd} \simeq T$. During the second listening/signaling period, the strategy considers the case in which there are $\text{MaxRnd} - 1$ neighbors, and thus, the transmission probability is $1/(\text{MaxRnd} - 1)$, and so on. This *sequential* selection of the listening/signaling period leads to a uniform choice of a listening/signaling neighbor. The above strategy is driven by a heuristic in which nodes signal with probability that is optimal for the case of $n \simeq T$, and thus, it depends on the number of competing neighbors.

Figure 7: Transition probability, ρ_i , for listening/signaling periods (line 19 in Fig. 2)

in Fig. 7. Note that the *sequential* selection of the broadcasting rounds with probability $1/(\text{MaxRnd} - k + 1)$ leads to the uniform selection $\rho_k = 1/\text{MaxRnd}$.

Proposition 10.2 considers p_i 's chances to be the only one to transmit in its neighborhood.

Proposition 10.2. *There is a nonzero probability, $\text{OnlyOne}_i(x)$ (cf. equation (27)), that only node p_i transmits in its broadcasting timeslot, t' , of broadcasting round $R(x)$.*

$$\begin{aligned} \text{OnlyOne}_i(x) \big|_{m_i > 0} &= \rho_1(1 - \rho_1)^{m_i} + \rho_2(1 - \rho_1 - \rho_2)^{m_i} \\ &+ \dots + \rho_{n-1}(1 - \sum_{\ell=1}^{n-1} \rho_\ell)^{m_i} \end{aligned} \quad (27)$$

Proof. We show that there is a nonzero probability that only node p_i transmits in its broadcasting timeslot, t' , of broadcasting round $R(x)$. Let us look at p_i and the nodes in $M_i(x)$ while they attempt to broadcast in the steps $a_i^{\text{timeslot}, t'}(x)$ and $a_k^{\text{timeslot}, t'}(x)_{|k \in M_i(x)}$. All of these steps include the execution of line 19, viz., each node chooses to transmit in listening/signaling period $\ell \in [0, \text{MaxRnd}]$ with probability $\rho_\ell = 1/(\text{MaxRnd} - \ell)$. Therefore, for any $\text{MaxRnd} > 0$, there is a nonzero probability, $\text{OnlyOne}_i(x)$ that, during timeslot t' , node p_i transmits in the listening/signaling period $a \in \text{MaxRnd}$ and no node in $M_i(x)$ transmits in round a (or in an earlier one).

We note that the fact that p_i transmits first during timeslot t' implies that it is the only to transmit during t' . This is because once p_i transmits a beacon in step

$a_i^{\text{timeslot}, t'}(x)$ (which includes the execution of line 21) node $p_j \in \mathcal{N}_i \supseteq M_i(x)$ raises the event $\text{carrier_sense}(t')$ immediately after $a_i^{\text{timeslot}, t'}(x)$. Thus, $\forall p_j \in M_i(x)$ we have that immediately after step $a_i^{\text{timeslot}, t'}(x)$, node p_j takes step $a_j^{\text{carrier.sense}, t'}(x)$, which includes the execution of lines 26 and 27 that assigns \perp to s_j and false to signal_j . Thus, p_j leaves the (listening/signaling) competition for timeslot t' (see line 18) and does not transmit its DATA packet (see line 23).

We now turn to calculate $\text{OnlyOne}_i(x)$. Let the variable $m_i = |M_i(x)|$ denote the number of nodes that select the same timeslot as p_i in configuration $c_0^{\text{timeslot: } s \neq \perp}(x)$. The value of $\text{OnlyOne}_i(x)$ depends on the value of m_i and we denote this dependence with the notation $q(i) \mid_{m_i}$ (conditional probability). It means the value of $\text{OnlyOne}_i(x)$ depends on the value of m_i . The value of $\text{OnlyOne}_i(x)$ for $m_i = 0$ is $\text{OnlyOne}_i(x) \mid_{m_i=0} = 1$. For the case of $m_i > 0$, $\text{OnlyOne}_i(x)$'s value is given by equation (27) (that appears again below), where ρ_j is the probability for transmitting in the j -th listening/signaling period.

$$\begin{aligned} \text{OnlyOne}_i(x) \mid_{m_i > 0} &= \rho_1(1 - \rho_1)^{m_i} + \rho_2(1 - \rho_1 - \rho_2)^{m_i} \\ &+ \dots + \rho_{n-1}(1 - \sum_{\ell=1}^{n-1} \rho_\ell)^{m_i} \quad [\text{clone of equation (27)}] \end{aligned}$$

We note that the j -th term in equation (27), is the probability that node p_i selects the j -th listening/signaling period and all its neighbors select a later listening/signaling period. \square

Proposition 10.3 shows that once a node is the only one in its neighborhood to transmit during its broadcasting timeslot, it enters the relative state *Allocated*.

Proposition 10.3. $M_i(x) = \emptyset$ (or having none of the nodes in $M_i(x)$ transmitting during timeslot t') implies that node p_i is in the relative state *Allocated* in $c_0^{\text{timeslot}}(x+1)$.

Proof. We need to show that, in $c_0^{\text{timeslot}}(x+1)$, we have that $s_i = t' \neq \perp$ and $\forall p_j \in \mathcal{N}_i : s_i \neq s_j$.

Showing that $s_i = t' \neq \perp$ in $c_0^{\text{timeslot}}(x+1)$ The proposition assumes that $t' \neq \perp$ in $c_0^{\text{timeslot}}(x)$. We wish to show that $s_i = t'$ in $c_0^{\text{timeslot}}(x+1)$, which implies that $s_i \neq \perp$ holds in $c_0^{\text{timeslot}}(x+1)$ and throughout $R(x+1)$.

Since the variable s_i is assigned only in lines 11 (when $t_i = 0$) and 26 (when $t_i = t'$), it is sufficient to show that line 26 is not executed by any step during timeslot t' of broadcasting round $R(x)$, i.e., $a_i^{\text{carrier.sense}, t'}(x) \notin R(x)$.

Node p_i raises the event carrier_sense only during timeslots in which p_i 's neighbor, p_j , transmits. By the proposition assumptions that, during timeslot t' of broadcasting round $R(x)$, none of p_i 's neighbors transmits, we have $a_i^{\text{carrier.sense}, t'}(x) \notin R(x)$. Moreover, $a_i^{\text{timeslot}, t'}(x+1)$ does not includes an execution of line 11 that changes the value of s_i , because $s_i = t' \neq \perp$ in $c_0^{\text{timeslot}}(x+1)$.

Showing that $\forall p_j \in \mathcal{N}_i : s_i \neq s_j$ in $c_0^{\text{timeslot}}(x+1)$ The proposition assumes that $\forall p_j \in \mathcal{N}_i : s_i \neq s_j$ in $c_0^{\text{timeslot}}(x)$. We wish to show that the same holds in $c_0^{\text{timeslot}}(x+1)$. Since the variable s_j is assigned to a non- \perp value only in line 11 when $t_i = 0$, it is

sufficient to show that when line 11 is executed in step $a_j^{\text{timeslot},0}(x+1)$ the function `select_unused()` considers a set that does not include p_i 's timeslot, s_i . This is implied by the facts that $\forall p_j \in \mathcal{N}_i : \text{unused}_j[t'] = \text{false}$ (Claim 10.1) and $s_i = t'$ (first item of (II) of this proof) in $c_0^{\text{timeslot}}(x+1)$. \square

11 Theorem 4.2

Theorem 4.2 shows that all nodes are allocated eventually with timeslots (convergence) and once all nodes are allocated, they stay this way (closure).

Theorem 4.2 (Self-Stabilization) *The MAC algorithm in Fig. 2 is a self-stabilizing algorithm with respect to the task τ_{TDMA} .*

Proof. After the above proof of propositions, we can demonstrate this theorem.

• **Convergence** We need to show that properties (1) to (5) eventually hold in configuration $c_0^{\text{timeslot}}(x+y)$ for a finite value of $y > 0$. Propositions 9.1, 9.2 and 9.3 imply that properties (1), (2), and respectively, (3) within two broadcasting round.

Propositions 10.1, 10.2 and 10.3 show that there is a nonzero probability that node p_i enters the relative state Allocated from either Ready or Obtaining within one broadcasting round. Thus, by the analyzing the expected time of the scheduler-luck games [16, 18], we have y has a finite value. Further analysis of y appears in theorems 5.1 and 6.2.

• **Closure** Suppose that $c_0^{\text{timeslot}}(x) \in R$ is a safe configuration and let $p_i \in P$ be any node. By the assumption that $c_0^{\text{timeslot}}(x)$, we have that p_i is in the relative state Allocated, i.e., properties (1) to (5) hold for any node p_i . We need to show that properties (1) to (5) holds in configuration $c_0^{\text{timeslot}}(x+1)$.

Propositions 9.1, 9.2 and 9.3 imply that properties (1), (2), and respectively, (3) (within one complete broadcasting round).

Properties (4) to (5) are implied by Proposition 10.3 and the fact that Properties (4) to (5) holds in $c_0^{\text{timeslot}}(x)$, i.e., $M(x) = \emptyset$. \square

12 Bounding $OnlyOne_i(x)$

Propositions 5.3 and 12.2 bound $OnlyOne_i(x)$'s value, where $R(x)$ is the x -th broadcasting round in execution R of the MAC algorithm in Fig. 2. We assume that properties (1) to (5) holds in the first configuration, $c_0^{\text{timeslot}}(x)$, of $R(x)$. These bounds are obtained by computing the expectation of $q_i |_{m_i}$ with respect to m_i , where $M_i(x) = \{p_k \in \mathcal{N}_i : s_k = t'\}$ in $c_0^{\text{timeslot}}(x)$ and $m_i = |M_i(x)|$. The reason is that m_i is a random variable, i.e., $q_i = E(OnlyOne_i(x) |_{m_i})$, where the expectation is computed with respect to the random variable m_i .

We note that all the terms in equation (27) are convex functions of m_i . This means that by Jensen's inequality, we obtain a lower bound of q_i in equation (28) by evaluating the expression $q_i |_{m_i}$ at m_i 's expectation, $E(m_i)$ [27].

$$q_i = E(q_i |_{m_i}) \geq q_i |_{E(m_i)} \quad (28)$$

The expression on the right side of the inequality can be again lower bounded if we estimate an upper bound for $E(m_i)$. We proceed to the computations in the proof of the Proposition 12.2 after demonstrating Proposition 12.1 which shows that $E(m_i)$ is bounded by the ratio d_i/T , which is rather intuitive but, needs to be proved.

Proposition 12.1. *In configuration $c_0^{\text{timeslot}}(x)$ it holds that $E(m_i) \leq d_i/T$, where $m_i = |M_i(x)|$.*

Proof. We show that $E(m_i) = d_i/T$ by considering configuration $c_0^{\text{timeslot}}(x)$. The maximal number of p_i 's neighbors that might choose the same timeslot as p_i in configuration $c_0^{\text{timeslot}}(x)$ is $\sum_{p_j \in \mathcal{N}_i} 1_{\{s_j = \perp\}}$, because any node, $p_j \in \mathcal{N}_i$, that chooses a new broadcasting timeslot immediately before $c_0^{\text{timeslot}}(x)$ must have $s_j = \perp$ in configuration $c_0^{\text{timeslot}}(x)$. We compute the expected value of m_i in equation (29) as a function of the number of empty timeslots, e_i , that p_i selects from when choosing a new broadcasting timeslot, where $e_i = |\text{unused_set}_i|$ in configuration $c_0^{\text{timeslot}}(x)$.

$$\begin{aligned}
E(m_i) &= \tag{29} \\
&\sum_{t \in E_i} E(m_i | s_i = t) \Pr(s_i = t) = \\
&\sum_{t \in E_i} \frac{1}{e_i} E(m_i | s_i = t) = \\
&\sum_{t \in E_i} \frac{1}{e_i} E \left(\sum_{p_j \in \mathcal{N}_i} 1_{\{p_j \text{ chooses timeslot } t\}} | s_i = t \right) = \\
&\sum_{t \in E_i} \frac{1}{e_i} \sum_{p_j \in \mathcal{N}_i} \frac{1}{|E_j|} 1_{\{t \in E_j\}} 1_{\{s_j = \perp\}}
\end{aligned}$$

Our assumption that $d_i \leq T - 1$ implies that $e_i > 0$. Using that $d_i = \sum_{p_j \in \mathcal{N}_i} (1_{\{s_j \neq \perp\}} + 1_{\{s_j = \perp\}})$ and, $e_i \geq T - \sum_{p_j \in \mathcal{N}_i} 1_{\{s_j \neq \perp\}}$, we obtain equation (30).

$$\begin{aligned}
E(m_i) &\leq \sum_{t \in E_i} \frac{1}{T - d_i + \sum_{p_j \in \mathcal{N}_i} 1_{\{s_j = \perp\}}} \sum_{p_j \in \mathcal{N}_i} \frac{1_{\{t \in E_j\}} 1_{\{s_j = \perp\}}}{|E_j|} \\
&= \frac{1}{T - d_i + \sum_{p_j \in \mathcal{N}_i} 1_{\{s_j = \perp\}}} \sum_{p_j \in \mathcal{N}_i} \frac{1_{\{s_j = \perp\}}}{|E_j|} \underbrace{\sum_{t \in E_i} 1_{\{t \in E_j\}}}_{|E_i \cap E_j|} \\
&\leq \frac{\sum_{p_j \in \mathcal{N}_i} 1_{\{s_j = \perp\}}}{T - d_i + \sum_{p_j \in \mathcal{N}_i} 1_{\{s_j = \perp\}}} \leq \frac{d_i}{T} \tag{30}
\end{aligned}$$

□

Proposition 12.2.

$$q_i \geq \sum_{k=1}^n \rho_k \left(1 - \sum_{\ell=1}^k \rho_\ell \right)^{\frac{d_i}{T}}$$

[clone of equation (6)]

Proof. Proposition 12.1 shows that $E(m_i) \leq d_i/T$. The proposition is demonstrated by evaluating expression (27) at $E(m_i) = d_i/T$, see equation (28). \square

Proposition 5.3 considers the concrete transmission probability $\rho_i = 1/MaxRnd$.

Proposition 5.3 *Let $\rho_i = 1/MaxRnd$. Equation (16) bounds from below the probability q_i .*

Proof. In this proof, we use the letter n instead of $MaxRnd$ for reason of space. We replace ρ_i with $1/n$ in equation (6) to obtain equation (31).

$$q_i \geq \sum_{k=1}^n \frac{1}{n} \left(1 - \frac{k}{n} \right)^{\frac{d_i}{T}} \quad (31)$$

Equation (32) is more compact than equation (31) and it is obtained by the fact that the function $(1-x)^s$ is convex.

$$\begin{aligned} q_i &\geq & (32) \\ &\sum_{k=1}^n \frac{1}{n} \left(1 - \frac{k}{n} \right)^{\frac{d_i}{T}} = \\ &\frac{1}{2n} \sum_{k=1}^n \left[\left(1 - \frac{k}{n} \right)^{\frac{d_i}{T}} + \left(1 - \frac{n-k+1}{n} \right)^{\frac{d_i}{T}} \right] \geq \\ &(\text{convexity}) \frac{1}{n} \sum_{k=1}^n \left(1 - \frac{n+1}{2n} \right)^{\frac{d_i}{T}} = \\ &\left(1 - \frac{n+1}{2n} \right)^{\frac{d_i}{T}} \end{aligned}$$

Another way to bound equation (31) is by considering the decreasing function $y \rightarrow (1-y)^x$, as in equation (33).

$$\begin{aligned} q_i &\geq \sum_{k=1}^n \frac{1}{n} \left(1 - \frac{j}{n} \right)^{\frac{d_i}{T}} \geq \int_{\frac{1}{n}}^1 (1-y)^{\frac{d_i}{T}} dy & (33) \\ &= \frac{1}{\frac{d_i}{T} + 1} \left(1 - \frac{1}{n} \right)^{\frac{d_i}{T} + 1} \end{aligned}$$

\square