

CRITICAL DENSITY OF ACTIVATED RANDOM WALKS ON TRANSITIVE GRAPHS

BY ALEXANDRE STAUFFER^{*,†} AND LORENZO TAGGI^{†,§}

University of Bath[†] and Technische Universität Darmstadt[§]

We consider the activated random walk model on general vertex-transitive graphs. A central question in this model is whether the critical density μ_c for sustained activity is strictly between 0 and 1. It was known that $\mu_c > 0$ on \mathbb{Z}^d , $d \geq 1$, and that $\mu_c < 1$ on \mathbb{Z} for small enough sleeping rate. We show that $\mu_c \rightarrow 0$ as $\lambda \rightarrow 0$ in all vertex-transitive transient graphs, implying that $\mu_c < 1$ for small enough sleeping rate. We also show that $\mu_c < 1$ for any sleeping rate in any vertex-transitive graph in which simple random walk has positive speed. Furthermore, we prove that $\mu_c > 0$ in any vertex-transitive amenable graph, and that $\mu_c \in (0, 1)$ for any sleeping rate on regular trees.

1. Introduction. We consider the activated random walk (ARW) model on a graph $G = (V, E)$. This is a continuous-time interacting particle system with conserved number of particles, where each particle can be in one of two states: A (active) or S (inactive, sleeping). Initially, the number of particles at each vertex of G is an independent Bernoulli random variable of parameter $\mu \in (0, 1]$, usually called the *particle density*, and all particles are of type A. Each A-particle performs an independent, continuous time random walk on G with jump rate 1, and with each jump being to a uniformly random neighbor. Moreover, every A-particle has a Poisson clock of rate $\lambda > 0$ (called the *sleeping rate*). When the clock of a particle rings, if the particle does not share the site with other particles, the transition $A \rightarrow S$ occurs (that is, the particle becomes of type S); otherwise nothing happens. Each S-particle does not move and remains sleeping until another particle jumps into its location. At such an instant, the S-particle turns into type A, giving the transition $A+S \rightarrow 2A$.

For any given λ , it is expected that ARW undergoes a phase transition as μ varies. For example, if μ is very small, there is a lot of empty space

*Supported by a Marie Curie Career Integration Grant PCIG13-GA-2013-618588 DSRELIS, and an EPSRC Early Career Fellowship.

†Supported by a German Research Foundation Grant BE 5267/1.

MSC 2010 subject classifications: Primary 82C22, 60K35; secondary 82C26

Keywords and phrases: interacting particle systems, random walks, absorbing states phase transitions

between particles, which allows each particle to eventually fall asleep (that is, turn into type S) and never become active again. When this happens, we say that *ARW fixates*. When this does not happen, we say that *ARW is active*. This case is expected to occur when μ is large, since active particles will repetitively jump on top of other particles, “waking up” the ones that had turned into type S .

In a seminal paper, Rolla and Sidoravicius [7] showed that this process satisfies a 0-1 law (i.e., the process is either active or fixated with probability 1) and is monotone with respect to μ . This gives the existence of a critical value

$$(1.1) \quad \mu_c = \mu_c(\lambda) := \inf \{ \mu \geq 0 : P(\text{ARW is active}) > 0 \}$$

such that ARW is active almost surely for all $\mu > \mu_c$, and fixates almost surely for all $\mu < \mu_c$. Though [7], as well as almost all existing works, are restricted to the case of G being \mathbb{Z}^d , the above properties hold for any vertex-transitive graph. A graph G is called *vertex transitive* if, for any two vertices $u, v \in V$, there exists a graph automorphism of G mapping u onto v . Throughout this paper we always consider that G is an infinite graph that is locally finite and *vertex transitive*, which ensures the existence of μ_c .

Our definition above implies that $\mu_c \leq 1$ since particles are initially distributed as Bernoulli random variables. However, even if we replace this with any product measure of density $\mu > 0$, it is intuitive that $\mu_c \leq 1$, since at most one particle can fall asleep at any given vertex. This has been established for a large class of graphs [7, 9, 1]. A fundamental and very important problem in activated random walks [7, 3] is whether

$$(1.2) \quad \mu_c \in (0, 1) \text{ for all } \lambda > 0 \text{ and all vertex-transitive graphs.}$$

This problem is widely open, and both sides of the above question (i.e., whether $\mu_c > 0$ or $\mu_c < 1$) turned out to be very challenging. In fact, even showing that $\mu_c < 1$ for *some* $\lambda > 0$ is already quite difficult, and is only known to hold on \mathbb{Z} , thanks to a very recent paper by Basu, Ganguly and Hoffman [2]. Our first theorem establishes this result in all vertex-transitive *transient* graphs, which includes \mathbb{Z}^d , $d \geq 3$.

THEOREM 1.1. *For any vertex-transitive, transient graph, it holds that*

$$\mu_c \rightarrow 0 \text{ as } \lambda \rightarrow 0.$$

More specifically, we have that $\lim_{\lambda \downarrow 0} \frac{\mu_c}{\lambda^{1/4}} < \infty$.

Regarding the question of whether $\mu_c < 1$ for *all* $\lambda > 0$, this has until now not been established for any single graph. A positive answer to this question has only been given for a variant of ARW where particles move according to *biased* random walks on \mathbb{Z}^d ; see Taggi [11]. Rolla and Tournier [8] further extended this result by proving that, for biased ARW on \mathbb{Z}^d , we have $\mu_c(\lambda) \rightarrow 0$ as $\lambda \rightarrow 0$. In our second theorem we give a positive answer to this question for the original, unbiased model, and for all graphs where simple random walk has positive speed. If $(X_t)_{t \in \mathbb{N}}$ is a random walk on G starting from a vertex x , and $|X_t|$ denotes the distance between X_t and x , we say that a random walk on G has *positive speed* if $\liminf_{t \rightarrow \infty} \frac{|X_t|}{t} > 0$ almost surely. This includes, for example, all non-amenable graphs that are vertex transitive.

THEOREM 1.2. *For any vertex-transitive graph such that a random walk on it has positive speed α , it holds that*

$$\mu_c < 1 \quad \text{for all } \lambda > 0.$$

More specifically, we obtain that $\mu_c < 1 - \frac{\alpha\delta}{1+\lambda}$, where δ is the probability that a random walk does not return to the origin.

We prove the theorem above by providing general sufficient conditions for ARW to be active, which as a consequence establishes an upper bound on μ_c . We believe this result is of independent interest and state it in Theorem 5.1.

For the other side of (1.2) (i.e., whether $\mu_c > 0$), there has been a bit more progress. It has been settled when G is \mathbb{Z}^d thanks to the seminal work of Rolla and Sidoravicius [7] for $d = 1$, and an elaborate proof of Sidoravicius and Teixeira [10] for $d \geq 2$. Our next theorem establishes that $\mu_c > 0$ in any vertex-transitive *amenable* graph, which includes \mathbb{Z}^d , $d \geq 1$. We remark that not only our result generalizes the ones in [7, 10], but also provides the additional information that $\mu_c \rightarrow 1$ as $\lambda \rightarrow \infty$. In addition, our proof is quite short in comparison to [10].

THEOREM 1.3. *For any vertex-transitive, amenable graph, we have*

$$\mu_c > 0 \quad \text{for all } \lambda > 0.$$

More specifically, we have $\mu_c \geq \frac{\lambda}{1+\lambda}$.

REMARK 1.4. Our lower bound is sharp, in the sense that there are no better lower bounds for μ_c which are just a function of λ and hold for any vertex-transitive amenable graph and any jump distribution. Indeed, μ_c is known to be equal to $\frac{\lambda}{1+\lambda}$ on \mathbb{Z} with totally asymmetric jumps [6].

REMARK 1.5. Our Theorems 1.1 and 1.3 hold in more generality, for any distribution of the initial location of the particles and for any jump distribution (biased or unbiased) which is translation invariant and has finite support.

Note that Theorems 1.2 and 1.3 provide a final answer to (1.2) in vertex-transitive graphs that are amenable but for which a random walk has positive speed; for example, the so-called lamplighter graphs. In our final result, we also establish (1.2) for the case of regular trees, excluding \mathbb{Z} .

THEOREM 1.6. *When G is a regular tree of degree at least 3, we have*

$$\mu_c \in (0, 1) \quad \text{for all } \lambda > 0.$$

In addition, we have $\mu_c \rightarrow 0$ as $\lambda \rightarrow 0$.

We now give a brief description of our proof techniques. The traditional strategy to establish bounds on μ_c is to consider a ball $B_L \subset V$ of some large radius L , centered at a given vertex $x \in V$, and *stabilize* ARW inside this ball. This consists of letting the process run (i.e., particles move and fall asleep) inside B_L , deleting every particle that exits B_L . This procedure will eventually end. At this point, each vertex of B_L will either contain a sleeping particle or contain no particle; such a vertex is usually called *stable*. It was shown in [7] that, roughly speaking, ARW is active if and only if the number of times particles visit x during the stabilization of B_L goes to infinity with L . In this paper, we introduce a new point of view on such stabilization procedure by focusing on some vertex $y \in B_L$, and carrying out what we call a *weak stabilization of B_L with respect to y* . Intuitively, in the weak stabilization we perform the steps of a stabilization procedure until each vertex of $B_L \setminus \{y\}$ is stable while y is allowed to be either stable or host exactly one active particle. This strategy allows us to estimate the probability that, at the end of a stabilization procedure, y contains a sleeping particle. In principle, the density of sleeping particles should correspond to μ_c , and it is by controlling such probability that we obtain estimates on μ_c . We believe that our weak stabilization procedure and our point of view of estimating the density of sleeping particles have the potential to foster even more substantial progress in this model. In fact, we believe our estimate on the probability that a sleeping particle ends at some vertex is of independent interest, and we state it in Theorem 3.1.

The remaining of the paper is organized as follows. In Section 2, we describe the so-called Diaconis-Fulton representation of ARW and its properties, which we employ in all of our proofs. Then, in Section 3, we introduce

the weak stabilization procedure and estimate the probability of having a sleeping particle at a given vertex (Theorem 3.1). Next we turn to the proofs of our main results: we prove Theorem 1.1 in Section 4, Theorem 1.2 in Section 5, Theorem 1.3 in Section 6, and Theorem 1.6 in Section 7.

2. Diaconis-Fulton representation. In this section we describe the Diaconis-Fulton graphical representation for the dynamics of ARW, following [7]. For a graph $G = (V, E)$, the state of configurations is $\Omega = \{0, \rho, 1, 2, 3, \dots\}^V$, where a vertex being in state ρ denotes that the vertex has one sleeping particle, while being in state $i \in \{0, 1, 2, \dots\}$ denotes that the vertex contains i active particles. We employ the following order on the states of a vertex: $0 < \rho < 1 < 2 < \dots$. In a configuration $\eta \in \Omega$, a site $x \in V$ is called *stable* if $\eta(x) \in \{0, \rho\}$, and it is called *unstable* if $\eta(x) \geq 1$. We fix an array of *instructions* $\tau = (\tau^{x,j} : x \in V, j \in \mathbb{N})$, where $\tau^{x,j}$ can either be of the form τ_{xy} or $\tau_{x\rho}$. We let τ_{xy} with $x, y \in V$ denote the instruction that a particle from x jumps to vertex y , and $\tau_{x\rho}$ denote the instruction that a particle from x falls asleep. Henceforth we call τ_{xy} a *jump instruction* and $\tau_{x\rho}$ a *sleep instruction*. Therefore, given any configuration η , performing the instruction τ_{xy} in η yields another configuration η' such that $\eta'(z) = \eta(z)$ for all $z \in V \setminus \{x, y\}$, $\eta'(x) = \eta(x) - \mathbf{1}(\eta(x) \geq 1)$, and $\eta'(y) = \eta(y) + \mathbf{1}(\eta(x) \geq 1)$. We use the convention that $1 + \rho = 2$. Similarly, performing the instruction $\tau_{x\rho}$ to η yields a configuration η' such that $\eta'(z) = \eta(z)$ for all $z \in V \setminus \{x\}$, and if $\eta(x) = 1$ we have $\eta'(x) = \rho$, otherwise $\eta'(x) = \eta(x)$.

Let $h = (h(x) : x \in V)$ count the number of instructions used at each site. We say that we *use* an instruction at x (or that we *topple* x) when we act on the current particle configuration η through the operator Φ_x , which is defined as,

$$(2.1) \quad \Phi_x(\eta, h) = (\tau^{x, h(x)+1} \eta, h + \delta_x).$$

The operation Φ_x is *legal* for η if x is unstable in η and *illegal* otherwise.

Properties. We now describe the properties of this representation. Later we discuss how they are related to the stochastic dynamics of ARW. For a sequence of vertices $\alpha = (x_1, x_2, \dots, x_k)$, we write $\Phi_\alpha = \Phi_{x_k} \Phi_{x_{k-1}} \dots \Phi_{x_1}$ and we say that Φ_α is *legal* for η if Φ_{x_ℓ} is legal for $\Phi_{(x_{\ell-1}, \dots, x_1)}(\eta, h)$ for all $\ell \in \{1, 2, \dots, k\}$. Let $m_\alpha = (m_\alpha(x) : x \in V)$ be given by, $m_\alpha(x) = \sum_\ell \mathbf{1}(x_\ell = x)$, the number of times the site x appears in α . We write $m_\alpha \geq m_\beta$ if $m_\alpha(x) \geq m_\beta(x) \forall x \in V$. Analogously we write $\eta' \geq \eta$ if $\eta'(x) \geq \eta(x)$ for all $x \in V$. We also write $(\eta', h') \geq (\eta, h)$ if $\eta' \geq \eta$ and $h' = h$.

Let η, η' be two configurations, x be a vertex in V and τ be an array of instructions. Let V' be a finite subset of V . A configuration η is said to be *stable* in V' if all the sites $x \in V'$ are stable. We say that α is contained in V' if all its elements are in V' , and we say that α *stabilizes* η in V' if every $x \in V'$ is stable in $\Phi_\alpha \eta$. We now state some fundamental properties of the Diaconis-Fulton representation. For the proof we refer to [7].

LEMMA 2.1 (Least Action Principle). *If α and β are legal sequences of topplings for η such that β is contained in V and α stabilizes η in V , then $m_\beta \leq m_\alpha$.*

LEMMA 2.2 (Abelian Property). *Given any $V' \subset V$, if α and β are both legal sequences for η that are contained in V' and stabilize η in V' , then $m_\alpha = m_\beta$. In particular, $\Phi_\alpha \eta = \Phi_\beta \eta$.*

For any finite subset $V' \subset V$, any $x \in V'$, any particle configuration η , and any array of instructions τ , we denote by $m_{V', \eta, \tau}(x)$ the number of times that x is toppled in the stabilization of V' starting from configuration η and using the instructions in τ . Note that by Lemma 2.2, we have that $m_{V', \eta, \tau}$ is well defined.

LEMMA 2.3 (Monotonicity). *If $V' \subset V'' \subset V$ and $\eta \leq \eta'$, then $m_{V', \eta, \tau} \leq m_{V'', \eta', \tau}$.*

By monotonicity, given any growing sequence of subsets $V_1 \subseteq V_2 \subseteq V_3 \subseteq \dots \subseteq V$ such that $\lim_{m \rightarrow \infty} V_m = V$, the limit

$$m_{\eta, \tau} = \lim_{m \rightarrow \infty} m_{V_m, \eta, \tau},$$

exists and does not depend on the particular sequence $\{V_m\}_m$.

We now introduce a probability measure on the space of instructions and of particle configurations. We denote by \mathcal{P} the probability measure according to which, for any $x \in V$ and any $j \in \mathbb{N}$, $\mathcal{P}(\tau^{x,j} = \tau_{x\rho}) = \frac{\lambda}{1+\lambda}$ and $\mathcal{P}(\tau^{x,j} = \tau_{xy}) = \frac{1}{d(1+\lambda)}$ for any $y \in V$ neighboring x , where d is the degree of each vertex of G and the $\tau^{x,j}$ are independent across different values of x or j . Finally we denote by $P^\nu = \mathcal{P} \otimes \nu$ the joint law of η and τ , where ν is a distribution on Ω giving the law of η . Let \mathbb{P}^ν denotes the probability measure induced by the ARW process when the initial distribution of particles is given by ν . We shall often omit the dependence on ν by writing P and \mathbb{P} instead of P^ν and \mathbb{P}^ν . The following lemma relates the dynamics of ARW to the stability property of the representation.

LEMMA 2.4 (0-1 law). *Let ν be an automorphism invariant, ergodic distribution with finite density. Let $x \in V$ be any given vertex of G . Then $\mathbb{P}^\nu(\text{ARW fixates}) = P^\nu(m_{\eta,\tau}(x) < \infty) \in \{0, 1\}$.*

Roughly speaking, the next lemma gives that removing a sleep instruction, cannot decrease the number of instructions used at a given vertex for stabilization. In order to state the lemma, consider an additional instruction ι besides τ_{xy} and $\tau_{x\rho}$. The effect of ι is to leave the configuration unchanged; i.e., $\iota\eta = \eta$. Then given two arrays $\tau = (\tau^{x,j})_{x,j}$ and $\tilde{\tau} = (\tilde{\tau}^{x,j})_{x,j}$, we write $\tau \leq \tilde{\tau}$ if for every $x \in V$ and $j \in \mathbb{N}$, we either have $\tilde{\tau}^{x,j} = \tau^{x,j}$ or we have $\tilde{\tau}^{x,j} = \iota$ and $\tau^{x,j} = \tau_{x\rho}$.

LEMMA 2.5 (Monotonicity with enforced activation). *Let τ and $\tilde{\tau}$ be two arrays of instructions such that $\tau \leq \tilde{\tau}$. Then, for any finite subset $V' \subset V$ and configuration $\eta \in \Omega$, we have $m_{V',\eta,\tau} \leq m_{V',\eta,\tilde{\tau}}$.*

When we average over η and τ using the measure P , we will simply write $m_{V'}$ instead of $m_{V',\eta,\tau}$.

3. Weak stabilization. In this section we introduce our method of weak stabilization and use it to derive upper and lower bounds on the probability that a given vertex contains an S -particle at the end of the stabilization of some set. This is the content of Theorem 3.1 below, which will play a fundamental role in the proofs of our main results. For any finite set $K \subset V$ and any vertex $x \in K$, let $Q(x, K)$ be the probability that there is one S -particle at x at the end of the stabilization of K .

THEOREM 3.1. *Consider ARW on a vertex-transitive graph $G = (V, E)$. Then, for any $K \subset V$ and any $x \in K$, we have*

$$(3.1) \quad Q(x, K) \geq \frac{\lambda}{1 + \lambda} P(m_K(x) \geq 1).$$

Moreover, if G is a vertex-transitive, transient graph, then

$$(3.2) \quad Q(x, K) \leq 3\sqrt{\lambda(C_G(1 + \lambda) + 1)},$$

where C_G is the expected number of times a simple random walk on G starting from x visits x .

In the proof of the theorem above we will employ the notion of *weakly stable configurations* and *weak stabilization*.

DEFINITION 3.2 (weakly stable configurations). We say that a configuration η is *weakly stable* in a subset $K \subset V$ with respect to a vertex $x \in K$ if $\eta(x) \leq 1$ and $\eta(y) \leq \rho$ for all $y \in K \setminus \{x\}$. In words, this means that all vertices in $K \setminus \{x\}$ are stable, and x is either stable or hosts at most one active particle. For conciseness, we just write that η is weakly stable for (x, K) .

DEFINITION 3.3 (weak stabilization). Given a subset $K \subset V$ and a vertex $x \in K$, the *weak stabilization* of (x, K) is a sequence of topplings of unstable sites of $K \setminus \{x\}$ and of topplings of x whenever x has at least two active particles, until a weakly stable configuration for (x, K) is obtained. The order of the topplings of a weak stabilization can be arbitrary.

We now formulate the Least Action Principle for weak stabilization of (x, K) . In order to state the lemma, we need to extend the notion of unstable vertex and of legal operations to weak stabilization of (x, K) . We call a vertex y *WS-unstable* (that is, unstable for weak stabilization) in $\eta \in \Omega$ if $\eta(y) \geq 1 + \delta_x(y)$, where $\delta_x(y) = 1$ if $x = y$ and $\delta_x(y) = 0$ otherwise. We call a vertex y *WS-stable* in $\eta \in \Omega$ if it is not WS-unstable. We call the operation Φ_y defined in (2.1) *WS-legal* for η if y is WS-unstable in η . Note that a WS-legal operation is always legal but a legal operation is not necessarily WS-legal. For a sequence of vertices $\alpha = (x_1, x_2, \dots, x_k)$, we say that Φ_α is WS-legal for η if Φ_{x_ℓ} is WS-legal for $\Phi_{(x_{\ell-1}, \dots, x_1)}(\eta, h)$ for all $\ell \in \{1, 2, \dots, k\}$. We say that α *stabilizes* η weakly in (x, K) if every $x \in V$ is WS-stable in $\Phi_\alpha \eta$.

LEMMA 3.4 (Least Action Principle for weak stabilization of (x, K)). *If α and β are sequences of topplings for η such that α is legal and stabilizes η weakly in (x, K) and β is WS-legal and is contained in K , then $m_\beta \leq m_\alpha$.*

PROOF. We follow [7]. The proof consists in observing that all topplings performed by β are necessary for weak stabilization of (x, K) . Let β be a WS-legal sequence contained in K . Assume $m_\alpha \not\leq m_\beta$. Write $\beta = (x_1, \dots, x_k)$ and $\beta^{(j)} = (x_1, \dots, x_j)$ for $j \leq k$. Let $\ell = \max\{j : m_{\beta^{(j)}} \leq m_\alpha\} < k$ and $y = x_{\ell+1} \in K$. Now y is WS-unstable in $\Phi_{\beta^{(\ell)}} \eta$ since β is WS-legal, moreover $m_{\beta^{(\ell)}} \leq m_\alpha$ and $m_{\beta^{(\ell)}}(y) = m_\alpha(y)$ by definition of ℓ . Now note that if y is WS-unstable for $\Phi_{\beta^{(\ell)}} \eta$, then y is WS-unstable also for $\Phi_\alpha \eta$, since toppling sites of $K \setminus \{y\}$ cannot decrease the number of particles at y . Then α does not stabilize η weakly in (x, K) . \square

REMARK 3.5. Consider any finite subset $K \subset V$ and any $x \in K$. The Abelian property (Lemma 2.2), Monotonicity (Lemma 2.3), and Monotonicity with enforced activation (Lemma 2.5) hold true for weak stabilization of (x, K) as well with no change in the proof.

The main idea of the proof of Theorem 3.1 is to perform a certain sequence of topplings to stabilize K that will allow us to control whether there is a sleeping particle at x . From the Abelian property (Lemma 2.2), in order to stabilize K we can perform the topplings in any order we want. We will stabilize K by first weakly stabilizing (x, K) , which gives a weakly stable configuration η_1 for (x, K) . Then either η_1 is stable for K , in which case we finish the stabilization procedure, or $\eta_1(x) = 1$. In the latter case, we topple x and weakly stabilize (x, K) again, obtaining a configuration η_2 . We repeat the above procedure until we obtain a stable configuration for K , concluding the stabilization. We will refer to this stabilization procedure as a *stabilization via weak stabilization*.

3.1. *Proof of the lower bound in Theorem 3.1.* Note that, in a stabilization via weak stabilization, after each weakly stable configuration η_i we obtain, if η_i is not stable, then with probability $\frac{\lambda}{1+\lambda}$ we encounter a sleep instruction at x , transforming η_i into a stable configuration. With this we can derive the lower bound (3.1) in Theorem 3.1.

PROOF OF (3.1) IN THEOREM 3.1. We apply the stabilization of K via weak stabilizations of (x, K) . Let η_1 be the first weakly stable configuration for (x, K) that is obtained in this procedure. As discussed above, if η_1 is not stable for K , then we obtain a stable configuration for K if the next instruction at x is sleep. Hence,

$$Q(x, K) \geq P(\eta_1 \text{ is not stable for } K) \frac{\lambda}{1 + \lambda}.$$

The proof is concluded by noting that the event that η_1 is not stable for K is equivalent to the event that x is toppled at least once. This is true because of the following. If η_1 is not stable for K , then $\eta_1(x) = 1$ which implies that x will be toppled at least once. In the other direction, if x is toppled at least once, then this happens either before η_1 is obtained or because $\eta_1(x) = 1$. But if x was toppled before η_1 was obtained, this must have happened at a time when x had at least two particles. From this time onwards, x will have at least one active particle until η_1 is obtained. Hence, η_1 is not stable. \square

3.2. *Proof of the upper bound in Theorem 3.1.* Our proof of the upper bound (3.2) for $Q(x, K)$ is a bit longer than the proof of the lower bound. We will perform the stabilization of K via weak stabilization as described above. The idea is to estimate the probability that, for any $i \geq 1$, we obtain a stable configuration for K after the i th weak stabilization of (x, K) . We do this by relating this probability to the probability that a random walk starting from x never returns to x . It is at this step that we use that G is transient.

After the i th time we perform the weak stabilization of (x, K) , we let $m_{(x,K)}^i(y)$ be the number of instructions that have been used at $y \in K$ up to this time, and denote by η_i the configuration we then obtained. Also, let $T_{(x,K)}$ denote the number of weak stabilizations of (x, K) we perform until a stable configuration in K is obtained. Note that $\eta_{T_{(x,K)}}$ is either a stable configuration, which implies that $\eta_{T_{(x,K)}}(x) = 0$, or $\eta_{T_{(x,K)}}$ is weakly stable for (x, K) with $\eta_{T_{(x,K)}}(x) = 1$ and the next instruction used at x was a sleep instruction, thereby concluding the stabilization of K . For consistency, for any $i > T_{(x,K)}$, let η_i be the stable configuration obtained after stabilizing K and, for any $y \in K$, define $m_{(x,K)}^i(y) = m_K(y)$, which is the total number of instructions used at y for the complete stabilization of K . By the Abelian property, the quantities $T_{(x,K)}$ and $m_{(x,K)}^i$ are all well defined.

Below we state a lemma, and then show how this lemma implies the upper bound on $Q(x, K)$.

LEMMA 3.6. *Given any vertex-transitive, transient graph $G = (V, E)$, any subset $K \subset V$ and any vertex $x \in K$, and letting C_G be the expected number of visits to x of a random walk on G starting from x , we have*

$$E[T_{(x,K)}] \leq C_G(1 + \lambda) + 1,$$

where the expectation is with respect to the measure P .

PROOF OF (3.2) IN THEOREM 3.1. For simplicity, write $\eta' = \eta_{T_{(x,K)}+1}$ for the configuration obtained after complete stabilization of K . Then the following expression holds, as the sum is over disjoint events,

$$(3.3) \quad Q(x, K) = P(\eta'(x) = \rho) = \sum_{k=1}^{\infty} P(T_{(x,K)} = k, \eta'(x) = \rho).$$

Now observe that

$$(3.4) \quad P(T_{(x,K)} = k, \eta'(x) = \rho) \leq \left(\frac{1}{1 + \lambda}\right)^{k-1} \frac{\lambda}{1 + \lambda}.$$

The previous inequality follows from independence of instructions: the event in the left-hand side implies that after each weak stabilization we have an active particle at x , and moreover we encounter a jump instruction at x after each of the first $k - 1$ weak stabilizations, and a sleep instruction at x after the last weak stabilization. Hence, for any $H \geq 1$ we can write

$$\begin{aligned} Q(x, K) &\leq \frac{\lambda}{1+\lambda} \sum_{k=1}^H \left(\frac{1}{1+\lambda} \right)^{k-1} + P(T_{(x,K)} > H) \\ &\leq 1 - \left(\frac{1}{1+\lambda} \right)^H + \frac{E[T_{(x,K)}]}{H}, \end{aligned}$$

where in the last step we used Markov's inequality. From Lemma 3.6, we obtain

$$\begin{aligned} Q(x, K) &\leq 1 - \left(\frac{1}{1+\lambda} \right)^H + \frac{C_G(1+\lambda) + 1}{H} \\ &\leq 1 - (1-\lambda)^H + \frac{C_G(1+\lambda) + 1}{H} \\ &\leq \lambda H + \frac{C_G(1+\lambda) + 1}{H}. \end{aligned}$$

Observe that our estimate holds for any integer $H \geq 1$ and that $C_G \geq 1$. Then, by setting $H = \lfloor \sqrt{\frac{C_G(1+\lambda)+1}{\lambda}} \rfloor$, we get that for any positive λ ,

$$Q(x, K) \leq 3\sqrt{\lambda(C_G(1+\lambda) + 1)}.$$

In the above calculations we used $\frac{x}{\lfloor x \rfloor} \leq 2$ for $x \geq 1$. \square

3.3. Proof of Lemma 3.6. In this section we establish the upper bound on $E[T_{(x,K)}]$ from Lemma 3.6. Let $x \in V$ be a given vertex. Let E^x denote the expectation E conditioned on the initial particle configuration having one active particle at x , and E_x denote the expectation conditioned on the initial particle configuration having no particle at x .

LEMMA 3.7. *For any finite subset $K \subset V$ and vertex $x \in K$ we have*

$$E_x[m_K(x)] \leq E^x[m_{(x,K)}^1(x)].$$

PROOF. Consider an initial particle configuration η having no particle at x , and the particle configuration η^x obtained from η by adding an active particle at x . We will show a stronger result saying that, by using the

same instruction array for both η and η^x , $m_K(x)$ starting from η is at most $m_{(x,K)}^1(x)$ starting from η^x . We stabilize K starting from η via weak stabilization of (x, K) , and do the same topplings for η^x . Since η and η^x differ only at x , until the first weak stabilization of η is concluded, the same topplings can be carried out in η^x as well. At this point, if there is a particle at x in η , there are two particles at x in η^x . Then if the next instruction at x is a jump instruction, we can perform the same toppling in η and η^x , and we repeat this procedure until another weakly stable configuration is obtained in η . On the other hand, if the next instruction at x is a sleep instruction, then the stabilization of η is concluded, but the weak stabilization of η^x continues. Finally, if there is no particle at x at the end of a weak stabilization of η , then the stabilization of η and the weak stabilization of η^x are concluded. Therefore, under this coupling, the weak stabilization of η concludes no later than that of η^x , concluding the proof. \square

PROOF OF LEMMA 3.6. The crucial observation is the following. Assume that $T_{(x,K)} \geq 2$. After each of the first $T_{(x,K)} - 1$ weak stabilizations of (x, K) , we must perform at least one toppling at x , and this toppling happens after the first weak stabilization of (x, K) , so it is not counted in $m_{(x,K)}^1(x)$. This gives that

$$(3.5) \quad T_{(x,K)} - 1 \leq m_K(x) - m_{(x,K)}^1(x).$$

The above bound also holds when $T_{(x,K)} = 1$ since $m_K(x) \geq m_{(x,K)}^1(x)$. Then the lemma follows by claiming that

$$(3.6) \quad E[m_K(x)] \leq E[m_{(x,K)}^1(x)] + C_G(1 + \lambda).$$

First we prove (3.6) with E replaced with E^x . Denote the particle that starts at x by z . From Lemma 2.5, we have that if we ignore some sleep instructions during the stabilization of K (i.e., we replace some sleep instructions in the instruction array τ with neutral instructions ι), the value of $m_K(x)$ can only increase. Therefore, we can bound $m_K(x)$ from above by carrying out a two-step stabilization procedure. In the first step, we move z ignoring any sleep instruction seen until z exits K . We call \mathcal{V} the expected number of topplings at x up to this point. Then, in the second step, we stabilize K in an arbitrary manner. Using Lemma 2.5 as mentioned above, we conclude that

$$E^x[m_K(x)] \leq \mathcal{V} + E_x[m_K(x)].$$

Note that $\mathcal{V} = C_G(1 + \lambda)$, as every time the particle visits x , we find a geometrically distributed number of sleep instructions (which are replaced

by instructions ι) before the particle jumps out of x . The expected number of sleep instructions found at x after every visit is $1 + \lambda$. With this we obtain

$$E^x[m_K(x)] \leq C_G(1 + \lambda) + E_x[m_K(x)] \leq C_G(1 + \lambda) + E^x[m_{(x,K)}^1(x)],$$

where the last step follows from Lemma 3.7.

Now we establish (3.6) with E replaced with E_x . Using Lemma 3.7, we have

$$E_x[m_K(x)] \leq E^x[m_{(x,K)}^1(x)].$$

Now for the term $E^x[m_{(x,K)}^1(x)]$, let η be an initial particle configuration having an active particle at x , and call z the particle that starts at x . Let η_x be the particle configuration obtained from η by removing z . We carry out a two-step stabilization procedure, as in the previous case. In the first step, we move z ignoring any sleep instruction seen until z exits K . We call \mathcal{V} the expected number of toppings at x up to this point. In the second step, we perform a weak stabilization of (x, K) starting from the particle configuration η_x . Note that by Lemma 2.5, Remark 3.5, and Lemma 3.4, we obtain that, after ignoring some sleep instructions and performing some legal toppings, the value of $m_{(x,K)}^1(x)$ can only increase. Thus, we conclude that $E^x[m_{(x,K)}^1(x)] \leq \mathcal{V} + E_x[m_{(x,K)}^1(x)]$. As in the previous case, we have that $\mathcal{V} = C_G(1 + \lambda)$. Putting everything together, we have

$$E_x[m_K(x)] \leq E^x[m_{(x,K)}^1(x)] \leq C_G(1 + \lambda) + E_x[m_{(x,K)}^1(x)],$$

which concludes the proof. \square

4. Proof of Theorem 1.1. Let L be a positive integer, and let $x \in V$ be a fixed vertex. Let B_L be the ball of radius L centered at x . For any $y \in B_L$, let p_y be the probability that a random walk starting from y visits x before exiting B_L .

LEMMA 4.1. *For any vertex-transitive, transient graph, we have*

$$\sum_{y \in B_L} p_y \rightarrow \infty \text{ as } L \rightarrow \infty.$$

PROOF. We can lower bound p_y by \tilde{p}_y , the probability that a random walk starting from y visits x before exiting B_L or returning to y . By symmetry, \tilde{p}_y is equal to the probability that a random walk starting from x visits y before returning to x and before exiting B_L . Therefore, $\sum_{y \in B_L} \tilde{p}_y$ is the expected number of vertices visited by a random walk starting from x before

returning to x and before exiting B_L . In a transient graph, this random walk has a positive probability of never returning to x , in which case it visits at least L vertices. This establishes the lemma. \square

PROOF OF THEOREM 1.1. We will stabilize B_L and show that, for any fixed $\mu > 0$ there exists a fixed $\lambda > 0$ small enough such that the number of topplings at x goes to infinity with L . This implies that $\mu_c \rightarrow 0$ as $\lambda \rightarrow 0$.

Let η be the initial particle configuration inside B_L and let η_s be the particle configuration inside B_L obtained after stabilization of B_L . Then η_s only contains sleeping particles. For each particle of η_s , we start a so-called *ghost particle* which performs independent simple random walk steps until exiting B_L . Let W_L be the number of visits to x by particles or ghosts, and let R_L be the number of times that x was visited by ghosts. So $W_L - R_L$ is the number of topplings at x during the stabilization of B_L . Let N_0 be the number of visits to x of a random walk that starts from x and is killed upon exiting B_L . For simplicity, let $q = q(\lambda) = 3\sqrt{\lambda(C_G(1 + \lambda) + 1)}$, the upper bound in the second part of Theorem 3.1. Hence,

$$(4.1) \quad E[W_L - R_L] = \sum_{y \in B_L} (\mu - Q(y, B_L)) p_y E[N_0] \geq (\mu - q) E[N_0] \sum_{y \in B_L} p_y.$$

Note that N_0 is a geometric random variable and, for any transient graph, it holds that $E[N_0] < \infty$ as $L \rightarrow \infty$. Also, Lemma 4.1 gives that for any $\mu > q$, $E[W_L - R_L] \rightarrow \infty$ as $L \rightarrow \infty$. We want to show that

$$(4.2) \quad P\left(W_L - R_L \leq \frac{E[W_L - R_L]}{3}\right) \leq c < 1,$$

for some constant c independent of L . This implies that

$$\liminf_{L \rightarrow \infty} P\left(W_L - R_L > \frac{E(W_L - R_L)}{3}\right) > 0.$$

By the 0-1 law, we then obtain that $W_L - R_L$ goes to infinity almost surely, concluding the proof.

In order to establish (4.2), note that

$$(4.3) \quad \begin{aligned} & P\left(W_L - R_L \leq \frac{E[W_L - R_L]}{3}\right) \\ &= P\left(W_L - E[W_L] + \frac{E[W_L - R_L]}{3} \leq R_L - E[R_L] - \frac{E[W_L - R_L]}{3}\right) \\ &\leq P\left(|W_L - E[W_L]| \geq \frac{E[W_L - R_L]}{3}\right) + P\left(|R_L - E[R_L]| \geq \frac{E[W_L - R_L]}{3}\right). \end{aligned}$$

We now use Chebyshev's inequality, which gives

$$(4.4) \quad P\left(W_L - R_L \leq \frac{E[W_L - R_L]}{3}\right) \leq 9 \frac{\text{Var}(W_L)}{E^2[W_L - R_L]} + 9 \frac{\text{Var}(R_L)}{E^2[W_L - R_L]}.$$

We claim that

$$(4.5) \quad \lim_{L \rightarrow \infty} \frac{\text{Var}(W_L)}{E^2[W_L - R_L]} = 0,$$

and that for any $\mu > 0$ and for any small enough λ ,

$$(4.6) \quad \limsup_{L \rightarrow \infty} \frac{\text{Var}(R_L)}{E^2[W_L - R_L]} \leq \frac{q}{(\mu - q)^2}.$$

Note that the above bound goes to 0 as $\lambda \rightarrow 0$. Putting (4.5) and (4.6) into (4.4) establishes (4.2), which concludes the proof of the theorem.

It remains to establish (4.5) and (4.6). For any 3 independent random variables A, B, C note that

$$(4.7) \quad \text{Var}(ABC) = E[A^2]E[B^2]E[C^2] - E^2[A]E^2[B]E^2[C].$$

Then using independence we can write $\text{Var}(W_L) = \sum_{y \in B_L} \text{Var}(\mathbf{1}(\eta(y) = 1) I_y N_0)$, where I_y is the indicator that a random walk starting from y visits x before exiting B_L ; hence, $p_y = E[I_y]$. Now applying (4.7), we obtain

$$\begin{aligned} \text{Var}(W_L) &= \sum_{y \in B_L} (\mu p_y E[N_0^2] - \mu^2 p_y^2 E^2[N_0]) \\ &= \mu E[N_0^2] \sum_{y \in B_L} p_y \left(1 - \mu p_y \frac{E^2[N_0]}{E[N_0^2]}\right) \leq \mu E[N_0^2] \sum_{y \in B_L} p_y. \end{aligned}$$

Therefore, using (4.1),

$$\frac{\text{Var}(W_L)}{E^2(W_L - R_L)} \leq \frac{\mu E[N_0^2]}{(\mu - q)^2 E^2[N_0] \sum_{y \in B_L} p_y} \rightarrow 0,$$

since $\sum_{y \in B_L} p_y \rightarrow \infty$ by Lemma 4.1, while all the other terms are bounded away from both infinity and zero.

Now we turn to (4.6). For $y \in B_L$, write

$$S_y = \mathbf{1}(\eta_s(y)), \quad s_y = E[S_y] = Q(y, B_L), \quad \text{and} \quad s_{x,y} = E[S_x S_y].$$

Using this notation, we have $R_L = \sum_{y \in B_L} S_y I_y N_0$. Since $\text{Var}(R_L) = E[R_L^2] - E^2[R_L]$, we write

$$ER_L^2 = \sum_{y \in B_L} E[S_y I_y N_0^2] + \sum_{y, z \in B_L, y \neq z} E(S_y S_z I_y I_z N_0 N_0'),$$

where N_0, N'_0 are independent and identically distributed. Using independence, we have

$$ER_L^2 = \sum_{y \in B_L} s_y p_y E[N_0^2] + \sum_{y, z \in B_L, y \neq z} s_{y,z} p_y p_z E^2[N_0].$$

Hence,

$$\begin{aligned} \text{Var}(R_L) &= \sum_{y \in B_L} s_y p_y E[N_0^2] + \sum_{y, z \in B_L, y \neq z} s_{y,z} p_y p_z E^2[N_0] - \left(\sum_{y \in B_L} s_y p_y E[N_0] \right)^2 \\ &= \sum_{y \in B_L} (s_y p_y E[N_0^2] - s_y^2 p_y^2 E^2[N_0]) + \sum_{y, z \in B_L, y \neq z} (s_{y,z} - s_y s_z) p_y p_z E^2[N_0] \\ &\leq \sum_{y \in B_L} s_y p_y E[N_0^2] + \sum_{y, z \in B_L, y \neq z} (s_y - s_y s_z) p_y p_z E^2[N_0] \\ &\leq q E[N_0^2] \sum_{y \in B_L} p_y + q E^2[N_0] \sum_{y, z \in B_L, y \neq z} p_y p_z. \end{aligned}$$

Finally, we obtain

$$\begin{aligned} \frac{\text{Var}(R_L)}{E^2(W_L - R_L)} &\leq \frac{q E[N_0^2] \sum_{y \in B_L} p_y + q E^2[N_0] \sum_{y, z \in B_L, y \neq z} p_y p_z}{(\mu - q)^2 E^2[N_0] \left(\sum_{y \in B_L} p_y \right)^2} \\ &\leq \frac{q E[N_0^2]}{(\mu - q)^2 E^2[N_0] \sum_{y \in B_L} p_y} + \frac{q}{(\mu - q)^2}. \end{aligned}$$

Note that for any fixed $\lambda > 0$ the first fraction goes to 0 with L since $\sum_x p_x \rightarrow \infty$ and all the other terms are bounded away from zero and infinity. The second term can be made arbitrarily small since $q \rightarrow 0$ as $\lambda \rightarrow 0$. In particular, if $\mu > q + \sqrt{q}$, the second term is smaller than 1, so ARW is active almost surely. This establishes (4.6). \square

5. Proof of Theorem 1.2. We prove Theorem 1.2 by first establishing general sufficient conditions that give $\mu_c < 1$ (Theorem 5.1 below), and then showing that graphs of positive speed for random walks satisfy those conditions.

Let $x \in V$ be a fixed vertex of G , which we refer to as the *origin*. Let $\{X(t)\}_{t \in \mathbb{N}}$ denote a simple random walk on G starting from the origin, and let $\{Y(t)\}_{t \in \mathbb{N}}$ be independent random variables such that, for any $t \in \mathbb{N}$, we have $Y(t) = 0$ with probability $\frac{1}{1+\lambda}$ and $Y(t) = 1$ with probability $\frac{\lambda}{1+\lambda}$.

Let B_L be the ball of radius L centered at x , and let A_L be the vertices at distance L from x . For any set $V' \subset V$, let

$$\tau_{V'} := \min\{t \in \mathbb{N} : X(t) \in V'\}$$

be the first hitting time of the random walk to V' and

$$\tau_{V'}^+ := \min\{t \geq 1 : X(t) \in V'\}$$

be the first return time of the random walk to V' . Finally, let

$$\tau_{V'}^k := \min\{t \geq 1 : X(t) \notin V' \text{ and } Y(t) = 1\}.$$

We can interpret the above quantity by considering that the random walk is “killed” outside V' at times t when $Y(t) = 1$; using this, $\tau_{V'}^k$ gives the time the random walk is killed.

Here we consider that the initial particle configuration, denoted by η , is given by any product of identical measures on \mathbb{N}^V with density $E[\eta(x)] = \mu$. We assume that the graph G is vertex-transitive. Note that the assumption of positive speed of the random walk on G is not required in the next theorem. Let $\nu_0 = P(\eta(x) = 0)$ be the probability that a vertex is empty at time 0, which is the same for all vertices.

THEOREM 5.1. *Given positive integers $n < L$, set $\Lambda = V \setminus B_L$ and let*

$$N_n^L := |\{t \in \mathbb{N} : X(t) \in A_n \text{ and } t < \tau_\Lambda \wedge \tau_x^+\}|$$

be the number of visits of $X(t)$ to A_n before $X(t)$ enters Λ or returns to the origin. Let

$$\tilde{N}_n^L := |\{t \in \mathbb{N} : X(t) \in A_n \text{ and } t < \tau_\Lambda \wedge \tau_x^+ \wedge \tau_{B_{n-1}}^k\}|$$

be the number of visits of $X(t)$ to A_n before $X(t)$ enters Λ , returns to the origin or is “killed” outside B_{n-1} . Let also $M_L := \sum_{n=0}^L N_n^L$ and $\tilde{M}_L := \sum_{n=0}^L \tilde{N}_n^L$. If given μ and λ we have

$$(5.1) \quad \liminf_{L \rightarrow \infty} \frac{E[\tilde{M}_L]}{E[M_L]} > \frac{\nu_0}{\mu + \nu_0}.$$

then ARW is active almost surely.

PROOF. We will define a stabilization procedure for B_L and show that the number of topplings at the origin goes to infinity with L . We will do the stabilization by moving particles located at different levels step by step. At the first step we move all particles which are located in A_L , at the next step we move all particles which are located in A_{L-1} , and so on. The same particle might be moved several times in the course of the whole procedure. We now define such steps.

First step. Let η be the initial particle configuration, and let Z_L be the particles of η which are in A_L . Order the particles in Z_L in some arbitrary manner. Consider the first particle in the order and move that particle until one of the following events occur:

1. the particle reaches the origin,
2. the particle reaches an empty site in B_{L-1} ,
3. the particle “uses” a sleep instruction in $V \setminus B_{L-1}$,
4. the particle reaches Λ .

Then, take the second particle in the order and move it several times until one of the four events above occurs. After that, take the third particle in the order and do the same. Repeat this procedure until all particles of Z_L have been moved. We obtain a new particle configuration that we denote by η^1 .

Second step. Let Z_{L-1} be the particles of η^1 which are in A_{L-1} . Note that Z_L and Z_{L-1} are not necessarily disjoint, since particles in Z_L could have ended in A_{L-1} after they were moved in the first step. Order the particles of Z_{L-1} in some arbitrary order. Now move the first particle in the order of Z_{L-1} until one of the following events occur:

1. the particle reaches the origin,
2. the particle reaches an empty site in B_{L-2} ,
3. the particle “uses” a sleep instruction in $V \setminus B_{L-2}$,
4. the particle reaches Λ .

Then, take the second particle in the order and move it several times until one of the four events above occurs. After that, take the third particle in the order and do the same. Repeat the same procedure until all particles of Z_{L-1} have been moved. We obtain a new particle configuration that we denote by η^2 .

Next steps. We repeat the procedure above, analogously defining the set of particles Z_{L-i} , $i \in \{2, 3, 4, \dots, L-1\}$, and obtaining the particle configuration η^{i+1} .

Note that η^L may not be a stable configuration. However, letting G_L be the total number of particles that stop at the origin during the procedure described above, the Abelian property (Lemma 2.2) implies that $m_{B_L}(x) \geq G_L$. Our goal is to prove that there exists a constant $c > 0$ independent of L such that the probability that $G_L > cL$ is bounded away from 0, which implies that ARW is active by the 0-1 law. In order to estimate G_L , we

introduce *ghost particles* as in Section 4. Ghost particles can be created at any step of our procedure. Consider the $(L - n + 1)$ th step, where we move particles from the set Z_n , the set of particles of η^{L-n} that are located in A_n . Let w be one of the particles that is moved at this step. Let $z \in A_n$ be its starting vertex. We create a ghost particle if the two next conditions hold:

- (i) $\eta(z) = 0$ (i.e., z is empty for the initial particle configuration),
- (ii) the motion of w stops because it “uses” a sleep instruction at some site $y \in V \setminus B_{n-1}$ (i.e., the motion of w stops due to condition 3 in the procedure above).

The ghost particle is then created at $y \in V \setminus B_{n-1}$, the site where the particle w uses the sleep instruction. We call the site $z \in A_n$ above *the site that is associated to the ghost*. A crucial point to observe is that, in order for w to create a ghost during step $L - n + 1$, it is necessary that w is in $V \setminus B_n$ for the initial particle configuration η , and that at some previous step w is moved until reaching the site z , which was empty at that time (so w is stopped according to condition 2 in the procedure above). Note that every particle creates at most one ghost in the course of the whole procedure. Indeed, when this happens, the particle that is responsible for the generation of the ghost is not moved any more at any subsequent step. After being created, each ghost particle performs independent simple random walk steps until reaching $\Lambda \cup \{x\}$, when it then stops.

Let W_L be the number of particles and ghosts visiting the origin, and let R_L be the number of ghosts visiting the origin. Then,

$$G_L = W_L - R_L.$$

We now estimate the terms W_L and R_L separately. For any $j \in \mathbb{N}$ and $z \in V$, let $(X^{(z,j)}(t))_{t \in \mathbb{N}}$ be an independent random walk on V starting from z and $(Y^{(z,j)}(t))_{t \in \mathbb{N}}$ be an infinite sequence of i.i.d. random variables such that $Y^{(0,0)}(0) = 1$ with probability $\frac{\lambda}{1+\lambda}$ and $Y^{(0,0)}(0) = 0$ with probability $\frac{1}{1+\lambda}$. Let $\tau_S^{(z,j)}$ be the first time the random walk $(X^{(z,j)}(t))_{t \in \mathbb{N}}$ visits the set $S \subset V$ and let us write simply $\tau_y^{(z,j)}$ if $S = \{y\}$. Then,

$$W_L \text{ stochastically dominates } \tilde{W}_L := \sum_{n=1}^L \sum_{z \in A_n} \sum_{j=1}^{\eta(z)} \mathbf{1} \left(\mathcal{A}^{(z,j)} \cap \mathcal{B}^{(z,j)} \right), \quad (5.2)$$

where $\mathcal{A}^{(z,j)} := \{\tau_x^{(z,j)} < \tau_\Lambda^{(z,j)}\}$, $\mathcal{B}^{(z,j)} := \{Y^{(z,j)}(t) = 0 \text{ for any } t \leq \tau_x^{(z,j)}\}$ such that $X^{(z,j)}(t) \notin B_{|z|-1}$, and $|z|$ denotes the distance between z and x .

Now we make a crucial observation for the estimation of R_L . Recall that every ghost can be associated to the site where the particle starts at the step it uses the sleep instruction and generates that ghost. From the definition of our procedure, it follows that for every site $z \in B_L$ such that $\eta(z) = 0$, *there exists at most one ghost that can be associated to z* . It also follows that if $z \in B_L$ is such that $\eta(z) = 1$, then no ghost can be associated to that site. Thus, if from every site $z \in B_L$ with $\eta(z) = 0$ we start a *sleeping random walk* $(X^{(z,0)}(t), Y^{(z,0)}(t))_{t \in \mathbb{N}}$ and we count \tilde{R}_L , the number of them which hit the origin before entering Λ and such that $Y(t) = 1$ somewhere in $V \setminus B_{|z|-1}$, we conclude that

$$(5.3) \quad \tilde{R}_L \text{ stochastically dominates } R_L.$$

Hence, we write,

$$(5.4) \quad \tilde{R}_L := \sum_{n=1}^L \sum_{z \in A_n} \mathbf{1}(\eta(z) = 0) \cdot \mathbf{1}\left(\mathcal{A}^{(z,0)} \cap \bar{\mathcal{B}}^{(z,0)}\right),$$

where for clarity we denote by $\bar{\mathcal{B}}^{(z,0)} := (\mathcal{B}^{(z,0)})^c = \{Y^{(z,0)}(t) = 1 \text{ for some } t \leq \tau_x^{(z,0)} \text{ such that } X^{(z,0)}(t) \notin B_{|z|-1}\}$ the complement of $\mathcal{B}^{(z,0)}$. As the initial particle configuration is distributed according to a product measure, from (5.2) and (5.4) it follows that,

$$(5.5) \quad \begin{aligned} E[\tilde{W}_L] - E[\tilde{R}_L] &= \sum_{n=1}^L \sum_{z \in A_n} \left[\mu \cdot P\left(\mathcal{A}^{(z,0)} \cap \mathcal{B}^{(z,0)}\right) - \nu_0 \cdot P\left(\mathcal{A}^{(z,0)} \cap \bar{\mathcal{B}}^{(z,0)}\right) \right] \\ &= \sum_{n=1}^L \sum_{z \in A_n} \left[(\mu + \nu_0) P\left(\mathcal{A}^{(z,0)} \cap \mathcal{B}^{(z,0)}\right) - \nu_0 \cdot P\left(\mathcal{A}^{(z,0)}\right) \right]. \end{aligned}$$

To simplify the notation, we will henceforth drop the 0's from the superscript in the terms above. When analyzing the term $P(\mathcal{A}^z \cap \mathcal{B}^z)$, consider the last time t that the random walk starting from $z \in A_n$ visits A_n before reaching the origin. We will denote by y the vertex of A_n where the random walk is in its last visit to A_n . Hence, decomposing in y and t , we have

$$(5.6) \quad P(\mathcal{A}^z \cap \mathcal{B}^z) = \sum_{y \in A_n} \sum_{t=1}^{\infty} P(\mathcal{C}^{z,y,t} \cap \mathcal{D}^{z,t} \cap \mathcal{E}^{z,t}) \cdot P(\tau_x^y < \tau_{A_n,+}^y),$$

where $\mathcal{C}^{z,y,t} := \{X^z(t) = y\}$, $\mathcal{D}^{z,t} := \{Y^z(i) = 0 \text{ for any } i \leq t \text{ such that } X^z(i) \notin B_{|z|-1}\}$, $\mathcal{E}^{z,t} := \{\tau_{\{x\} \cup \Lambda}^z > t\}$, $\tau_{S,+}^z$ is the first return time of the

random walk starting from z to the set $S \subset V$. Now since graph is transitive, any path of a random walk from a vertex z_1 to z_2 occurs with the same probability as the reversed path for a random walk going from z_2 to z_1 . This gives that, for $y, z \in A_n$,

$$(5.7) \quad P(\tau_x^y < \tau_{A_n,+}^y) = P(\{X^x(\tau_{A_n}^x) = y\} \cap \{\tau_{A_n}^x < \tau_{x,+}^x\});$$

that is, the event $\tau_x^y < \tau_{A_n,+}^y$ is equivalent to the event that a random walk starting from x visits A_n before returning to x , and visits A_n for the first time at y . Also, for $y, z \in A_n$ and $t \in \mathbb{N}$,

$$(5.8) \quad P(\mathcal{C}^{z,y,t} \cap \mathcal{D}^{z,t} \cap \mathcal{E}^{z,t}) = P(\mathcal{C}^{y,z,t} \cap \mathcal{D}^{y,t} \cap \mathcal{E}^{y,t}).$$

Now plug (5.7) and (5.8) into (5.6). Summing over $z \in A_n$ first and then over y and t , and using the Markov property for the random walk, we conclude that

$$(5.9) \quad \sum_{z \in A_n} P(\mathcal{A}^z \cap \mathcal{B}^z) = \sum_{y \in A_n} \sum_{t=0}^{\infty} E \left[\sum_{z \in A_n} \mathbf{1}(\mathcal{C}^{y,z,t} \cap \mathcal{D}^{y,t} \cap \mathcal{E}^{y,t}) \right] \cdot P(\{X^x(\tau_{A_n}^x) = y\} \cap \{\tau_{A_n}^x < \tau_{x,+}^x\}) = E[\tilde{N}_n^L].$$

Similarly to (5.9), we obtain

$$(5.10) \quad \sum_{z \in A_n} P(\mathcal{A}^z) = E[N_n^L].$$

Hence, plugging (5.9) and (5.10) into (5.5), we have

$$(5.11) \quad \begin{aligned} E[\tilde{W}_L] - E[\tilde{R}_L] &= \left(\sum_{n=0}^L (\mu + \nu_0) E[\tilde{N}_n^L] - \nu_0 E[N_n^L] \right) \\ &= (\mu + \nu_0) E[\tilde{M}_L] - \nu_0 E[M_L]. \end{aligned}$$

Note now that $E[M_L]$ goes with L to the expectation of the return time of the random walk, which is infinite on any infinite, connected graph (see for example Theorem 1.1 in [5]). Then, from our assumption in (5.1), it follows that the lower bound above diverges with L . It remains to prove that this implies that $G_L \rightarrow \infty$ with L with positive probability, which in turn implies

that ARW is active almost surely by the 0-1 law (Lemma 2.4). For this, we use the same derivation as in (4.3) and (4.4), which gives that

$$(5.12) \quad \begin{aligned} P\left(\tilde{W}_L - \tilde{R}_L < \frac{E[\tilde{W}_L - \tilde{R}_L]}{3}\right) &\leq 9 \frac{\text{Var}(\tilde{W}_L)}{E^2[\tilde{W}_L - \tilde{R}_L]} + 9 \frac{\text{Var}(\tilde{R}_L)}{E^2[\tilde{W}_L - \tilde{R}_L]} \\ &\leq 9 \frac{E[\tilde{W}_L]}{E^2[\tilde{W}_L - \tilde{R}_L]} + 9 \frac{E[\tilde{R}_L]}{E^2[\tilde{W}_L - \tilde{R}_L]}, \end{aligned}$$

where in the last step we use that $\text{Var}(\tilde{W}_L) \leq E[\tilde{W}_L]$ and $\text{Var}(\tilde{R}_L) \leq E[\tilde{R}_L]$ since \tilde{W}_L and \tilde{R}_L are defined as a sum of independent Bernoulli random variables. Note that (5.11) and (5.1) imply that

$$(5.13) \quad E[\tilde{W}_L - \tilde{R}_L] > K E[M_L] \quad \text{for some constant } K > 0 \text{ and all large enough } L.$$

Hence we obtain that $E[\tilde{W}_L] \geq E[\tilde{R}_L]$ for all large enough L . In addition, from the derivation of (5.5) and (5.11) we have

$$E[\tilde{W}_L] \leq (\mu + \nu_0) E[\tilde{M}_L].$$

Using these facts, we obtain

$$E[\tilde{W}_L + \tilde{R}_L] \leq 2E[\tilde{W}_L] \leq 2(\mu + \nu_0)E[\tilde{M}_L].$$

Plugging this into (5.12), and using (5.13), we get

$$P\left(\tilde{W}_L - \tilde{R}_L < \frac{E[\tilde{W}_L - \tilde{R}_L]}{3}\right) \leq \frac{18(\mu + \nu_0)E[\tilde{M}_L]}{K^2 E^2[M_L]} \leq \frac{18(\mu + \nu_0)}{K^2 E[M_L]}.$$

The last term converges to 0 with L . Hence, $P(\tilde{W}_L - \tilde{R}_L \geq \frac{E[\tilde{W}_L - \tilde{R}_L]}{3})$ is bounded away from 0 and this concludes the proof. \square

PROOF OF THEOREM 1.2. We show that for any $\lambda > 0$ and $\mu > 1 - \frac{\alpha\delta}{1+\lambda}$ the condition in (5.1) is satisfied. Observe that, conditioning on the non-return of the random walk to the origin, \tilde{N}_n^L is stochastically larger than a random variable which takes value 1 with probability $\frac{1}{1+\lambda}$ and 0 with probability $\frac{\lambda}{1+\lambda}$, as the random walk hits A_n at least one time. Hence,

$$(5.14) \quad E[\tilde{N}_n^L] \geq \frac{\delta}{1+\lambda} \quad \text{and, consequently,} \quad E[\tilde{M}_L] \geq \frac{\delta}{1+\lambda} L.$$

Our goal is to use that the random walk has a positive speed α , which is to say that

$$(5.15) \quad \lim_{t \rightarrow \infty} \frac{|X(t)|}{t} = \alpha \quad \text{almost surely,}$$

to show that, for any $\epsilon > 0$, there exists $L_0 = L_0(\epsilon)$ large enough so that

$$(5.16) \quad E[M_L] \leq \frac{L}{\alpha - \epsilon} \quad \text{for all } L \geq L_0.$$

To do this, note that (5.15) implies that, for any $\xi > 0$, there exists t_0 large enough so that

$$P(X(t) > (1 - \xi)\alpha t \quad \text{for all } t \geq t_0) > 1 - \xi.$$

Now let $\Delta_L = \lceil \frac{L}{(1-\xi)\alpha} \rceil$, so for L large enough we have that $X(\Delta_L)$ is outside B_L with probability at least $1 - \xi$. If that does not happen, then $X(\Delta_L)$ is at some random vertex $y \in B_L$. Then after an additional time of Δ_{2L} , with probability at least $1 - \xi$, the walker exits the ball of radius $2L$ centered at y ; consequently, it also exits B_L . This gives that $P(X(\Delta_{3L}) \in B_L) \leq \xi^2$. Iterating this argument, we have

$$\begin{aligned} E[M_L] &= \sum_{i \geq 1} P(M_L \geq i) \leq \Delta_L + \sum_{k=1}^{\infty} (\Delta_{(2k+1)L} - \Delta_{(2k-1)L}) \cdot P(X(\Delta_{(2k-1)L}) > L) \\ &\leq \Delta_L + \sum_{k=1}^{\infty} \left(1 + \frac{2L}{(1-\xi)\alpha}\right) \xi^k \\ &= \Delta_L + \left(1 + \frac{2L}{(1-\xi)\alpha}\right) \frac{\xi}{1-\xi}. \end{aligned}$$

Then (5.16) follows by taking ξ small enough with respect to ϵ . Hence, we conclude that for all L large enough,

$$\frac{E[\tilde{M}_L]}{E[M_L]} \geq \frac{\delta(\alpha - \epsilon)}{1 + \lambda}.$$

Thus, the condition in (5.1) is satisfied when $\nu_0 = 1 - \mu$ as long as $\mu > 1 - \frac{\alpha\delta}{1+\lambda}$. \square

6. Proof of Theorem 1.3.

PROOF OF THEOREM 1.3. Since G is amenable and vertex transitive, we can take a sequence of subsets $\{V_n\}_{n \geq 1}$ of V such that $V_n \rightarrow V$ as $n \rightarrow \infty$, there exists a vertex $x \in \bigcap_{n=1}^{\infty} V_n$, and

$$\frac{|\partial V_n|}{|V_n|} \text{ is non-increasing and goes to 0 as } n \rightarrow \infty,$$

where ∂V_n denotes the external boundary of V_n ; that is, the set of vertices in $V \setminus V_n$ that have an edge incident to V_n . Let B_K be the ball of radius K centered at x , and recall that $m_{B_K}(x)$ is the number of instructions used at x to stabilize B_K . If we assume that $\mu > \mu_c$, then the 0-1 law (Lemma 2.4) implies that $\Pr(m_{B_K}(x) \geq 1) \rightarrow 1$ as $K \rightarrow \infty$. By monotonicity of this probability, for any fixed $\epsilon > 0$, we can find $K = K(\epsilon)$ large enough such that

$$\Pr(m_{B_K}(x) \geq 1) \geq 1 - \epsilon.$$

For any set $V_n \subset V$, let V_n^K be the set obtained by taking the union of balls of radius K centered at each vertex of V_n . Hence $V_n^K \supset V_n$. Let $N_{n,K}$ be the number of particles inside V_n^K prior to the stabilization of V_n^K and let $N_{n,K}^s$ be the number of sleeping particles in V_n^K after the stabilization of V_n^K . Clearly, $N_{n,K}^s \leq N_{n,K}$ almost surely. Let d denotes the degree of each vertex of G ; so B_K has at most d^K vertices. Note that

$$E[N_{n,k}] = |V_n^K| \mu \leq (|V_n| + d^K |\partial V_n|) \mu = \left(1 + d^K \frac{|\partial V_n|}{|V_n|}\right) |V_n| \mu.$$

Also, from (3.1) in Theorem 3.1, we have

$$E[N_{n,K}^s] \geq \sum_{y \in V_n} Q(y, V_n^K) \geq \left(\frac{\lambda}{1+\lambda}\right) \sum_{y \in V_n} \Pr(m_{V_N^K}(y) \geq 1).$$

Since V_N^K contains a ball of radius K centered at y , by monotonicity and transitivity we obtain

$$E[N_{n,K}^s] \geq |V_n| \left(\frac{\lambda}{1+\lambda}\right) \sum_{y \in V_n} \Pr(m_{B_K}(x) \geq 1) \geq |V_n| \left(\frac{\lambda}{1+\lambda}\right) (1 - \epsilon).$$

Since $E[N_{n,k}] \geq E[N_{n,K}^s]$, placing the two inequalities together yields

$$\mu \geq \left(\frac{\lambda}{1+\lambda}\right) (1 - \epsilon) \left(1 + d^K \frac{|\partial V_n|}{|V_n|}\right)^{-1}.$$

Now set $n = n(d, K)$ large enough such that $\mu \geq \left(\frac{\lambda}{1+\lambda}\right) (1 - \epsilon)^2$. Therefore, assuming that $\mu > \mu_c$ implies that $\mu \geq \left(\frac{\lambda}{1+\lambda}\right) (1 - \epsilon)^2$, which completes the proof since $\epsilon > 0$ is arbitrary. \square

7. Proof of Theorem 1.6. In order to show that $\mu_c > 0$ for any $\lambda > 0$ when G is a d -regular tree, we will relate a stabilization procedure to a certain branching process in \mathbb{Z} . To avoid ambiguity we will refer to the particles of the branching process as *tokens*. Initially the branching process starts with d tokens at position 1. We will show that $\mu_c > 0$ holds if with positive probability we have that no token ever visits a position $k \leq 0$.

We start defining this branching process. Start with d tokens at position 1. The process evolves in discrete steps, where at each step we update the position of each token independently. Given a token at position $k \in \mathbb{Z}$, we update it as follows. With probability α , the token advances one position, jumping to position $k + 1$. With probability $1 - \alpha$, there is a *branching*. This means that the token is deleted and is replaced by ℓd tokens at position $k - \ell$, where $\ell \geq 1$ is an independent geometric random variable of success probability β ; i.e., $P(\ell = z) = (1 - \beta)^{z-1} \beta$. The value d here will later be the same as the degree of the tree, that is why we choose to use the same letter, while α and β are some additional parameters that will be related to μ and λ .

LEMMA 7.1. *For all $d \geq 1$ and all $\beta \in (0, 1)$, there exists $\alpha_0 \in (0, 1)$ large enough so that, for all $\alpha \in (\alpha_0, 1]$, with positive probability there will never be a token in positions $k \leq 0$.*

PROOF. Let $\gamma = \sqrt{1 - \beta}$. At time t , if k_1, k_2, \dots are the positions of the tokens at that time, define the function

$$\Psi_t = \sum_{i \geq 1} \gamma^{k_i}.$$

Let R be the smallest value such that $d\gamma^{R+1} < 1/5$, and consider the event

$$(7.1) \quad \mathcal{E} = \{\text{in the first } R \text{ steps all tokens advance and do not branch}\}.$$

Note that $P(\mathcal{E}) = \alpha^{dR}$, and \mathcal{E} implies that at time R all tokens are at position $R + 1$. Thus,

$$P(\Psi_R < \frac{1}{5}) \geq P(\mathcal{E}) = \alpha^{dR}.$$

Note that if any token reaches a position $k \leq 0$, then we have $\Psi_t \geq 1$. We show that with positive probability $\Psi_t < 1$ for all $t \geq R$. For this it suffices to show that Ψ_t is a supermartingale, provided α is large enough.

Let \mathcal{F}_t denote the filtration given by the position of the tokens at times $0, 1, \dots, t$. Now we compute the change in Ψ_t in one step. Define Δ_k as the

expected change in Ψ_t caused by moving a token from position k , assuming that there is at least one token at position k at that time. By the form of Ψ_t , we have that Δ_k does not depend on t . Given a token at position k , since the token advances one position with probability α , and gets replaced by ℓd tokens at position $k - \ell$ with probability $(1 - \alpha)(1 - \beta)^{\ell-1}\beta$, we have that

$$\begin{aligned}\Delta_k &= -\gamma^k + \alpha\gamma^{k+1} + (1 - \alpha) \sum_{\ell=1}^{\infty} \ell d \beta (1 - \beta)^{\ell-1} \gamma^{k-\ell} \\ &= -\gamma^k \left(1 - \alpha\gamma - \frac{d\beta(1 - \alpha)}{1 - \beta} \sum_{\ell=1}^{\infty} \ell \left(\frac{1 - \beta}{\gamma} \right)^\ell \right).\end{aligned}$$

Since $\frac{1 - \beta}{\gamma} = \gamma < 1$, the sum above converges to $\frac{\gamma}{(1 - \gamma)^2}$. This and replacing $1 - \beta$ with γ^2 yield

$$\Delta_k = -\gamma^k \left(1 - \alpha\gamma - \frac{d(1 - \gamma^2)(1 - \alpha)}{\gamma(1 - \gamma)^2} \right) = -\gamma^k \left(1 - \alpha\gamma - \frac{d(1 + \gamma)(1 - \alpha)}{\gamma(1 - \gamma)} \right).$$

Hence,

$$\begin{aligned}E(\Psi_{t+1} | \mathcal{F}_t) &= \Psi_t + \sum_{i \geq 1} \Delta_{k_i} = \Psi_t - \sum_{i \geq 1} \gamma^{k_i} \left(1 - \alpha\gamma - \frac{d(1 + \gamma)(1 - \alpha)}{\gamma(1 - \gamma)} \right) \\ &= \Psi_t \left(\alpha\gamma + (1 - \alpha) \frac{d(1 + \gamma)}{\gamma(1 - \gamma)} \right).\end{aligned}$$

Note that we can make the term inside the parenthesis as close to γ as possible by having α close to 1. So, since $\gamma < 1$, by having α close enough to 1 we obtain that $\{\Psi_t\}_t$ is a supermartingale. Let τ be the first time that $\Psi_{R+\tau} \geq 1$ and n be any positive integer. We apply the optional stopping theorem for the almost surely bounded stopping time $\tau \wedge n$, and obtain under the event $\{\Psi_R < 1/5\}$ that

$$\frac{1}{5} > \Psi_R \geq E(\Psi_{R+(\tau \wedge n)}) \geq E(\Psi_{R+\tau} | \tau \leq n) P(\tau \leq n) \geq P(\tau \leq n).$$

Since n is arbitrary, the probability that there will never be a token in positions $k \leq 0$ is at least $\alpha^{dR \frac{4}{5}}$. \square

Before proceeding to the proof of Theorem 1.6, we state a well-known lemma regarding random walks on regular trees.

LEMMA 7.2. *For any ℓ , let p_ℓ be the probability that a random walk starting at distance ℓ from the origin ever visits the origin. Then, for a d -regular tree we have $p_\ell = \left(\frac{1}{d-1} \right)^\ell$.*

PROOF. The lemma follows by checking that if we set $p_\ell = a^\ell$ for some $a > 0$, then $a = \frac{1}{d-1}$ is the only solution in $(0, 1)$ of the recursion $a^\ell = \frac{1}{d}a^{\ell-1} + \left(\frac{d-1}{d}\right)a^{\ell+1}$. \square

PROOF OF THEOREM 1.6. Since a simple random walk in a d -regular tree with $d \geq 3$ has positive speed, Theorem 1.2 gives that $\mu_c < 1$ for any $\lambda > 0$. Also, since a d -regular tree with $d \geq 3$ is a transient graph, Theorem 1.1 gives that $\lim_{\lambda \downarrow 0} \mu_c = 0$. It remains to show that $\mu_c > 0$ for any $\lambda > 0$.

For any $\rho \in (0, 1)$, let Geo_ρ be a geometric random variable of success probability ρ . We assume that, at each vertex v , the number of particles initially located at v is distributed independently according to the distribution of $\text{Geo}_{1-\mu}^* = \text{Geo}_{1-\mu} - 1$. This is enough for our purposes, because $P(\text{Geo}_{1-\mu}^* = 0) = 1 - \mu$, so $\text{Geo}_{1-\mu}^*$ stochastically dominates a Bernoulli random variable of mean μ . Then using monotonicity of ARW (cf. Lemma 2.3), if for a given $\mu > 0$ ARW almost surely fixates starting from this initial configuration of particles, ARW almost surely fixates starting from a Bernoulli field of particles of density μ . This will establish that $\mu_c > 0$.

We will employ a beautiful stabilization procedure developed by Rolla and Sidoravicius [7] for the one-dimensional lattice \mathbb{Z} . We will need to carry out a much more delicate analysis for the case of a d -regular tree. Let x_1, x_2, \dots be the particles ordered according to their initial distance to the origin, with x_1 being the closest particle to the origin. Let L be an arbitrarily large integer, and consider the finite system inside B_L , the ball of radius L around the origin. Our goal is to show that with positive probability we stabilize B_L without any particle visiting the origin. The idea is to move particles in order, ignoring some sleep instructions and stopping them when they see a sleep instruction near the origin. We do this to pack the particles as close as possible to the origin in such a way that the gap between the particles that have already been moved and the particles that have not yet been moved increases with time. This creates more room for particles to fixate, allowing the stabilization procedure to be carried out until the end without activating any particle that was moved before.

Now we describe the stabilization procedure in details. We will define sets $C_k \subset V$, $k \geq 0$. Let C_0 consist of only the origin. We start by moving particle x_1 repetitively, ignoring all sleep instructions, until it either reaches $V \setminus B_L$ or C_0 . If it reaches $V \setminus B_L$, then we set $C_1 = C_0$. Otherwise, let $z_1, z_2, \dots, z_T \in V$ be the sequence of vertices visited by x_1 , with z_1 being the initial location of x_1 and $z_T \in C_0$. Define τ to be the largest integer so that x_1 ignored a sleep instruction at z_τ ; if x_1 never ignored a sleep instruction until it reaches the origin, we declare that the procedure failed.

If the procedure has not failed, set $C_1 = C_0 \cup \{z_\tau, z_{\tau+1}, \dots, z_{T-1}\}$. Using the terminology in [7], we see C_1 as the set of *corrupted* vertices after x_1 is moved. If after defining C_1 we have that at least one of the subsequent particles x_2, x_3, \dots is located inside C_1 , we declare that the procedure fails.

The idea behind the definition of C_1 is the following. We would like to move x_1 as close as possible to C_0 , to the point that we stop x_1 at the last sleep instruction it sees before visiting C_0 . However, in order to observe that z_τ is the vertex where the last sleep instruction is seen by x_1 , we need to observe the instructions at $z_{\tau+1}, z_{\tau+2}, \dots, z_{T-1}$. This corrupts the array of instructions at the vertices $z_{\tau+1}, z_{\tau+2}, \dots, z_{T-1}$, so we cannot use these arrays of instructions when we move the subsequent particles. These vertices, together with x_τ and C_0 , are the ones forming C_1 .

We then repeat the procedure above. After having moved x_{k-1} , we move x_k repetitively until it either reaches $V \setminus B_L$ (in which case we set $C_k = C_{k-1}$) or it reaches C_{k-1} (in which case we define C_k as the vertices in C_{k-1} plus all vertices visited by x_k since the last sleep instruction x_k sees). The procedure fails if x_k visits C_{k-1} before $V \setminus B_L$ and before seeing any sleep instruction, or if at least one of the subsequent particles x_{k+1}, x_{k+2}, \dots is located inside C_k . For each $k \geq 1$, let E_k be the event that the procedure does not fail when we move x_k . Our goal is to show that there exists a positive constant c so that, for all $L \geq 1$, we have

$$(7.2) \quad P\left(\bigcap_{k=1}^{n_L} E_k\right) \geq c > 0,$$

where n_L is the number of particles initially inside B_L . When (7.2) holds, we have that this procedure stabilizes B_L , without using any instruction at the origin. By the zero-one law (Lemma 2.4), this implies that ARW fixates almost surely.

In order to establish (7.2), we will relate this stabilization procedure with a branching process \mathcal{B} on \mathbb{Z} , and compare \mathcal{B} with the branching process defined in the beginning of the section, which we denote by \mathcal{B}' . The processes \mathcal{B} and \mathcal{B}' start with the same number of tokens, and at the same locations. Then we couple \mathcal{B} and \mathcal{B}' to show that at any time we can associate each token ι of \mathcal{B} with a distinct token ι' of \mathcal{B}' such that

$$(7.3) \quad \text{the position of } \iota \text{ is not smaller than that of } \iota'.$$

If at any time the stabilization procedure fails, then we halt \mathcal{B} . We will show that this can only happen when \mathcal{B}' has a token at some position $k \leq 0$. This will allow us to use Lemma 7.1 to establish (7.2), as the construction

of \mathcal{B} and its coupling with \mathcal{B}' will imply that

$$(7.4) \quad P\left(\bigcap_{k=1}^{n_L} E_k\right) \geq P(\text{no token of } \mathcal{B}' \text{ visits position } k \leq 0).$$

The initial configuration of \mathcal{B} will be d tokens at position 1. Each token is associated with one connected component of the graph obtained from B_L by removing C_0 ; we denote this graph by $G \setminus C_0$. Since G is a tree, a particle that starts in one component of $G \setminus C_0$ cannot jump to a vertex in another component without visiting C_0 . This will imply that tokens evolve independently of one another. The initial configuration of \mathcal{B}' will be identical to that of \mathcal{B} , and we associate each token of \mathcal{B} to a distinct token of \mathcal{B}' . Note that, for this initial configuration (7.3) holds.

Let $\mathcal{L} = \{v_1, v_2, \dots\}$ be an ordered list of the vertices of $B_L \setminus C_0$, where the vertices are sorted according to the order they are visited in a breadth-first search in G starting from the origin. Thus, for any $i < j$, v_i is not furthest away from the origin than v_j . Given any subset of vertices S of G , and any vertex v of G that is not in S , we denote by $d_G(v, S)$ as the distance between v and S ; that is,

$$d_G(v, S) = \min\{\text{distance between } v \text{ and } u \text{ in } G : u \in S\}.$$

One important point is that we will not sample yet the locations of the particles x_1, x_2, \dots . The procedure for updating \mathcal{B} is to consider the vertex v that is at the front of the \mathcal{L} , and check whether v hosts a particle. If the vertex does host a particle, then we move that particle according to the stabilization procedure, which will cause the token corresponding to the component of v to move and/or branch. Depending on the outcome, we may remove v from the list. Then we iterate this procedure.

We will now describe precisely how \mathcal{B} is updated, and later will show how to couple \mathcal{B}' with \mathcal{B} . Assume that, at some moment, the vertex that is at the front of the list is v , and that we have already discovered and moved the particles x_1, x_2, \dots, x_j . Assume that the number of tokens of \mathcal{B} is the number of connected components of $G \setminus C_j$, that \mathcal{B}' has at least as many tokens as \mathcal{B} , and that property (7.3) holds. Let S be the connected component of v in $G \setminus C_j$, and let ι be the token of \mathcal{B} corresponding to S . Assume that ι is located at position $d_G(v, C_j)$, and that ι' is the token of \mathcal{B}' associated with ι . Note that all the properties above hold for the initial configuration.

We start the update of \mathcal{B} by checking whether $v \in C_j$. If this is the case, the stabilization procedure failed, so we halt \mathcal{B} . Otherwise, we update \mathcal{B} in two phases. Recall that the initial number of particles at v is distributed according to $\text{Geo}_{1-\mu}^*$. We will observe the particles at v one-by-one. This

means that in the first time we consider v we will check whether v hosts at least one particle, which happens with probability μ . If this is the case, we move this particle and update \mathcal{B} according to the two phases described below. Then, in the next update of \mathcal{B} , we will consider v again, and ask whether v hosts at least two particles. The conditional probability of this event given that v hosts at least one particle is again μ . If this happens, we then move this second particle as we will describe below. We iterate this, each time checking whether v hosts another particle, until we find out that v does not host another particle. At this point we move to another vertex. We now describe the two phases for the update of \mathcal{B} . The first phase starts by checking whether v hosts another particle.

First phase, case one: no other particle at v . As explained above, this happens with probability $1 - \mu$, regardless of how many times v has been checked before. In this case, we remove v from \mathcal{L} , do not change \mathcal{B} , and jump to the second phase.

First phase, case two: there is a new particle x_{j+1} at v . This happens with probability μ . We move x_{j+1} according to the stabilization procedure (see Figure 1). If x_{j+1} does not visit C_j before leaving B_L , nothing happens, and we jump to the second phase. Otherwise, let M_{j+1} be the number of vertices that x_{j+1} visits from the last sleep instructions it sees until visiting C_j . If the stabilization procedure fails¹, then we halt \mathcal{B} . Otherwise, we obtain a set C_{j+1} which satisfies $C_{j+1} \setminus C_j \subset S$. We update \mathcal{B} by removing the token ι and replacing it by as many tokens as the number of connected components of $S \setminus C_{j+1}$. For each such component S' of $S \setminus C_{j+1}$, let the token corresponding to that component be located at position $\min_{u \in \mathcal{L} \cap S'} d_G(u, C_{j+1})$. Since each instruction is a sleep instruction with probability $\frac{\lambda}{1+\lambda}$, we have that

$$(7.5) \quad M_{j+1} \text{ is stochastically dominated by } \text{Geo}_{\frac{\lambda}{1+\lambda}}.$$

Also, note that

$$(7.6) \quad C_{j+1} \text{ splits } S \text{ into at most } d M_{j+1} \text{ connected components.}$$

The reason is that for each new connected component S' , there must exist an edge in G from S' to $C_{j+1} \setminus C_j$, which is a set with M_{j+1} vertices. Also, each new token created in \mathcal{B} has position at least $d_G(v, C_j) - M_{j+1}$, where

¹Note that one of the reasons for the stabilization procedure to fail is because the set of newly corrupted sites contains a particle that has not yet been moved. We cannot detect that the procedure fails because of this at this moment, because the locations of the particles that have not yet been moved are not known. But this can be detected in future updates of \mathcal{B} , when checking whether $v \in C_j$. If this is the case, we only halt \mathcal{B} at that time.

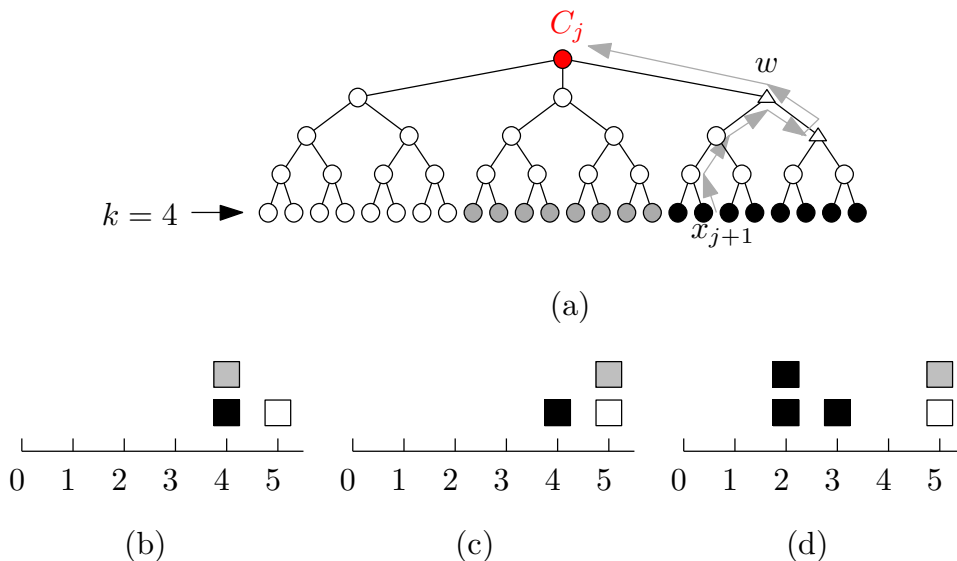


FIG 1. Illustration of how to update \mathcal{B} when observing vertices at distance $k = 4$ from C_j (red vertex). After all white vertices in (a) have been observed, \mathcal{B} is as illustrated in (b), with a white/grey/black square corresponding to the token of the component with white/grey/black vertices at distance k from C_j . Then, after observing all grey vertices, which led to no particle visiting C_j , the grey token in \mathcal{B} advances one position, as in (c). Next, when observing the black vertices we find particle x_{j+1} , which moves according to the grey arrows and sees a sleep instruction in its first visit to w . Then we add the triangular vertices to the set of corrupted vertices, and the black token branches into three tokens, one for each connected component created by removing the triangular vertices from the component of the black vertices. After this, \mathcal{B} becomes the configuration in (d).

we recall that $d_G(v, C_j)$ was the position of ι in \mathcal{B} . Then we go to the second phase without removing v from the list.

Second phase. Let C be the set of corrupted sites after the end of the first phase; that is, $C = C_j$ if there were no other particle at v (first case above), or $C = C_{j+1}$ (second case above). We check whether \mathcal{L} contains any vertex u in the same component of the graph $G \setminus C$ as v , and whose distance (in G) to the origin of G is the same as v . If this is the case (which includes the case that v remains in \mathcal{L} after the end of the first phase), then nothing is done and we end the second phase. Otherwise, and this happens only if there was no additional particle at v , we take the token ι and advance it by one position, moving it to position $d_G(v, C) + 1$. Refer to Figure 1(a–c). This concludes the second phase.

We then iterate the two phases above until \mathcal{L} becomes empty. Since the

vertex at the front of \mathcal{L} is removed at each iteration with probability $1 - \mu$, each vertex is processed a finite number of times, almost surely, and the procedure ends almost surely.

Now we couple \mathcal{B}' with \mathcal{B} and show that property (7.3) continues to hold. Recall that \mathcal{B}' is defined in terms of three parameters α, β and d , where d is the degree of vertices in G . For the other parameters, we set

$$\alpha = \exp\left(-\frac{\mu}{(1-\mu)(d-1)}\right) \quad \text{and} \quad \beta = \frac{\lambda}{1+\lambda}.$$

First we estimate the probability that a token ι is advanced from position k to $k+1$. Let C be the set of corrupted sites at the time the token is advanced, and let $u_1, u_2, \dots, u_\kappa$ be the vertices from the component of $G \setminus C$ that is associated to ι and satisfy $d_G(u_j, C) = k$ for all j ; note that $\kappa = (d-1)^{k-1}$. Note that ι only advances by one position if each $u_1, u_2, \dots, u_\kappa$ has no particle that reaches C before leaving B_L . If there is a particle at some u_j , then the probability that this particle visits C before leaving B_L is at most p_k , where p_k is defined in Lemma 7.2. Therefore, the probability that ι advances to $k+1$ is at least

$$\begin{aligned} \left(\sum_{i \geq 0} \mu^i (1-\mu)(1-p_k)^i\right)^{(d-1)^{k-1}} &= \left(\frac{1-\mu}{1-\mu+\mu p_k}\right)^{(d-1)^{k-1}} \\ &\geq \exp\left(-\frac{\mu p_k}{1-\mu} \cdot (d-1)^{k-1}\right) \\ &= \alpha, \end{aligned}$$

where in the inequality we used that $\frac{1}{1+\epsilon} \geq \exp(-\epsilon)$ for all $\epsilon > 0$. Therefore, we can couple ι and ι' such that if ι' advances by one position, so does ι .

Now if ι' branches, then ι may branch or advance. If ι advances, then property (7.3) continues to hold regardless of how ι' branches, since new tokens of \mathcal{B}' will all have positions smaller than that of ι . If ι also branches, then by (7.5) and the value of β , the number of new tokens created by the branch of ι' stochastically dominates dM_{j+1} , which is not smaller than the number of new tokens created by the branch of ι . The positions of the new tokens of \mathcal{B}' are not larger than $d_G(v, C_j) - M_{j+1}$, which is a lower bound for the position of the new tokens of \mathcal{B}' . Therefore, we can couple the branch of ι and the branch of ι' such that each new token of \mathcal{B} can be associated with one distinct new token of \mathcal{B}' , and property (7.3) continues to hold. We will update \mathcal{B}' even if the stabilization procedure for the branch of ι failed because the particle encountered no sleep instruction. Note that M_{j+1} is still well defined in this case, and we can perform the coupling between the branch of ι' and M_{j+1} as described above.

To conclude the proof, we show that if the stabilization procedure fails, then \mathcal{B}' has a token at some position $k \leq 0$. Assume that the particles x_1, x_2, \dots, x_j have already been observed, and C_j is the current set of corrupted sites. We perform the two phases above repetitively until we find the first particle x_i , $i > j$, which visits C_j before exiting B_L . Note that $C_j = C_{i-1}$. Let v be the vertex where x_i started from, and let $k = d_G(v, C_j)$. Note that k is the position of the token ι of \mathcal{B} corresponding to the component of v in $G \setminus C_j$, and the corresponding token ι' of \mathcal{B}' is at position at most k by (7.3). The procedure cannot fail before finding x_i , and there are two events that can make the procedure fail during the move of x_i . The first is if x_i visits C_j before seeing any sleep instruction. If this happens, then $M_i \geq k$, and all tokens produced from the branching of ι' will have position not larger than $d_G(v, C_j) - M_{j+1} \leq 0$. The second event is if x_i sees sleep instructions, but the new set of corrupted vertices C_i hosts at least one particle that has not yet been moved. This implies that C_i contains a vertex u that is still in \mathcal{L} . But by the ordering of the vertices in \mathcal{L} , this implies that $d_G(u, C_j) \geq d_G(v, C_j)$ which gives that $M_i \geq k$. Therefore the token in \mathcal{B} corresponding to the component of u will be placed at position at most 0, implying that \mathcal{B}' will have a token at position at most 0.

It is important to remark that the coupling between \mathcal{B} and \mathcal{B}' described above suggests that the tokens of \mathcal{B}' are not updated in the same order as described in the beginning of the section, where all tokens of \mathcal{B}' were moving or branching once in each discrete step. Indeed, a token from \mathcal{B} could branch many times before another token from \mathcal{B} is moved or branches, because for example a vertex v could host more than one particle that visits the set of corrupted sites before leaving B_L . However, one can still apply Lemma 7.1 by reordering the updates of \mathcal{B} and \mathcal{B}' accordingly. Letting A_k denote the vertices of G at distance k from the origin, this is possible since tokens move or branch independently of one another, and because for any given k it will take only a finite number of steps to remove from \mathcal{L} all vertices of A_k . For any k , let \mathcal{I}'_k be the set of tokens of \mathcal{B}' at the first time that \mathcal{L} contains no vertex from A_k . Thus, for any k , at the moment that \mathcal{L} has no vertex from A_k we obtain that all the tokens from \mathcal{I}'_{k-1} will have moved or branched at least once, giving that each token of \mathcal{B}' will move or branch in a finite time. To conclude the proof, note that β depends only on λ . Hence, given any d and β , we can find μ small enough to make α close enough to 1. Then Lemma 7.1 gives that $P(\text{no token of } \mathcal{B}' \text{ visits position } k \leq 0)$ is bounded away from 0, which with (7.4) concludes the proof. \square