Adv. Appl. Prob. **50,** 302–326 (2018)
*doi:*10.1017/apr.2018.14
© *Applied Probability Trust* 2018

DEGREE-DEPENDENT THRESHOLD-BASED RANDOM SEQUENTIAL ADSORPTION ON RANDOM TREES

THOMAS M. M. MEYFROYT,* *Eindhoven University of Technology*

Abstract

We consider a special version of random sequential adsorption (RSA) with nearest-neighbor interaction on infinite tree graphs. In classical RSA, starting with a graph with initially inactive nodes, each of the nodes of the graph is inspected in a random order and is irreversibly activated if none of its nearest neighbors are active yet. We generalize this nearest-neighbor blocking effect to a degree-dependent threshold-based blocking effect. That is, each node of the graph is assumed to have its own degree-dependent threshold and if, upon inspection of a node, the number of active direct neighbors is less than that node's threshold, the node will become irreversibly active. We analyze the activation probability of nodes on an infinite tree graph, given the degree distribution of the tree and the degree-dependent thresholds. We also show how to calculate the correlation between the activity of nodes as a function of their distance. Finally, we propose an algorithm which can be used to solve the inverse problem of determining how to set the degree-dependent thresholds in infinite tree graphs in order to reach some desired activation probabilities.

Keywords: Analytical model; counter-based gossip protocol; interacting particle system; random sequential adsorption; jamming limit; random tree graph

2010 Mathematics Subject Classification: Primary 82C22; 60K35
Secondary 68M10; 90B18

1. Introduction

Processes where particles are deposited successively at random positions in an initially empty space, subject to a blocking effect whereby deposited particles can cooperatively prevent new particles from being deposited nearby, are referred to as random sequential adsorption (RSA). The simplest type of RSA models are those with *hard-core* blocking effects, where each deposited particle will prevent any new particle from being deposited in its immediate neighborhood. Note that these processes are irreversible, meaning that the system will eventually reach a jammed state, where no new particles can be deposited. Typically, the analysis of RSA deals with the calculation of the time evolution of the fraction of space, in a continuum, or the fraction of vertices, in a graph, occupied by particles.

One classic example of RSA with hard-core interaction is known as Rényi's parking problem, where cars of a certain length arrive at random positions on the real line and park if they do not overlap with other already parked cars. The limiting fraction of space occupied by cars when the system has reached the jammed state is known as Rényi's parking constant [18].

Received 19 January 2017; revision received 15 August 2017.
* Postal address: Eindhoven University of Technology, MetaForum 4.086, P.O. Box 513, 5600 MB Eindhoven, The Netherlands. Email address: tmeyfroyt@gmail.com

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

Classically, RSA was studied on lattices and in Euclidean space, due to its many applications in biology, physics, and chemistry. For an extensive survey on the analysis of sequential adsorption processes in these fields; see [7]. However, in recent years, mostly motivated by applications in communication sciences and in wireless networks, the study of RSA on random graphs has received much attention. For example, RSA with hard-core blocking can be used to model wireless networks using carrier-sense multiple access-like broadcasting algorithms, where a node is only allowed to transmit data if none of its neighbors is already transmitting [1]. In the context of wireless networks, the deposition of particles is often referred to as the activation of nodes. Likewise, occupied and empty nodes are referred to as active and inactive nodes.

In [5] and [20], RSA with hard-core interaction on random trees was studied. In these papers, it was shown how the fraction of vertices occupied by particles can be expressed in terms of the generating function of the degree distribution of the tree. In [8], multilayer parking on random trees was considered. Similarly, in [2] and [3], RSA with hard-core interaction on random graphs with a given degree distribution was studied. The Erdős–Rényi random graph model was studied in [6] and [19], where, in [19], hard-core blocking was considered and, in [6], generalizations of RSA were considered, including multilayer parking with and without screening.

In this paper we study a new version of RSA with a *threshold* blocking effect, generalizing RSA with hard-core nearest-neighbor blocking effects, on infinite random trees. In our model, each node of the graph has its own random threshold, which we allow to depend on the degree of the node. Then, when a node attempts to activate, it is only able to do so if the number of active neighbors of the node is less than that node's threshold.

The key contributions of this paper are as follows. First, motivated by the study of a certain class of wireless communication protocols, we introduce RSA with a threshold blocking effect, generalizing the hard-core blocking effect. For this model we derive formulae for the time evolution of the probability that a node is active and for the covariance between the activity states of nodes as a function of their distance from each other. Additionally, we propose the activation threshold algorithm (Algorithm 7.1), which can be used to determine how to set the degree-dependent thresholds in order to achieve some target activation probabilities on the infinite tree. Finally, we show that the thresholds generated by this algorithm also perform remarkably well when applied to finite tree graphs and random spatial graphs, which are generally not trees.

1.1. Organization of the paper

The remainder of this paper is organized as follows. Before presenting the model and its analysis, we first provide a brief motivation for studying RSA with a threshold blocking effect in Section 2. In Section 3 we define the mathematical model describing the RSA process with a threshold blocking effect on tree graphs, which is the main focus of this paper. In Section 4 we analyze the probability that a node in this graph is unable to activate, due to the number of active neighbors exceeding its threshold value. In Section 5 we then show how one can also calculate the correlation between the states of nodes as a function of their distance from each other and conclude that these correlations decrease at a superexponential rate as this distance grows larger. Section 6 is devoted to the treatment of some interesting special cases, which admit a detailed analysis. In Section 7 we propose an algorithm which allows us to determine how to set the thresholds of nodes in order to achieve some target activation probabilities. In Section 8 we show by simulation that the thresholds obtained from this algorithm, which

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

assumes an infinite tree graph, also yield good results in finite and nontree graphs. Finally, in Section 9 we briefly summarize our results and make some concluding remarks.

2. Motivation

For many networks in the internet of things, broadcasting is a fundamental operation used to support many applications, such as neighbor discovery, code dissemination, data collection, and so on. Typically, these networks are decentralized and consist of a large number of devices, requiring reliable and scalable distributed broadcasting protocols. A class of protocols which has been shown to be a powerful technique for implementing broadcast in such networks is the class of counter-based gossip protocols [15].

In counter-based schemes, each node keeps track of a counter which counts the number of broadcasts it has received from its neighbors. Then, whenever the node itself wants to broadcast, it will do so only if its counter is below some predefined threshold value. The reasoning behind this is that if many of your neighbors have already sent out their data, your broadcast is probably redundant and should be suppressed. These simple rules lead to scalable broadcasting schemes, where the number of broadcasts per time unit remains bounded, even when the network becomes more and more dense [14].

In particular, a counter-based gossip protocol called Trickle [12] has become an increasingly popular mechanism for broadcasting in the context of the internet of things. While originally designed for propagating and maintaining code updates in wireless sensor networks, Trickle has proven to be a powerful mechanism applicable to many protocol design problems. Most notably, it serves as the basic mechanism for broadcasting in protocols such as the IPv6 multicast protocol for low power and lossy networks (MPL) [9] and the IPv6 routeing protocol for low power and lossy networks (RPL) [22].

The original specification of Trickle defines a fixed global threshold which is the same for all nodes in the network. While this works well in networks with a uniform node density, it has several disadvantages in networks with a nonuniform node density [10], [21]. First, it is not clear how to set this threshold, especially since the network topology might be unknown or could change. Second, with a global threshold the transmission load is not evenly distributed among nodes in networks with a nonuniform node density, which can lead to less efficient data dissemination. For example, broadcasts of high-degree nodes might be suppressed often, even though these nodes are able to reach many other nodes.

Therefore, two recent papers introduce degree-dependent thresholds for Trickle in which nodes set their threshold based on the number of neighbors that they have [4], [13]. These approaches are shown to reduce unfairness among nodes, thus increasing the efficiency of data dissemination using Trickle.

While the introduction of degree-dependent thresholds for counter-based gossip schemes is natural, the question of how to set these thresholds remains. Degree-dependent counter-based broadcasting schemes have, to the best of the author's knowledge, not yet been mathematically analyzed, not even for simple network topologies. Therefore, the goal of this paper is to provide a first step towards the analysis and optimization of degree-dependent counter-based gossiping schemes.

3. Model description and dynamics

We stick as much as possible to the notation of [17], in which RSA with a hard-core blocking effect on a deterministic tree was treated.

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

We consider an infinite tree graph $G = (V, E)$, where each node $v \in V$ has D_v neighbors. Let the D_v be independent and identically distributed (i.i.d.) random variables and write $a_n := \mathbb{P}[D_v = n]$ for $n \in \mathbb{N}$. We require that $a_0 = a_1 = 0$, to be sure that the tree is infinite and does not have any leaf nodes. Additionally, each node v has an activation threshold B_v, which we allow to depend on the node's degree. We write $b_k^n := \mathbb{P}[B_v = k \mid D_v = n]$. Finally, each node v has an activation time T_v, where the T_v are i.i.d. uniform[0, 1] random variables.

Consider now a fixed realization of the tree. We will define a blocking process for all induced subgraphs of G. Let $W \subseteq V$. The dynamics of the blocking process on the subgraph of G induced by W are described by a stochastic jump process $(X_t^W,\ 0 \le t \le 1)$ taking values in $\{0, 1\}^W$. Here $X_t^W(v)$ is the state of node v at time t for the blocking process on the subgraph of G induced by the vertex set W: 0 if node v is inactive at time t and 1 if it is active. Initially, all nodes are inactive, that is, $X_0^W(v) = 0$ for all $v \in W$. Then, if at time T_v, fewer than B_v nodes in W that are adjacent to node v are active, node v will become irreversibly active, otherwise the node remains inactive forever.

More formally, denote by $N^W(v)$ the subset of nodes in W that are adjacent to node v on the graph G and define $S_t^W(v) = \sum_{w \in N^W(v)} X_t^W(w)$ to be the number of active neighbors of v at time t which are in W. Then, for $t < T_v$, $X_t^W(v) = 0$ for all v and, for $t \ge T_v$, $X_t^W(v) = 1$ if $S_{T_v}^W(v) < B_v$, and $X_t^W(v) = 0$ otherwise. For ease of notation, we will write $N(v) \equiv N^V(v)$, $S_t(v) \equiv S_t^V(v)$, and $X_t(v) \equiv X_t^V(v)$ whenever we are considering the complete node set V.

While it is clear that $X_t^W(v)$ is well defined for finite W, in the following proposition we show that, assuming D_v has a finite first moment, the process $(X_t,\ 0 \le t \le 1)$ on the infinite tree is also well defined.

Proposition 3.1. *Let $v \in V$ and define $V_n := \{w \in V \mid d(v, w) \le n\}$, where $d(v, w)$ denotes the number of edges in the path connecting v and w. Assuming that $\mathbb{E}[D_v] < \infty$, the random variable $X_t(v)$ is well defined and given by the limit*

$$X_t(v) := \lim_{n \to \infty} X_t^{V_n}(v). \tag{3.1}$$

Proof. Let $n_1 < n_2$. Then it should hold that $X_t^{V_{n_1}}(v) = X_t^{V_{n_2}}(v)$ unless there exists a path of vertices $v_0, v_1, \ldots, v_m$, starting at $v_0 = v$ and ending at some $v_m \in V_{n_2} \setminus V_{n_1}$, such that $T_{v_0} > T_{v_1} > \cdots > T_{v_m}$, since the only way that a vertex in $V_{n_2} \setminus V_{n_1}$ can influence v is if the activation of v_m somehow influences the activity of v_1, which, in turn, influences the activity of $v_0 = v$.

Hence, a sufficient condition for the existence of (3.1) is the existence of a finite number $R(v)$, such that there is no path $v_0, v_1, \ldots, v_m$ in V with $v_0 = v$, $m > R(v)$, and with $T_{v_0} > T_{v_1} > T_{v_2} > \cdots > T_{v_m}$, since then $X_t^{V_m}$ will be the same for all $m > R(v)$.

Let $Z_m = |V_m \setminus V_{m-1}|$ denote the number of vertices at distance m from v. Noting that the probability that $T_{v_0} > T_{v_1} > \cdots > T_{v_m}$ is equal to $1/(m + 1)!$, we can write

$$\mathbb{P}[X_t^{V_{m-1}}(v) \ne X_t(v) \mid Z_m = z] \le \mathbb{P}[R(v) \ge m \mid Z_m = z] \le \frac{z}{(m+1)!}.$$

Here, for the second inequality, we have assumed the activation timers of all z paths to be independent. Unconditioning, we find

$$\mathbb{P}[X_t^{V_{m-1}}(v) \ne X_t(v)] \le \mathbb{P}[R(v) \ge m] \le \frac{\mathbb{E}[Z_m]}{(m+1)!}.$$

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

This finally implies that

$$\sum_{m=0}^{\infty} \mathbb{P}[X_t^{V_m}(v) \neq X_t(v)] \leq \sum_{m=1}^{\infty} \mathbb{P}[R(v) \geq m] \leq \sum_{m=1}^{\infty} \frac{\mathbb{E}[Z_m]}{(m+1)!} < \infty,$$

since $\mathbb{E}[Z_m] = \mathbb{E}[D_v]\mathbb{E}[D_v - 1]^{m-1}$. Hence, by the Borel–Cantelli lemma, this implies almost sure convergence of the random variable $X_t^{V_m}(v)$. □

Finally, it is also important to note that the functions $N^W(v)$ and $S_t^W(v)$ are also defined for $v \notin W$. Moreover, in the rest of the paper, with some abuse of notation, we will write $X_t^W(W') = 0$ to denote the event that $X_t^W(w) = 0$ for all $w \in W' \subseteq W$ and, similarly, we will write $X_t^W(W') = 1$ to denote the event that $X_t^W(w) = 1$ for all $w \in W' \subseteq W$.

We are interested in the calculation of the following three quantities. First, we are interested in determining the time evolution of the activation probability $p_n(t)$ that a node with a given degree n is active at time t for the blocking process on G. Second, we want to find the correlation between the activity states of two nodes in V as a function of their distance. Finally, given the node degree distribution $\{a_n\}_{n=0}^{\infty}$ of the tree, we want to determine how to choose the threshold distribution b_k^n in order to achieve some desired values for the activation probabilities $p_n(1)$.

4. Activation probability

Consider an arbitrary vertex $v \in V$. Denote by $\alpha(t)$ the probability of finding v inactive at time $t \in [0, 1]$. That is,

$$\alpha(t) = \mathbb{P}[X_t(v) = 0].$$

Similarly, denote by $\beta(t)$ the probability of finding v inactive at time $t \in [0, 1]$ for the blocking process on the graph obtained by randomly removing one of the branches at node v in G. The probability $\beta(t)$ can be interpreted as the probability of finding v inactive at time t, given $T_w > t$ for some $w \in N(v)$. So, letting $w \in N(v)$, we can write

$$\beta(t) = \mathbb{P}[X_t^{V\setminus\{w\}}(v) = 0] = \mathbb{P}[X_t(v) = 0 \mid T_w > t].$$

The following proposition will be key in analyzing these functions.

Proposition 4.1. *Let $W \subseteq V$ and let W' and W'' be disjoint finite subsets of W. Then, for $0 \leq t < 1$,*

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}&\mathbb{P}[X_t^W(W') = 0, X_t^W(W'') = 1] \\ &= \sum_{v\in W''} \mathbb{P}[S_t^{W\setminus\{v\}}(v) < B_v, X_t^{W\setminus\{v\}}(W') = 0, X_t^{W\setminus\{v\}}(W'' \setminus \{v\}) = 1] \\ &\quad - \sum_{v\in W'} \mathbb{P}[S_t^{W\setminus\{v\}}(v) < B_v, X_t^{W\setminus\{v\}}(W' \setminus \{v\}) = 0, X_t^{W\setminus\{v\}}(W'') = 1]. \qquad (4.1)\end{aligned}$$

Proof. See Appendix A. □

Note that Proposition 4.1 is a generalization of Proposition 2 of [17], which applied only to hard-core blocking. Proposition 4.1 allows us to write differential equations for the probabilities of having particular configurations of active and inactive nodes. The first term on the right-hand side of (4.1) yields the rate at which the configuration is created and the second term the rate

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

at which it is destroyed. Moreover, Proposition 4.1 tells us that in order to analyze the time evolution of the probabilities of having particular configurations of active and inactive nodes on a certain graph, we have to analyze the blocking process on the same graph where we have removed certain nodes. This turns out to be a useful property, since by removing a node from a tree graph, we always end up with several unconnected trees which can be analyzed separately. This is illustrated by the following analysis.

We apply Proposition 4.1 to $\beta'(t) = (\mathrm{d}/\mathrm{d}t)\mathbb{P}[X_t^{V\setminus\{w\}}(v) = 0]$ with $w \in N(v)$. Since this configuration can be destroyed only by the activation of node v, we have to consider only the second term on the right-hand side of (4.1). Note that by removing v from our graph we create $D_v - 1$ identical unconnected copies of graphs; see Figure 1(a). Moreover, these graphs are identical in distribution to the original graph where only a branch at v had been removed. Hence, this allows us to write

$$\begin{aligned}\beta'(t) &= -\mathbb{P}[S_t^{V\setminus\{v,w\}}(v) < B_v] \\ &= -\sum_{n=1}^{\infty} a_n \sum_{k=1}^{\infty} b_k^n \mathbb{P}[S_t^{V\setminus\{v,w\}}(v) < k \mid D_v = n, B_v = k] \\ &= -\sum_{n=1}^{\infty} a_n \sum_{k=1}^{\infty} b_k^n \sum_{m=0}^{k-1} \binom{n-1}{m} \beta(t)^{n-m-1}(1-\beta(t))^m, \end{aligned} \tag{4.2}$$

where in the second step we condition on the degree and threshold of v and in the third step we use the fact that the activation probabilities of the remaining neighbors of v are independent and identical to $\beta(t)$. Combined with the initial condition $\beta(0) = 1$, we can conclude that $\beta(t)$ is implicitly given by

$$\int_{\beta(t)}^{1} \left(\sum_{n=1}^{\infty} a_n \sum_{k=1}^{\infty} b_k^n \sum_{m=0}^{k-1} \binom{n-1}{m} s^{n-m-1}(1-s)^m \right)^{-1} \mathrm{d}s = t. \tag{4.3}$$

Similarly, Proposition 4.1 can be applied to the function $\alpha'(t) = (\mathrm{d}/\mathrm{d}t)\mathbb{P}[X_t(v) = 0]$. Since this configuration can also be destroyed only by the activation of node v, we again have to consider only the second term on the right-hand side of (4.1). Note that by removing v from our graph, we now create D_v identical unconnected copies of graphs; see Figure 1(b). Hence,

$$\begin{aligned}\alpha'(t) &= -\mathbb{P}[S_t^{V\setminus\{v\}}(v) < B_v] \\ &= -\sum_{n=1}^{\infty} a_n \sum_{k=1}^{\infty} b_k^n \mathbb{P}[S_t^{V\setminus\{v\}}(v) < k \mid D_v = n, B_v = k] \\ &= -\sum_{n=1}^{\infty} a_n \sum_{k=1}^{\infty} b_k^n \sum_{m=0}^{k-1} \binom{n}{m} \beta(t)^{n-m}(1-\beta(t))^m, \end{aligned} \tag{4.4}$$

which yields, combined with the initial condition $\alpha(0) = 1$,

$$\alpha(t) = 1 - \int_0^t \sum_{n=1}^{\infty} a_n \sum_{k=1}^{\infty} b_k^n \sum_{m=0}^{k-1} \binom{n}{m} \beta(s)^{n-m}(1-\beta(s))^m \,\mathrm{d}s. \tag{4.5}$$

Finally, making the change of variables $x = \beta(s)$, (4.5) becomes

$$\alpha(t) = 1 - \int_{\beta(t)}^{1} \frac{\sum_{n=1}^{\infty} a_n \sum_{k=1}^{\infty} b_k^n \sum_{m=0}^{k-1} \binom{n}{m} x^{n-m}(1-x)^m}{\sum_{n=1}^{\infty} a_n \sum_{k=1}^{\infty} b_k^n \sum_{m=0}^{k-1} \binom{n-1}{m} x^{n-m-1}(1-x)^m} \,\mathrm{d}x. \tag{4.6}$$

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

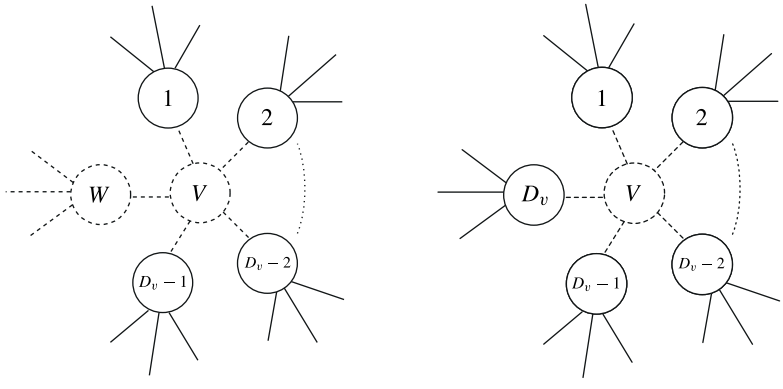


FIGURE 1: (a) Relevant configuration for $\beta'(t)$ and (b) relevant configuration for $\alpha'(t)$.

Note that, as is obvious from their definition, $\alpha(t)$ and $\beta(t)$ are nonincreasing functions of t, with $\alpha(0) = \beta(0) = 1$.

Finally, in (4.4) we could also have conditioned on the degree of node v. Doing so, we find that the expected fraction of nodes with degree n that are active at time t is given by

$$p_n(t) = \int_{\beta(t)}^{1} \frac{\sum_{k=1}^{\infty} b_k^n \sum_{m=0}^{k-1} \binom{n}{m} x^{n-m}(1-x)^m}{\sum_{n=1}^{\infty} a_n \sum_{k=1}^{\infty} b_k^n \sum_{m=0}^{k-1} \binom{n-1}{m} x^{n-m-1}(1-x)^m} \, \mathrm{d}x. \tag{4.7}$$

5. Activation covariance

In this section we derive a coupled set of differential equations, whose solutions allow us to determine the covariance between the activity states of two nodes that are at distance m from each other. First, this will provide some insight into the spatial distribution of active nodes. Second, knowing how fast correlations between node activities decrease as a function of distance allows us to gain an insight into the importance of having a tree-like graph. If correlations decrease relatively fast, this suggests that the analysis of the previous section might be successfully used to approximate the activation probabilities on graphs which are locally tree-like, such as the Erdős–Rényi random graph.

Let $v_1, v_2 \in V$ be a pair of nodes with $d(v_1, v_2) = m$ and let $v_1' \in N(v_1)$ and $v_2' \in N(v_2)$ be two nodes that are not on the path connecting v_1 with v_2. In order to obtain a closed system of equations, we define, for $m = 1, 2, \ldots$, the following correlation functions:

$$\begin{gathered} \alpha_m(t) = \mathbb{P}[X_t(v_1) = 0, X_t(v_2) = 0], \qquad \rho_m(t) = \mathbb{P}[X_t(v_1) = 1, X_t(v_2) = 0], \\ \zeta_m(t) = \mathbb{P}[X_t(v_1) = 1, X_t(v_2) = 1], \end{gathered} \tag{5.1}$$

$$\begin{aligned} \beta_m(t) &= \mathbb{P}[X_t^{V\setminus\{v_1'\}}(v_1) = 0, X_t^{V\setminus\{v_1'\}}(v_2) = 0], \\ \lambda_m(t) &= \mathbb{P}[X_t^{V\setminus\{v_1'\}}(v_1) = 1, X_t^{V\setminus\{v_1'\}}(v_2) = 0], \\ \mu_m(t) &= \mathbb{P}[X_t^{V\setminus\{v_1'\}}(v_1) = 0, X_t^{V\setminus\{v_1'\}}(v_2) = 1], \\ \eta_m(t) &= \mathbb{P}[X_t^{V\setminus\{v_1'\}}(v_1) = 1, X_t^{V\setminus\{v_1'\}}(v_2) = 1], \end{aligned} \tag{5.2}$$

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

and

$$\begin{aligned}\gamma_m(t) &= \mathbb{P}[X_t^{V\setminus\{v_1',v_2'\}}(v_1) = 0,\, X_t^{V\setminus\{v_1',v_2'\}}(v_2) = 0],\\ \kappa_m(t) &= \mathbb{P}[X_t^{V\setminus\{v_1',v_2'\}}(v_1) = 1,\, X_t^{V\setminus\{v_1',v_2'\}}(v_2) = 0],\\ \theta_m(t) &= \mathbb{P}[X_t^{V\setminus\{v_1',v_2'\}}(v_1) = 1,\, X_t^{V\setminus\{v_1',v_2'\}}(v_2) = 1].\end{aligned} \tag{5.3}$$

Our function of interest is

$$A_m(t) := \operatorname{cov}[X_t(v_1), X_t(v_2)] = \alpha_m(t) - \alpha(t)^2, \tag{5.4}$$

which will give us the correlation between the states of two nodes that are at distance m from each other.

Our approach to analyzing (5.4) is as follows. In Subsection 5.1 we illustrate how one can apply Proposition 4.1 to obtain a closed set of differential equations for the quantities in (5.1)–(5.3). This requires us to first introduce functions whose derivatives are similar in spirit to the derivatives in (4.2) and (4.4), capturing the probability that a node is unable to activate, given that one of its neighbors is already active. In Subsection 5.2, using probabilistic interpretations of (5.1)–(5.3), we derive some additional useful relations between these functions. Finally, transforming the differential equations from Subsection 5.1 and exploiting the relations from Subsection 5.2, we derive a differential equation for the covariance function $A_m(t)$ in Subsection 5.3. Note that since the analysis of (5.4) is relatively involved, the reader may decide to skip the remainder of the section at first reading.

5.1. Relating (5.1)–(5.3)

We apply Proposition 4.1 to obtain differential equations for the quantities in (5.1)–(5.3), but first we need to define some additional functions. Let $v \in V$. First, denote by $\gamma(t)$ the probability of finding node v inactive at time t for the blocking process on the graph obtained by randomly removing two of the branches at node v in G. Then, similar to the derivation of (4.2), using Proposition 4.1,

$$\gamma'(t) = -\sum_{n=1}^{\infty} a_n \sum_{k=1}^{\infty} b_k^n \sum_{m=0}^{k-1} \binom{n-2}{m} \beta(t)^{n-m-2}(1-\beta(t))^m. \tag{5.5}$$

Additionally, define $\beta_*(t)$ as the probability of finding node v, assuming its threshold has been decreased by 1, inactive at time t for the blocking process on the graph obtained by randomly removing one of the branches at node v in G. Similarly, denote by $\gamma_*(t)$ the probability of finding node v, assuming its threshold has been decreased by 1, inactive at time t for the blocking process on the graph obtained by randomly removing two of the branches at node v in G. Then $\beta_*'(t)$ and $\gamma_*'(t)$ are given by the same expressions as in (4.2) and (5.5), except that the threshold has been decreased by 1:

$$\begin{aligned}\beta_*'(t) &= -\sum_{n=1}^{\infty} a_n \sum_{k=2}^{\infty} b_k^n \sum_{m=0}^{k-2} \binom{n-1}{m} \beta(t)^{n-m-1}(1-\beta(t))^m,\\ -3pt\gamma_*'(t) &= -\sum_{n=1}^{\infty} a_n \sum_{k=2}^{\infty} b_k^n \sum_{m=0}^{k-2} \binom{n-2}{m} \beta(t)^{n-m-2}(1-\beta(t))^m.\end{aligned}$$

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

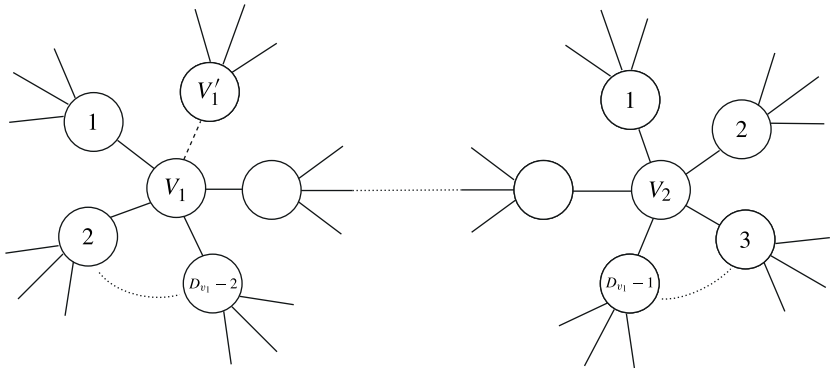


FIGURE 2: Local representation of the configuration relevant for $\lambda_m(t)$.

As an example, we illustrate how to apply Proposition 4.1 to the function $\lambda_m(t)$. Direct application of Proposition 4.1 yields

$$\begin{aligned}\lambda_m'(t) &= \frac{\mathrm{d}}{\mathrm{d}t}\mathbb{P}[X_t^{V\setminus\{v_1'\}}(v_1) = 1, X_t^{V\setminus\{v_1'\}}(v_2) = 0] \\ &= \mathbb{P}[S_t^{V\setminus\{v_1',v_1\}}(v_1) < B_v, X_t^{V\setminus\{v_1',v_1\}}(v_2) = 0] \\ &\quad - \mathbb{P}[S_t^{V\setminus\{v_1',v_2\}}(v_2) < B_v, X_t^{V\setminus\{v_1',v_2\}}(v_1) = 1]. \end{aligned} \tag{5.6}$$

In order to simplify these expressions, it might be useful to keep Figure 2 in mind.

For convenience, we will often drop the notation for dependence on t in the remainder of this section. Conditioning on the state of the neighbor of v_1 on the path between v_1 and v_2, we find for the first term:

$$\mathbb{P}[S_t^{V\setminus\{v_1',v_1\}}(v_1) < B_v,\ X_t^{V\setminus\{v_1',v_1\}}(v_2) = 0] = -\gamma'\beta_{m-1} - \gamma_*'\lambda_{m-1}. \tag{5.7}$$

For the second term, conditioning on the state of the neighbor of v_2 on the path between v_1 and v_2, we find

$$\mathbb{P}[S_t^{V\setminus\{v_1',v_2\}}(v_2) < B_v, X_t^{V\setminus\{v_1',v_2\}}(v_1) = 1] = -\beta'\kappa_{m-1} - \beta_*'\theta_{m-1}. \tag{5.8}$$

From (5.6)–(5.8), we conclude that $\lambda_m' = -\gamma'\beta_{m-1} - \gamma_*'\lambda_{m-1} + \beta'\kappa_{m-1} + \beta_*'\theta_{m-1}$. Applying Proposition 4.1 in the same way to (5.1), we obtain the following system of equations:

$$\begin{gathered}\alpha_m' = 2\beta'\beta_{m-1} + 2\beta_*'\lambda_{m-1}, \qquad \rho_m' = -\beta'\beta_{m-1} - \beta_*'\lambda_{m-1} + \beta'\mu_{m-1} + \beta_*'\eta_{m-1}, \\ \zeta_m' = -2\beta'\mu_{m-1} - 2\beta_*'\eta_{m-1},\end{gathered} \tag{5.9}$$

where $\alpha_0(t) := \alpha(t)$, $\rho_0(t) := 0$, and $\zeta_0(t) := 1 - \alpha(t)$.

Applying Proposition 4.1 to (5.2), we obtain

$$\begin{aligned}\beta_m' &= \gamma'\beta_{m-1} + \gamma_*'\lambda_{m-1} + \beta'\gamma_{m-1} + \beta_*'\kappa_{m-1}, \\ \lambda_m' &= -\gamma'\beta_{m-1} - \gamma_*'\lambda_{m-1} + \beta'\kappa_{m-1} + \beta_*'\theta_{m-1}, \\ \mu_m' &= -\beta'\gamma_{m-1} - \beta_*'\kappa_{m-1} + \gamma'\mu_{m-1} + \gamma_*'\eta_{m-1}, \\ \eta_m' &= -\beta'\kappa_{m-1} - \beta_*'\theta_{m-1} - \gamma'\mu_{m-1} - \gamma_*'\eta_{m-1},\end{aligned} \tag{5.10}$$

where $\beta_0(t) := \beta(t)$, $\lambda_0(t) := \mu_0(t) = 0$, and $\eta_0(t) := 1 - \beta(t)$.

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

Finally, applying Proposition 4.1 to (5.3), we obtain

$$\gamma_m' = 2\gamma'\gamma_{m-1} + 2\gamma_*'\kappa_{m-1}, \qquad \kappa_m' = -\gamma'\gamma_{m-1} - \gamma_*'\kappa_{m-1} + \gamma_*'\theta_{m-1} + \gamma'\kappa_{m-1}, \\ \theta_m' = -2\gamma_*'\theta_{m-1} - 2\gamma'\kappa_{m-1}, \tag{5.11}$$

where $\gamma_0(t) := \gamma(t)$, $\kappa_0(t) := 0$, and $\theta_0(t) := 1 - \gamma(t)$.

5.2. Some preliminary observations

In this section we derive some additional relations which will help us in deriving $A_m(t)$ from (5.9)–(5.11). Define $u(t) := \gamma(t) - \gamma_*(t)$ and $v(t) := \beta(t) - \beta_*(t)$. Then, using (5.10) and $\beta_m + \lambda_m + \mu_m + \eta_m = 1$, we can write

$$\beta_m' + \mu_m' = (\beta_{m-1} + \mu_{m-1})u' + \gamma_*'.$$

Noting that, for all $m \in \mathbb{N}$, we have $\beta = \beta_m + \mu_m$ and $1 - \beta = \lambda_m + \eta_m$, we find the following useful relation:

$$\beta' = \gamma_*' + \beta u' = \gamma' - (1 - \beta)u'. \tag{5.12}$$

Similarly, (5.9) yields

$$\alpha_m' + \rho_m' = (\beta_{m-1} + \mu_{m-1})v' + \beta_*'.$$

Noting that $\alpha_m + \rho_m = \alpha$ for all $m \in \mathbb{N}$ then yields

$$\alpha' = \beta_*' + \beta v' = \beta' - (1 - \beta)v'. \tag{5.13}$$

Now, define $f_m(t) = \gamma_m(t) + \kappa_m(t) = \mathbb{P}[X_t^{V\setminus\{v_1', v_2'\}}(v_2') = 0]$. Then (5.11) yields

$$f_m' = \gamma_m' + \kappa_m' = (\gamma_{m-1} + \kappa_{m-1})u' + \gamma_*' = f_{m-1}u' + \gamma_*'. \tag{5.14}$$

Also define $g_m(t) = \beta_m(t) + \lambda_m(t) = \mathbb{P}[X_t^{V\setminus\{v_1'\}}(v_2') = 0]$. Then (5.10) yields

$$g_m' = \beta_m' + \lambda_m' = (\gamma_{m-1} + \kappa_{m-1})v' + \beta_*' = f_{m-1}v' + \beta_*'. \tag{5.15}$$

5.3. The covariance function

Finally, to obtain a differential equation for $A_m(t)$, we define the following functions:

$$F_m(t) := f_m(t) - \beta(t), \qquad G_m(t) := g_m(t) - \alpha(t), \\ \Gamma_m(t) := \gamma_m(t) - \beta(t)^2, \qquad B_m(t) := \beta_m(t) - \alpha(t)\beta(t).$$

Note that all these functions converge to 0 as m tends to ∞. We introduce the following shorthand notation:

$$\int f \mathrm{d}^m g := \int_0^t \int_0^{x_1} \cdots \int_0^{x_{m-1}} f(x_m)\mathrm{d}g(x_m) \cdots \mathrm{d}g(x_2)\mathrm{d}g(x_1),$$

and $\int f \mathrm{d}g = \int f \mathrm{d}^1 g$. Then, combining (5.12) and (5.14), we find that

$$F_m' = u' F_{m-1}. \tag{5.16}$$

Moreover, by (5.12), we have $F_0(t) = \gamma(t) - \beta(t) = \int (1 - \beta)\,\mathrm{d}u$ with $u(t) := \gamma(t) - \gamma_*(t)$ and, therefore,

$$F_m = \int (\gamma - \beta)\,\mathrm{d}^m u = \int (1 - \beta)\,\mathrm{d}^{m+1} u. \tag{5.17}$$

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

Similarly, combining (5.13) and (5.15) yields

$$G'_m = \nu' F_{m-1}, \tag{5.18}$$

which implies that

$$G_m = \int\int (1-\beta)\, \mathrm{d}^m u \mathrm{d}v. \tag{5.19}$$

Now consider $\Gamma_m(t) = \gamma_m(t) - \beta(t)^2$. Some manipulation of (5.11) yields

$$\Gamma'_m = 2u'\Gamma_{m-1} + 2\gamma'_* F_{m-1}, \tag{5.20}$$

where $\Gamma_0(t) = \gamma(t) - \beta^2(t)$. Similarly, by manipulating (5.10), we obtain, for $B_m(t) = \beta_m(t) - \alpha(t)\beta(t)$,

$$B'_m = u' B_{m-1} + \nu'\Gamma_{m-1} + \gamma'_* G_{m-1} + \beta'_* F_{m-1}, \tag{5.21}$$

where $B_0(t) = \beta(t)(1-\alpha(t))$. Finally, for $A_m(t) = \alpha_m(t) - \alpha(t)^2$, we find that

$$A'_m = 2\nu' B_{m-1} + 2\beta'_* G_{m-1}, \tag{5.22}$$

where $A_0(t) = \alpha(t)(1-\alpha(t))$.

Note that while (5.16)–(5.22) form a closed set of recursive differential equations, in general they do not admit an explicit solution. However, as partially demonstrated in Appendix B, the solution to (5.22) can be simplified in such a way that $A_m(t)$ can be calculated by numerical integration for moderate values of m. Moreover, as also discussed in Appendix B, from (5.22), we can deduce that, in general, the correlations between nodes decay at a superexponential rate as their distance m increases. Finally, note that for the hard-core blocking case $B_v = 1$ for all $v \in V$, (5.16)–(5.22) simplify even more, as we will see in the following section.

6. Special cases

This section is devoted to the treatment of some special cases, which admit a detailed analysis.

6.1. The Bethe lattice

In this subsection we will consider the deterministic case that $D_v = z + 1$ for all $v \in V$, that is, G is a Bethe lattice with coordination number $z + 1$; see [16].

6.1.1. *Hard-core blocking.* Let $B_v = 1$ for all $v \in V$, that is, the activation of a node will block the activation of all its neighbors. Then, using (4.2), (4.4) can be written as $\alpha'(t) = \beta(t)\beta'(t)$. It follows then that (4.6) simplifies to the result from [17] (and earlier work):

$$\alpha(t) = \tfrac{1}{2}(1 + \beta(t)^2), \tag{6.1}$$

where solving (4.2) yields

$$\beta(t) = \begin{cases} \mathrm{e}^{-t} & \text{for } z = 1, \\ (1 + (z-1)t)^{-1/(z-1)} & \text{otherwise.} \end{cases}$$

Additionally, note that for this case $\gamma(t) = 1 + u(t)$, since $\gamma_*(t) = 1$, and (5.12) yields $\beta(t) = \mathrm{e}^{u(t)}$. Hence, (5.20) simplifies to

$$\Gamma'_m = 2u'\Gamma_{m-1}.$$

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

Now note that (5.13) yields $v' = u'\mathrm{e}^u$, so (5.21) becomes

$$B'_m = u'(B_{m-1} - \mathrm{e}^u \Gamma_{m-1}).$$

Finally, (5.22) yields

$$A'_m = 2u'\mathrm{e}^u B_{m-1}.$$

As shown by the result of Theorem 13 of [17], this coupled set of differential equations can be solved in terms of $u(t) = \log[\beta(t)]$ and has the following solution:

$$A_m(t) = -\frac{\beta^2(t)}{2} \sum_{n=0}^{\infty} \frac{(2\log[\beta(t)])^{2n+m+1}}{(2n+m+1)!}. \tag{6.2}$$

Note that, as was concluded in the previous section, (6.2) confirms that correlations between nodes become small very fast as their distance from each other increases, since the functions $A_m(t)$ decay at a superexponential rate.

One can also use the function $A_m(t)$ to calculate the variance of the scaled number of active nodes at time t. Let $v \in V$ and recall that $V_n := \{w \in V \mid d(v, w) \le n\}$. For finite $W \subset V$, define $Y_t(W) = \sum_{w\in W} X_t(w)$. Write

$$\sigma_z^2(t) := \lim_{n\to\infty} (|V_n|)^{-1} \operatorname{var}[Y_t(V_n)].$$

Denoting by $P_z^{n,m}$ the number of directed self-avoiding paths of length m on the graph induced by V_n, we have, by translation invariance,

$$\sigma_z^2(t) = \lim_{n\to\infty} (|V_n|)^{-1} \sum_{v,w\in V_n} A_{d(v,w)}(t) = A_0(t) + \lim_{n\to\infty} \sum_{m=1}^{\infty} \frac{P_z^{n,m}}{|V_n|} A_m(t). \tag{6.3}$$

We can now state the following proposition concerning a central limit theorem for $Y_t(V_n)$.

Proposition 6.1. *Let $D_v = z + 1$ for all $v \in V$. Then*

$$\frac{Y_t(V_n)}{|V_n|} \to \mathcal{N}(\alpha(t), \sigma_z^2(t)), \qquad n \to \infty. \tag{6.4}$$

Also, if $B_v = 1$ for all $v \in V$, then, for $z = 1$, we have $\sigma_1^2(t) = t\mathrm{e}^{-4t}$ and, for $z > 1$,

$$\sigma_z^2(t) = \frac{\beta(t)^4}{z-1} + \frac{(1-\sqrt{z})^2\beta(t)^{2(1-\sqrt{z})} - (1+\sqrt{z})^2\beta(t)^{2(1+\sqrt{z})}}{4\sqrt{z}(z-1)}.$$

Proof. See Appendix C. □

6.1.2. *Deterministic thresholds.* Consider again the deterministic case of the Bethe lattice, that is, $D_v = z + 1$ for all nodes $v \in V$, and let $B_v = k$ for all nodes $v \in V$. Then (4.6) can be written as

$$\alpha(t) = 1 - \int_{\beta(t)}^{1} \frac{\sum_{m=0}^{k-1} \binom{z+1}{m} x^{z-m+1}(1-x)^m}{\sum_{m=0}^{k-1} \binom{z}{m} x^{z-m}(1-x)^m} \,\mathrm{d}x,$$

where

$$\int_{\beta(t)}^{1} \left(\sum_{m=0}^{k-1} \binom{z}{m} s^{z-m}(1-s)^m \right)^{-1} \mathrm{d}s = t.$$

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

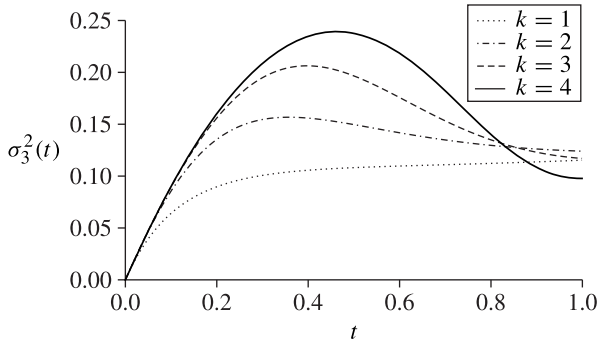


FIGURE 3: Plots of $\sigma_3^2(t)$ for $k = 1, 2, 3, 4$.

Simplifying the solution to (5.22) is difficult for this more general case. However, it can be rewritten such that it can be calculated numerically; see Appendix B. In combination with (6.3), this then allows us to numerically evaluate the normalized variance function for a combination of z and k. To illustrate this, we present the plot of $\sigma_z^2(t)$ for $z = 3$ and $k = 1, 2, 3, 4$ in Figure 3. It is interesting to note that, perhaps counterintuitively, there does not exist a strict ordering of the functions $\sigma_3^2(t)$ for $k = 1, 2, 3, 4$.

6.2. The random tree

In this subsection we will consider the more general case of the random tree. Note that the cases for which $B_v = 1$ for all nodes $v \in V$ and D_v follow a Poisson, geometric, or binomial distribution was treated in [20].

6.2.1. *Hard-core blocking.* Consider the case that $B_v = 1$ for all nodes $v \in V$. Since in this case (4.4) can again be written as $\alpha'(t) = \beta(t)\beta'(t)$, (4.6) simplifies to the same expression as in (6.1). Moreover, as was also shown in [5], writing $G(s) = \sum a_n s^n$, $\beta(t)$ is given by

$$\int_{\beta(t)}^{1} \frac{s}{G(s)}\,\mathrm{d}s = t.$$

Additionally, Theorem 13 of [17] can also be easily extended to the case of a random tree, leading to the same expression as in (6.2), that is,

$$A_m(t) = -\frac{\beta^2(t)}{2}\sum_{n=0}^{\infty}\frac{(2\log[\beta(t)])^{2n+m+1}}{(2n+m+1)!}.$$

6.2.2. *Threshold equal to node degree.* Consider the case that $B_v \equiv D_v$ for all nodes $v \in V$. Then, we have $\beta(t) = 1 - t$ and, hence, (4.5) simplifies to

$$\alpha(t) = 1 - \int_0^t \sum_{n=1}^{\infty} a_n(1 - s^n)\,\mathrm{d}s = 1 - t + \int_0^t G(s)\,\mathrm{d}s.$$

Note that this implies that $\alpha(1) = \sum_{n=0}^{\infty} a_n/(n+1)$, which makes sense, since the only way that a node remains inactive is if it tries to activate after all its neighbors.

Consider, for example, the case where $D_v - 2$ follows a Poisson distribution with mean λ. In this case, we have $G(s) = s^2 \mathrm{e}^{\lambda(s-1)}$, which allows explicit calculation of $\alpha(t)$:

$$\alpha(t) = 1 - t + \lambda^{-3}(2 - \lambda t(2 - \lambda t))\mathrm{e}^{-\lambda(1-t)} - 2\lambda^{-3}\mathrm{e}^{-\lambda}.$$

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

6.2.3. *Uniformly distributed thresholds.* Let

$$b_k^n = \mathbb{P}[B_v = k \mid D_v = n] = \begin{cases} \dfrac{1}{n} & \text{if } 1 \le k \le n, \\ 0 & \text{otherwise,} \end{cases}$$

then (4.7) simplifies, for all n, to

$$p_n(t) = \int_{\beta(t)}^1 \frac{x}{x(1-c)+c}\,\mathrm{d}x = \frac{1 - c(1-c)t - \mathrm{e}^{-(1-c)t}}{(1-c)^2},$$

where

$$c = \int_0^1 \frac{G[z]}{z}\,\mathrm{d}z \quad \text{and} \quad \beta(t) = \frac{\mathrm{e}^{-(1-c)t} - c}{1-c}.$$

This implies that a very straightforward scheme to give all nodes an equal probability of activating is for each node v to set its threshold uniformly between 1 and D_v.

7. Configuring the threshold

In Section 4 we have focused on how to determine the probability that a node will be able to activate, given the degree distribution of the graph and degree-dependent thresholds. In this section we focus on solving the inverse problem: given some desired degree-dependent activation probabilities $p_n(1)$, we want to choose the thresholds in such a way that we achieve these probabilities. For example, with scalability in mind, one could be interested in setting the thresholds such that the probability that a node is able to activate is inversely proportional to its degree.

The structure of this section is as follows. First, we introduce a continuous version of a deterministic degree-dependent threshold. Second, we prove that for any vector of desired degree-dependent activation probabilities there exist such degree-dependent thresholds, which, when applied to the infinite tree, achieve these desired activation probabilities. Finally, we propose an efficient iterative algorithm in order to determine these thresholds.

We introduce a continuous version of a deterministic degree-dependent activation threshold. Define $\mathcal{N} := \{n : a_n > 0\}$, that is, the set of node degrees that appear in G. For practical purposes, we assume $|\mathcal{N}|$ to be finite. Given a threshold vector $\boldsymbol{k} = (k_n)_{n\in\mathcal{N}}$, we now set the b_k^n for all $n \in \mathcal{N}$ as follows:

$$b_k^n = \mathbb{P}[B_v = k \mid D_v = n] = \begin{cases} k_n - \lfloor k_n \rfloor & \text{if } k = \lceil k_n \rceil, \\ 1 - (k_n - \lfloor k_n \rfloor) & \text{if } k = \lfloor k_n \rfloor, \\ 0 & \text{otherwise,} \end{cases} \tag{7.1}$$

Naturally, in the case that $\lceil k_n \rceil = \lfloor k_n \rfloor$, we set $b_{k_n}^n = 1$. Note that this implies $\mathbb{E}[B_v \mid D_v = n] = k_n$ and that a node with degree n will either have $\lceil k_n \rceil$ or $\lfloor k_n \rfloor$ as its activation threshold.

Let $\hat{\boldsymbol{p}} = (\hat{p}_n)_{n\in\mathcal{N}}$, with $p_n \in [0,1]$ for all $n \in \mathcal{N}$ and $\sum_{n\in\mathcal{N}} p_n = 1$, be the desired activation probability vector. Our goal is to set the thresholds k_n such that we achieve $p_n(1) = \hat{p}_n$. We first introduce some additional notation. For convenience, we will sometimes use the representation $k_n := (n+1)x_n$ with $x_n \in [0,1]$. Note that this allows us to pick any nontrivial set of activation thresholds, since for all $k_n > n+1$ we have $p_n(1) = 1$. Now, for $\boldsymbol{x} = \{x_n\}_{n\in\mathcal{N}} \in [0,1]^{|\mathcal{N}|}$, let the function $\pi_n(\boldsymbol{x})$ be equal to $p_n(1)$ as in (4.7), where the b_k^n are set in accordance with (7.1) with $k_n = (n+1)x_n$. Thus, $\pi_n(\boldsymbol{x})$ yields the probability that

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

a node with n neighbors activates, if we should use $k_n = (n+1)x_n$ as the degree-dependent activation thresholds. Similarly, let $\beta(t, \boldsymbol{x})$ be equal to $\beta(t)$ as in (4.3), with again the b_k^n set in accordance to (7.1) with $k_n = (n+1)x_n$.

We are now in a position to formulate the following proposition on the existence of threshold values for any desired vector of activation probabilities.

Proposition 7.1. *For every degree distribution $\{a_i\}_{i=0}^{\infty}$ and the activation probability vector $(\hat{p}_n)_{n\in\mathcal{N}}$, there exists a vector $\{x_n\}_{n\in\mathcal{N}} = \boldsymbol{x} \in [0,1]^{|\mathcal{N}|}$ such that $\pi_n(\boldsymbol{x}) = \hat{p}_n$ for all $n \in \mathcal{N}$.*

Proof. Let $\boldsymbol{x} \in [0,1]^{|\mathcal{N}|}$ and consider the vector-valued function $\mathcal{P}(\boldsymbol{x}) = (\pi_n(\boldsymbol{x}) - \hat{p}_n)_{n\in\mathcal{N}}$, which is continuous in $\boldsymbol{x}$. Note that, for any $n \in \mathcal{N}$, we have $\pi_n(\boldsymbol{x}) = 0$ if $x_n = 0$ and $\pi_n(\boldsymbol{x}) = 1$ if $x_n = 1$. Therefore, by the Poincaré–Miranda theorem [11], which is a generalization of the intermediate value theorem, there exists a point $\boldsymbol{x} \in [0,1]^{|\mathcal{N}|}$ for which the function $\mathcal{P}(\boldsymbol{x})$ vanishes. Therefore, there exists a vector $\boldsymbol{x} \in [0,1]^{|\mathcal{N}|}$ such that $\pi_n(\boldsymbol{x}) = \hat{p}_n$ for all $n \in \mathcal{N}$. This completes the proof. □

In order to determine the vector $\boldsymbol{x}$ we propose an iterative algorithm. First, we introduce the following function:

$$\phi(\boldsymbol{x}, k, n) = \int_0^1 \Bigg(\sum_{m=0}^{\lfloor k \rfloor - 1} \binom{n}{m} \beta(s, \boldsymbol{x})^{n-m} (1 - \beta(s, \boldsymbol{x}))^m + (k - \lfloor k \rfloor) \binom{n}{\lfloor k \rfloor} \beta(s, \boldsymbol{x})^{n - \lfloor k \rfloor} (1 - \beta(s, \boldsymbol{x}))^{\lfloor k \rfloor} \Bigg) \mathrm{d}s.$$

It is important to note that $\phi(\boldsymbol{x}, (n+1)x_n, n) = \pi_n(\boldsymbol{x})$. Additionally, consider the situation where we have changed the activation threshold distribution of a single node with degree n to (7.1) with $k_n = k$, while all other nodes in V use the thresholds according to (7.1) with $k_n = (n+1)x_n$. Then, $\phi(\boldsymbol{x}, k, n)$ yields the probability that the node with the changed threshold distribution will be able to activate.

We now introduce the activation threshold algorithm.

Algorithm 7.1. (*Activation threshold algorithm.*) Set $i = 0$ and start with the vector $\boldsymbol{x}^{(0)} = (1)_{n\in\mathcal{N}}$. Iterate between the following steps until the desired precision is reached.

1. For all $n \in \mathcal{N}$, determine $k_n^{(i+1)}$ such that
$$\phi(\boldsymbol{x}^{(i)}, k_n^{(i+1)}, n) = \hat{p}_n. \tag{7.2}$$
2. Set $\boldsymbol{x}^{(i+1)} = (k_n^{(i+1)}/(n+1))_{n\in\mathcal{N}}$ and increase i by 1.

Essentially what we are doing in step 2 of the algorithm is the following. Given the current threshold values, we determine how we should reset the threshold of a single node with degree n, in order that it has an activation probability equal to $\hat{p}_n$, if all other nodes continue using the current threshold values. Having calculated this for all $n \in \mathcal{N}$, we then give all nodes the new thresholds and repeat the same calculation.

Theorem 7.1. *The sequence $\{\boldsymbol{x}^{(i)}\}_{i=0}^{\infty}$ generated by Algorithm 7.1 converges to a vector $\boldsymbol{x}^*$, such that $\pi_n(\boldsymbol{x}^*) = \hat{p}_n$ for all $n \in \mathcal{N}$.*

Proof. First, note that $\pi_n(\boldsymbol{x}^{(0)}) = 1 \geq \hat{p}_n$ for all n, which implies that $x_n^{(1)} \leq x_n^{(0)}$ and, consequently, $\pi_n(\boldsymbol{x}^{(1)}) \leq \pi_n(\boldsymbol{x}^{(0)})$ for all $n \in \mathcal{N}$. Moreover, since $x_n^{(1)} \leq x_n^{(0)}$ for all $n \in \mathcal{N}$,

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

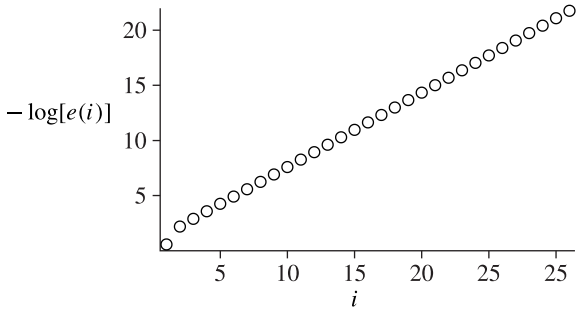


FIGURE 4: Logarithmic plot of the maximum absolute error function as a function of the number of iterations of Algorithm 7.1.

this implies, by (4.3), that $\beta(t, \boldsymbol{x}^{(1)}) \geq \beta(t, \boldsymbol{x}^{(0)})$ for all $t \in [0, 1]$. Now note that this also implies that, for any $n \in \mathcal{N}$, $k \leq n$, and $t \in [0, 1]$,

$$\sum_{m=0}^{k} \binom{n}{m} \beta(t, \boldsymbol{x}^{(1)})^{n-m} (1 - \beta(t, \boldsymbol{x}^{(1)}))^m \geq \sum_{m=0}^{k} \binom{n}{m} \beta(t, \boldsymbol{x}^{(0)})^{n-m} (1 - \beta(t, \boldsymbol{x}^{(0)}))^m. \quad (7.3)$$

This is best seen in a probabilistic way. Let X_j, $j = 1, \ldots, n$, be i.i.d. random variables uniformly distributed in $[0, 1]$. Then the left-hand side of (7.3) is equal to the probability that $X_j \leq \beta(t, \boldsymbol{x}^{(1)})$ for at least $n - k$ values of j. Clearly, since $\beta(t, \boldsymbol{x}^{(1)}) \geq \beta(t, \boldsymbol{x}^{(0)})$, this probability is greater than the probability that $X_j \leq \beta(t, \boldsymbol{x}^{(0)})$ for at least $n - k$ values of j, which is the right-hand side of (7.3).

Hence, from (7.3), we can conclude that

$$\pi_n(\boldsymbol{x}^{(1)}) = \phi(\boldsymbol{x}^{(1)}, k_n^{(1)}, n) \geq \phi(\boldsymbol{x}^{(0)}, k_n^{(1)}, n) = \hat{p}_n.$$

Therefore, since $\pi_n(\boldsymbol{x}^{(1)}) \geq \hat{p}_n$ for all $n \in \mathcal{N}$, this implies that $x_n^{(2)} \leq x_n^{(1)}$ and, consequently, $\pi_n(\boldsymbol{x}^{(1)}) \geq \pi_n(\boldsymbol{x}^{(2)}) \geq \hat{p}_n$. More generally, we can then conclude that $x_n^{(i+1)} \leq x_n^{(i)}$ and since $x_n^{(i+1)}$ is bounded it must converge to some x_n^* and, hence, by (7.2), we have $\pi_n(\boldsymbol{x}^*) = \phi(\boldsymbol{x}^*, (n+1)x_n^*, n) = \hat{p}_n$. □

To conclude this section, we show, by means of an example, that $\boldsymbol{x}^{(n)}$ of Algorithm 7.1 converges to a vector $\boldsymbol{x}^*$ exponentially fast. Assume that $D_v - 2$ follows a binomial distribution with parameters 18 and $\frac{1}{2}$, that is,

$$a_n = \binom{18}{n-2} \left(\frac{1}{2}\right)^{18} \quad \text{for } n = 2, 3, \ldots, 20.$$

Furthermore, assume that the desired activation probabilities are given by $\hat{p}_n = 1/2$, for $n = 2, 3, \ldots, 20$. Define the maximum absolute error function $e(i) := \max_{n \in \mathcal{N}} |\pi_n(\boldsymbol{x}^{(i)}) - \hat{p}_n|$. In Figure 4 we present the logarithm of the error function $e(i)$ as a function of the number of iterations of Algorithm 7.1. We see that $\pi_n(\boldsymbol{x}^{(i)})$ converges exponentially fast to the desired activation probabilities.

8. Configuring thresholds in finite networks

Suppose now that we wish to apply the thresholds generated by Algorithm 7.1 to an actual wireless network using a counter-based broadcasting protocol. In practice such networks are not infinite in size. Moreover, in general, we cannot rely on the network having a tree structure.

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

However, assuming that the network is relatively large and that the connectivity graph does have some tree-like properties, we could treat it as an infinite tree. That is, we set the a_n to be the fraction of nodes in our graph with node degree n and apply Algorithm 7.1 to generate an activation threshold vector $\boldsymbol{k}$, as if the network were an infinite tree. We then apply these thresholds to our original network, which is finite and not a tree, in the hope that we still achieve our target probabilities $\hat{p}_n$. In this section we conduct several simulation experiments, showing that this approach achieves a relatively good performance.

8.1. A finite tree

We first study a network graph which does have a tree structure, but is finite in size. Note that such network structures are relatively common in the context of the internet of things. For example, such structures often arise in sensor networks, where nodes need to periodically send sensor data, which is to be collected at some central node. An example of a connectivity graph of a network consisting of 100 nodes and its corresponding degree distribution is shown in Figure 5.

Let us first assume that for the network in Figure 5 our target activation probabilities are given by $\hat{p}_n = \frac{1}{2}$, that is, we want each node in the network to have the same probability of activating. Requiring a precision of five digits, running Algorithm 7.1 with these target probabilities produces the thresholds k_n, presented in Figure 6(a), after only 18 iterations. Recall that the values k_n determine the activation threshold distribution as defined by (7.1). Applying these thresholds to our original network and simulating the blocking process for 10^5 rounds produces the graph in Figure 6(b). In this graph each cross represents a node and its activation probability averaged over all rounds, hence there are 100 crosses. The dashed line represents our target probabilities.

We can conclude that the average activation probabilities of all nodes are remarkably close to the desired value of $\frac{1}{2}$, even though the network is not infinite. Furthermore, it is worth noting that the thresholds found by Algorithm 7.1 appear to depend linearly on the node degree. This is intuitive, since in this example all nodes aim to have an activation probability of $\frac{1}{2}$ and, therefore, the threshold of a node with n neighbors should roughly equal $\frac{1}{2}n$. This is in agreement with the trend that we see in Figure 6(a).

As a second example, assume now that our target activation probabilities are given by $\hat{p}_n = 2/(n+1)$, that is, we want the probability that a node is able to activate to be inversely proportional to its degree. Requiring a precision of five digits, Algorithm 7.1 produces the thresholds plotted in Figure 7(a) after eight iterations. Applying this sequence to our network and simulating the blocking process for 10^5 rounds produces the graph in Figure 7(b). In this

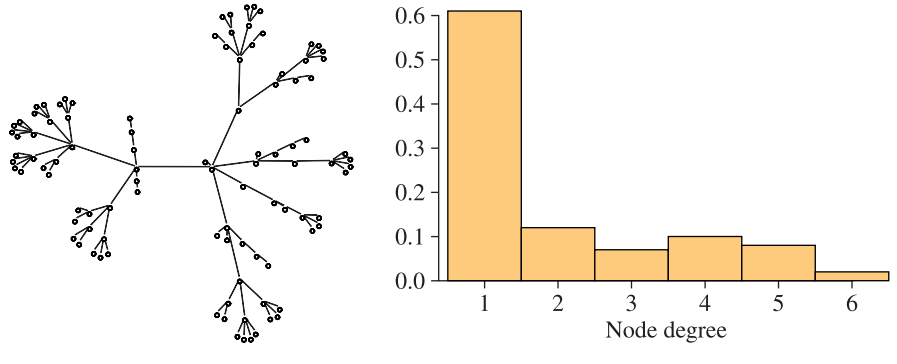


FIGURE 5: A connectivity graph (*left*) and the corresponding degree distribution (*right*).

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

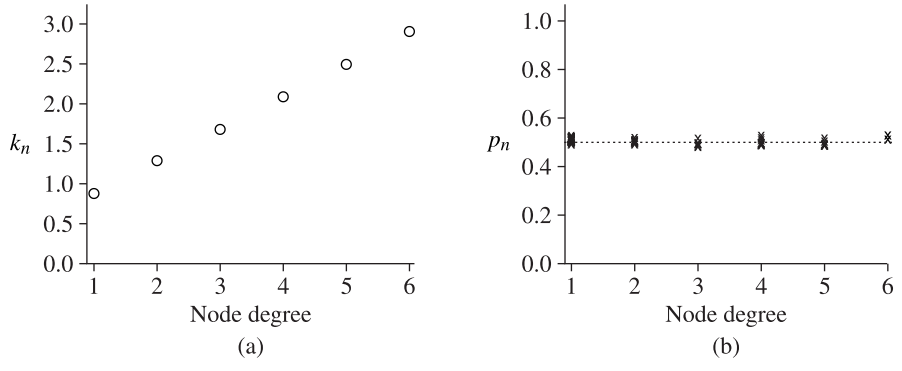


FIGURE 6: (a) Thresholds generated by Algorithm 7.1 for $\hat{p}_n = \frac{1}{2}$. (b) Scatter plot of average activation probabilities for $\hat{p}_n = \frac{1}{2}$.

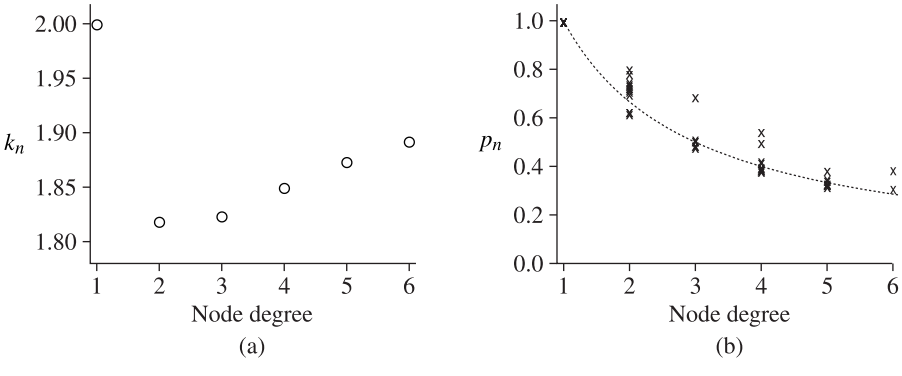


FIGURE 7: (a) Thresholds generated by Algorithm 7.1 for $\hat{p}_n = 2/(n+1)$. (b) Scatter plot of average activation probabilities for $\hat{p}_n = 2/(n+1)$.

graph each cross represents a node and its activation probability averaged over all rounds and the dashed line represents the target probabilities.

Also, here we see that the activation probabilities are very close to the desired values. Note that in this case we see a little more variation in the achieved activation probabilities. This is to be expected since some nodes want to achieve different activation probabilities than others, depending on their degree, therefore there is more variation in the activity of neighboring nodes. In other words, some nodes may have many neighbors with high activation probabilities and some nodes may have many neighbors with low activation probabilities. Therefore, since we set thresholds based solely on the number of neighbors a node has, this results in more variation in the achieved activation probabilities.

8.2. A random geometric graph

We will now consider networks represented by a random geometric graph, which are neither infinite nor a tree. These types of graph are of interest since they often serve as a basic model for connectivity in wireless networks. An example of a connectivity graph of such a network consisting of 200 uniformly placed nodes in a square area and its corresponding degree distribution is presented in Figure 8.

First, assume that, for the network in Figure 8, our target activation probabilities are again given by $\hat{p}_n = \frac{1}{2}$, that is, we want each node in the network to have the same probability of

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

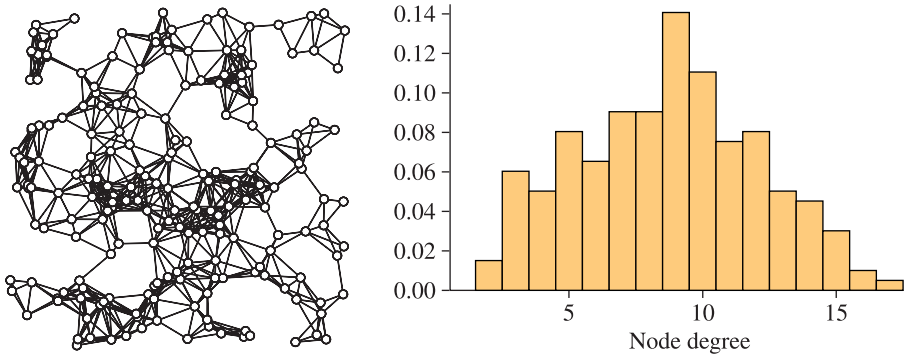


FIGURE 8: A connectivity graph (*left*) and the corresponding degree distribution (*right*).

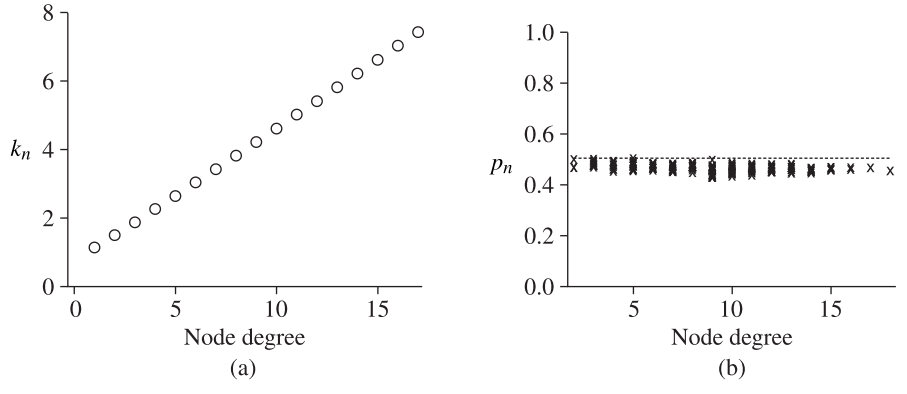


FIGURE 9: (a) Thresholds generated by Algorithm 7.1 for $\hat{p}_n = \frac{1}{2}$. (b) Scatter plot of average activation probabilities for $\hat{p}_n = \frac{1}{2}$.

activating. Requiring again a precision of five digits, Algorithm 7.1 produces the thresholds, plotted in Figure 9(a), after 15 iterations. Applying this sequence to our original network and simulating the blocking process for 10^5 rounds produces the scatter plot presented in Figure 9(b).

We find that even though our network is far from tree-like and has many clusters and loops, the average activation probabilities are still very close to the desired values. A possible explanation for this is that, in general, correlations between the activity states of nodes decay extremely fast as a function of their distance, as demonstrated in Appendix B, and, hence, having loops in the network has only a small influence. Additionally, note again that the thresholds in this example seem to depend linearly on the node degree.

As a final example, assume that, for the same network, our target activation probabilities are now given by $\hat{p}_n = 3/(n+1)$. Requiring a precision of five digits, running Algorithm 7.1 with these target probabilities produces the thresholds, presented in Figure 10(a), after 18 iterations. Applying this sequence to the network of Figure 8 and again simulating the blocking process for 10^5 rounds produces the graph presented in Figure 10(b).

Again we can conclude that the average activation probabilities of all nodes are relatively close to the desired values, even though the network is neither tree-like nor infinite. Additionally, note that in this case we see a little more variation in the achieved activation probabilities, as was also the case for the final example on the finite tree of the previous subsection. Here we also

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

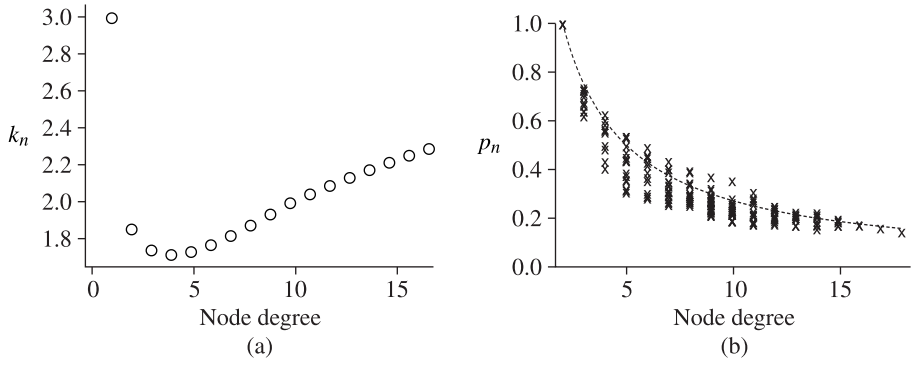


FIGURE 10: (a) Thresholds generated by Algorithm 7.1 for $\hat{p}_n = 3/(n+1)$. (b) Scatter plot of average activation probabilities for $\hat{p}_n = 3/(n+1)$.

see that this variation appears to be greater for low-degree nodes than for high-degree nodes. This seems reasonable, since a low-degree node can easily have only neighbors with high target activation probabilities or only neighbors with low target activation probabilities. This issue should play less of a role for high-degree nodes, where this effect evens out more easily.

9. Conclusion

Counter-based gossiping has proven to be a reliable and scalable approach to implement broadcasting. While traditional counter-based schemes generally use a fixed global activation threshold for all nodes in the network, recent work has shown that the use of degree-dependent activation thresholds can greatly improve the performance of these schemes, especially when used in networks with nonuniform node density. However, little is known about how one should configure these degree-dependent activation thresholds. Most of the papers on this topic rely on experimental results and simulations for determining how to configure the thresholds, which has the drawback in that they consider only a few specific network topologies and the results are difficult to generalize.

By modeling counter-based broadcasting schemes as a special case of RSA, generalizing hard-core blocking RSA, we hope to provide a first step towards a better understanding of counter-based gossip protocols. For infinite tree graphs, we have shown how to determine the time evolution of the probability that a node is able to activate, given its degree. Moreover, we have developed a general method for determining the correlation between the activities of nodes as a function of their distance. These correlations are shown to fall off extremely fast, suggesting that the analysis of the infinite tree network can provide good approximations for finite nontree networks.

Additionally, we have introduced a continuous version of deterministic degree-dependent activation thresholds and have proposed the activation threshold algorithm (Algorithm 7.1) which, given some desired degree-dependent activation probabilities, allows us to determine how to set these activation thresholds such that we achieve these probabilities. For the infinite tree, we have shown that

1. for any desired degree-dependent activation probability vector, there exist activation thresholds achieving those probabilities and
2. the activation threshold algorithm is able to determine those thresholds quickly.

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

As future work, we plan to study the correlation function $A_m(t)$ in more detail. In particular, achieving a better understanding about the decay of $A_m(t)$ as m decreases could provide additional insight into the applicability of our analysis to nontree networks. Moreover, it is interesting to analyze the effect of introducing loops in a tree network on the activation probabilities and correlation function, as it could lead to error bounds on the achieved activation probabilities when applying thresholds generated by the activation threshold algorithm to nontree networks. Finally, it would also be interesting to investigate the impact of not knowing the exact degree distribution of a network and how using an approximate distribution influences the performance of the activation threshold algorithm. This is especially relevant in the case of communication networks, where one can generally not assume that the exact degree distribution is known.

Appendix A.

Proof of Proposition 4.1. The steps of the proof are similar to those in the proof of Proposition 2 of [17]. For $v \in W'$ and $0 \le t < t+h < 1$, we have

$$\begin{aligned}
&\mathbb{P}[X_{t+h}^W(v) = 1, X_t^W(W') = 0, X_t^W(W'') = 1] \\
&\quad = \mathbb{P}[T_v \in (t, t+h), S_t^W(v) < B_v, X_t^W(W') = 0, X_t^W(W'') = 1] + o(h) \\
&\quad = h\mathbb{P}[S_t^W(v) < B_v, X_t^W(W') = 0, X_t^W(W'') = 1 \mid T_v \in (t, t+h)] + o(h) \\
&\quad = h\mathbb{P}[S_t^{W\setminus\{v\}}(v) < B_v, X_t^{W\setminus\{v\}}(W' \setminus \{v\}) = 0, X_t^{W\setminus\{v\}}(W'') = 1] + o(h),
\end{aligned}$$

where the second equality follows from the fact that T_v follows a uniform distribution. Summing over $v \in W'$, we obtain

$$\begin{aligned}
&\sum_{v\in W'} \mathbb{P}[X_{t+h}^W(v) = 1, X_t^W(W') = 0, X_t^W(W'') = 1] \\
&\qquad = h\left(\sum_{v\in W'} \mathbb{P}[S_t^{W\setminus\{v\}}(v) < B_v, X_t^{W\setminus\{v\}}(W' \setminus \{v\}) = 0, X_t^{W\setminus\{v\}}(W'') = 1]\right) + o(h).
\end{aligned}$$

Similarly, for $0 < h < t$,

$$\begin{aligned}
&\sum_{v\in W'} \mathbb{P}[X_t^W(v) = 1, X_{t-h}^W(W') = 0, X_{t-h}^W(W'') = 1] \\
&\qquad = h\left(\sum_{v\in W'} \mathbb{P}[S_{t-h}^{W\setminus\{v\}}(v) < B_v, X_{t-h}^{W\setminus\{v\}}(W' \setminus \{v\}) = 0, X_{t-h}^{W\setminus\{v\}}(W'') = 1]\right) + o(h) \\
&\qquad = h\left(\sum_{v\in W'} \mathbb{P}[S_t^{W\setminus\{v\}}(v) < B_v, X_t^{W\setminus\{v\}}(W' \setminus \{v\}) = 0, X_t^{W\setminus\{v\}}(W'') = 1]\right) + o(h),
\end{aligned}$$

where the last equality follows from the left-continuity in t. In a similar fashion, we can deduce that

$$\begin{aligned}
&\sum_{v\in W''} \mathbb{P}[X_t^W(v) = 0, X_{t+h}^W(W') = 0, X_{t+h}^W(W'') = 1] \\
&\qquad = h\left(\sum_{v\in W''} \mathbb{P}[S_t^{W\setminus\{v\}}(v) < B_v, X_t^{W\setminus\{v\}}(W') = 0, X_t^{W\setminus\{v\}}(W'' \setminus \{v\}) = 1]\right) + o(h),
\end{aligned}$$

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

and

$$\sum_{v\in W''} \mathbb{P}[X_{t-h}^{W}(v)=0, X_t^W(W')=0, X_t^W(W'')=1]$$
$$= h\Bigg(\sum_{v\in W''} \mathbb{P}[S_t^{W\setminus\{v\}}(v) < B_v, X_t^{W\setminus\{v\}}(W')=0, X_t^{W\setminus\{v\}}(W''\setminus\{v\})=1]\Bigg) + o(h).$$

Combining all results completes the proof. □

Appendix B. Simplifying the covariance function

In this appendix we demonstrate how one can derive expressions for F_m, G_m, Γ_m, B_m, and A_m, which allow for numerical calculation of these functions. First, by the Cauchy formula for repeated integration, we can write (5.17) as

$$\begin{aligned} F_m(t) &= \int_0^t F_0(x)\,\mathrm{d}^m u(x) \\ &= \int_0^t \frac{(u(t)-u(x))^m}{m!}\,\mathrm{d}F_0(x) \\ &= \int_0^t \frac{(u(t)-u(x))^m}{m!}(1-\beta(x))\,\mathrm{d}u(x). \end{aligned}$$

From this, it follows that (5.19) becomes

$$G_m(t) = \int_0^t \left(\int_0^{x_1} \frac{(u(x_1)-u(x_2))^m}{m!}\,\mathrm{d}(1-\beta(x_2))\right)\mathrm{d}\nu(x_1).$$

Consider now (5.20). Using $\Gamma_0(t) = \gamma(t) - \beta^2(t)$ yields

$$\Gamma_m = 2^m \int (\gamma-\beta^2)\,\mathrm{d}^m u + \sum_{j=1}^{m} 2^j \int\int F_{m-j}\,\mathrm{d}\gamma_*\,\mathrm{d}^{j-1}u. \tag{B.1}$$

The first term on the right-hand side of (B.1) can be treated in a similar fashion as the treatment of (5.17) and (5.19) above. Rewriting the second term on the right-hand side of (B.1) leads to

$$\begin{aligned} &\sum_{j=1}^{m} 2^j \int_0^t \int_0^{x_1} F_{m-j}(x_2)\,\mathrm{d}\gamma_*(x_2)\,\mathrm{d}^{j-1}u(x_1) \\ &\quad= \sum_{j=1}^{m} 2^j \int_0^t \int_0^{x_1} \frac{(u(t)-u(x_1))^{j-2}}{(j-2)!} F_{m-j}(x_2)\,\mathrm{d}\gamma_*(x_2)\,\mathrm{d}u(x_1) \\ &\quad= \sum_{j=1}^{m} 2^j \int_0^t \int_{x_2}^t \frac{(u(t)-u(x_1))^{j-2}}{(j-2)!} F_{m-j}(x_2)\,\mathrm{d}u(x_1)\,\mathrm{d}\gamma_*(x_2) \\ &\quad= \sum_{j=1}^{m} 2^j \int_0^t \frac{(u(t)-u(x_2))^{j-1}}{(j-1)!} F_{m-j}(x_2)\,\mathrm{d}\gamma_*(x_2) \end{aligned}$$

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

$$= \sum_{j=1}^{m} 2^j \int_0^t \int_0^{x_1} \frac{(u(t)-u(x_1))^{j-1}}{(j-1)!} \frac{(u(x_1)-u(x_2))^{m-j}}{(m-j)!} \, \mathrm{d}F_0(x_2) \, \mathrm{d}\gamma_*(x_1)$$
$$= 2 \int_0^t \left(\int_0^{x_1} \frac{(2u(t)-u(x_1)-u(x_2))^{m-1}}{(m-1)!} \, \mathrm{d}F_0(x_2) \right) \mathrm{d}\gamma_*(x_1),$$

which then allows us to write, for $m \geq 1$,

$$\Gamma_m(t) = \int_0^t \frac{(2u(t)-2u(x))^m}{m!} \, \mathrm{d}\Gamma_0(x) + 2 \int_0^t \left(\int_0^{x_1} \frac{(2u(t)-u(x_1)-u(x_2))^{m-1}}{(m-1)!} \, \mathrm{d}F_0(x_2) \right) \mathrm{d}\gamma_*(x_1)$$

and $\Gamma_0(t) = \gamma(t) - \beta(t)^2$. The solutions to (5.21) and (5.22) can be treated in a similar fashion as the treatment of (5.20) and its solution above.

Finally, note that since all functions $\Gamma_0(x)$, $F_0(x)$, and $\gamma_*(t)$ are bounded by 1 in absolute value, we can write

$$|\Gamma_m(t)| \leq \int_0^t \frac{|(2u(t)-2u(x))^m|}{m!} \, \mathrm{d}x + 2 \int_0^t \left(\int_0^{x_1} \frac{|(2u(t)-u(x_1)-u(x_2))^{m-1}|}{(m-1)!} \, \mathrm{d}x_2 \right) \mathrm{d}x_1.$$

Furthermore, since $0 \leq u(t) \leq 1$ for all t, we have

$$|\Gamma_m(t)| \leq \int_0^t \frac{2^m}{m!} \, \mathrm{d}x + \int_0^t \left(\int_0^{x_1} \frac{2^m}{(m-1)!} \, \mathrm{d}x_2 \right) \mathrm{d}x_1,$$

from which one can conclude that $\Gamma_m(t)$ goes to 0 at a superexponential rate as m grows large. A similar treatment of the function $A_m(t)$ will show that it also goes to 0 at a superexponential rate as m grows large. This then implies that the covariance between the activity states of nodes decays at a superexponential rate as m grows large.

Appendix C.

Proof of Proposition 6.1. Equation (6.4) is a direct consequence of Theorem 10 of [17]. It remains to show that the expressions for $\sigma_z^2(t)$ hold. We first prove the following lemma.

Lemma C.1. *Fix z, then for $m \in \mathbb{N}^+$,*

$$\lim_{n\to\infty} \frac{P_z^{n,2m-1}}{|V_n|} = 2z^{m-1} \quad \textit{and} \quad \lim_{n\to\infty} \frac{P_z^{n,2m}}{|V_n|} = (z+1)z^{m-1}.$$

Proof. For the $z = 1$ case, the statement is easily verified. Assume that $z > 1$. We prove the statement by induction. First, note that, since there are $(z+1)z^{n-1}$ leaf nodes in the graph induced by V_n,

$$P_z^{n,1} = (z+1)|V_{n-1}| + (z+1)z^{n-1},$$

which yields, in combination with $|V_n| = 1 + (z+1)((1-z^n)/(1-z))$,

$$\lim_{n\to\infty} \frac{P_z^{n,1}}{|V_n|} = 2,$$

which will serve as our induction basis.

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

Assume now that $m > 1$. Define $Q_z^{n,m}$ to be the number of directed self-avoiding paths of length m on the graph induced by V_n *ending* on a leaf node. Equivalently, we could let $Q_z^{n,m}$ be the number of directed self-avoiding paths of length m on the graph induced by V_n *originating* on a leaf node. Then, we can write

$$P_z^{n,m} = z(P_z^{n,m-1} - Q_z^{n,m-1}). \tag{C.1}$$

Let us count the number of directed self-avoiding paths of length m on the graph induced by V_n *originating* on a leaf node, that is, $Q_z^{n,m}$, assuming $n \gg m$. For a path originating at a leaf node to stay within the graph induced by V_n, it has to move towards the root, the first $\lceil m/2 \rceil$ steps after which it can go in any self-avoiding direction for the last steps. Hence, since there are $(z+1)z^{n-1}$ leaf nodes,

$$Q_z^{n,m} = (z+1)z^{n-1+\lfloor m/2 \rfloor}. \tag{C.2}$$

The rest of the proof by induction follows from (C.1) and (C.2). □

For the $z = 1$ case, applying Lemma C.1 in combination with (6.2) to (6.3) yields $\sigma_1^2(t) = t\mathrm{e}^{-4t}$; see Theorem 11 of [17]. For $z > 1$, Lemma C.1 yields, in combination with (6.2) and (6.3),

$$\begin{aligned}
\sigma_z^2(t) &= A_0(t) - \beta(t)^2 \sum_{m=0}^{\infty} z^m \sum_{n=m+1}^{\infty} \frac{(2\log[\beta(t)])^{2n}}{(2n)!} \\
&\quad - \frac{(z+1)\beta^2(t)}{2} \sum_{m=1}^{\infty} z^{m-1} \sum_{n=m}^{\infty} \frac{(2\log[\beta(t)])^{2n+1}}{(2n+1)!} \\
&= A_0(t) - \beta(t)^2 \sum_{n=1}^{\infty} \frac{(2\log[\beta(t)])^{2n}}{(2n)!} \sum_{m=0}^{n-1} z^m \\
&\quad - \frac{(z+1)\beta^2(t)}{2} \sum_{n=1}^{\infty} \frac{(2\log[\beta(t)])^{2n+1}}{(2n+1)!} \sum_{m=1}^{n} z^{m-1} \\
&= A_0(t) - \frac{\beta(t)^2}{z-1}\left(\sum_{n=1}^{\infty}(z^n-1)\frac{(2\log[\beta(t)])^{2n}}{(2n)!} - (z^n-1)\frac{(z+1)}{2}\frac{(2\log[\beta(t)])^{2n+1}}{(2n+1)!}\right).
\end{aligned}$$

Finally, using $A_0(t) = \alpha(t)(1-\alpha(t)) = \frac{1}{4}(1-\beta(t)^4) = -\frac{1}{2}\beta(t)^2 \sinh[2\log[\beta(t)]]$, we find that, after some manipulation and gathering of terms,

$$\begin{aligned}
\sigma_z^2(t) &= \frac{\beta(t)^4}{z-1} - \frac{\beta(t)^2}{z-1}\left(\cosh[2\sqrt{z}\log[\beta(t)]] + \frac{(z+1)}{2\sqrt{z}}\sinh[2\sqrt{z}\log[\beta(t)]]\right) \\
&= \frac{\beta(t)^4}{z-1} + \frac{(1-\sqrt{z})^2\beta(t)^{2(1-\sqrt{z})} - (1+\sqrt{z})^2\beta(t)^{2(1+\sqrt{z})}}{4\sqrt{z}(z-1)}.
\end{aligned}$$

□

Acknowledgements

The author wishes to express his sincerest gratitude to Sem Borst and Onno Boxma who provided valuable comments that greatly assisted this research and the writing of this paper.

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press

References

[1] Baccelli, F. and Blaszczyszyn, B. (2009). *Stochastic Geometry and Wireless Networks, Vol. 1, Theory, Foundation and Trends in Networking.* Now Publishers, Breda.

[2] Bermolen, P., Jonckheere, M. and Moyal, P. (2017). The jamming constant of uniform random graphs. *Stoch. Process. Appl.* **127,** 2138–2178.

[3] Bermolen, P., Jonckheere, M., Larroca, F. and Moyal, P. (2014). Estimating the spatial reuse with configuration models. Preprint. Available at https://arxiv.org/abs/1411.0143.

[4] Coladon, T., Vučinić, M. and Tourancheau, B. (2015). Improving trickle fairness: locally-calculated redundancy constants. In *Proceedings of the International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS)*, IEEE, pp. 1–6.

[5] Dehling, H. G., Fleurke, S. R. and Külske, C. (2008). Parking on a random tree. *J. Statist. Phys.* **133,** 151–157.

[6] Dhara, S., van Leeuwaarden, J. S. H. and Mukherjee, D. (2016). Generalized random sequential adsorption on Erdős–Rényi random graphs. *J. Statist. Phys.* **164,** 1217–1232.

[7] Evans, J. W. (1993). Random and cooperative sequential adsorption. *Rev. Mod. Phys.* **65,** 1281–1329.

[8] Fleurke, S. R. and Külske, C. (2010). Multilayer parking with screening on a random tree. *J. Statist. Phys.* **139,** 417–431.

[9] Hui, J. and Kelsey, R. (2016). Multicast protocol for low power and lossy networks (MPL). Preprint 7731, Internet RFC.

[10] Kermajani, H. and Gomez, C. (2014). On the network convergence process in RPL over IEEE 802.15.4 multihop networks: improvement and trade-offs. *Sensors* **14,** 11993–12022.

[11] Kulpa, W. (1997). The Poincaré-Miranda theorem. *Amer. Math. Monthly* **104,** 545–550.

[12] Levis, P., Patel, N., Culler, D. and Shenker, S. (2004). Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *Proceedings of the First Symposium on Networked Systems Design and Implementation*, ACM, New York, pp. 15–28.

[13] Meyfroyt, T. M. M., Stolikj, M. and Lukkien, J. J. (2015). Adaptive broadcast suppression for Trickle-based protocols. In *Proceedings of the 16th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, IEEE, 9 pp.

[14] Meyfroyt, T. M. M., Borst, S. C., Boxma, O. J. and Denteneer, D. (2015). On the scalability and message count of trickle-based broadcasting schemes. *Queueing Systems* **81,** 203–230.

[15] Ni, S.-Y., Tseng, Y.-C., Chen, Y.-S. and Sheu, J.-P. (1999). The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, ACM, New York, pp. 151–162.

[16] Ostilli, M. (2012). Cayley trees and Bethe lattices: a concise analysis for mathematicians and physicists. *Physica A* **391,** 3417–3423.

[17] Penrose, M. D. and Sudbury, A. (2005). Exact and approximate results for deposition and annihilation processes on graphs. *Ann. App. Prob.* **15,** 853–889.

[18] Rényi, A. (1958). On a one-dimensional problem concerning random space filling. *Magyar Tud. Akad. Mat. Kutató Int. Közl.* **3,** 109–127.

[19] Sanders, J., Jonckheere, M. and Kokkelmans, S. (2015). Sub-Poissonian statistics of jamming limits in ultracold Rydberg gases. *Phys. Rev. Lett.* **115,** 043002.

[20] Sudbury, A. (2009). Random sequential adsorption on random trees. *J. Statist. Phys.* **136,** 51–58.

[21] Vallati, C. and Mingozzi, E. (2013). Trickle-F: fair broadcast suppression to improve energy-efficient route formation with the RPL routing protocol. In *Proceedings of Sustainable Internet and ICT for Sustainability (SustainIT)*, IEEE, 9 pp.

[22] Winter, T. *et al*. (2012). RPL: IPv6 routing protocol for low-power and lossy networks. Preprint 6550, Internet RFC.

https://doi.org/10.1017/apr.2018.14 Published online by Cambridge University Press