PRINCIPLES OF ROBUST MEDIUM ACCESS AND AN APPLICATION TO LEADER ELECTION*

BARUCH AWERBUCH[1], ANDREA RICHA[2], CHRISTIAN SCHEIDELER[3],
STEFAN SCHMID[4], JIN ZHANG[2]

[1] DEPT. COMPUTER SCIENCE, JOHN HOPKINS UNIVERSITY, BALTIMORE, MD 21218,
USA; BARUCH@CS.JHU.EDU

[2] COMPUTER SCIENCE AND ENGINEERING, SCIDSE, ARIZONA STATE UNIVERSITY,
TEMPE, AZ 85287, USA; {ARICHA,JZHANG82}@ASU.EDU

[3] DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF PADERBORN, D-33102
PADERBORN, GERMANY; SCHEIDELER@UPB.DE

[4] TELEKOM INNOVATION LABORATORIES (T-LABS) & TU BERLIN, D-10587 BERLIN,
GERMANY; STEFAN@NET.T-LABS.TU-BERLIN.DE

Abstract. This article studies the design of medium access control (MAC) protocols for wireless networks that are provably robust against arbitrary and unpredictable disruptions, e.g., due to unintentional external interference from co-existing networks or due to jamming. We consider a wireless network consisting of a set of n honest and reliable nodes within transmission (and interference) range of each other, and we model the external disruptions with a powerful, adaptive adversary. This adversary may know the protocol and its entire history and can use this knowledge to jam the wireless channel at will at any time. It is allowed to jam a $(1-\epsilon)$-fraction of the time steps, for an arbitrary constant $\epsilon > 0$ unknown to the nodes. The nodes cannot distinguish between the adversarial jamming or a collision of two or more messages that are sent at the same time. We demonstrate, for the first time, that there is a local-control MAC protocol requiring only very limited knowledge about the adversary and the network that achieves a constant (asymptotically optimal) throughput for the non-jammed time periods under any adversarial strategy above. The derived principles are also useful to build robust applications on top of the MAC layer, and we present an exemplary study for leader election, one of the most fundamental tasks in distributed computing.

1. Introduction. The efficient use of a shared medium is arguable one of the most relevant but also most complex problems in distributed computing. First, a wireless network requires distributed access coordination mechanisms which minimize the *internal* interference due to simultaneous transmissions from wireless devices in the same network. In addition, the availability of the wireless medium can vary significantly over time due to the *external* interference, e.g., due to disturbances from other sources such as microwaves, due to transmissions of co-existing (potentially mobile) networks, or due to intentional or even adversarial interruptions. Adversarial attacks constitute a major threat especially since they often do not require any special hardware and may be implemented by simply listening to the open medium and broadcasting in the same frequency band as the network.

This article studies the design of distributed medium access schemes which are robust even against a powerful adversary who can block the medium at arbitrary and unpredictable times, and in an adaptive manner (i.e., depending on the protocol history). This adversarial model is used to capture a wide range of interference scenarios. Despite the adversary's power, we show that provably robust medium access solutions exist in the sense that in the time periods where the medium is available, there are many successful transmissions.

1.1. Our Model. We attend to a wireless network consisting of n reliable and honest nodes within each other's transmission (and interference) range. All of the nodes are continuously contending for sending a packet on the wireless channel. We assume that time proceeds in synchronous time steps and in each time step any node may decide to transmit a packet. A node may either transmit a message or sense the channel at a time step, but it cannot do both, and there is no immediate feedback mechanism telling a node whether its transmission

*Preliminary versions of this article appeared at the 27th ACM Symposium on Principles of Distributed Computing (PODC 2008) [5] and at 12th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2011) [36].

was successful. A node which is sensing the channel may either (i) sense an *idle* channel (in case no other node is transmitting at that time), (ii) sense a *busy* channel (in case two or more nodes transmit at the time step), or (iii) *receive* a packet (in case exactly one node transmits at the time step).

In addition to these nodes there is an adversary. We allow the adversary to know the protocol and its entire history and to use this knowledge in order to jam the wireless channel at will at any time (i.e, the adversary is *adaptive*). Whenever it jams the channel, all nodes will notice a busy channel. However, the nodes cannot distinguish between the adversarial jamming or a collision of two or more messages that are sent at the same time. We assume that the adversary is only allowed to jam a $(1-\epsilon)$-fraction of the time steps, for an arbitrary constant $\epsilon > 0$ unknown to the honest nodes.

We allow the adversary to perform bursty jamming. More formally, an adversary is called $(T, 1-\epsilon)$-*bounded* for some $T \in \mathbb{N}$ and $0 < \epsilon < 1$ if for any time window of size $w \geq T$ the adversary can jam at most $(1-\epsilon)w$ of the time steps in that window. A MAC protocol is called *c-competitive* against some $(T, 1-\epsilon)$-bounded adversary (with high probability[1] or on expectation) if, for any sufficiently large number of time steps, the nodes manage to perform successful message transmissions in at least a c-fraction of the time steps not jammed by the adversary (with high probability or on expectation).

Our goal is to design a *symmetric local-control* MAC protocol that is *constant competitive* against any $(T, 1-\epsilon)$-bounded adversary, i.e., there is no central authority controlling the nodes, and the nodes have symmetric roles at any point in time. The nodes do not know ϵ, but we do allow them to have a very rough upper bound of their number n and T. More specifically, we will assume that the nodes have a common parameter $\gamma = O(1/(\log T + \log\log n))$. Such an estimate leaves room for a superpolynomial change in n and a polynomial change in T over time, so it does not make the problem trivial (as it would be the case if the nodes knew constant factor approximations of n or T).

1.2. Our Contributions. This article introduces techniques for the design of robust medium access protocols. In particular, it presents the first MAC protocol that is constant competitive w.h.p., under any $(T, 1-\epsilon)$-bounded adversary, given that the protocol is executed sufficiently long. The protocol does not need to know ϵ, and ϵ can be an arbitrarily small constant. The developed principles can also be used to build robust applications on top of the MAC layer. In this respect, we present a new solution to the leader election problem—an evergreen in the distributed computing. Our solution is not only robust to interference, but it is also self-stabilizing in the sense that it converges to a correct state from any initial state. This is particularly interesting in dynamic environments. We are not aware of any similarly robust solution to the leader election problem.

1.3. Related Work. Wireless network jamming has been extensively studied in the applied networking domain (e.g., [1, 8, 10, 26, 28, 29, 31, 32, 40, 42, 43, 44]). Mechanisms for launching jamming attacks (e.g., [10, 26, 28, 44]) as well as defense mechanisms against these attacks (e.g., [1, 10, 42, 28, 29, 31, 8, 44]) have been proposed and validated through simulations and experiments.

Traditional defenses against jamming primarily focus on the design of physical layer technologies, such as spread spectrum [29, 31, 38]. While widely spread frequencies could potentially help in guarding against physical layer jamming, spread spectrum techniques cannot be used effectively in the relatively narrow frequency bands used by the 802.11 standard.

More recent work has also focused on various MAC layer strategies in order to handle jamming, including coding strategies [10], channel surfing and spatial retreat [45, 1], or

[1] "With high probability", or short "w.h.p.", means a probability of at least $1 - 1/n^c$ for any constant $c > 0$.

mechanisms to hide messages from a jammer, evade its search, and reduce the impact of corrupted messages [42]. Most of these strategies have only been evaluated experimentally and would not help against the jammers considered in this article.

A recent study [6] shows both theoretically and experimentally that an adaptive jammer, such as the one proposed here, can dramatically reduce the throughput of the standard random backoff MAC protocol of the IEEE802.11 standard with only limited energy cost on the adversary side (please also refer to [6] for other references on jamming in 802.11).

Adversarial jamming has also been studied theoretically. There are two basic approaches in the literature. The first assumes that messages may be corrupted at random (e.g. [34]), and the second bounds the number of messages that the adversary can transmit or disrupt due to, for example, a limited energy budget (e.g. [16, 21]). In a single hop wireless network (like ours), messages will not be corrupted independently at random (every time the jammer transmits, all messages in that time step will be corrupted); moreover, an adaptive adversary seems more powerful than one that jams uniformly at random [6]. Hence, we focus on the second line of theoretical work since it is more relevant to the results in this article.

The latest results in [16, 21] address adversarial jamming at both the MAC and network layers, where the adversary may not only be jamming the channel but also introducing malicious (fake) messages (possibly with address spoofing). The results in [16] only consider the scenario that the nodes have one message to transmit (e.g., a broadcast operation). When translated to our continuous data stream scenario, the protocol presented in [16] would not be able to sustain a constant-competitive ratio if the adversary is allowed to jam more than half of the time steps (i.e., if $\epsilon < 1/2$), given the fact that their single message broadcast algorithm takes at least twice as many steps as the number of time steps utilized by the jammer. Moreover, [16] assumes that the nodes have knowledge of n and of the fact that the adversary has a bounded number of messages it can transmit (in contrast, we only need the nodes to have an estimate on $\log\log n$ and $\log T$).

In [21], the authors consider a wireless network in which node positions form a grid where multiple (at most t) adversarial nodes are allowed in the direct neighborhood of a node. If t is at most a suitably small constant, then they give a protocol for reliable broadcast of a single message given that there is a fixed bound on the number of time steps the adversary is disrupting communication (if t is large, no broadcast protocol is guaranteed to terminate). The authors only show that eventually the broadcast operation will be completed, but give no bounds on how long that will take. Moreover, their algorithms will clearly deplete the energy of the non-faulty nodes at a higher rate than that of the faulty nodes.

Most of the theoretical work on the design of efficient MAC protocols has focused on random backoff protocols (e.g., [7, 11, 17, 18, 25, 35]) that do not take jamming activity into account and therefore are not robust against it. MAC protocols have also been designed in the context of broadcasting (e.g., [12]) and clustering (e.g., [23]). Most of them use random backoff or tournaments in order to handle interference and thereby achieve a fast runtime.

In general terms, in a random backoff protocol, each node periodically attempts to transmit a message starting with a certain probability p. In case the message transmission is unsuccessful (due to interference), the node will retry sending the message in the next time steps with monotonically decreasing probabilities (for example, $p^2, p^4, p^8, \ldots$) until the message is successfully transmitted or the minimum allowable probability is reached. In a dense network (as in our single-hop scenario), an adversary with knowledge of the MAC protocol would simply wait until the nodes have reached transmission probabilities that are inversely proportional to the number of close-by nodes to start jamming the channel, forcing the nodes to lower their transmission probabilities by so much that a constant throughput is not achievable.

There is also a large body on the leader election application considered in this article. Leader election is an evergreen in distributed algorithms research and there exist many theoretical and practical results [4, 14, 22, 27, 30, 33, 39, 41]. The following two book chapters provide a good introduction: Chapter 3 in [3] and Chapter 8 in [19]. A leader election algorithm should be as flexible as possible in the sense that a correct solution is computed *independently* of the initial network state. For instance, the algorithm should be able to react to a leader departure, or be able to cope with situations where for some reasons, multiple nodes consider themselves leaders. *Self-stabilization* [13] is an attractive concept to describe such self-repairing properties of an algorithm, and it has been intensively studied already, not only in terms of eventual stabilization but also in terms of guaranteed convergence times (see e.g., the works on time-adaptive self-stabilization such as [24]). Several self-stabilizing leader election protocols have been devised, e.g., [2, 9, 20] (see also the fault-contained solutions such as [15]). However, none of these approaches allows us to elect a leader in a wireless network that is exposed to harsh interference or even adaptive jamming. But such interruptions of communication are often unavoidable in wireless systems, and we believe that electing a leader can be particularly useful in such harsh environments.

1.4. Organization. The remainder of this article is organized as follows. Section 2 introduces the main principles of our approach and presents the robust medium access protocol (Section 2.1). We prove competitive throughput in Section 2.2 and also show that the number of useless message transmission attempts in times of high external interference is small (i.e., the protocol does not waste transmission energy). Section 3 then attends to the specific application of leader election, and presents a protocol (Section 3.1) together with a proof of the robustness properties (Section 3.2). Section 4 concludes the paper.

2. Robust Medium Access. In this section we present and analyze our MAC protocol. We start with a description of our basic ideas behind the protocol, and then provide the formal listing of the protocol and analyze its competitiveness.

Our MAC protocol is based on a simple idea. Suppose that each node v decides to send a message at the current time step with probability p_v with $p_v \leq \hat{p}$ for some small constant $0 < \hat{p} < 1$. Let $p = \sum_v p_v$, q_0 be the probability that the channel is idle and q_1 be the probability that exactly one node is sending a message. Then the following claim holds.

CLAIM 2.1. $q_0 \cdot p \leq q_1 \leq \frac{q_0}{1-\hat{p}} \cdot p$.

Proof. It holds that $q_0 = \prod_v (1-p_v)$ and $q_1 = \sum_v p_v \prod_{w \neq v}(1-p_w)$. Hence,

$$q_1 \leq \sum_v p_v \frac{1}{1-\hat{p}} \prod_w (1-p_w) = \frac{q_0 \cdot p}{1-\hat{p}} \text{ and } q_1 \geq \sum_v p_v \prod_w (1-p_w) = q_0 \cdot p\ .$$

□

Hence, if the nodes observe that the number of time steps in which the channel is idle is essentially equal to the number of time steps in which exactly one message is sent, then $p = \sum_v p_v$ is likely to be around 1. Otherwise, they know that they need to adapt their probabilities. Therefore, if we had sufficiently many cases in which an idle channel or exactly one message transmission is observed (which is the case if the adversary does not heavily jam the channel and p is not too large), then one can adapt the probabilities p_v just based on these two events and ignore all cases in which the wireless channel is blocked (either because the adversary is jamming it or at least two messages interfere with each other). Essentially, the following strategy could be used at every node for some small enough $\gamma > 0$:

In each time step, every node v is sending a message with probability p_v. If it decides not to send a message, it checks the following two cases:

- If the wireless channel is idle, then $p_v := (1+\gamma)p_v$.

- If exactly one message is sent, then $p_v := (1+\gamma)^{-1} p_v$.

The beauty of the algorithm is that it ignores blocked time steps, which makes it more robust against adversarial jamming: the access probabilities are maintained. However, there is a catch to this strategy because it only works well as long as p does not get too high. If p is initially very high or by chance gets very high, it will be extremely unlikely for the nodes to observe one of the two cases above. Hence, further ideas are necessary.

Our idea is to use a threshold T_v for each node v that cuts its time into time intervals. If v does not observe a successful message transmission for T_v many steps, then p_v is decreased. In this way, eventually p will become small. However, since the algorithm is not aware of T, the time window of the adversary, p may be decreased too quickly or too slowly in this way. Hence, we need proper rules for adapting T_v over time. It turns out that the following rules work: whenever v senses a successful transmission, T_v is decreased by 1, and whenever v does not sense a successful transmission for T_v time steps, T_v is increased by 1 for the next time interval considered by v. One may ask why T_v should not be decreased as well if an idle channel is sensed, but interestingly this is not a good rule, as will come out in the analysis. Next, we give a formal description of our MAC protocol.

2.1. Description of the MAC Protocol. In our MAC protocol, each node v maintains a probability value p_v, a threshold T_v and a counter c_v. The parameter γ is the same for every node and is set to some sufficiently small value in $O(1/(\log T + \log\log n))$. Thus, we assume that the nodes have some polynomial estimate of T and even rougher estimate of n. Let $\hat{p}$ be any constant so that $0 < \hat{p} \leq 1/24$. Initially, every node v sets $T_v := 1$, $c_v := 1$ and $p_v := \hat{p}$. Afterwards, the protocol works in synchronized time steps. We assume synchronized time steps for the analysis, but a non-synchronized execution of the protocol would also work as long as all nodes operate at roughly the same speed.

In each step, each node v does the following. v decides with probability p_v to send a message. If it decides not to send a message, it checks the following two conditions:

1. If v senses an idle channel, then $p_v := \min\{(1+\gamma)p_v, \hat{p}\}$.
2. If v successfully receives a message, then $p_v := (1+\gamma)^{-1} p_v$ and $T_v := \max\{1, T_v - 1\}$.

Afterwards, v sets $c_v := c_v + 1$. If $c_v > T_v$ then it does the following: v sets $c_v := 1$, and if there was no step among the past T_v time steps in which v sensed a successful message transmission, then $p_v := (1+\gamma)^{-1} p_v$ and $T_v := T_v + 1$.

2.2. Robustness. Let $N = \max\{T, n\}$. In this section, we will prove the following theorem.

THEOREM 2.2. *For $n \geq 2$ the MAC protocol is constant competitive w.h.p. under any $(T, 1-\epsilon)$-bounded adversary if the protocol is executed for at least $\Theta(\frac{1}{\epsilon} \log N \max\{T, \frac{1}{\epsilon\gamma^2} \log^3 N\})$ many time steps.*

Notice that for $n = 1$ a node will never experience a time step with a successful transmission. Hence, it would just keep reducing its access probability in our protocol, thereby reaching a dormant state, which is the best it can do in this case as there is no one else to communicate with. Thus, it only makes sense to consider the case $n \geq 2$.

The proof of the theorem will frequently use the following general form of the well-known Chernoff bounds, which may be of independent interest. They are derived from Chernoff bounds presented in [37].

LEMMA 2.3. *Consider any set of binary random variables $X_1, \ldots, X_n$. Suppose that there are values $p_1, \ldots, p_n \in [0,1]$ with $\mathbb{E}[\prod_{i \in S} X_i] \leq \prod_{i \in S} p_i$ for every set $S \subseteq$*

$\{1,\ldots,n\}$. *Then it holds for* $X=\sum_{i=1}^n X_i$ *and* $\mu=\sum_{i=1}^n p_i$ *and any* $\delta>0$ *that*

$$\mathbb{P}[X \geq (1+\delta)\mu] \leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu} \leq e^{-\frac{\delta^2\mu}{2(1+\delta/3)}}$$

If, on the other hand, it holds that $\mathbb{E}[\prod_{i\in S} X_i] \geq \prod_{i\in S} p_i$ *for every set* $S\subseteq\{1,\ldots,n\}$*, then it holds for any* $0<\delta<1$ *that*

$$\mathbb{P}[X \leq (1-\delta)\mu] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\mu} \leq e^{-\delta^2\mu/2}$$

Let V be the set of all nodes. For the proof of the theorem we will consider all possible decompositions of V into a single node v_0 and $U=V\setminus\{v_0\}$. Let $p_t(v)$ be node v's access probability p_v at the beginning of the t-th time step. Furthermore, let $p_t=\sum_{v\in U} p_t(v)$ (i.e., without node v_0) and $L=\Omega(\frac{1}{\epsilon}\log N \max\{T, \frac{1}{\epsilon\gamma^2}\log^3 N\})$ be the number of time steps for which we study the competitiveness of the protocol. If $L\geq N$, we will redefine N to $N=\max\{T,n,L\}$ in order to cover long runtimes. If we can prove a constant competitiveness for any such L, Theorem 2.2 follows.

We prove the theorem by induction over sufficiently large time frames. Let I be a time frame consisting of $\frac{\alpha}{\epsilon}\log N$ *subframes* I' of size $f=\max\{T,\frac{\alpha\beta^2}{\epsilon\gamma^2}\log^3 N\}$, where α and β are sufficiently large constants. Let $F=\frac{\alpha}{\epsilon}\log N\cdot f$ denote the size of I. We assume that at the beginning of I, $p_t\geq 1/(f^2(1+\gamma)^{2\sqrt{f}})$ and $T_v\leq\sqrt{F}/2$ for every node v. Our goal is to show that in this case the MAC protocol is constant competitive for I w.r.t. every subset $U=V\setminus\{v_0\}$ and at the end of I, $p_t\geq 1/(f^2(1+\gamma)^{2\sqrt{f}})$ and $T_v\leq\sqrt{F}/2$ for every node v with probability at least $1-1/N^c$ for any constant $c>0$ (which we will also call *with high probability* or *w.h.p.* in the following). Since initially $T_v=1$ and $p_v=\hat{p}$ for every v, this implies that the MAC protocol achieves a constant competitiveness in the first time frame, w.h.p., and due to the properties on T_v and p_v, this also holds for polynomially many time frames, w.h.p.

The proof for time frame I proceeds as follows. Consider some fixed subset $U=V\setminus\{v_0\}$. A time step t or subframe I' of I with starting time t is called *good* if $p_t\leq 9$. Otherwise, it is called *bad*. First, we show that for any subframe I' in which initially $p_t\geq 1/(f^2(1+\gamma)^{2\sqrt{f}})$, also afterwards $p_t\geq 1/(f^2(1+\gamma)^{2\sqrt{f}})$, w.h.p. (Lemma 2.4). Then we show that for any subframe I' with $T_v\leq(3/4)\sqrt{F}$ for every node $v\in U$ at the beginning of I', the subsequent subframe is good with probability at least $1-1/f^c$ for any constant $c>0$ (which we will call *with moderate probability* or *w.m.p.*) (Lemma 2.7). Based on the insights gained in the proof, we show that in a good subframe I', all non-jammed time steps in I' are good w.m.p. (Corollary 2.11). After that, we prove that a constant fraction of the time steps in such a subframe also have probabilities lower bounded by a constant (Lemma 2.12), w.h.p., which implies that the MAC protocol is constant competitive for I' w.m.p. (Lemma 2.13). If at the beginning of frame I, $T_v\leq\sqrt{F}/2$ for every node $v\in U$, then during the first eighth of I, called J, $T_v\leq(3/4)\sqrt{F}$, no matter what happens to the nodes in J. This allows us to show that a constant fraction of the subframes of J are constant competitive w.h.p., which implies that the MAC protocol is constant competitive for J w.h.p. (Lemma 2.14). With that insight we can show that if at the beginning of J, $T_v\leq\sqrt{F}/2$ for every node $v\in U$, then this also holds at the end of J w.h.p. (Lemma 2.15). Hence, all eighths of I have a constant competitiveness, w.h.p., which implies that I has a constant competitiveness and at the end of I, $T_v\leq\sqrt{F}/2$ for every node v, w.h.p. Applying these results inductively over all time frames I yields Theorem 2.2.

At the end of this subsection, we also study the recovery properties of our MAC protocol (Theorem 2.16). It turns out that the MAC protocol can get quickly out of any set of (p_v, c_v, T_v)-values, which implies that it also works well if the nodes enter the network at arbitrary times and with arbitrary values instead of starting the protocol at the same time and with the same values, which is not realistic in practice.

LEMMA 2.4. *For any subframe I' in which initially $p_{t_0} \geq 1/\ (f^2(1+\gamma)^{2\sqrt{f}})$, the last time step t of I' satisfies $p_t \geq 1/(f^2(1+\gamma)^{2\sqrt{f}})$, w.h.p.*

Proof. We start with the following claim about the maximum number of times nodes decrease their probabilities in I' due to $c_v > T_v$.

CLAIM 2.5. *If in subframe I' the number of successful message transmissions is at most k, then every node v increases T_v at most $k + \sqrt{2f}$ many times.*

Proof. Only successful message transmissions reduce T_v. If there is no successful message transmission within T_v many steps, T_v is increased. Suppose that $k = 0$. Then the number of times a node v increases T_v is upper bounded by the largest possible ℓ so that $\sum_{i=T_v^0}^{T_v^0+\ell} i \leq f$, where T_v^0 is the initial size of T_v. For any $T_v^0 \geq 1$, $\ell \leq \sqrt{2f}$, so the claim is true for $k = 0$. At best, each additional successful transmission allows us to reduce all thresholds for v by 1, so we are searching for the maximum ℓ so that $\sum_{i=T_v^0-k}^{T_v^0-k+\ell} \max\{i, 1\} \leq f$. This ℓ is upper bounded by $k + \sqrt{2f}$, which proves our claim. □

This claim allows us to show the following claim.

CLAIM 2.6. *Suppose that for the first time step t_0 in I', $p_{t_0} \in [1/(f^2(1+\gamma)^{2\sqrt{f}}), 1/f^2]$. Then there is a time step t in I' with $p_t \geq 1/f^2$, w.h.p.*

Proof. Suppose that there are g non-jammed time steps in I'. Let k_0 be the number of these steps with an idle channel and k_1 be the number of these steps with a successful message transmission. Furthermore, let k_2 be the maximum number of times a node v increases T_v in I'. If all time steps t in I' satisfy $p_t < 1/f^2$, then it must hold that

$$k_0 - \log_{1+\gamma}(1/p_{t_0}) \leq k_1 + k_2$$

This is because no v has reached a point with $p_t(v) = \hat{p}$ in this case, which implies that for each time step t' with an idle channel, $p_{t'+1} = (1+\gamma)p_{t'}$. Furthermore, at most $\log_{1+\gamma}(1/p_{t_0})$ increases of p_t due to an idle channel would be needed to get p_t to $1/f^2$, and then there would have to be a balance between further increases and decreases of p_t in order to avoid the case $p_t \geq 1/f^2$. We know from Claim 2.5 that $k_2 \leq k_1 + \sqrt{2f}$. Hence,

$$k_0 \leq 2\log_{1+\gamma} f + 2\sqrt{f} + 2k_1 + \sqrt{2f}$$

Suppose that $2\log_{1+\gamma} f + 4\sqrt{f} \leq \epsilon f/2$, which is true if $f = \Omega(1/\epsilon^2)$ is sufficiently large (resp. $\epsilon = \Omega(1/\log^3 N)$). Since $g \geq \epsilon f$ due to our adversarial model, it follows that we must satisfy $k_0 \leq 2k_1 + g/2$.

For any time step t with $p_t \leq 1/f^2$,

$$\begin{aligned}\mathbb{P}[\geq 1 \text{ message transmitted at } t] &\leq \sum_v p_v(t) = p_t + \hat{p} \\ &\leq 1/f^2 + \hat{p}\end{aligned}$$

where $\hat{p}$ is due to node v_0 not considered in p_t. Hence, $\mathbb{E}[k_0] \geq (1 - 1/f^2 - \hat{p})g$ and $\mathbb{E}[k_1] \leq (1/f^2 + \hat{p})g$. In order to prove bounds on k_0 and k_1 that hold w.h.p., we can use the general Chernoff bounds stated above. For any step t, let the binary random variable X_t be 1 if and only if the channel is idle at step t or $p_t \geq 1/f^2$. Then

$$\begin{aligned}\mathbb{P}[X_t = 1] &= \mathbb{P}[\text{channel idle and } p_t \leq 1/f^2] + \mathbb{P}[p_t > 1/f^2] \\ &= \mathbb{P}[p_t \leq 1/f^2] \cdot \mathbb{P}[\text{channel idle} \mid p_t \leq 1/f^2] + \mathbb{P}[p_t > 1/f^2] \\ &\geq \mathbb{P}[p_t \leq 1/f^2](1 - 1/f^2 - \hat{p}) + \mathbb{P}[p_t > 1/f^2] \\ &\geq 1 - 1/f^2 - \hat{p}\end{aligned}$$

and since this probability bound holds irrespective of prior steps and is *independent* of the adversarial jamming decision at time t, it follows for any set S of time steps prior to some time step t that

$$\mathbb{P}[X_t = 1 \mid \prod_{s \in S} X_s = 1] \geq 1 - 1/f^2 - \hat{p}$$

Thus, for any set of time steps S it holds that $\mathbb{E}[\prod_{s \in S} X_s] \geq (1 - 1/f^2 - \hat{p})^{|S|}$. Together with the fact that $g \geq \epsilon f \geq \alpha \log N$, the Chernoff bounds imply that, w.h.p., either $k_0 > 3g/4$ (given that $\hat{p} \leq 1/24$) or we have a time step t with $p_t \geq 1/f^2$.

On the other hand, let the binary random variable Y_t be 1 if and only if exactly one message is sent at time t and $p_t \leq 1/f^2$. Then

$$\begin{aligned}\mathbb{P}[Y_t = 1] &= \mathbb{P}[p_t \leq 1/f^2] \cdot \mathbb{P}[\text{one msg sent} \mid p_t \leq 1/f^2] \\ &\leq 1/f^2 + \hat{p}\end{aligned}$$

and it holds for any set S of time steps prior to some time step t that

$$\mathbb{P}[Y_t = 1 \mid \prod_{s \in S} Y_s = 1] \leq 1/f^2 + \hat{p}$$

Thus, the Chernoff bounds imply that $k_1 < g/8$, w.h.p. (given that $\hat{p} \leq 1/24$). That, however, would violate the condition that $k_0 \leq 2k_1 + g/2$.

Note that the choice of g is not oblivious as the adversary may *adaptively* decide to set g based on the history of events. Hence, we need to sum up the probabilities over all adversarial strategies of selecting g in order to show that none of them succeeds, but since there are only f many, and for each the claimed property holds w.h.p., the claim follows. □

So suppose that there is a time step t in I' with $p_t \geq 1/f^2$. If t belongs to one of the last $\beta \log N$ non-jammed steps in I', then it follows for the probability $p_{t'}$ at the end of I' that

$$p_{t'} \geq \frac{1}{f^2} \cdot (1+\gamma)^{-2\beta \log N + \sqrt{2f}} \geq \frac{1}{f^2(1+\gamma)^{2\sqrt{f}}}$$

given that $\epsilon = \Omega(1/\log^3 N)$ as at most $\beta \log N$ decreases of p_t can happen due to a successful transmission and at most $\beta \log N + \sqrt{2f}$ decreases of p_t can happen due to exceeding T_v.

Suppose, on the other hand, that there is no time step t among the last $\beta \log N$ non-jammed steps in I' with $p_t \geq 1/f^2$. In this case, we assume that a specific step t in I' outside of these last steps is the last time step with $p_t \geq 1/f^2$. When defining k_0, k_1 and k_2 as above but from that point on it follows that $p_{t'}$ at the end of I' is still bounded from below by $1/(f^2(1+\gamma)^{2\sqrt{f}})$ as long as $k_0 \geq k_1$. Our analysis above implies that this is true w.h.p. (see Claim 2.8 for similar arguments in the other direction), which finishes the proof of Lemma 2.4. □

LEMMA 2.7. *For any subframe I' with $T_v \leq (3/4)\sqrt{F}$ for all nodes v at the beginning of I', the last time step t of I' satisfies $p_t \leq 9$ w.m.p.*

Proof. We first show that there is a time step t in I' with $p_t \leq 6$, w.h.p. Let the time steps in which the adversary does not jam the channel and at most one message is sent by the nodes be called *useful*. Suppose that there are g useful time steps in I'. Let k_0 be the number of these steps with an idle channel and k_1 be the number of these steps with a successful message transmission. In order to establish a relationship between k_0 and k_1 we need the following claims.

CLAIM 2.8. *If all time steps $t \in I'$ satisfy $p_t > 6$, then it holds for any $g \geq \delta \log N$ for a sufficiently large constant δ that $k_1 \geq k_0$ w.h.p.*

Proof. Let $q_0(t)$ be the probability of an idle channel and $q_1(t)$ be the probability of a successful message transmission at a useful step t. If $p_t > 6$, then it follows from Claim 2.1 that

$$\begin{aligned}\mathbb{P}[\text{channel idle}] = \frac{q_0(t)}{q_0(t)+q_1(t)} &\leq \frac{q_0(t)}{q_0(t)+p_t \cdot q_0(t)} \\ &\leq \frac{1}{1+6} = \frac{1}{7}\end{aligned}$$

irrespective of what happened at previous time steps. Hence, $\mathbb{E}[k_0] \leq g/7$ under the assumption that all useful time steps t satisfy $p_t > 6$. Thus, our Chernoff bounds yield $k_0 \leq g/2$ w.h.p. (given that δ is a sufficiently large constant), which implies that $k_1 \geq k_0$. □

Now we are ready for the following claim.

CLAIM 2.9. *If all time steps in I' satisfy $p_t > 6$, then it must hold w.h.p. that*

$$k_1 - 2\log_{1+\gamma} N \leq (5/4)k_0$$

Proof. If exactly one message is sent at a step t, then $p_{t+1} \geq (1+\gamma)^{-1}p_t$ and

$$p_{t+1} \leq (1+\gamma)^{-1}(p_t - \hat{p}) + \hat{p} \leq (1+\gamma)^{-1}p_t + \gamma(1+\gamma)^{-1}\hat{p}$$

because only the sending node does not decrease its probability, and for this node the maximum probability is $\hat{p}$. For $p_t > 6$ it follows that $p_{t+1} \in [(1+\gamma)^{-1}p_t, (1+\gamma)^{-4/5}p_t]$. From Claim 2.8 we now that after the first $\delta \log N$ useful steps, there must have been more steps with a successful transmission than with an idle channel for any one of the remaining useful steps, w.h.p, which implies that for each of them, $p_v < \hat{p}$ for all nodes v. Thus, whenever there is an idle channel for these steps, $p_{t+1} = (1+\gamma)p_t$. Hence, if we start with $p_t = 6$ after the first $\delta \log N$ useful steps, then in order to avoid a step t' with $p_{t'} \leq 6$ in I' we must have that $k_1 \leq (5/4)k_0$. Since p_t might be as high as $\hat{p}n$ initially, we can allow at most $(5/4)\log_{1+\gamma} N$ further events of a successful message transmission without having a step t' with $p_{t'} \leq 6$.

Since $\log_{1+\gamma} N = \omega(\log N)$, it holds that

$$\delta \log N + (5/4)\log_{1+\gamma} N \leq 2\log_{1+\gamma} N$$

for a sufficiently large N, which implies the claim. □

Also, $k_0 + k_1 = g$. Suppose that $g \geq \delta \log_{1+\gamma} N$ for a sufficiently large constant δ. It holds that

$$(g - k_0) - 2g/\delta \leq (5/4)k_0 \quad \Leftrightarrow \quad k_0 \geq (4/9)(1 - 2/\delta)g$$

We know from the proof of Claim 2.8 that for any useful step t with $p_t > 6$, $\mathbb{P}[\text{channel idle}] \leq \frac{1}{7}$. Hence, $\mathbb{E}[k_0] \leq g/7$. Since random decisions are made independently in each step, our Chernoff bounds imply that $k_0 < (4/9)(1-2/\delta)g$ w.h.p. if δ is sufficiently large.

Thus, if I' contains at least $\delta \log_{1+\gamma} N$ useful steps, we are done. Otherwise, notice that for every node v it follows from the MAC protocol and the choice of f and F that if initially $T_v \leq (3/4)\sqrt{F}$, then T_v can be at most $\sqrt{F}$ during I'. Let us cut I' into m intervals of size $2\sqrt{F}$ each. It is easy to check that if β in the definition of f is sufficiently large compared to δ, then $m \geq 3\delta \log_{1+\gamma} N$. If there are less than $\delta \log_{1+\gamma} N$ useful steps, then at least $2\delta \log_{1+\gamma} N$ of these intervals do not contain any useful step, which implies that p_v is reduced by at least $(1+\gamma)^{-1}$ by each v in each of these intervals.

Hence, altogether, every p_v gets reduced by a factor of at least $(1+\gamma)^{-2\delta \log_{1+\gamma} N}$ during I'. The useful time steps can only raise that by $(1+\gamma)^{\delta \log_{1+\gamma} N}$, so altogether we must have $p_t \leq 6$ at some time point during I', w.h.p.

In the following, let t_0 denote any time in I' with $p_{t_0} \leq 6$. We finally prove the following claim.

CLAIM 2.10. *For any useful time step t after a step t_0 in I' with $p_{t_0} \leq \phi$ for some $\phi \geq 6$ and any constant $\delta > 0$ it holds that*

$$\mathbb{P}[p_t \geq (1+\delta)\phi] \leq 8 \cdot (1+\delta)^{-1/(6\gamma)}$$

Proof. Suppose that t_0 is the last useful time step before step t in I' with $p_{t_0} \leq \phi$. Let g be the number of useful time steps from t_0 to t. Then $g \geq \ln(1+\delta)/\ln(1+\gamma)$ because otherwise it is not possible that $p_t \geq (1+\delta)\phi$. Recall that for any useful step r with $p_r \geq 6$, $\mathbb{P}[p_{r+1} = (1+\gamma)p_r] \leq 1/7$. If exactly one message is sent at a useful step, then $p_{r+1} \in [(1+\gamma)^{-1}p_r, (1+\gamma)^{-4/5}p_r]$. Let k_0 be the number of useful steps with an idle channel and k_1 be the number of useful steps with a successful message transmission. It must hold that $k_0 \geq (4/5)k_1 + \ln(1+\delta)/\ln(1+\gamma)$ so that $p_t \geq (1+\delta)\phi$. Also, $k_0 + k_1 = g$. Hence, $k_0 \geq (4/9)g + (5/9)\ln(1+\delta)/\ln(1+\gamma) \geq \max\{(4/9)g, \ln(1+\delta)/\ln(1+\gamma)\}$. It holds that $\mathbb{E}[k_0] \leq g/7$, so the Chernoff bounds imply that

$$\begin{aligned}\mathbb{P}[k_0 \geq (4/9)g] &\leq \mathbb{P}[k_0 \geq (1+2)g/7] \\ &\leq e^{-[2^2/(2(1+2/3))](g/7)} = e^{-g/6}\end{aligned}$$

Hence,

$$\begin{aligned}\mathbb{P}[p_t \geq (1+\delta)\phi] &\leq \sum_{g \geq \frac{\ln(1+\delta)}{\ln(1+\gamma)}} \mathbb{P}[k_0 \geq (4/9)g] \leq \sum_{g \geq \frac{\ln(1+\delta)}{\ln(1+\gamma)}} e^{-g/6} \\ &\leq 8(1+\delta)^{-\frac{1}{6\ln(1+\gamma)}} \leq 8(1+\delta)^{-1/(6\gamma)}\end{aligned}$$

□

Since we assume that $\gamma = O(1/\log f)$, it follows that w.m.p., $p_t \leq (1+\delta)6$ for any particular time step t after t_0, resulting in the lemma with $\delta = 1/2$. □

Claim 2.10 with $\phi = 9$ and $\delta = 1/3$ implies the following result.

COROLLARY 2.11. *For any good subframe I', all non-jammed time steps t of I' satisfy $p_t \leq 12$ w.m.p.*

We also need to show that for a constant fraction of the non-jammed time steps in a good subframe, p_t is also lower bounded by a constant. Recall that $\hat{p} \leq 1/24$.

LEMMA 2.12. *For any subframe I' in which initially $p_t \geq 1/\left(f^2(1+\gamma)^{2\sqrt{f}}\right)$, at least $1/8$ of the non-jammed steps t satisfy $p_t \geq \hat{p}$, w.h.p.*

Proof. Let G be the set of all non-jammed time steps in I' and S be the set of all steps t in G with $p_t < \hat{p}$. Let $g = |G|$ and $s = |S|$. If $s \leq 7g/8$, we are done. Hence, consider the case that $s \geq 7g/8$.

Suppose that p_t must be increased k_0 many times to get from its initial value up to a value of $\hat{p}$ and that p_t is decreased k_1 many times in S due a successful message transmission. Furthermore, let k_2 be the maximum number of times a node v decreases p_v due to $c_v > T_v$ in the MAC protocol. For S to be feasible (i.e., probabilities can be assigned to each $t \in S$ so that $p_t < \hat{p}$) it must hold for the number ℓ of times in S in which the channel is idle that

$$\ell \leq k_0 + k_1 + k_2$$

For the special case that $k_0 = k_2 = 0$ this follows from the fact that whenever there is a successful message transmission, p_t is reduced to $p_{t+1} \geq (1+\gamma)^{-1} p_t$. On the other hand, whenever there is an idle channel, it holds that $p_{t+1} = (1+\gamma)p_t$ because of $p_t < \hat{p}$. Thus, if $\ell > k_1$, then one of the steps in S would have to have a probability of at least $\hat{p}$, violating the definition of S. k_0 comes into the formula due to the startup cost of getting to a value of $\hat{p}$, and k_2 comes into the formula since the reductions of the $p_t(v)$ values due to $c_v > T_v$ in the MAC protocol allow up to k_2 additional increases of p_t for S to stay feasible.

First, we bound ℓ. If $p_t < \hat{p}$, then $\mathbb{P}[\text{idle channel at step } t] \geq 1 - \hat{p} - \hat{p}$ (where the second $\hat{p}$ is due to node v_0), irrespective of prior time steps, Hence, $\mathbb{E}[\ell] \geq (1-2\hat{p})s$. For $\hat{p} \leq 1/24$ our Chernoff bounds imply (because of $s \geq 7g/8 \geq (7/8)\epsilon f$) that $\ell \geq s/2$ w.h.p. If at the beginning of I', $p_t \geq 1/(f^2(1+\gamma)^{2\sqrt{f}})$ then $k_0 \leq 2\log_{1+\gamma} f + 2\sqrt{f}$. Moreover, $k_2 \leq g/8 + k_1 + \sqrt{2f}$ because of Claim 2.5. Hence, $k_0 + k_1 + k_2 \leq 2\log_{1+\gamma} f + 2\sqrt{f} + 2k_1 + g/8 + \sqrt{2f}$, which must be at least $s/2$ so that $\ell \leq k_0 + k_1 + k_2$ (given that $\ell \geq s/2$). Suppose that $2\log_{1+\gamma} f + 4\sqrt{f} \leq \epsilon f/16$ (which is true if $f = \Omega(1/\epsilon^2)$ is large enough). Then for this to be true it must hold that

$$2k_1 + g/8 + g/16 \geq (7g/8)/2 \quad \Leftrightarrow \quad k_1 \geq g/8$$

If $k_1 \geq g/8$ then also $k_1 \geq s/8$, so our goal will be to show that $k_1 < s/8$ w.h.p.

If $p_t < \hat{p}$, then $\mathbb{P}[\text{successful message transmission at step } t] \leq 2\hat{p}$, irrespective of prior time steps. Hence, $\mathbb{E}[k_1] \leq 2\hat{p}s$. Furthermore, for $\hat{p} \leq 1/24$ our Chernoff bounds imply because of $s \geq 7g/8 \geq (7/8)\epsilon f$ that $k_1 < s/8$ w.h.p. Since there are at most f^2 ways (for the adversary) of choosing g and s, this holds for any combination of g and s, which yields the lemma. □

Combining the results above, we get:

LEMMA 2.13. *For any good subframe I' the MAC protocol is constant competitive in I' w.m.p.*

Proof. From Corollary 2.11 and Lemma 2.12 we know that in a good subframe at least $1/8$ of the non-jammed time steps t have a constant probability value p_t w.m.p. For these steps there is a constant probability that a message is successfully sent. Using the Chernoff bounds results in the lemma. □

Consider now the first eighth of frame I, called J.

LEMMA 2.14. *If at the beginning of J, $p \geq 1/(f^2(1+\gamma)^{2\sqrt{f}})$ and $T_v \leq \sqrt{F}/2$ for all nodes v, then we also have $p \geq 1/(f^2(1+\gamma)^{2\sqrt{f}})$ at the end of J and the MAC protocol is constant competitive for J, w.h.p.*

Proof. The bound for p at the end of J directly follows from Lemma 2.4. Suppose, as a worst case, that initially $T_v = \sqrt{F}/2$ for some v. Clearly, T_v assumes the maximum possible value at the end of J if T_v is never decreased in J. Since T_v can be increased at most $(F/8)/(\sqrt{F}/2) = \sqrt{F}/4$ many times in J, T_v can reach a maximum value of at most $(3/4)\sqrt{F}$ inside of J, so we can apply Lemma 2.7.

Recall that J consists of $k = \frac{\alpha}{8\epsilon}\log N$ many subframes, numbered $I_1, \ldots, I_k$. For each I_i, let the binary random variable X_i be 1 if and only if I_i is good. From Lemma 2.7 it follows that for any $i \geq 1$ and any set $S \subseteq \{1, \ldots, i-1\}$,

$$\mathbb{P}[X_i = 1 \mid \prod_{j \in S} X_j = 1] \geq 1 - 1/f^c$$

for some constant c that can be made arbitrarily large. Hence, for any set $S \subseteq \{1, \ldots, k\}$, $\mathbb{E}[\prod_{i \in S} X_i] \geq (1 - 1/f^c)^{|S|}$. Our Chernoff bounds therefore imply that at most $(\alpha/24\epsilon) \log N$ of the subframes in J are bad, w.h.p, if α is sufficiently large. According to Lemma 2.13, each of the good subframes is constant competitive w.m.p., where the probability bounds are only based on events in the subframes themselves and therefore hold irrespective of the other subframes (given that each of them is good). So the Chernoff bounds imply that at most $(\alpha/24\epsilon) \log N$ of them do not result in a constant competitiveness of the MAC protocol, w.h.p. The remaining $(\alpha/24\epsilon) \log N$ subframes in J achieve constant competitiveness, which implies that the MAC protocol is constant competitive on J, w.h.p. $\square$

We finally need the following lemma that bounds T_v. The proof of this lemma requires considering all possible decompositions of V into a node v_0 and $U = V \setminus \{v_0\}$ so that every node experiences many successful transmissions.

LEMMA 2.15. *If at the beginning of J, $T_v \leq \sqrt{F}/2$ for all v, then it holds that also $T_v \leq \sqrt{F}/2$ at the end of J, w.h.p.*

Proof. We know from Lemma 2.14 that for any node v our protocol is constant competitive for $V \setminus \{v\}$ w.h.p. Hence, every node v notices $\Omega(\epsilon|J|)$ successful message transmissions in J w.h.p. T_v is maximized at the end of J if all of these successful transmissions happen at the beginning of J, which would get T_v down to 1. Afterwards, T_v can raise to a value of at most t for the maximum t with $\sum_{i=1}^{t} i \leq |J|$. Since such a t can be at most $\sqrt{2|J|}$, it follows that T_v can be at most $\sqrt{2F/8} = \sqrt{F}/2$ at the end of J, w.h.p. $\square$

Inductively using Lemmas 2.13 and 2.15 on the eighths of frame I implies that our MAC protocol is constant competitive on I and at the end of I, $p_v \geq 1/(f^2(1+\gamma)^{2\sqrt{f}})$ and $T_v \leq \sqrt{F}/2$ for all v w.h.p. Hence, our MAC protocol is constant competitive for L many time steps, w.h.p., for any $L = \Omega(\frac{1}{\epsilon} \log N \max\{T, \frac{1}{\epsilon\gamma^2} \log^3 N\})$, which implies Theorem 2.2.

Finally, we show that our protocol can quickly recover from any setting of the (T_v, c_v, p_v)-values.

THEOREM 2.16. *For any p_{t_0} and $\hat{T} = \max_v T_v$ it takes at most $O(\frac{1}{\epsilon} \log_{1+\gamma}(1/p_{t_0}) + \hat{T}^2)$ many time steps, w.h.p., until the MAC protocol satisfies again $p_t \geq 1/(f^2(1+\gamma)^{2\sqrt{f}})$ and $\max_v T_v \leq \sqrt{F}/2$ for the original definitions of F and f above.*

Proof. Suppose that $p_{t_0} < 1/(f^2(1+\gamma)^{2\sqrt{f}})$ for some time point t_0. Then it follows from the constraints of the adversary and the Chernoff bounds that it takes at most $\delta/\epsilon \log_{1+\gamma}(1/p_{t_0})$ steps for some sufficiently large constant δ to get the system from p_{t_0} up to $p_{t_0}^{1/2}$, w.h.p. (in fact, with a probability of at least $1 - p_{t_0}^c$ for any constant c, irrespective of $\hat{T}$). Another $\frac{\delta}{2\epsilon} \log_{1+\gamma}(1/p_{t_0})$ steps will then get the system from $p_{t_0}^{1/2}$ to $p_{t_0}^{1/4}$, w.h.p. (in fact, with probability at least $1 - (p_{t_0}^{1/2})^c$ for any constant c). Continuing these arguments in order to get from $p_{t_0}^{1/2^i}$ to $p_{t_0}^{1/2^{i+1}}$ it follows that altogether at most $\frac{2\delta}{\epsilon} \log_{1+\gamma}(1/p_{t_0})$ steps are needed to get the system from p_{t_0} to a probability $p_t \geq \frac{1}{f^2(1+\gamma)^{2\sqrt{f}}}$, w.h.p. (or more precisely, with probability at least $1 - 1/N^c$).

It remains to bound the time to get T_v down to $\sqrt{F}/2$ for every v. It holds that $\hat{T} \leq \sqrt{F}/2$ if and only if $F \geq 4\hat{T}^2$. Hence, consider a time frame I of size $F' = \max\{F, 4\hat{T}^2\}$ for the old definition of F above, where I starts at the point at which the probabilities p_v have recovered to $p_t \geq 1/(f^2(1+\gamma)^{2\sqrt{f}})$. Then all the proofs above go through and imply that I

is constant competitive. Moreover, when cutting I into pieces of size $|I|/32$ instead of $|I|/8$, the proof of Lemma 2.15 implies that at the end of the first $1/32$-piece J of I, $T_v \leq \sqrt{F'}/4$, w.h.p. Hence, the time frames of the nodes shrank by a factor of at least 2 in J. Inductively using this bound, it follows that also at the end of I, $T_v \leq \sqrt{F'}/4$ for all v, w.h.p. This allows us to reduce F' by a factor of 2 for the next frame I. Also for this F', we get $T_v \leq \sqrt{F'}/4$ for all v, w.h.p., so we can keep shrinking I by a factor of 2 until $|I| = F$ for the original F considered in our proofs above. Altogether, the recovery to $\hat{T} \leq \sqrt{F}/2$ for all v takes at most $O(\hat{T}^2)$ time.

Combining the two upper bounds for the recovery time yields the theorem. □

Interestingly, we can show that our MAC protocol is also efficient under adversarial attacks in terms of transmitted messages. The first lemma follows directly from our earlier insights.

LEMMA 2.17. *For any time frame I of size F as defined above, the total number of transmitted messages by all the nodes is bounded by $O(F)$ w.h.p.*

If the adversary performs permanent jamming, the number of message transmissions converges, i.e., our MAC protocol reaches a dormant stage.

LEMMA 2.18. *Consider any time step t_0 with $\sum_v p_v \leq p$ and $\max_v T_v \leq \hat{T}$ for some values $p > 0$ and $\hat{T} \geq 1/\gamma$. Then for any continuous jamming attack starting at t_0 the total number of message transmissions during the entire attack is at most $O(p \cdot \hat{T}/\gamma + \log N)$ w.h.p.*

Proof. First, we determine the expected number of transmissions of a single node v. Let $p_v(t)$ be the probability that v transmits a message in round $t_0 + t$. Due to our MAC protocol, $p_v(t)$ decreases by $(1+\gamma)^{-1}$ at latest for $t = \hat{T}$, then another time after $\hat{T} + 1$ further steps, another time after $\hat{T} + 2$ further steps, and so on. Hence, the total expected number of transmissions of v for any continuous jamming attack is at most

$$\begin{aligned}
&\sum_{T_v \geq \hat{T}} T_v \cdot p_v(t_0)(1+\gamma)^{T_v - \hat{T}} \\
&= p_v(t_0) \sum_{i \geq 0} (\hat{T} + i)(1+\gamma)^{-i} \\
&\leq \frac{1+\gamma}{\gamma} \cdot \hat{T} \cdot p_v(t_0) + \left(\frac{1+\gamma}{\gamma}\right)^2 \cdot p_v(t_0) \\
&= O(p_v(t_0)\hat{T}/\gamma)
\end{aligned}$$

Summing up over all nodes, we obtain a total of $O(p \cdot \hat{T}/\gamma)$ transmissions. Since all transmission decisions are done independently at random, the Chernoff bounds imply a total of at most $O(p \cdot \hat{T}/\gamma + \log N)$ w.h.p. □

In our MAC protocol, beyond f steps after any initial choice of the access probabilities, $p = O(\log N)$, w.h.p. This is due to the proof of Lemma 2.7 and the fact that for $p \geq c \log N$, the probability that an idle channel is experienced is at most $1/N^c$, so further increasing p has a polynomially small probability. Furthermore, $\hat{T} = O(\log^2 N/\gamma)$ w.h.p. for any constant ϵ given that all nodes v start with $T_v = 1$. Hence, the total number of transmissions of our MAC protocol under a permanent attack that starts after f steps would be bounded by $O(\log^3 N/\gamma^2)$ w.h.p.

3. An Application to Leader Election. Robust medium access techniques can constitute an important building block for many robust applications. In this section, we provide an exemplary application to the classic leader election problem where n nodes need to agree on a single leader among them. Concretely, our goal is to design a leader election protocol that is self-stabilizing despite adversarial jamming.

Following the usual notation in the self-stabilization literature, the *system state* is determined by the state of all *variables* in the system. That is, the protocol and any constants used by the protocol are assumed to be immutable and not part of the system state. A system is called *self-stabilizing* if and only if (1) when starting from any state, it is guaranteed to eventually reach a legal state (convergence) and (2) given that the system is in a legal state, it is guaranteed to stay in a legal state (closure), provided that there are no faults or membership changes in the system. In our case, roughly speaking, the legal state is the state in which we have exactly one leader.

We will define the set of legal states more formally when we introduce our protocol. While our protocol is randomized and the leader election has to be performed under adversarial jamming, our protocol is still guaranteed to eventually elect exactly one leader from any initial state.

3.1. The SELECT Protocol. Our leader election algorithm (called SELECT for *SElf-stabilizing Leader EleCTion*) is based on the ideas introduced for the medium access protocol. Again, each node v maintains a parameter p_v which describes v's probability of accessing the medium at a given moment of time. The nodes adapt and *synchronize* their p_v values over time in a multiplicative increase multiplicative decrease manner, i.e., the value is lowered in times of high interference or increased during times where the channel is idling. However, p_v will never exceed $\hat{p}$, for some constant $0 < \hat{p} < 1$.

In addition, each node maintains two variables, a threshold variable T_v and a counter variable c_v. Again, T_v is used to estimate the adversary's time window T: a good estimation of T can help the nodes recover from a situation where they experience high interference in the network. In times of high interference, T_v will be increased and the sending probability p_v will be decreased.

Initially, every node v sets $c_v := 1$ and $p_v := \hat{p}$. Note however that while we provide some initial values for the variables in our description, our protocol is self-stabilizing and works for *any* initial variable values, as we will show in our proofs.

SELECT distinguishes between two node roles: *follower* and *leader*. We use s_v to indicate the role of the node: $s_v = 1$ means that node v is a leader, whereas $s_v = 0$ means v is a follower. The basic idea of our protocol is to divide time into intervals of a small number of rounds specified by the constant parameter $b > 5$ (we use the variable mc as a modulo counter); in the following, we will refer to a sequence of rounds between two consecutive $mc = 0$ events as a b-*interval*. (Of course, it can happen that all b slots of an interval are jammed.)

Our protocol is based on the concept of so-called *leader slots*, special rounds—in each b-interval through which SELECT cycles—in which leaders are obliged to send an alive message (a so-called *leader message*) and in which followers keep silent. The idea is that the followers learn that the leader has left in case of an idling medium during a leader slot (of course, the leader slots may be jammed!) and a new election is triggered automatically.

SELECT uses four *leader slots*:[2] ls_1, ls_2, ls_3 and ls_4. Of course, in the beginning, all nodes may have different ls values and may disagree on which slots during the b-interval are leader slots. However, over time, the nodes synchronize their states and a consistent view emerges. For the synchronization, five temporary variables ls'_0, ls'_1, ls'_2, ls'_3, and ls'_4 are used, which store future ls values.

Depending on whether the node is of type follower or leader, the leader slots are updated

[2] It is an open question whether a protocol with less leader slots can be devised.

Algorithm 1 Leader Election: Follower

```
$mc := c_v \mod b$
if $mc = 0$ then
    $ls_1 := ls'_0, ls_2 := ls'_1, ls_3 := ls'_2, ls_4 := ls'_3$
    $s_v := s'_v$
end if
if ($ls_3 =$ undefined) or ($mc \neq ls_1$ and $mc \neq ls_2$ and $mc \neq ls_3$ and $mc \neq ls_4$) then
    $v$ decides with $p_v$ to send a follower message
    if $v$ sends a follower message then
        the message contains:
        $cc_1 := ls'_0, cc_2 := ls'_1, cc_3 := ls'_2, cc_4 := ls'_3, c_{new} := c_v, T_{new} := T_v, p_{new} := p_v$
    end if
end if
if $v$ does not send a follower message then
    $v$ senses the channel
    if channel is idle then
        if $mc = ls_3$ then
            $s'_v := 1$
            $p_v := \hat{p}$
        else
            $p_v := \min\{(1+\gamma)p_v, \hat{p}\}$
        end if
    else if $v$ receives 'LEADER' then
        $s'_v := 0$
        $ls_3 := undefined$
        $ls'_2 := undefined$
    else if $v$ receives a tuple of $\{cc_1, cc_2, cc_3, cc_4, c_{new}, T_{new}, p_{new}\}$ then
        $T_v := T_{new}$
        $p_v := (1+\gamma)^{-1} p_{new}$
        $c_v := c_{new}$
        $ls'_0 := random(0, b-1)$
        $ls'_1 := cc_1, ls'_2 := cc_2, ls'_3 := cc_3, ls'_4 := cc_4$
    end if
end if
$c_v := c_v + 1$
if $c_v \geq b \cdot T_v$ then
    $c_v := 0$
    if (not CONDITION) then
        $p_v := (1+\gamma)^{-1} p_v, T_v := T_v + 1$
        $ls'_0 := undefined, ls'_1 := undefined, ls'_2 := undefined, ls'_3 := undefined, ls'_4 := undefined$
    else
        $T_v := \max\{T_v - 1, 4\}$
    end if
end if
```

Algorithm 2 Leader Election: Leader

```
$mc := c_v \mod b$
if $mc = 0$ then
    $ls_1 := ls'_1, ls_2 := ls'_2, ls_3 := ls'_3, ls_4 := ls'_4$
end if
if $mc = ls_1$ or $mc = ls_2$ or $mc = ls_3$ or $mc = ls_4$ then
    $v$ sends the leader message 'LEADER'
else
    $v$ decides with $p_v$ to send 'LEADER'
    if $v$ does not send 'LEADER' then
        $v$ senses the channel
        if channel is idle then
            $p_v := \min\{(1+\gamma)^2 p_v, \hat{p}\}$
        else if $v$ receives a message then
            $p_v := (1+\gamma)^{-1} p_v$
            if message is 'LEADER' then
                $s_v := 0, s'_v := 0$
                $ls_3 := undefined, ls'_2 := undefined$
            else if message is a follower message, i.e., a tuple of $\{cc_1, cc_2, cc_3, cc_4, c_{new}, T_{new}, p_{new}\}$ then
                $c_v := c_{new}, T_v := T_{new}$
                $ls'_1 := cc_1, ls'_2 := cc_2, ls'_3 := cc_3, ls'_4 := cc_4$
            end if
        end if
    end if
end if
$c_v := c_v + 1$
if $c_v \geq b \cdot T_v$ then
    $c_v := 0$
    if (not CONDITION) then
        $p_v := (1+\gamma)^{-1} p_v, T_v := T_v + 1$
        $ls'_0 := undefined, ls'_1 := undefined, ls'_2 := undefined, ls'_3 := undefined, ls'_4 := undefined$
    else
        $T_v := \max\{T_v - 1, 4\}$
    end if
end if
```

FIG. 3.1. *Algorithm for followers (*left*) and leaders (*right*).*

differently: At the beginning of a new b-interval, a leader copies its ls'_i values to the ls_i values. A follower on the other hand copies the ls' values "diagonally" in the sense that ls'_i is copied to ls'_{i+1} for $i \in \{0, 1, 2, 3\}$. As we will see, this mechanism ensures that an elected leader covers the leader slot ls_3 *of each follower*. (SELECT guarantees that the adaptive adversary has no knowledge about the ls_3 slots at all until it is already too late to prevent a successful election.) Another special slot besides ls_3 is ls'_0 which is a random seed to mix the execution for increased robustness.

In Figure 3.1 we give the detailed formal description of the follower and the leader protocol, respectively. Recall that our algorithms can tolerate any initial values of mc, p_v, T_v, c_v, s_v, s'_v, ls_1, ls_2, ls_3, ls_4, ls'_0, ls'_1, ls'_2, ls'_3, ls'_4. For instance, in the beginning, all nodes v may be leaders and for all v, $s_v = 1$. However, the fixed parameters used by the algorithms, namely $\hat{p}$, γ, or b, are assumed to be immutable.

Both the follower and the leader algorithm consist of three main parts. The b-interval wise update (Lines $2-4$) makes sure that ls values are refreshed frequently. Lines $6-33$ (in case of a follower) and Lines $5-24$ (in case of a leader) are used for medium access in order to synchronize the nodes' states (by a message that includes c_v, T_v, and p_v values) and give nodes the chance to become or remain leader (by a 'LEADER' message). The last sections of the algorithms are used to react to high interference (by reducing p_v) and to reset leader slots. The reason for checking whether ls_3 is undefined in Line 6 of the follower protocol is to keep the leader slots hidden from the adaptive adversary until it is already too late to prevent a successful leader election.[3]

Both the follower and the leader protocol depend on the following crucial CONDITION.

DEFINITION 3.1 (CONDITION). *We define* CONDITION *(Line* 37 *for followers, and Line* 28 *for leaders) as the event that at least one 'LEADER' message was received during the past* $b \cdot T_v$ *steps.*

The idea is that if CONDITION is fulfilled, we know that the protocol is already in a good state. Moreover, we will see that the adversary cannot prevent CONDITION to become true for a long time as the T_v values would continue to increase.

Finally, also note that leaders increase p_v faster (i.e., by larger multiplicative factors) during idle rounds than followers. With this mechanism, SELECT improves the likelihood that a 'LEADER' message gets through and hence that a unique leader is elected.

3.2. Analysis. This section shows that the randomized SELECT protocol is guaranteed to eventually reach a situation where there is exactly one leader and $n-1$ followers. Concretely, we will derive the following theorem.

THEOREM 3.2. *Given an arbitrary initial configuration and in the absence of state faults, our leader election protocol reaches a state where there is exactly one leader and* $n-1$ *followers, despite an adaptive* $(T, 1-\epsilon)$*-bounded jammer, for any* T *and any constant* $\epsilon > 0$.

We make use of the following definitions. First, we define the system state.

DEFINITION 3.3 (State and System State). *The* state of node v *is determined by the state of the variables* p_v, T_v, c_v, s_v, s'_v, mc, ls'_0, ls_1, ls'_1, ls_2, ls'_2, ls_3, ls'_3, ls_4 *and* ls'_4. *The* state of the system *is the set of the states of all nodes.*

We use the following LS_L set to describe the union of all possible leader slot values present in the system.

DEFINITION 3.4 (The LS_L State Set). *For any given system state, let* $LS_L = \{ls_1(v), ls_2(v), ls_3(v), ls_4(v) \mid v \text{ is leader}\} \setminus \{undefined\}$.

The system can be in several special states which are formalized next: follower states, pre-leader states, and leader states. Let $[b] = \{0, \dots, b-1\}$.

DEFINITION 3.5 (Follower State). *A state* S *is called a* follower state*, denoted by* $S \in$ FOLLOWER*, if all the following conditions hold.* ***(i)*** *All nodes are followers* $(\forall v \in V : s_v = 0)$*;* ***(ii)*** *for every node v:* $ls_1(v), ls_2(v), ls_3(v), ls_4(v) \in [b] \cup \{undefined\}$, $ls'_1(v), ls'_2(v), ls'_3(v), ls'_4(v) \in [b] \cup \{undefined\}$, $ls'_0(v) \in [b]$*;* ***(iii)*** *the follower nodes can be partitioned into two sets* $\{v\}$ *and* $V \setminus \{v\}$*, according to their* ls' *values (*v *is the node that successfully sent the last follower message); for each* $w \in V \setminus \{v\}$*:* $ls'_1(w) = ls'_0(v)$, $ls'_2(w) = ls'_1(v)$, $ls'_3(w) = ls'_2(v)$, $ls'_4(w) = ls'_3(v)$*, and* $ls_2(w) = ls_1(v)$, $ls_3(w) = ls_2(v)$*, and* $ls_4(w) = ls_3(v)$*;* ***(iv)*** *for any pair of follower nodes* $v, w \in V$ *with* $ls'_2(v) \in [b]$ *and* $ls_3(v) \in [b]$, $c_v = c_w$ *and* $T_v = T_w$.

We use the concept of so-called *pre-leader states*, i.e., states that result from follower states before some nodes become leaders.

[3] This check allows the adversary to be even reactive.

DEFINITION 3.6 (Pre-leader State). *A state S is called a* pre- leader state*, denoted by $S \in \mathrm{PRE-LEADER}$, if it is a follower state, and at least one follower node v has $s'_v = 1$.*

While in the beginning, the leader sets may be large as each node regards different slots during the b-interval as the "leader slots", over time the values synchronize and the LS sets become smaller. This facilitates a fast leader (re-) election.

DEFINITION 3.7 (Leader State). *A state S is called a* leader state*, denoted by $S \in \mathrm{LEADER}$, if all the following conditions are satisfied:*

***(i)** There is at least one leader, i.e., $|\{v|v \in V : s_v = 1\}| \geq 1$; **(ii)** for every node v, $ls_1(v), ls_2(v), ls_3(v), ls_4(v) \in [b] \cup \{undefined\}$, $ls'_1(v), ls'_2(v), ls'_3(v), ls'_4(v) \in [b] \cup \{undefined\}$, $ls'_0(v) \in [b]$; **(iii)** let v be any follower and let w be any follower or leader, then $ls_3(v) \in \{ls_1(w), ls_2(w), ls_3(w), ls_4(w)\} \cup \{undefined\}$, $ls'_2(v) \in \{ls'_0(w), ls'_1(w), ls'_2(w), ls'_3(w)\} \cup \{undefined\}$; **(iv)** $|LS_L| \leq 5$; **(v)** for every follower w with $ls_3(w) \in [b]$ or $ls'_2(w) \in [b]$, $c_w = c_v$ and $T_w = T_v$ for any leader v.*

So in a leader state, it holds that any follower's ls_3 and ls'_2 slots are covered by either another follower's ls and ls' slots, or a leader's ls and ls slots (cf Condition (iii)).

Finally, it is useful to define safe and legal states.

DEFINITION 3.8 (Safe and Legal State). *A system state S is called* safe *(denoted by $S \in \mathrm{SAFE}$) if $S \in \mathrm{FOLLOWER}$ or $S \in \mathrm{LEADER}$, and* legal *(denoted by $S \in \mathrm{LEGAL}$) if S is safe and there is exactly one node v with $s_v = 1$.*

Thus, according to our definitions, any legal state is also a safe state. In the following, let $\mathcal{S}$ be the set of all possible system states, $\mathrm{SAFE} \subset \mathcal{S}$ be the set of all *safe* system states and $\mathrm{LEGAL} \subset \mathrm{SAFE}$ be the set of all *legal* system states.

The proof of Theorem 3.2 unfolds in a number of lemmas. An interesting property of our randomized algorithm is that it is *guaranteed* to be correct, in the sense that deterministically exactly one leader is elected; only the runtime is probabilistic (i.e., depends on the random choices made by SELECT).

First, we study leader messages.

LEMMA 3.9. *For any network state it holds that if a leader successfully transmits a 'LEADER' message, the system will immediately enter a legal state.*

Proof. When a node (either follower or leader) receives a 'LEADER' message, it sets ls_3 and ls'_2 to $undefined$ (Lines $22-25$ in Figure 3.1 *left*; after Lines $15-17$ of Figure 3.1 *right*), and considers itself a follower. Thus, in the new state, there is exactly one leader (the sender of the 'LEADER' message) and $n-1$ followers. The state is also a safe state, namely a leader state: Conditions (i) and (ii) are fulfilled trivially. Condition (iv) also holds as there is only one leader that has four slots. Condition (iii) is fulfilled because nodes receiving a 'LEADER' message reset their slots ls_3 and ls'_2; since ls_3 and ls'_2 are undefined for a follower, also Condition (v) holds. □

We next consider what happens if nodes hear a message sent by a follower.

LEMMA 3.10. *For any network state it holds that when a follower successfully transmits a message, the system is guaranteed to enter a safe state at the beginning of the next b-interval.*

Proof. First note that if a leader message gets through before the next b-interval, the claim holds trivially due to Lemma 3.9.

Otherwise we distinguish two cases: (A) For every node v, $s'_v = 0$ (not pre-leader) and $s_v = 0$ (not leader) by the end of current b-interval. (B) There is at least one node v with either $s'_v = 1$ (pre-leader) or $s_v = 1$ (leader) by the end of current b-interval.

In Case (A), after the follower message has been successfully sent, there are still n followers and no leaders or pre-leaders. We will show that the system enters the follower state at the beginning of the next b-interval. Let us refer to the follower node that sent the message by v and to any remaining node by w. When w receives the message from v (Lines $26 - 32$ in Figure 3.1 *left*), it sets $ls'_1(w) := ls'_0(v)$, $ls'_2(w) := ls'_1(v)$, $ls'_3(w) := ls'_2(v)$, and $ls'_4(w) := ls'_3(v)$. The c values become the same ($c_w = c_v$), and $T_w := T_v$. The new state therefore fulfills the follower state conditions: Clearly, Conditions $(i), (ii)$, and (iv) are fulfilled immediately, and Condition (iii) holds as well, as for all followers w that did not send a message and follower v which sent a message, at the beginning of the next b-interval: $ls_3(w) = ls'_2(w) = ls'_1(v) = ls_2(v)$, $ls_3(v) = ls'_2(v) = ls'_3(w) = ls_4(w)$, and $ls_1(v) = ls_2(w) = ls'_0(v) = ls'_1(w)$.

For Case (B), observe that during the remainder of the b-interval the number of pre-leader nodes with $s'_v = 1$ cannot decrease, and hence there will be at least one leader at the beginning of the next b-interval. We now show that the new state will indeed be a leader state as nodes "synchronize" with the follower node that sent the message. Without loss of generality, assume that node u is the last follower that successfully sent a follower message in the current b-interval. Let us refer to the other follower nodes by v_1 and to the leader nodes or the pre-leader nodes (i.e., the followers v with $s'_v = 1$) by v_2. Again, Conditions (i) and (ii) are fulfilled trivially. As for Condition (iii), we need to consider two sub-cases:

(Case 1) No node experienced an idle channel in its ls_3 slot after the message has been successfully sent. If this is the case and follower u is not a pre-leader, it holds that for follower v_1: $ls'_2(v_1) = ls'_2(v_2) = ls'_1(u)$ in the current b-interval, and $ls_3(v_1) = ls_2(v_2) = ls_2(u)$ at the beginning of the next b-interval; on the other hand, if follower u is a pre-leader, then in the current b-interval it holds that for follower v_1: $ls'_2(v_1) = ls'_2(v_2) = ls'_1(u)$, and $ls_3(v_1) = ls_2(v_2) = ls_1(u)$ at the beginning of the next b-interval. Hence, Condition (iii) holds. Regarding the cardinality of the leader set LS_L, observe that at the beginning of the next b-interval, if u is not a pre-leader, all leaders will have $ls_1 = ls'_0(u), ls_2 = ls'_1(u), ls_3 = ls'_2(u), ls_4 = ls'_3(u)$, and hence $LS_L = \{ls'_0(u), ls'_1(u), ls'_2(u), ls'_3(u)\}$, therefore $|LS_L| \leq 5$; otherwise, if u is a pre-leader, then $LS_L = \{ls'_0(u), ls'_1(u), ls'_2(u), ls'_3(u), ls'_4(u)\}$, therefore $|LS_L| \leq 5$.

(Case 2) One or more nodes experienced an idle channel in their ls_3 slots after the message has been successfully sent. In the following, we prove this case correct assuming that u is a follower and not a pre-leader. If u is a pre-leader, the proof is analogous.

1. If v_1 experienced the idle channel at its ls_3 time slot, and became a pre-leader:
   Note that a node v_1 may experience an idle channel after receiving the message from u and hence become a pre-leader, however Condition (iii) is still satisfied, as it holds that for follower u: $ls'_2(u) = ls'_3(v_2) = ls'_3(v_1)$ in the current b-interval and $ls_3(u) = ls_3(v_2) = ls_3(v_1)$ at the beginning of the next b-interval. As for the cardinality of the leader set LS_L, observe that at the beginning of the next b-interval, all leaders will have $ls_1 = ls'_0(u), ls_2 = ls'_1(u), ls_3 = ls'_2(u), ls_4 = ls'_3(u)$, and hence $LS_L = \{ls'_0(u), ls'_1(u), ls'_2(u), ls'_3(u)\}$, therefore $|LS_L| \leq 5$.
2. If u experienced the idle channel at its ls_3 time slot, and became a pre-leader:
   If node u experienced an idle channel after successfully sending the message, u became a pre-leader, and we have for a follower v_1, $ls'_2(v_1) = ls'_2(v_2) = ls'_1(u)$ in the current b-interval and $ls_3(v_1) = ls_2(v_2) = ls_1(u)$ at the beginning of the next b-interval. Hence, Condition (iii) is satisfied. As for $|LS_L|$, observe that at the beginning of the next b-interval, for a leader v_2, $ls_1 = ls'_0(u), ls_2 = ls'_1(u), ls_3 = ls'_2(u), ls_4 = ls'_3(u)$, while for the remaining leader u, it holds that $ls_1 = ls'_1(u), ls_2 = ls'_2(u), ls_3 = ls'_3(u), ls_4 = ls'_4(u)$. Hence, also in this case, we have that $|LS_L| \leq 5$.

Finally, Condition (v) is true for both of the sub-cases, because the c_v and T_v values are "synchronized" when the follower message is received (Lines 27 and 29 in Figure 3.1 *left*;

Line 19 in Figure 3.1 *right*). □

An important property of SELECT is that once it is in a safe state, it will remain so in future (given that there are no external changes). Similar properties can be derived for other states, as we will see.

LEMMA 3.11. *Once the system is in a safe state, it will remain in a safe state in the future.*

Proof. We study what can happen in one round, and show that in each case, the safety properties are maintained. In a round, (A) either a 'LEADER' message is successfully sent, (B) a follower message is successfully sent, (C) there are collisions or the channel is jammed, or (D) there is an idle channel.

In Case (A), the claim directly follows from Lemma 3.9 and from the fact that safe states are a super set of the legal states (SAFE $\supset$ LEGAL). In Case (B), the claim follows from Lemma 3.10 and by the fact that the system is in the safe state already.

In Case (C), if the channel is blocked, follower nodes (even those which sent a message in this round) do not change their state except for the synchronized rounds in Lines $35-43$, and similarly for the leaders in Lines $26-34$. Our protocols guarantee that the leaders have the same c_v and T_v values as the followers when ls_3 and ls'_2 are valid, and since the leaders experience the same number of successful transmissions and idle time steps as the followers do (single-hop network), the claim follows.

If there is an idle channel (Case (D)), all nodes v for which $ls_3(v) = mc$ will set $s'_v = 1$ in the current b-interval, while other values remain the same. It is clear that from this point on until the end of the current b-interval, the claim holds. Moreover, as we show next, the claim is still true at the beginning of next b-interval. If $ls_3(v)$ is undefined, then the claim holds trivially, as no states will change in this case. If $ls_3(v) = mc$ for any node v and the nodes experience an idle channel, there is no leader since, if there was a leader, according to Condition (iii) of the leader state definition (Definition 3.7), a follower's ls_3 slot would always be covered by a leader slot of a leader, which yields the contradiction. Hence, the current safe state must be a pre-leader state. Let v denote the followers that have $s'_v = 0$ (i.e., they are not pre-leaders); let u denote the followers with $s'_u = 1$ (pre-leaders). In the current b-interval, we have $ls'_2(v) \in \{ls'_0(u), ls'_1(u), ls'_2(u), ls'_3(u)\} \cup \{undefined\}$, which is true according to Condition (iii) of the follower state definition (Definition 3.5). Then, at the beginning of next b-interval, u will become a leader, and hence we have $ls_3(v) = ls'_2(v)$, $ls_1(u) = ls'_1(u)$, $ls_2(u) = ls'_2(u)$, and $ls_3(u) = ls'_3(u)$. This implies that $ls_3(v) \in \{ls'_0(u), ls_1(u), ls_2(u), ls_3(u)\}$, which satisfies Condition (iii) of the leader state Definition 3.7. Conditions (i) and (ii) are clearly satisfied. Condition (iv) holds simply because we have shown (in Lemma 3.10, Case (B)), when there is an idle time step, $|LS_L| \leq 5$. Condition (v) is true because we always synchronize the c_v and T_v values.□

LEMMA 3.12. *Once a system is in a leader state, it will remain in a leader state in the future.*

Proof. Lemma 3.11 tells us that the system will never leave a safe state. Therefore, it remains to prove that there will always be at least one node v with $s_v = 1$. This clearly holds as the only way a leader can become a follower again is by receiving a 'LEADER' message (see Lines $15-17$), which of course implies that another leader is still active and remains to be a leader. Also, since we are in a leader state, Condition (v) holds and it further implies that leaders will never invalidate their ls slots before the followers. This guarantees that the protocol will never get out of a leader state. □

LEMMA 3.13. *Once a system is in a legal state, it will remain in a legal state in the future.*

Proof. By Lemma 3.11, we know that our system will never leave a safe state again, and hence, we only need to prove that there will always be exactly one node v with $s_v = 1$. This is true because in the safe state, a follower node w can never become a leader, as its $ls_3(w)$ slot is covered by the leader v: $ls_3(w) \in \{ls_1(v), ls_2(v), ls_3(v), ls_4(v)\}$ and $ls'_2(w) \in \{ls'_0(v), ls'_1(v), ls'_2(v), ls'_3(v)\}$ (Condition (iii) of leader state). Since a follower will never send a 'LEADER' message, v will remain a leader forever, which proves the claim. □

Regarding convergence, note that the system quickly enters a safe state, deterministically.

LEMMA 3.14. *For any initial system state with $\hat{T} = \max_v T_v$, it takes at most $b \cdot \hat{T}$ rounds until the system is in a safe state.*

Proof. We distinguish three cases: if a leader message gets through sometimes in these rounds, then the claim holds by Lemma 3.9; if a follower message gets through, then the claim holds by Lemma 3.10. If within $\max_v T_v b$ rounds neither a follower message nor a leader message gets through, all nodes will have to reset their ls slots (since CONDITION in Line 37 (Figure 3.1 *left*) resp. Line 28 (Figure 3.1 *right*) is not met). This however constitutes the safe state (all conditions fulfilled trivially), which is maintained according to Lemma 3.11. □

Armed with these results, we can prove convergence.

LEMMA 3.15. *For any safe state,* SELECT *will eventually reach a legal state.*

Proof. We divide the proof in two phases: the phase where the protocol transitions to the leader state from the follower state, and the phase where it transitions to the legal state from the leader state.

1. Follower state to leader state
   If CONDITION is fulfilled, we know that a 'LEADER' message got through and the system is in a legal state (and hence also in a leader state). As long as CONDITION is not fulfilled, T_v is increasing for each node v. So eventually, $\hat{T} = \max_v T_v \geq 2T/b$. We can also provide a lower bound on the cumulative probability p. W.l.o.g. suppose that $T \geq (3/\epsilon)\log_{1+\gamma} n$ (a smaller T will only make the jammer less flexible and weaker). Suppose that p is at most $\epsilon/4$ throughout some T-interval I. Then it follows from the standard Chernoff bounds that there are at most $\epsilon T/3$ busy steps in I with high probability. If this is true, then no matter how the adversary jams during I, at least $(1-\epsilon/3)T - (1-\epsilon)T = 2\epsilon T/3$ non-jammed steps will be idle, which implies that the cumulative probability at the end of I will be by a factor of at least $(1+\gamma)^{\epsilon T/3} \geq n^3$ higher than at the beginning of I. Using this insight, it follows that eventually a T-interval is reached with $p > \epsilon/4$. Once such a T-interval has been reached, it is easy to show that p will not get below $1/n^2$ any more w.h.p. so that for every T-interval afterwards there is a time point t with $p > \epsilon/4$ w.h.p. So infinitely often the following event can take place with some lower-bounded, positive probability:
   Consider two consecutive T-intervals I_1 and I_2 starting at a time when $c_v = 0$ for every node v. Suppose that I_1 just consists of busy steps and I_2 just consists of idle time steps. Then the adversary has to leave ϵT busy time steps in I_1 non-jammed and ϵT idle time steps in I_2 non-jammed. For I_1, there is a positive probability in this case that exactly 3 messages from different nodes are successfully sent in 3 different b-intervals. In this case, all but one follower respect the leader slots (as their ls_3-value is defined) while the follower that sent the last successful message may still send out messages at *all* time steps (as its ls_3-value is still undefined, see Line 6 of the follower protocol). Thus, it is indeed possible that all time steps in I_1 are busy. Up to that point, the adversary has not learned anything about the leader slots. In I_2, there is also a positive probability that none of the followers transmits a

message throughout I_2 so that all time steps are idle. As the adversary does not know which of them is a leader slot and has to leave ϵT non-jammed, there is a positive probability that ls_3 is non-jammed, and some of the followers become pre-leaders and then leaders.

Thus, the expected time to get from a follower to a leader state is finite.

2. Leader state to legal state

   If there is only one leader in the leader state, the system is already in a legal state by definition. If there is more than one leader, then we distinguish between the following cases. If CONDITION is fulfilled, we know that a 'LEADER' message got through and the system is in a legal state. Otherwise, the leaders will invalidate all of their ls slots once their c_v values are reset to 0. At this point there is a positive probability that for the next T steps a 'LEADER' message is successfully sent. As the adversary has to leave ϵT time steps non-jammed, at least one 'LEADER' message will be successfully transmitted within these T steps so that the system reaches a legal state.

   Analogous to the followers in the previous case, one can lower bound the cumulative probability of the leaders (in fact, the leaders will eventually reach a time point with a cumulative probability of $\Omega(\epsilon)$ as they increase their probabilities in case of an idle channel more aggressively than the followers) so that the chance above of successfully transmitting a 'LEADER' message repeats itself infinitely often with a lower-bounded positive probability. Thus, the expected time to get from a leader to a legal state is finite as well.

From these cases, the lemma follows. □

4. Conclusion. This article presented the first medium access scheme robust to a wide range of interference types which even include adaptive jamming, together with a rigorous analysis proving an asymptotically optimal, constant competitive throughput. We regard this result as an important step towards a better understanding of more complex protocol or physical models for signal propagation. Moreover, as we have shown for the case of leader election, such a protocol can also serve as a basis for robust applications. Another important direction for future research regards the study of dynamical aspects, e.g.: How can a MAC algorithm adapt to join and leave behavior or mobility of the nodes, and which rate is sustainable without losing a constant competitiveness?

REFERENCES

[1] G. Alnifie and R. Simon. A multi-channel defense against jamming attacks in wireless sensor networks. In *Proc. of Q2SWinet '07*, pages 95–104, 2007.

[2] Gheorghe Antonoiu, Gheorghe Antonoiu, Pradip K Srimani, and Pradip K Srimani. A self-stabilizing leader election algorithm for tree graphs. *Journal of Parallel and Distributed Computing*, 34:227–232, 1996.

[3] Hagit Attiya and Jennifer Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics (Chapter 3).* John Wiley & Sons, 2004.

[4] B. Awerbuch. Optimal distributed algorithms for minimum weight spanning tree, counting, leader election, and related problems. In *Proc. STOC*, 1987.

[5] Baruch Awerbuch, Andrea Richa, and Christian Scheideler. A jamming-resistant mac protocol for single-hop wireless networks. In *Proc. 27th ACM Symposium on Principles of Distributed Computing (PODC)*, 2008.

[6] E. Bayraktaroglu, C. King, X. Liu, G. Noubir, R. Rajaraman, and B. Thapa. On the performance of IEEE 802.11 under jamming. In *Proc. of IEEE Infocom '08*, 2008.

[7] Michael A. Bender, Martin Farach-Colton, Simai He, Bradley C. Kuszmaul, and Charles E. Leiserson. Adversarial contention resolution for simple channels. In *Proc. of SPAA '05*, pages 325–332, 2005.

[8] T. Brown, J. James, and A. Sethi. Jamming and sensing of encrypted wireless ad hoc networks. In *Proc. of MobiHoc '06*, pages 120–130, 2006.

[9] Shukai Cai, Taisuke Izumi, and Koichi Wada. Space complexity of self-stabilizing leader election in passively-mobile anonymous agents. In *Proc. 16th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, 2009.

[10] J.T. Chiang and Y.-C. Hu. Cross-layer jamming detection and mitigation in wireless broadcast networks. In *Proc. of MobiCom '07*, pages 346–349, 2007.

[11] Bogdan S. Chlebus, Dariusz R. Kowalski, and Mariusz A. Rokicki. Adversarial queuing on the multiple-access channel. In *Proc. of PODC '06*, pages 92–101, 2006.

[12] A. Czumaj and W. Rytter. Broadcasting algorithms in radio networks with unknown topology. *Journal of Algorithms*, 60(2):115 – 143, 2006.

[13] Edsger W. Dijkstra. Self-stabilization in spite of distributed control. *Communications of the ACM*, 17(11):643–644, 1974.

[14] R. G. Gallager, P. A. Humblet, and P. M. Spira. A distributed algorithm for minimum-weight spanning trees. *ACM Trans. Program. Lang. Syst.*, 5(1):66–77, 1983.

[15] Sukumar Ghosh and Arobinda Gupta. An exercise in fault-containment: self-stabilizing leader election. *Inf. Process. Lett.*, 59(5):281–288, 1996.

[16] S. Gilbert, R. Guerraoui, and C. Newport. Of malicious motes and suspicious sensors: On the efficiency of malicious interference in wireless networks. In *Proc. of OPODIS '06*, 2006.

[17] Leslie Ann Goldberg, Philip D. Mackenzie, Mike Paterson, and Aravind Srinivasan. Contention resolution with constant expected delay. *Journal of the ACM*, 47(6):1048–1096, 2000.

[18] Johan Hastad, Tom Leighton, and Brian Rogoff. Analysis of backoff protocols for mulitiple access channels. *SIAM Journal on Computing*, 25(4):740–774, 1996.

[19] J. Hromkovic, R. Klasing, A. Pelc, P. Ruzicka, and W. Unger. *Dissemination of Information in Communication Networks: Broadcasting, Gossiping, Leader Election, and Fault-Tolerance (Chapter 8)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[20] Gene Itkis, Chengdian Lin, and Janos Simon. Deterministic, constant space, self-stabilizing leader election on uniform rings. In *Proc. 9th International Workshop on Distributed Algorithms (WDAG)*, pages 288–302, 1995.

[21] C.Y. Koo, V. Bhandari, J. Katz, and N.H. Vaidya. Reliable broadcast in radio networks: The bounded collision case. In *Proc. of PODC '06*, 2006.

[22] E. Korach, S. Kutten, and S. Moran. A modular technique for the design of efficient distributed leader finding algorithms. *ACM Trans. Program. Lang. Syst.*, 12(1):84–101, 1990.

[23] Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. Radio network clustering from scratch. In *Proc. of ESA '04*, 2004.

[24] Shay Kutten and Boaz Patt-Shamir. Time-adaptive self stabilization. In *Proc. PODC*, 1997.

[25] Byung-Jae Kwak, Nah-Oak Song, and Leonard E. Miller. Performance analysis of exponential backoff. *IEEE/ACM Transactions on Networking*, 13(2):343–355, 2005.

[26] Y.W. Law, L. van Hoesel, J. Doumen, P. Hartel, and P. Havinga. Energy-efficient link-layer jamming attacks against wireless sensor network mac protocols. In *Proc. of SASN '05*, pages 76–88, 2005.

[27] Seungjoon Lee, Dave Levin, Vijay Gopalakrishnan, and Bobby Bhattacharjee. Backbone construction in selfish wireless networks. In *Proc. ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, 2007.

[28] M. Li, I. Koutsopoulos, and R. Poovendran. Optimal jamming attacks and network defense policies in wireless sensor networks. In *Proc. of Infocom '07*, pages 1307–1315, 2007.

[29] Xin Liu, Guevara Noubir, Ravi Sundaram, and San Tan. Spread: Foiling smart jammers using multi-layer agility. In *Proc. of Infocom '07*, pages 2536–2540, 2007.

[30] Koji Nakano and Stephan Olariu. Randomized leader election protocols in radio networks with no collision detection. In *ISAAC '00: Proceedings of the 11th International Conference on Algorithms and Computation*, pages 362–373, London, UK, 2000. Springer-Verlag.

[31] Vishnu Navda, Aniruddha Bohra, Samrat Ganguly, and Dan Rubenstein. Using channel hopping to increase 802.11 resilience to jamming attacks. In *Proc. of Infocom '07*, pages 2526–2530, 2007.

[32] R. Negi and A. Perrig. Jamming analysis of MAC protocols. Technical report, Carnegie Mellon University, 2003.

[33] Koji Nikano and Stephan Olariu. Uniform leader election protocols for radio networks. *IEEE Trans. Parallel Distrib. Syst.*, 13(5):516–526, 2002.

[34] A. Pelc and D. Peleg. Feasibility and complexity of broadcasting with random transmission failures. In *Proc. of PODC '05*, 2005.

[35] Prabhakar Raghavan and Eli Upfal. Stochastic contention resolution with short delays. *SIAM Journal on Computing*, 28(2):709–719, 1999.

[36] Andréa Richa, Christian Scheideler, Stefan Schmid, and Jin Zhang. Self-stabilizing leader election for single-hop wireless networks despite jamming. In *Proc. 12th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2011.

[37] J. Schmidt, A. Siegel, and A. Srinivasan. Chernoff-Hoeffding bounds for applications with limited indepen-

dence. *SIAM Journal on Discrete Mathematics*, 8(2):223–250, 1995.

[38] M. K. Simon, J. K. Omura, R. A. Schultz, and B. K. Levin. *Spread Spectrum Communications Handbook*. McGraw-Hill, 2001.

[39] Georgios Smaragdakis, Ibrahim Matta, and Azer Bestavros. Sep: A stable election protocol for clustered heterogeneous wireless sensor networks. In *Proc. 2nd International Workshop on Sensor and Actor Network Protocols and Applications (SANPA)*, 2004.

[40] David Thuente and Mithun Acharya. Intelligent jamming in wireless networks with applications to 802.11b and other networks. In *Proc. of MILCOM '06*, 2006.

[41] Dan E Willard. Log-logarithmic selection resolution protocols in a multiple access channel. *SIAM J. Comput.*, 15(2):468–477, 1986.

[42] A.D. Wood, J.A. Stankovic, and G. Zhou. DEEJAM: Defeating energy-efficient jamming in IEEE 802.15.4-based wireless networks. In *Proc. of SECON '07*, 2007.

[43] W. Xu, K. Ma, W. Trappe, and Y. Zhang. Jamming sensor networks: attack and defense strategies. *IEEE Network*, 20(3):41–47, 2006.

[44] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proc. of MobiHoc '05*, pages 46–57, 2005.

[45] W. Xu, T. Wood, and Y. Zhang. Channel surfing and spatial retreats: defenses against wireless denial of service. In *Proc. of Workshop on Wireless Security*, 2004.