Analysis of Push-type Epidemic Data Dissemination in Fully Connected Networks

Mine Caglar

Department of Mathematics, Koc University, Istanbul, Turkey

Ali Devin Sezer

Université d'Evry, Laboratoire Analyse et Probabilités and Middle East Technical University, Institute of Applied Mathematics

Abstract

Consider a fully connected network of nodes, some of which have a piece of data to be disseminated to the whole network. We analyze the following push-type epidemic algorithm: in each push round, every node that has the data, i.e., every infected node, randomly chooses c other nodes in the network and transmits, i.e., pushes, the data to them. We write this round dynamically as a random walk whose each step corresponds to a random selection of one of the infected nodes, which gives an efficient way of representing the distribution of the result of each round. We use this to compute the expected number of rounds so that a given percentage of the network is infected and continue a numerical comparison of the push algorithm and the pull algorithm (where the susceptible nodes randomly choose peers) initiated in an earlier work. We then derive the fluid and diffusion limits of the random walk as the network size goes to ∞ and derive a number of properties of the push algorithm: 1) the number of newly infected nodes in a push round, and the number of random selections needed so that a given percent of the network is infected, are both asymptotically normal 2) for large networks, starting with a nonzero proportion of infected nodes, a pull round infects slightly more nodes on average 3) the number of rounds until a given proportion λ of the network is infected converges to a constant for almost all $\lambda \in (0, 1)$. Numerical examples for all of the theoretical results are provided.

Keywords: peer to peer, pull, push, epidemics, epidemic algorithm, diffusion, fluid, approximation, asymptotic, analysis, data dissemination,

Email addresses: mcaglar@ku.edu.tr (Mine Caglar), alidevin.sezer@univ-evry.fr (Ali Devin Sezer)

URL: http://home.ku.edu.tr/~mcaglar (Mine Caglar), https://sites.google.com/site/devinsezer/ (Ali Devin Sezer)

fully connected, network, graph

1. Introduction

Epidemic algorithms mimic spread of infectious diseases to disseminate data in large networks [7, 17, 11, 16, 19, 20, 24]. As is common in the literature, let us call a node of a network *infected* if it holds the piece of data to be disseminated and *susceptible* otherwise. Two of the main types of epidemic algorithms are *push* and *pull*. Both of these progress in discrete stages called *rounds*; in a push round, each infected node randomly selects $c > 0$ nodes uniformly and without repetition among the rest of the nodes and uploads, i.e, pushes, the data to these nodes; in a pull round, each susceptible node randomly selects c nodes and if any of these nodes is infected, the selecting node downloads, i.e., pulls, the data from the infected node to which it has connected. The parameter c is called the *fanout*.

One calls a network fully connected if each of its nodes can directly connect to any other node in the network. If the network is represented as a graph, the network is fully connected if and only if its graph is complete. Analysis over fully connected networks is a natural first step in the study of algorithms on networks. [16, 20, 12, 4] study epidemic algorithms on fully connected networks and [14, 3, 8] study a range of other stochastic algorithms on them; see Section 5 for more on fully connectedness and for comments on other topologies. The aim of the present work is a thorough analysis of the push algorithm over fully connected networks; the following paragraphs explain the elements of this analysis.

A key random variable in epidemic algorithms is the number Y of newly infected nodes after an epidemic round. The paper [18] studies this random variable for fully connected networks and observes that it is binomial for the pull[1] round when conditioned on the number of infected nodes in the network right before the round begins. For the distribution of Y in the push round, [18] assumes $c = 1$ and derives the formula (2.1), by counting all digraphs which correspond to each realization of Y. The direct computation of (2.1) requires high precision and lengthy arithmetic and this restricts its use to small networks ($n \leq 200$). In Section 2 we represent Y in a completely different way for general c using a random walk S with linear and state dependent dynamics. Remember that within a push round each infected node randomly selects c peers and transmits its data to these peers. Each step of the walk S corresponds to one of these random selections; see (2.2),

[1] [18], following [4], reverses the roles of the words "pull" and "push"; what is called "pull" here is called "push" in these works. We always use these words in the sense explained in the first paragraph. One should keep this reversal in mind when comparing the results of the present paper with those in [18, 4].

(2.3) and (2.6) for the exact dynamics. The position S_k of the walk at its k^{th} step (k being the number of infected nodes before the round begins) is our desired representation of Y. Subsection 2.1 explains how S can be used as a model for the whole push algorithm when it is allowed to take an unlimited number of steps. The next subsection computes the first and second moments of S, which gives, in particular, those of Y.

The random walk S is also a discrete time Markov chain and its dynamics therefore can also be expressed as its one step transition matrix. The equation (2.5) writes the distribution of Y as the first row of the k^{th} power of this matrix. This formula is valid for all n, k and c and gives a fast way of computing Y's distribution. We use this result in Section 3 to compute, for $n = 500$, and $c \in \{1, 7\}$, the expected number of push rounds needed so that the proportion of the infected nodes in the network reaches $\lambda \in (0, 1)$. This expectation is simple to compute for the pull algorithm, because Y of that algorithm is binomial. We carry out these computations for the pull algorithm as well to compare them with the results of the push; the results are given in Figures 2 and 3.

Section 4 contains the main results of our analysis: here we compute the diffusion and fluid limits of the random walk S and derive a number of properties of the push epidemic algorithm from these limits. For the asymptotic analysis to make sense we set the initial number of infected nodes to k_n such that $\lim_n k_n/n = (1-\mu) \in (0, 1)$. Theorem 1 shows that as n goes to ∞, the scaled random walk S/n behaves like $t \to \Gamma_t + \frac{1}{\sqrt{n}} X_t$ where Γ is the deterministic process $t \to \mu(1 - e^{-ct})$ and X is a time discounted Wiener integral of a function of Γ (see (4.23)). The t variable here is the continuous scaled time and $t = (1 - \mu) = \lim_n k_n/n$ corresponds to the $k_n \approx (1 - \mu)n^{th}$ step of the random walk and hence to the end of the first push round. This establishes that $\sqrt{n}(Y/n - \Gamma_{1-\mu})$ converges to a zero mean normal random variable whose variance is given by the quadratic variation of X ((4.26) and (4.27)). Subsection 4.1 uses Theorem 1 to compare the pull and the push algorithms for large fully connected networks. In particular, (4.29) says that a pull round always infects slightly more nodes on average, for large networks and starting with a nonzero proportion of infected nodes. The difference disappears with increasing c. When the network is initially half infected, for $c > 15$ a single round of push or pull suffices to infect almost all of the nodes.

The last observation suggests that one study more carefully what happens in a single round. The random walk representation and its limits allow exactly this. In subsection 4.2 we use the asymptotic limit of S derived in Theorem 1 to compute the asymptotics of the number $\tau_\lambda^{(n)}$ of random selections needed so that the proportion of infected nodes in the network reaches $\lambda \in (0, 1)$. Theorems 2 and 3 say that this quantity is also asymptotically normal and provide its mean and variance. We would like to emphasize

that $\tau_\lambda^{(n)}$ is not just a function of the value of S at a particular deterministic point in time but of its whole path. Thus, a stochastic process level analysis of S is inevitable in the study of $\tau_\lambda^{(n)}$.

Subsection 4.3 uses Γ to derive the fluid limit of a sequence of push rounds. Let us denote by $\nu_\lambda^{(n)}$ the (random) number of push *rounds* needed so that the proportion of the network is above $\lambda \in (0,1)$. The final result of our analysis is Theorem 4, which says that as the network size increases to ∞, $\nu_\lambda^{(n)}$ converges to a constant integer, if λ is not one of the deterministic levels derived in subsection 4.3 that the fluid limit of the rounds go through.

Although we have not seen in the prior literature the diffusion analysis of the random walk S, the proof of Theorem 1 is based on results and ideas in [10] and is relegated to the appendix. We have not been able to find in the prior literature analyses and proofs similar to the ones we give in subsections 4.2 and 4.4 and therefore the proofs in these subsections follow the statements of the theorems.

The problems we treat and their solution have connections to a vast literature in communication systems, databases, applied probability, queueing theory, stochastic biological models among others. The following review only touches a small subset of this literature which directly relates to our analysis and of which we happen to be aware of. [16] studies a number of epidemic algorithms in a fully connected network. The one most related to the current paper is an algorithm in which *all* nodes randomly connect to peers rather than only infected or only susceptible and all connections use both pull and push. The paper uses Chernoff's bound to derive bounds on the tail probabilities on the number of rounds this algorithm needs to spread a piece of data to the whole network with high probability. [20] studies the effect of dividing the data to be transferred into pieces on the performance of the epidemic algorithms in a fully connected network and derives asymptotic bounds on the tail probabilities of the number of rounds to disseminate the data to the whole network. As with [16], in [20] the rounds are considered atomic and during each round all nodes randomly connect to peers. The main mathematical tool is again large deviation bounds on independent and identically distributed (iid) sums, similar to Chernoff's bound. [12] studies a continuous time Markov process model similar to actual epidemic models and uses a coupling argument to find bounds on the expected time to total dissemination in terms of the largest eigenvalue of the adjacency matrix of the network graph and applies its results to a number of graph structures, including complete graphs. [24] studies epidemic algorithms as a model for the spread of computer viruses; in this context it makes sense to allow nodes to "recover." Then a natural quantity of interest is the time limit of the probability of each node being infected or susceptible. [24] proves the existence of these limits and derives conditions under which convergence occurs exponentially fast. [1] compares pull, push and and a "pull-push" algorithm

in the context of sensor networks using software that is used in actual sensors and a simulation environment which can run this software. This allows its authors to study in detail several aspects of the performance of these algorithms in practical systems.

To the best of our knowledge the present paper is the first to use diffusion limits in the study of epidemic algorithms in networks. However, in the biologic epidemics literature diffusion limits are a basic tool; the classical reference on this subject is [10, Chapter 11]; a recent review is [6]. In all biological epidemic models that we are aware of, the network graph is implicitly taken to be fully connected by assuming that all members of the population somehow are able to interact with each other, similar to chemicals interacting in a liquid mixture. There are two classes of epidemic models: continuous time and discrete time [9]. The continuous time assumption of the first class leads to a limit process (see [10, Theorem 2.3, page 458]) which is different from the asymptotics of the push algorithm. Among the works which assume a discrete time, the analysis of [21] is closest to this work. The authors of [21] study a discrete time epidemic process which allows recovery. Infection mechanism of the model corresponds to the pull algorithm (susceptibles randomly choose peers). Besides allowing recovery the novelty of [21] is that it allows a random fanout for each individual. [21, Section 3.2] derives a diffusion limit for this model for constant fanout and no recovery.

Another branch of research related to epidemic algorithms is urn models in applied probability. The paper [7] uses this connection in finding asymptotic bounds on the tail distribution of the the number of rounds until most nodes are infected, for the pull, push and a "push-pull" algorithms over graphs defined by the classical preferential attachment model [5]. After our analysis we have noticed that [13] treats an urn model that corresponds to the push algorithm over fully connected networks. In particular, it derives the asymptotic limit of the random variable Y using a completely different set of tools from the ones used in this work (namely, probability generating functions (pgfs) and a result of [22] which characterizes pgfs arising from a Bernoulli sequence). [13] notes that a recursive characterization of Y, similar to the one we give in Section 2 goes all the way back to [23], which casts the problem in terms of a sequence of trials of an event "presuming that the probability of the event on a given trial depends only on the number of the previous successes."

Further comments on our results and future research are in Section 5.

2. Random Walk Representation

We will begin our analysis by considering a single push round on a network with n total nodes and k infected nodes. Let $Y(n, k)$ denote the number of the number of newly infected nodes after the round is over. The paper

[18], which only considers the case $c = 1$, finds the following formula for the distribution of $Y(n, k)$

$$\mathbb{P}(Y(n,k) = i) = \frac{\binom{n-k}{i} i! \sum_{k_1=i}^{k} \binom{k}{k_1} (k-1)^{k-k_1} \mathbf{s}(k_1, i)}{(n-1)^k} \tag{2.1}$$

where $\mathbf{s}$ denotes the Stirling numbers of the second kind. This formula comes from representing the result of a round as a graph and counting those graphs that give $Y(n, k) = i$. The use of (2.1) raises two issues: 1) it is valid only for $c = 1$ and 2) the expressions that appear in it can be computed precisely for only small values of n (i.e., $n \leq 200$) , see [18] for more on these. In this section we derive a new dynamic representation of this round as a random walk with state dependent increments, whose each step corresponds to a random selection made by one of the infected nodes during the round.

Let us first consider the case $c = 1$; the extension to $c > 1$ will be straightforward. The random walk representation of the push round begins with thinking that the nodes do the their random selection of a peer from the rest of the network sequentially and one by one. The Bernoulli random variable X_1 denotes the result of the first selection ($X_1 = 1$ if the selected node is susceptible, 0 otherwise), X_2 the result of the second selection, and so on. Now define

$$S_{l+1} = S_l + X_{l+1} \qquad l = 0, 1, 2, \ldots \tag{2.2}$$

where the conditional distribution of X_{l+1} given $S_l = i \leq n - k$ is Bernoulli with success probability

$$\mathbb{P}(X_{l+1} = 1 |\ S_l = i) = (n - k - i)/(n - 1), \tag{2.3}$$

for $l = 1, \ldots, k-1$ and $S_0 = 0$. S_l is the number of newly infected nodes after the l^{th} random selection; $Y(n, k)$, then, is S_k. The conditional distribution (2.3) is written by noting that the only way for the $l + 1^{st}$ selection to increase the infected count is by choosing a susceptible node that has not been touched by the first l random selections. We also note $P(X_{l+1} = i|S_0, S_1, ..., S_l) = P(X_{l+1} = i|S_l)$ which makes S a Markov process and ensures that (2.3) completely determines the entire distribution of S.

Subsection 2.1 will explain how S can serve as a model of the whole push algorithm when it runs for an unlimited number of steps.

The dynamics (2.3) imply that S is a Markov chain with one step transition matrix

$$P_{i,i+j} = P(X_{l+1} = j|S_l = i). \tag{2.4}$$

The probability distribution of $Y(n, k)$ (i.e., the position of the chain at its k^{th} step) is the first row of P^k, i.e.,

$$\mathbb{P}(Y(n,k) = i) = P^k_{0,i} \qquad i = 0, 1, 2, \ldots, \min(k, n-k). \tag{2.5}$$

P^k can be computed quickly for relatively large values of n, because P is sparse.

For $c > 1$, one only generalizes the conditional distribution of X_{l+1} given $S_l = i \leq n - k$ from (2.3) to

$$\mathbb{P}(X_{l+1} = j |\ S_l = i) = \binom{k-1+i}{c-j}\binom{n-k-i}{j}\Big/\binom{n-1}{c}, \tag{2.6}$$

which is a hypergeometric distribution on $\{j : \max(0, c-(k+i-1)) \leq j \leq \min(c, n-(k+i))\}$. With this generalization, the formulas (2.4) and (2.5) continue to work for $c > 1$.

Note that (2.6), (2.4) and (2.5) imply that Y can take values from $\max(0, c+1-k)$ to $\min(k, n-k)$ with positive probability.

2.1. S as a model of the whole push algorithm

Now suppose that we would like to model a second round which follows the first round also with a random walk. Let us temporarily call this random walk $S^{(2)}$. (2.2) and (2.3) continue to describe the dynamics of $S^{(2)}$ if we only replace the k in (2.3) with $S_k + k$, which is the number of infected nodes in the network at the end of the first round. This and the Markov property of S imply that we don't actually need a second process $S^{(2)}$ to describe the second round; it suffices to simply run the original process S indefinitely; its first $I_0 = k$ steps will model the random selections of the infected nodes in the first round, its next $I_1 \doteq k + S_k$ steps will model the random selections in the second round, the next $I_2 \doteq k + S_{I_0+I_1}$ steps in the third round, the next $I_3 \doteq k + S_{I_0+I_1+I_2}$ steps in the fourth round and so on. The sequence $(I_0, I_1, I_2, ..)$ itself represents the number of infected nodes after the 0^{th} round, the 1^{st} round, the 2^{nd} round and so on. In this way we see that the single random walk S, when ran indefinitely, is another model for the entire push algorithm. The key difference from the traditional model of a sequence of push rounds is that S takes the random selections that occur in the rounds as the atomic operation of the push algorithm and not the rounds; the rounds are then expressed as recursive random increments of this walk as above. We will use this observation repeatedly in subsections 4.2 and 4.4 when we want to use results about S to get results on a sequence of push rounds and hence on the entire push algorithm. In the rest of the paper we will always assume that S_l is defined for all l, following the dynamics (2.2) and (2.3) or (2.6) (for $c \geq 1$).

2.2. Expectation and second moment of S and Y

By taking the expectation of both sides of (2.2) and of its square and using (2.3) one can find linear recursions for $\mathbb{E}[S_l]$ and $\mathbb{E}[S_l^2]$; setting $l = k$ gives $\mathbb{E}[Y]$ and $\mathbb{E}[Y^2]$. The results of this subsection will also be useful in

the fluid and diffusion limit analysis of Section 4. Let us start with fanout $c = 1$. Take first conditional expectation given S_l of both sides of (2.2)

$$\mathbb{E}[S_{l+1}|S_l] = \mathbb{E}[S_l|S_l] + \mathbb{E}[X_{l+1}|S_l].$$

The conditional distribution (2.3) of X_{l+1} given S_l implies that $\mathbb{E}[X_{l+1}|S_l] = (n - k - S_l)/(n - 1)$. Substituting this in the last display and taking now the ordinary expectation of both sides give

$$\gamma_{l+1} = a_1\gamma_l + a_0 \tag{2.7}$$

where $\gamma_l \doteq \mathbb{E}[S_l]$, and

$$a_1 \doteq (n-2)/(n-1), \quad a_0 \doteq (n-k)/(n-1). \tag{2.8}$$

By definition we set $\gamma_0 \doteq 0$. (2.7) is a linear recursion whose solution is

$$\gamma_l = (n-k)(1-a_1^l). \tag{2.9}$$

Y, the number of infected nodes at the end of the push round, equals S_k and therefore

$$\mathbb{E}[Y] = \mathbb{E}[S_k] = (n-k)(1-a_1^k). \tag{2.10}$$

The second moment of S_l is computed similarly. Let $\alpha_l \doteq \mathbb{E}[S_l^2]$;

$$\alpha_l = \mathbb{E}[S_l^2] = \mathbb{E}\left[(S_{l-1} + X_l)^2\right] = \mathbb{E}[S_{l-1}^2] + 2\mathbb{E}[S_{l-1}X_l] + \mathbb{E}[X_l^2].$$

The middle term can be written in terms of α_{l-1} and γ_{l-1} as follows:

$$\begin{aligned}\mathbb{E}[S_{l-1}X_l] &= \mathbb{E}[\mathbb{E}[S_{l-1}X_l|S_{l-1}]] = \mathbb{E}[S_{l-1}\mathbb{E}[X_l|S_{l-1}]] \\ &= \mathbb{E}\left[S_{l-1}\frac{n-k-S_{l-1}}{n-1}\right] = \frac{n-k}{n-1}\gamma_{l-1} - \frac{\alpha_{l-1}}{n-1},\end{aligned}$$

where we have again used the conditional distribution (2.3). Then, we have the following recursion for α_l:

$$\begin{aligned}\alpha_l &= \alpha_{l-1} + 2\left(\frac{n-k}{n-1}\gamma_{l-1} - \frac{\alpha_{l-1}}{n-1}\right) + \frac{n-k-\gamma_{l-1}}{n-1} \\ &= b_2\alpha_{l-1} + b_1\gamma_{l-1} + b_0\end{aligned} \tag{2.11}$$

where

$$b_2 \doteq \frac{n-3}{n-1}, b_1 \doteq \frac{2(n-k)-1}{n-1}, b_0 \doteq \frac{n-k}{n-1}. \tag{2.12}$$

This and (2.9) imply

$$\alpha_l = \left(1 - b_2^l\right) b_0 + b_1 \sum_{i=0}^{l-1} \gamma_i b_2^{l-1-i}. \tag{2.13}$$

The second moment of Y is

$$\mathbb{E}[Y^2] = \mathbb{E}[S_k^2] = \alpha_k. \tag{2.14}$$

(2.9) implies $\lim_l \gamma_l = (n-k)$. Hence, the last sum in (2.13) converges to $(n-k)\sum_{i=0}^{\infty} b_2^i = (n-k)(n-1)/2$ and this implies $\alpha_l \to (n-k)^2$. Therefore, for n and k fixed, $\mathrm{var}(S_l) \to 0$ as $l \to \infty$ and $S_l \to (n-k)$, i.e., if the random selections of the nodes continue indefinitely all nodes will eventually be infected almost surely. The graph of the variance $\mathrm{var}(S_l)$ is shown in Figure 1.

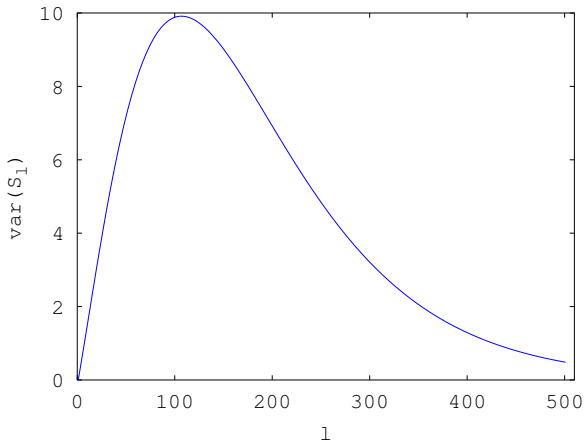


Figure 1: Graph of $\mathrm{var}(S_l)$ for $n = 100, k = 20$

The case $c > 1$ works exactly the same except that one uses the conditional distribution (2.6) rather than (2.3) in computing $\mathbb{E}[X_{l+1}^q | S_l]$, $q = 1, 2$; all of the equations (2.7), (2.9), (2.11) and (2.13) remain exactly as before except that one generalizes the definitions of the coefficients $\{a_i\}$ (2.8) and $\{b_i\}$ (2.12) to

$$\begin{aligned} &a_0 \doteq c(n-k)/(n-1), \quad a_1 \doteq (n-1-c)/(n-1), \\ &b_0 \doteq \frac{c((n-(c+1))+(n-k)(c-1))(n-k)}{(n-2)(n-1)}, \\ &b_1 \doteq \frac{c(n-(c+1))(2(n-k)-1)}{(n-2)(n-1)}, \quad b_2 \doteq 1 - \frac{2c}{n-1} + \frac{c(c-1)}{(n-1)(n-2)}. \end{aligned} \tag{2.15}$$

Note that these reduce to (2.8) and (2.12) for $c = 1$. We summarize the formulas derived about the distribution of $Y(n,k)$ in Table 1.

3. A numerical comparison of push and pull

Let us now use the results so far to numerically compare the push and the pull algorithms. The observations made in this section will also motivate the

	Distribution	Expectation	Second moment
$c = 1$	(2.4), (2.5)	(2.10)	(2.14)
change for $c \geq 1$	(2.3)→ (2.6)	(2.8) → (2.15)	(2.12) → (2.15)

Table 1: Equation numbers for the distribution and moments of $Y(n, k)$

theoretical results of the next chapter. As in subsection 2.1, let I_m denote the number of infected nodes after the m^{th} round. We have shown in that subsection how to write I_m in terms of S. The markov property of S implies that one can also write the same sequence as

$$I_m = I_{m-1} + Y_m, \tag{3.16}$$

where Y_m, conditioned on $I_{m-1} = j$, is independent of $(I_0, I_1, ..., I_{m-2})$ and has the same distribution as $Y(n, j)$. One of the key questions about the process $(I_0, I_1, I_2, ...)$, and hence about the push algorithm is this: how many rounds is required so that a given proportion of the network is infected? This is the quantity that all of [18, 16, 20, 7] analyze. The answer to this question is expressed as the following stopping time of I:

$$\nu_\lambda^{(n)} \doteq \inf\{m : I_m/n \geq \lambda\}, \lambda \in (0, 1). \tag{3.17}$$

We compute the weak limit of $\nu_\lambda^{(n)}$ as the network size goes to ∞ in Subsection 4.4. In the numerical study of the present section, we compute

$$N(\lambda, k) \doteq \mathbb{E}_k \left[\nu_\lambda^{(n)}\right],$$

for $n = 500$ and $c \in \{1, 7\}$; the k in the subscript of the expectation operator denotes that we condition on $I_0 = k$, i.e., the infected number of nodes in the network before the first round is k. $N(1, k)$ is called the mean total dissemination time. Because n is fixed (i.e., we are not taking any limits) throughout this section, there is no harm in assuming $\lambda \in \{j/n, 0 < j < n\}$, which is the set of all possible proportions of infected nodes for a finite network with n nodes. The dynamics (3.16) implies that for $\lambda > k/n$

$$N(\lambda, k) = 1 + N(\lambda, k)\mathbb{P}(Y(n, k) = 0) + \sum_{i=1}^{n\lambda-k-1} N(\lambda, k + i)\mathbb{P}(Y(n, k) = i). \tag{3.18}$$

The 1 on the right means that going from k infected nodes to $j = n\lambda$ infected nodes will take at least one round, the second term handles the case where no infections occur in the first round and the sum handles the cases where the first round infects at least 1 node but less than the $j - k$ needed to get a total of j infected nodes. Furthermore note that

$$N(\lambda, k) = 0 \tag{3.19}$$

when $\lambda \leq k/n$ because if there are k infected nodes initially then obviously $j = \lambda n \leq k$ of them are also infected already and we need no rounds. (3.19), (3.18) and (2.5) can be used to compute $N(j/n, k)$ for all j and k. Note that (3.18) and (3.19) are the same for the pull algorithm as well, the only change is in the distribution of $Y(n, k)$, which is binomial for the pull. Figure 2 shows $\lambda \to N(\lambda, k)$ for the push and the pull algorithms for $k = 50$, $n = 500$ and $c \in \{1, 7\}$. These values of k and n correspond to $k/n = 50/500 = 0.1$ initial proportion of infected nodes.

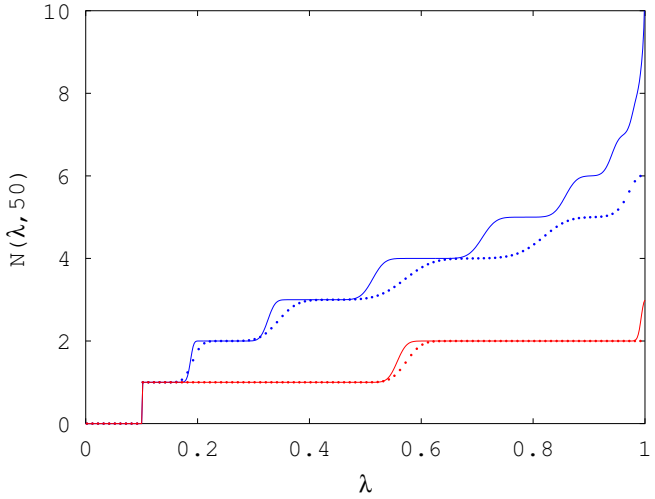


Figure 2: Graphs of $\lambda \to N(\lambda, k)$ for the push and the pull algorithms; the dotted curves are pull, the solid curves are push; $n = 500$, $k = 50$, $c \in \{1, 7\}$, the top couple is $c = 1$ and the bottom couple is $c = 7$

Figure 2 suggests that, for both push and pull, $\lambda \to N(\lambda, k)$ alternates between phases of constancy and rapid growth and that this behavior gets more marked with increasing c. One of the goals of the next section is to explain this behavior. For now, let us briefly comment that both of these algorithms have deterministic fluid limits (derived for the push algorithm in Theorem 1 and in subsection 4.3) and as the network size grows each round infects an almost deterministic proportion of the nodes. This can be used to prove that, in the limit, $\nu_\lambda^{(n)}$ becomes almost deterministic and as a function of λ it becomes a step function, increasing only at the levels of infection attained by the rounds of the fluid limit; the exact result on this is Theorem 4, proved in subsection 4.4. The more pronounced nature of the growth phases for greater values of c will again be explained by Theorem 1 which implies that the deviations from the fluid limit has a lower variance as c grows.

The N of the pull algorithm in Figure 2 lies on or below that of push. This suggests, for a large network with a nonzero initial proportion of infected nodes, on average, the pull reaches a given level in less or equal

number of rounds than the push. However, note that the difference between the algorithms in Figure 2 is not that great and grows only as the network is near being fully infected. The theoretical result which explains these observations is Proposition 1 in subsection 4.1, which compares the expected number of infected nodes in a pull and a push round.

[18, Section 6] uses the binomial distribution of Y under the pull algorithm to compute the dependence on the fanout c of the mean total dissemination time $N(1,1)$ starting with a single infected node for a network of 500 nodes. With (2.5) we are able to do the same also for the push algorithm. $N(1,1)$ as a function of c for $n = 500$ is given in Figure 3 for both algorithms. Although $N(1,1)$ is larger in the push algorithm for smaller fanout values, this value for both algorithm seem to converge for $c \geq 8$. Corollary 1 below partially explains this phenomenon.

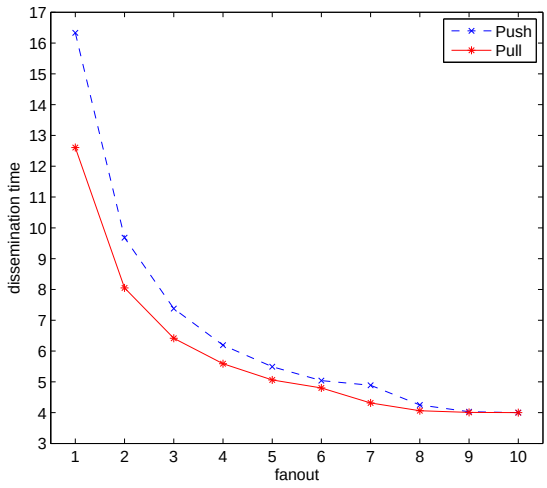


Figure 3: Expected total dissemination time in rounds versus fanout for $n = 500$, and $k = 1$

4. Fluid and Diffusion Limits

A great deal can be understood about the push algorithm by computing the fluid and diffusion limits of S as the network size n goes to ∞. To get meaningful limits, we allow the initial number of infected nodes k to depend on n in such a way that

$$0 < \mu = \lim_{n\to\infty} \frac{n - k_n}{n} < 1 \tag{4.20}$$

holds, where μ is the initial proportion of susceptible nodes in the network. The limits we talk of here are known as *weak limits* in probability theory and is almost always shown with the sign $\Rightarrow$, which we will also do below.

The quintessential weak convergence result is the central limit theorem and "diffusion approximations" are central limit theorems for the entire sample paths of processes. Two of the basic references on weak convergence are [2, 10].

To obtain the fluid and diffusion limits, we scale and center S and time as follows:

$$X_t^n \doteq \sqrt{n}\left(\frac{S_{\lfloor nt \rfloor}}{n} - \Gamma_t^n\right) \quad \Gamma_t^n \doteq \frac{\gamma_{\lfloor nt \rfloor}}{n}. \tag{4.21}$$

First let us note that the time variable t of the scaled processes Γ^n and X^n correspond to the $\lfloor nt \rfloor^{th}$ step of S; Γ^n is the expected proportion of newly infected nodes at time t and X^n is $\sqrt{n}$ times the deviation of the actual proportion from the expected proportion, again at time t. This is the standard scaling in all diffusion analyses of Markovian random walks with finite variance increments. Just as in the central limit theorem, the scaling by $\sqrt{n}$ puts the difference between the actual and the expected proportions at a scale that ensures weak convergence to a nontrivial (i.e., neither 0 nor ∞) limit.

The process (X^n, Γ^n) takes values in $D_{\mathbb{R}^2}[0, \infty)$, the vector space of right continuous functions with left limits from $[0, \infty)$ to $\mathbb{R}^2$. Define

$$\sigma(t) \doteq \sqrt{(\mu - \Gamma_t)(1 - (\mu - \Gamma_t))}. \tag{4.22}$$

The next theorem gives the fluid and diffusion limits of S.

Theorem 1. *Let (X^n, Γ^n) be defined as in (4.21) and (X, Γ) be given by*

$$\begin{aligned} X_t &\doteq X_0 e^{-ct} + e^{-ct} \int_0^t e^{cs} \sqrt{c}\sigma(s) dW_s, \\ \Gamma_t &\doteq (\Gamma_0 - \mu)e^{-ct} + \mu, \end{aligned} \tag{4.23}$$

where $c \in \mathbb{Z}_+$ and σ is as given in (4.22). Then

$$(X^n, \Gamma^n) \Rightarrow (X, \Gamma).$$

The proof, given in the appendix, is based on representing Markov processes and their weak convergence in terms of the semigroups that the processes define and the generators of these semigroups [10].

Theorem 1 implies that the random walk $\{S_l\}$ behaves more like $\left\{n\left(\Gamma_{l/n} + \frac{1}{\sqrt{n}} X_{l/n}\right)\right\}$ as n, the network size, increases. The left part of Figure 4 shows $\{n\Gamma_{l/n}\}$ and a sample path of $\{S_l\}$ and its right part shows their difference. The random path of S_l in this figure has been simulated using (2.3). Theorem 1 implies that the pathwise distribution of this difference gets closer to that of $\{\sqrt{n}X_{l/n}\}$ as n increases.

Remember that $Y(n, k_n)$ can be represented as the value S_{k_n} of S at the step k_n, which corresponds to time $1 - \mu$ in the scaled continuous time.

Theorem 1 then implies in particular that $n\Gamma_{1-\mu} + \sqrt{n}X_{1-\mu}$ is the normal approximation of $Y(n, k_n)$. Let us write this as

$$Y(n, k_n) \approx n\Gamma_{1-\mu} + \sqrt{n}X_{1-\mu}. \tag{4.24}$$

The random variable on the right is normally distributed with mean $n\Gamma_{1-\mu}$. To compute its variance we only need the second moment of $X_{1-\mu}$, which we derive now. It will be simpler to write everything in terms of

$$\bar{X}_t \doteq \int_0^t e^{cs}\sqrt{c}\sigma(s)dW_s. \tag{4.25}$$

The second moment of X_t is

$$\mathrm{var}(X_t) \doteq e^{-2ct}\mathbb{E}[\bar{X}_t^2]. \tag{4.26}$$

$\bar{X}_t$ is a stochastic integral with respect to a Brownian motion, and therefore $t \to \bar{X}_t$ is a continuous martingale whose quadratic variation equals [15, page 139]:

$$\langle \bar{X} \rangle_t = \int_0^t e^{2cs}c\sigma(s)^2 sds = \mu\left(e^{ct} - 1 - \mu ct\right), \tag{4.27}$$

which is a deterministic process. This implies $\mathbb{E}[\bar{X}_t^2] = \langle \bar{X} \rangle_t$ (see again [15, page 137]). This and setting $t = (1-\mu)$ in (4.26) gives

$$n\mu e^{-2c(1-\mu)}\left(e^{c(1-\mu)} - 1 - \mu(1-\mu)c\right) \tag{4.28}$$

as our approximation of the variance of $Y(n, k_n)$.

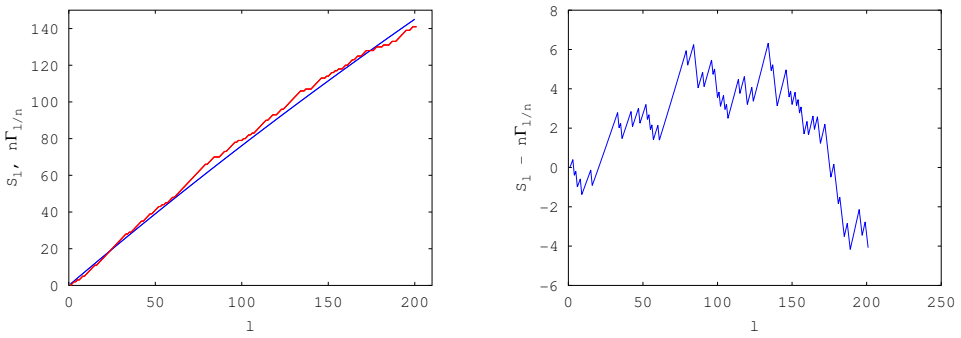


Figure 4: A sample path of S_l, $n\Gamma_{l/n}$ and their difference; $n = 500$, $k_n = 200$ and $c = 1$

4.1. Comparison of push and pull for large networks

Theorem 1 allows a simple comparison of the push and the pull algorithms when n is large. In this subsection it will be easier to use a separate symbol to denote the number of newly infected nodes in a pull round; let us use $\tilde{Y}(n, k)$ for this purpose. It is well known (see [18]) and simple to see that $\tilde{Y}(n, k)$ is Binomial$(1 - p_n, n - k)$ with failure probability $p_n = \binom{n-k-1}{c}/\binom{n-1}{c}$.

Proposition 1. *Let μ and k_n be as in* (4.20). *Then* $\mathbb{E}[\tilde{Y}(n, k_n)] > \mathbb{E}[Y(n, k_n)]$ *for n large.*

Proof. Theorem 1 says that the average proportion of newly infected nodes after a push round converges to $\Gamma_{1-\mu} = \mu(1 - e^{-c(1-\mu)})$. The expected proportion of newly infected nodes in a pull round will be $\mathbb{E}[\tilde{Y}(n, k_n)/n] = (1-p_n)(n-k_n)/n$, the mean of Binomial$(1-p_n, n-k_n)$ divided by n. Since c is fixed, $p_n \to \mu^c$. This and $(n - k_n)/n \to \mu$ imply $\lim_n \mathbb{E}[\tilde{Y}(n, k_n)/n] = \mu(1 - \mu^c)$. The inequality

$$\mu(1 - \mu^c) > \mu \left(1 - e^{-c(1-\mu)}\right) \tag{4.29}$$

for all μ and c implies the statement of the proposition. □

Corollary 1. *For all μ, if c is taken large enough, a single round of pull or push is enough to infect the whole network.*

Proof. Both sides of (4.29) converge to μ, the initial proportion of susceptible nodes, as $c \to \infty$. □

Several comments on these results and possible research directions that they suggest are as follows.

While the inequality (4.29) is obviously true, the difference between the two sides is at most 0.06 for $c = 1$ and decreases as c increases (simple calculus shows the truth of these statements). Thus, the performance of these rounds are on average similar, which explains the near performance of the pull and the push in the numerical example given in Figure 2.

There is an important caveat to Proposition 1, which we would like explain with an example. As with all values of μ, for μ close to 1, i.e., when initially most of the nodes are susceptible, a pull round infects on average more nodes than push, as indeed claimed by Proposition 1. *But* a push round takes merely $(1-\mu)n$ random selections whereas the pull takes μn; for large μ, a push round is a very small operation whereas a pull round involves almost the whole network. Thus, for a fairer comparison we think that it would be a good idea to take into account the sizes of these operations. Such a comparison can be undertaken in a future work.

An interesting comparison is when $\mu = 0.5$ when the initial number of the infected nodes equal the number of the susceptible ones. In this case push and pull will involve the same number of random selections. For $c = 1$, (4.29) implies that the pull round infects on average fifty percent of the susceptible nodes whereas the push approach infects around forty percent. For increasing values of c the difference quickly disappears and for $c \geq 15$ a single round of either algorithm is enough to infect the whole network.

The foregoing discussion suggests the following heuristic: in a network with few infected nodes, initially set c to a relatively high value (say between

10 and 15, if possible) use push until half the network is infected and then switch to pull and gradually decrease c. For smaller values of c, it will be more advantageous to switch to pull earlier. Obviously, to turn these ideas into a full fledged algorithm requires a good deal of more work which includes a specification of how the nodes detect the infection level in the network to do the switch. The design of such an algorithm and its analysis can also be the subject of future work.

4.2. First time to hit $n\Gamma_{1-\mu}$

We have seen in the previous subsection that for $\mu = 1/2$ and $c \geq 15$ one expects a single round to be enough to infect the whole network. In such cases, the number of rounds before the proportion of infected nodes hits a certain level becomes trivial (i.e. just 1), and "the number of random selections" before the same event becomes more useful and interesting. (4.24) implies that, under (4.20), the ratio of the number of newly infected nodes to the number of nodes in the whole network at the end of the first round has expectation approximately $\Gamma_{1-\mu}$. The number of random selections needed to hit this level corresponds to the following stopping time of S: $\tau_n \doteq \inf\{l : S_l \geq n\Gamma_{1-\mu}\}$. The goal of this subsection is to derive approximations to the distribution of τ_n using the diffusion approximation of Theorem 1. The results we obtain will also be useful in subsection 4.4 in finding the limits of $\nu_\lambda^{(n)}$, the number of *rounds* needed before the infection level of the network is λ. We would like to point out that τ_n cannot be studied if one represents the result of a push round as a single random variable; the ensuing analysis requires the use of the random walk representation.

Choose t_n so that it solves

$$\Gamma_{1-\mu} - \Gamma_{t_n} = C\sqrt{\frac{\mathrm{var}(X_{1-\mu})}{n}}, \tag{4.30}$$

where $C > 0$ is a large constant. (4.26) and (4.27) imply

$$t_n \doteq -\frac{1}{c}\log\left(\frac{1}{\mu}\left(\mu - \Gamma_{1-\mu} + \frac{C}{\sqrt{n}}\sqrt{\mathrm{var}(X_{1-\mu})}\right)\right). \tag{4.31}$$

Taylor expanding log in the last display around $1 - \Gamma_{1-\mu}/\mu$ gives

$$t_n = 1 - \mu - \frac{C\sqrt{\mathrm{var}(X_{1-\mu})}}{c(\mu - \Gamma_{1-\mu})\sqrt{n}} + O(1/n). \tag{4.32}$$

Proposition 2.

$$\mathbb{P}(\tau_n \leq nt_n) \leq \sqrt{\frac{2}{\pi}}\int_C e^{\frac{-x^2}{2}}\,dx, \tag{4.33}$$

for n large enough.

We refer the reader to the appendix for the proof. Proposition 2 implies that, once we choose C large enough, with very high probability $\tau_n > nt_n$ and, by (4.32), nt_n is only $O(\sqrt{n})$ steps away from $n(1-\mu) = k_n$. We will use this in the proof of the next theorem to focus our attention on a small neighborhood around k_n.

Theorem 2. $(\tau_n - k_n)/\sqrt{n} \Rightarrow N(0, v)$ *where* $v \doteq var(X_{1-\mu})/(c(\mu - \Gamma_{1-\mu}))^2$.

Proof. Fix a finite interval (a, b); our goal is to show

$$\lim_n \mathbb{P}\left(\frac{\tau_n - k_n}{\sqrt{n}} \in (a, b)\right) = \frac{1}{\sqrt{2\pi v}} \int_a^b e^{-x^2/2v} dx. \tag{4.34}$$

Choose C in (4.30) so that

$$C > \frac{c(\mu - \Gamma_{1-\mu})}{\sqrt{\text{var}(X_{1-\mu})}} \max(|a|, |b|).$$

and define $\mathbf{t}_n \doteq k_n/n - t_n$. Partition the event $E \doteq \{(\tau_n - n(1-\mu))/\sqrt{n} \in (a, b)\}$ as $(E \cap \{\tau_n < nt_n\}) \cup (E \cap \{\tau_n \geq nt_n\})$. Proposition 2 implies that, by increasing C, if necessary, the probability of the first of these sets can be made arbitrarily small. Furthermore, on the set E the greatest value that τ_n can take is $k_n + b\sqrt{n}$; by the choice of C this is bounded above by $k_n + n\mathbf{t}_n$. These imply that we can replace $\tau_n = \inf\{l : S_l \geq n\Gamma_{1-\mu}\}$ in (4.34) with

$$\tau_n' \doteq \inf\{l \in I_n : S_l \geq n\Gamma_{1-\mu}\},$$

where $I_n \doteq k_n + n\mathbf{t}_n(-1, 1)$. Thus in the rest of this argument we will prove

$$\lim_n \mathbb{P}\left(\frac{\tau_n' - k_n}{\sqrt{n}} \in (a, b)\right) = \frac{1}{\sqrt{2\pi v}} \int_a^b e^{-x^2/2v} dx. \tag{4.35}$$

For this, it is enough to study the asymptotics of the dynamics of S in the interval I_n. To do so, define the scaled process

$$\hat{X}^n(\hat{t}) \doteq \frac{1}{\sqrt{n}} \left(S_{\lfloor k_n + \hat{t}\sqrt{n} \rfloor} - n\Gamma_{1-\mu}\right).$$

The scaled time $\hat{\tau}_n \doteq \frac{\tau_n' - k_n}{\sqrt{n}}$ of (4.35) is the first time the process $\hat{X}^n$ hits 0. Thus to find its limit distribution it is enough to compute the weak limit of $\hat{X}^n$, which we will now do.

The time interval $J_n \doteq \sqrt{n}\mathbf{t}_n[-1, 1]$ for the process $\hat{X}^n$ corresponds exactly the time interval I_n for S and therefore, we will be studying $\hat{X}^n$ on J_n. Note that $t = 0$ is the middle of J_n and corresponds to time k_n of S, which is the last step of the first push round. The end of J_n is the time point $\theta_n \doteq \sqrt{n}\mathbf{t}_n$. $k_n/n \to (1-\mu)$ and (4.32) imply

$$\theta_n \to \theta \doteq \frac{C\sqrt{\text{var}(X_{1-\mu})}}{c(\mu - \Gamma_{1-\mu})}.$$

J_n is symmetric around 0, therefore, its starting point converges to $-\theta$. Then the limit process $\hat{X}$ of $\hat{X}^n$ will be running on the interval $[-\theta, \theta]$. Note that the initial point $\hat{X}^n(-\theta_n)$ is $\frac{1}{\sqrt{n}}(S_{t_n} - n\Gamma_{1-\mu})$. Theorem 1, $t_n/n \to (1-\mu)$, and (4.30) imply that this random variable converges weakly to a normal random variable with mean $-C\sqrt{\text{var}(X_{1-\mu})} < 0$ and variance $\text{var}(X_{1-\mu})$. Hence, this is the distribution of the limit $\hat{X}(-\theta)$.

To compute the dynamics of $\hat{X}$ one proceeds parallel to the proof of Theorem 1. Fix $(\hat{x}, \hat{t}) \in \mathbb{R} \times [-\theta, \theta]$ and define

$$\hat{T}_n \doteq \mathbb{E}_{(\hat{x},\hat{t})}\left[f\left(\hat{X}^{(n)}_{\hat{t}+1/\sqrt{n}}\right)\right], \hat{A}_n \doteq \sqrt{n}(\hat{T}_n - I),$$

where f is a smooth function on $\mathbb{R}$ with compact support. The subscript $(\hat{x}, \hat{t})$ of the expectation means that we are conditioning of $\hat{X}^{(n)}_t = \hat{x}$. It remains to compute $\hat{A}_n$. This computation is parallel to the arguments given in the proof of Theorem 1 with the most important difference now the time is scaled by $1/\sqrt{n}$ rather than $1/n$. Thus, we omit the details and directly write down the limit:

$$\lim_{n\to\infty} A_n f = f'(\hat{x})c(\mu - \Gamma_{1-\mu}).$$

The right side of the last display is the generator of the process

$$\hat{X}(\hat{t}) = \hat{X}(-\theta) + c(\mu - \Gamma_{1-\mu})(\hat{t} + \theta). \tag{4.36}$$

whose randomness is completely determined by its initial position. The exactly the same line of arguments as in the proof of Theorem 1 now imply $\hat{X}^{(n)} \Rightarrow \hat{X}$. We are interested in the limit of $\mathbb{P}(\hat{\tau}_n \in (a, b))$, the probability that $\hat{X}^{(n)}$ hits 0 between time points a and b. The weak limit we have just established implies that this limit equals $\mathbb{P}(\hat{\tau} \in (a, b))$ where $\hat{\tau}$ is the first time when the limit process $\hat{X}$ hits 0. (4.36) implies that it will take $\hat{X}$ $\frac{-\hat{X}_{-\theta}}{c(\mu-\Gamma_{1-\mu})}$ unit of time to hit 0. We subtract from this θ to convert it to the time unit of the limit time interval $[-\theta, \theta]$:

$$\tau' = \frac{-\hat{X}_{-\theta}}{c(\mu - \Gamma_{1-\mu})} - \theta.$$

But this is a random variable with mean 0 and variance v. Hence we have (4.35).

□

Here is a numerical example for Theorem 2. Suppose we have a network with $n = 5000$ nodes of which $k = 200$ are initially infected, i.e., $\mu = 0.96$. Suppose that fanout is $c = 5$. For this network, the v that appears in Theorem 2 is $v \approx 0.0012$. We know from Theorem 1 that on average at the end of the first push round the total number of infected nodes will be

$200 + n\Gamma_{1-\mu} \approx 970$. Now Theorem 2 says that the first push round will attain this infection level with probability approximately equal to 1/2: and if it does, this will almost certainly happen in the last $4\sqrt{5000v} \approx 10$ steps of the push round (which lasts a total of $k = 200$ steps); this network hitting 970 infected nodes before the last 10 random selections is as likely as a normal random variable being 4 standard deviations below its mean. The same theorem implies that with probability 1/2 this level will be attained approximately in the first 10 steps of the next round.

For $0 < \lambda < \mu$, define $\tau_\lambda^{(n)} \doteq \inf\{l : S_l \geq n\lambda\}$. $\tau_\lambda^{(n)}$ is the number of random selections before the proportion of newly infected nodes has reached λ; in terms of the random walk S, it is the first time S reaches the level $n\lambda$.

Define

$$\bar{\tau}_\lambda(\mu) \doteq -\frac{1}{c}\log\left(1 - \frac{\lambda}{\mu}\right) \tag{4.37}$$

$\bar{\tau}$, as a function of λ, is the inverse function of the fluid limit $\Gamma_t(\mu) = \mu(1-e^{-ct})$ with respect to t. Dynamically, $\bar{\tau}_\lambda$ is the first time Γ hits the level λ. Because Γ is deterministic, so is this hitting time. A quick examination of the proof of Theorem 2 reveals that if one replaces $1-\mu$ with $\bar{\tau}_\lambda$ and $\Gamma_{1-\mu}$ with λ, the proof continues to work exactly as is. This gives us the following generalization of Theorem 2:

Theorem 3. *Let $\tau_\lambda^{(n)}$ and $\bar{\tau}_\lambda$ be as above. $(\tau_\lambda^{(n)} - n\bar{\tau}_\lambda)/\sqrt{n} \Rightarrow N(0, v)$ with $v = var(X_{\bar{\tau}_\lambda})/(c(\mu-\lambda))^2$.*

4.3. Fluid limit of the whole push algorithm

As suggested in the beginning of this section, instead of the number of newly infected nodes, one can keep track of their proportion in the network. This amounts to dividing S by n. Theorem 1 gives us the following approximation for the proportion process: $S_l/n \approx \Gamma_{l/n} + \frac{1}{\sqrt{n}}X_{l/n}$ Thus as n goes to infinity, the proportion process S/n converges to its fluid limit Γ. In subsection 2.1 we have argued that the random walk S, when ran indefinitely, is a model for the whole push algorithm. This implies that $t \to \Gamma_t$ for $t \in (0, \infty)$ is a representation of the fluid limit of the whole push algorithm. To break it into rounds, we proceed as follows. Define $\varphi_0 \doteq (1-\mu)$, this is the initial proportion of infected nodes in the fluid limit. The first round in the prelimit lasts k_n steps; in the scaled limit time, this corresponds to the time point $\lim_n k_n/n = 1-\mu$, thus the first round of the fluid limit ends at time $1-\mu$ and the total proportion of infected nodes after the first round is $\varphi_1 \doteq \varphi_0 + \Gamma_{\varphi_0}$. The next fluid round will last a time interval of length φ_1 and will add an additional $\Gamma_{\varphi_0+\varphi_1} - \Gamma_{\varphi_0}$ proportion of infected nodes, bringing the total proportion of infected nodes at the end of the second round to $\varphi_2 \doteq \varphi_0 + \Gamma_{\varphi_1+\varphi_0}$ In general, the proportion of infected nodes at the end of the i^{th} round of the fluid limit will be

$$\varphi_i = \varphi_0 + \Gamma_{c_{i-1}}, \tag{4.38}$$

with $c_{i-1} \doteq \sum_{j=0}^{i-1} \varphi_j$. The construction of $\{\varphi_0, \varphi_1, \varphi_2, ...\}$ is the fluid limit version of that of the sequence $(I_0, I_1, I_2, ...)$ given in subsection 2.1; and indeed, Theorem 1 implies $I_i/n \Rightarrow \varphi_i$. The sequence $\{\varphi_i\}$ is increasing and deterministic; in the fluid limit, each round infects a deterministic proportion of the network. In the next subsection we will use the sequence $\{\varphi_i\}$ to compute the weak limit of $\nu_\lambda^{(n)}$, the number of push rounds in a network with n nodes before the proportion of the infected nodes in the network hits λ.

4.4. Weak limit of $\nu_\lambda^{(n)}$

Recall that $\nu_\lambda^{(n)}$, defined in (3.17), is the number of push rounds needed before the proportion of the infected nodes in the network reaches $\lambda \in (0, 1)$. Section 3 presents numerical computations of the expectation of this random variable for a network with 500 nodes and for different fanout values and makes several observations about the results. Here, we derive the weak limit of this random variable using the sequence $\{\varphi_i\}$ of (4.38) and Theorems 1 and 3.

Define

$$\bar{\nu}_\lambda \doteq \inf\{i : \varphi_i \geq \lambda\};$$

$\bar{\nu}$ is the number of rounds that the fluid limit network needs so that its proportion of infected nodes equals λ. Because φ is an increasing deterministic sequence, $\bar{\nu}_\lambda$ can be characterized as follows: $\bar{\nu}_\lambda = i$ for $\lambda \in [\varphi_i, \varphi_{i+1})$. Hence, as function of λ, $\bar{\nu}$ is right continuous, piecewise constant and it jumps precisely by 1 at the points $\{\varphi_1, \varphi_2, \varphi_3, ...\}$. As the final step of our analysis of the push algorithm we prove

Theorem 4. $\nu_\lambda^{(n)} \Rightarrow \bar{\nu}_\lambda$ *for* $\lambda \in (0, 1) - \{\varphi_0, \varphi_1, \varphi_2, ...\}$.

Proof. $\nu_\lambda = \bar{\nu}_\lambda = 0$ for $\lambda \leq \varphi_0 = 1 - \mu$; i.e., if λ is less than $1 - \mu$ (the initial proportion of infected nodes), the network already has more than λ proportion of infected nodes before any round begins.

To keep the proof short, we will treat the first two rounds; an argument that covers all rounds will involve the same ideas. Fix a $\lambda \in (\varphi_0, \varphi_1)$; we would like to show that $\nu_\alpha^{(n)} \Rightarrow \bar{\nu}_\lambda = 1$, i.e., as n goes to ∞, the infection level λ is attained in the first round with probability approaching 1. $\lambda < \varphi_1$, and the definitions (4.38) and (4.37) of φ_1 and $\bar{\tau}_\lambda$ imply

$$\bar{\tau}_\lambda < 1 - \mu; \tag{4.39}$$

this last inequality can also be expressed as follows: at time $1 - \mu$ the proportion of infected nodes in the fluid limit is φ_1; λ being strictly less than φ_1 and the fluid limit Γ being strictly increasing and deterministic, it must be that the first time the fluid limit has reached the infection level λ must be before time $1 - \mu$.

$\nu_\alpha^{(n)} = 1$ if and only if $\tau_\lambda^{(n)} \leq k_n$, which is the same inequality as

$$\frac{1}{\sqrt{n}}\left(\frac{1}{\sqrt{n}}\left(\tau_\lambda^{(n)} - n\bar{\tau}_\lambda\right)\right) \leq k_n/n - \bar{\tau}_\lambda. \tag{4.40}$$

The first term on the left is a constant and converges to 0; Theorem 3 says that the second term on the left converges weakly to a finite random variable. Thus, their product converges weakly to 0. (4.39) and $k_n/n \to (1-\mu)$ imply, on the other hand, that the limit of the right side is strictly greater than 0. Thus, the probability of the event expressed in this display, that is, the probability that $\nu_\lambda = 1$, indeed converges to 1.

For $\lambda \in (\varphi_1, \varphi_2)$, we would like to show $\nu_\lambda^{(n)}$ converges weakly to $\bar{\nu}_\lambda = 2$, i.e., we want to show that the probability of the event

$$k_n < \tau_\lambda < k_n + S_{k_n}$$

converges to 1 (k_n is the total number of random selections in the first round and $k_n + S_{k_n}$ is the same total after the second round). A rescaling and centering similar to (4.40) and Theorems 1, 3 imply this. □

An interesting question is the weak limit of $\nu_{\varphi_i}^{(n)}$. We have already covered the case $i = 1$ in the argument given in the numerical example following Theorem 2: $\nu_{\varphi_1}^{(n)} \Rightarrow T_1$ where $T_1 \in \{1, 2\}$ and takes these value with equal probability. The weak limit of the whole sequence $\left\{\nu_{\varphi_i}^{(n)}, i = 1, 2, 3, ...\right\}$ requires a longer analysis and we leave it to future work.

Figure 5 shows two graphs: first is that of $\bar{\nu}_\lambda$ for an initial infection rate of $\mu = 0.01$ and fanout $c = 3$; the second is $N(\lambda, 5 = 500\mu) = \mathbb{E}_5[\nu_\lambda^{(500)}]$; the graph of $N(\lambda, 50)$ for $c \in \{1, 7\}$ has been given earlier in Figure 2. Clearly, as Theorem 4 implies, these graphs overlap except at $\{\varphi_i\}$; at φ_i, $\bar{\nu}$ jumps by 1 while N rapidly goes from $i - 1$ to i. Theorem 4 also implies that the λ interval over which this transition occurs shrinks to the single point φ_i as the network size grows to ∞.

5. Conclusion

In the present paper we have tried to give a detailed analysis of the push type epidemic algorithm over fully connected networks. The starting point of our analysis was taking the random selections in a push round as the atomic operation of the push algorithm and defining a random walk whose each step corresponds to a random selection. The main body of the analysis consisted of computing weak limits of this random walk and its various functions. We expect the ideas of the present work to be directly applicable to other epidemic algorithms (such as those considered in [16, 20]) over fully connected networks.

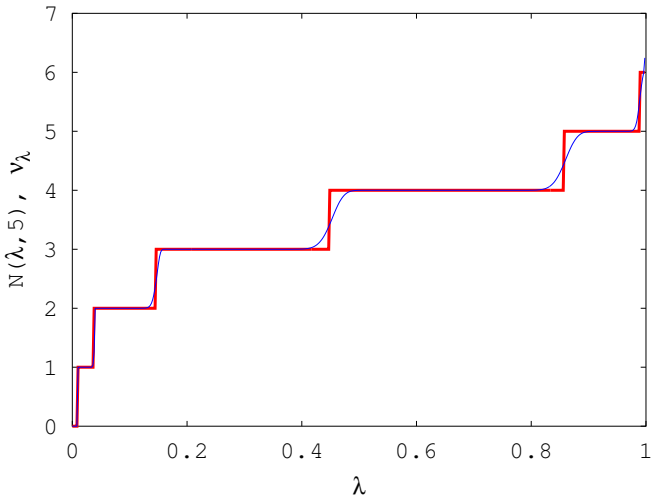


Figure 5: The graphs of $\bar{\nu}_\lambda$ (the thicker curve) and $N(\lambda, 5)$; both for $c = 3$

Fully connectedness is one of the natural limits of the collection of possible topologies over a given graph. As the topology of a network approaches full connectivity one expects the results for the latter to be good approximations for the former. One can further use results on quantities in fully connected networks as upper and lower bounds on the same quantities in other network topologies. After a detailed analysis of this basic case, an interesting and important direction is the generalization to more complex network structures, such as those considered in [7, 12].

To the best of our understanding, most of the literature on the asymptotic analysis of epidemic algorithms focus on obtaining upperbounds on the tail probabilities of the total dissemination time, $\nu_1^{(n)}$ in our notation. For this, authors often use large deviations results such as Chernoff's bound. However, most of the underlying processes in these models may have fluid and diffusion limits and these limits can give much more precise information about the distribution of key random variables, such as the total dissemination time. We hope that the present work provides an example of how this path can be followed in the context of a simple model.

Acknowledgement

Ali Devin Sezer's work on this article has been supported by the Rbuce-up European Marie Curie project, `http://www.rbuce-up.eu/`.

Appendix A. Proofs

Proof of Theorem 1. To avoid confusion between discrete and continuous time parameters, we will show continuous time parameters in parantheses,

e.g., we will write $X^n(t)$ instead of X^n_t. We begin by assuming that $c = 1$, the modifications for $c > 1$ will be straightforward. k_n is assumed to grow with n so that $\lim_n (n - k_n)/n = \mu \in (0, 1)$. Let $C_b(\mathbb{R}^2)$ denote the set of bounded and continuous functions on $\mathbb{R}^2$; and $C^2_0(\mathbb{R}^2)$ the set of twice differentiable functions with compact support with continuous Hessians. Define $T_n : C_b(\mathbb{R}^2) \to C_b(\mathbb{R}^2)$ as follows:

$$[T_n(f)](x) = \mathbb{E}_x[f(X^n(1/n), \Gamma^n(1/n))], x \in \mathbb{R}^2 \tag{A.1}$$

where the subscript x of $\mathbb{E}_x$ denotes conditioning on $(X^n(0), \Gamma^n(0)) = x = (x_1, x_2)$.

Define

$$A_n \doteq n(T_n - I). \tag{A.2}$$

Let us compute A_n explicitly for $f \in C^2_0(\mathbb{R}^2)$. The expectation in (A.1) is conditioned on $(X^n(0), \Gamma^n(0)) = (x_1, x_2)$, i.e., on

$$\frac{S_0 - \gamma_0}{\sqrt{n}} = x_1, \frac{\gamma_0}{n} = x_2. \tag{A.3}$$

This, (2.2), (2.7) and the definition of X^n imply

$$\begin{aligned} X^n(1/n) = \frac{S_1 - \gamma_1}{\sqrt{n}} &= \frac{S_0 - \gamma_0}{\sqrt{n}} + \frac{S_1 - S_0}{\sqrt{n}} - \frac{\gamma_1 - \gamma_0}{\sqrt{n}} \\ &= x_1 + \left(X_1 - \left(\frac{n - k_n}{n-1} - \frac{n}{n-1} x_2 \right) \right) / \sqrt{n} \\ \Gamma^n(1/n) = \frac{\gamma_1}{n} &= \frac{1}{n} \left(\gamma_0 \frac{n-2}{n-1} + \frac{n - k_n}{n-1} \right) \\ &= x_2 - \frac{x_2}{n-1} + \frac{n - k_n}{n(n-1)}. \end{aligned}$$

Define

$$\begin{aligned} Y_1 &\doteq X_1 - \left(\frac{n - k_n}{n-1} - \frac{n}{n-1} x_2 \right), \\ y_2 &\doteq -\frac{x_2}{n-1} + \frac{n - k_n}{n(n-1)}, \quad \Delta \doteq (Y_1/\sqrt{n}, y_2) \end{aligned} \tag{A.4}$$

where X_1 refers to the random variable given in (2.3). $\mathbb{E}_x[Y_1]$ and $E_x[Y_1^2]$ will be useful in what follows, so let us compute them first. Equations (A.3) and (2.3) give

$$\mathbb{E}_x[X_1] = \frac{n - k_n}{n-1} - \frac{n}{n-1} x_2 - \frac{\sqrt{n}}{n-1} x_1, \tag{A.5}$$

which gives

$$\mathbb{E}_x[Y_1] = \mathbb{E}_x[X_1] - \left(\frac{n - k_n}{n-1} - \frac{n}{n-1} x_2 \right) = -\frac{\sqrt{n}}{n-1} x_1. \tag{A.6}$$

On the other hand $\mathbb{E}_x[X_1] = \mathbb{E}_x[X_1^2] = (n - k_n - nx_2)/(n-1)$ implies

$$\begin{aligned}\mathbb{E}_x[Y_1^2] &= \mathbb{E}_x[X_1^2] + 2\mathbb{E}_x[Y_1]d_n - d_n^2 \\ &= (n - k_n - nx_2)/(n-1) - \frac{\sqrt{n}}{n-1}x_1 d_n - d_n^2\end{aligned} \tag{A.7}$$

where d_n refers to the constant $(-(n-k_n)+nx_2)/(n-1)$ in Y_1's definition.

The expectation that occurs in the definition (A.1) of T_n written in terms of the vector Δ is

$$\mathbb{E}_x[f(X^n(1/n), \Gamma^n(1/n))] = \mathbb{E}_x[f(x+\Delta)]$$

Let Df denote the gradient of c and Hf its Hessian; and let Δ^2 denote the tensor product $\Delta \otimes \Delta$. Using f's Taylor's expansion in the last display gives

$$\mathbb{E}_x[f(X^n(1/n), \Gamma^n(1/n))] = \mathbb{E}_x\left[f(x) + \langle Df(x), \Delta\rangle + \frac{1}{2}\langle Hf(x+\theta_n), \Delta^2\rangle\right] \tag{A.8}$$

where θ_n is a random vector that lies on the line segment connecting 0 to Δ. Let us deal with each of the terms that appear in (A.8) one by one. The function $f(x)$ is not random and so it comes out of the expectation. The first order term is:

$$\langle Df(x), \Delta\rangle = f_{x_1}(x)Y_1/\sqrt{n} + f_{x_2}(x)\, y_2.$$

That the second term is deterministic and (A.6) give

$$\mathbb{E}_x\left[\langle Df(x), \Delta\rangle\right] = -f_{x_1}(x)\frac{x_1}{n-1} + f_{x_2}(x)\, y_2. \tag{A.9}$$

$\mathbb{E}_x\left[\langle Hf(\theta_n), \Delta^2\rangle\right]$ is

$$\begin{aligned}&\frac{1}{n}\mathbb{E}_x\left[f_{x_1,x_1}(x+\theta_n)Y_1^2\right] + 2y_2\frac{1}{\sqrt{n}}\mathbb{E}_x\left[f_{x_1,x_2}(x+\theta_n)Y_1\right] \\ &\quad + (y_2)^2 f_{x_2,x_2}(x+\theta_n).\end{aligned}$$

Substituting these in (A.2) yields

$$\begin{aligned}A_n f = &-f_{x_1}\frac{n}{n-1}x_1 + f_{x_2}(x)(ny_2) + \frac{1}{2}\mathbb{E}_x\left[f_{x_1,x_1}(x+\theta_n)Y_1^2\right] \\ &+ y_2\sqrt{n}\mathbb{E}_x\left[f_{x_1,x_2}(x+\theta_n)Y_1\right] + \frac{1}{2}n(y_2)^2 f_{x_2,x_2}(x+\theta_n)\end{aligned} \tag{A.10}$$

Now let us compute the limits of each of the terms in the last sum as $n \to \infty$. The first term does not depend on n. The definition of y_2 and (4.20) imply

$$\lim_{n\to\infty} f_{x_2}(x)(ny_2) = f_{x_2}(\mu - x_2). \tag{A.11}$$

Note that

$$\theta_n \to 0 \tag{A.12}$$

uniformly, because Δ is bounded. This and the continuity of Hf imply

$$\lim_{n\to\infty} \frac{1}{2}\mathbb{E}_x\left[f_{x_1,x_1}(x+\theta_n)Y_1^2\right] = \frac{1}{2}f_{x_1,x_1}(x)\lim_{n\to\infty}\mathbb{E}_x\left[Y_1^2\right]. \tag{A.13}$$

The observation (A.12), continuity of Hf and $|(y_2)^2| = O(1/n^2)$ imply

$$\lim_{n\to 0} \frac{1}{2}n(y_2)^2 f_{x_2,x_2}(x+\theta_n) = 0. \tag{A.14}$$

Boundedness of Y_1, (A.6), (A.12), continuity of Hf an $|y_2| = O(1/n)$ imply

$$\lim_{n\to\infty} y_2\sqrt{n}\mathbb{E}_x\left[f_{x_1,x_2}(x+\theta_n)Y_1\right] = 0. \tag{A.15}$$

Letting $n \to \infty$ in (A.7) gives

$$\lim_{n\to\infty}\mathbb{E}_x\left[Y_1^2\right] = (\mu - x_2)(1-(\mu-x_2)). \tag{A.16}$$

The equations (A.10), along with (A.11), (A.13), (A.14), (A.15) and (A.16) yield $\lim_{n\to\infty} A_n f = Af$ where $Af \doteq \langle Df, (-x_1, \mu - x_2)\rangle + \frac{1}{2}(\mu - x_2)(1-(\mu-x_2))f_{x_1,x_1}$. One can check directly that A is the infinitesimal generator of the semigroup $T(\cdot)$ defined by the process (X_t, Γ_t). It follows from its definition that $T(\cdot)$ is a Feller semigroup on $C_b(\mathbb{R}^2)$. Furthermore, [10, Proposition 3.2, page 17] imply that $C^\infty(\mathbb{R}^2)$ forms a core for the generator A. Thus, [10, Theorem 1.2, page 31] and [10, Theorem 2.6, page 168] imply $(X^n, \Gamma^n) \Rightarrow (X, \Gamma)$.

Modifications for $c > 1$ are as follows. The variables Y_1 and y_2 are now defined as

$$Y_1 \doteq X_1 - c\left(\frac{n-k_n}{n-1} - \frac{n}{n-1}x_2\right) \qquad y_2 \doteq -\frac{x_2\, c}{n-1} + \frac{(n-k_n)\, c}{n(n-1)}$$

where X_1 has the hypergeometric distribution (2.6). Then

$$\mathbb{E}_x[Y_1] = -c\,\frac{\sqrt{n}}{n-1}x_1. \tag{A.17}$$

(A.7) becomes

$$\mathbb{E}_x[Y_1^2] = \mathbb{E}_x[X_1^2] + 2\mathbb{E}_x[Y_1]d_n - d_n^2 = \mathbb{E}_x[X_1^2] - \frac{c\sqrt{n}}{n-1}x_1 d_n - d_n^2$$

where d_n is now $c(-(n-k_n)+nx_2)/(n-1)$. For the asymptotic analysis, we only need the limit of the last display as $n \to \infty$. For n large, one can

approximate the hypergeometric X_1 as Binomial(p^*, c) with success probability $p^* = (k_n - nx_2)/n$, whose second moment is $cp^* + c(c-1)(p^*)^2$. Substituting this in the last display and letting $n \to \infty$ we get

$$\lim_{n\to\infty} \mathbb{E}_x[Y_1^2] = c\,(\mu - x_2)(1 - (\mu - x_2)) \tag{A.18}$$

which generalizes (A.16) to $c > 1$. On the other hand, we have

$$\mathbb{E}_x\left[\langle Df(x), \Delta\rangle\right] = -f_{x_1}(x)\, c\, \frac{x_1}{n-1} + f_{x_2}(x)\, y_2$$

(generalization of (A.9)). It follows that

$$\begin{aligned} A_n f = &-f_{x_1}\, c\, \frac{n}{n-1} x_1 + f_{x_2}(x)(ny_2) + \frac{1}{2}\mathbb{E}_x\left[f_{x_1,x_1}(x+\theta_n)Y_1^2\right] \\ &+ y_2\sqrt{n}\mathbb{E}_x\left[f_{x_1,x_2}(x+\theta_n)Y_1\right] + \frac{1}{2} n (y_2)^2 f_{x_2,x_2}(x+\theta_n). \end{aligned}$$

Same arguments as in the $c = 1$ case give

$$\lim_{n\to\infty} f_{x_2}(x)(ny_2) = f_{x_2}\, c\,(\mu - x_2).$$

The last two displays (A.17) and (A.18) imply $\lim_{n\to\infty} A_n f = Af$ where $Af \doteq \langle Df, c\,(-x_1, \mu - x_2)\rangle + \frac{1}{2}\, c\,(\mu - x_2)(1 - (\mu - x_2)) f_{x_1,x_1}$. The rest of the proof is the same as in the case of $c = 1$. □

Proof of Proposition 2. Define the stopping time

$$\tau_n' \doteq \inf\left\{ l : \frac{1}{\sqrt{n}}(S_l - \gamma_l) \geq \frac{n\Gamma_{1-\mu} - \gamma_{\lfloor nt_n \rfloor}}{\sqrt{n}} \right\}.$$

$S_l \geq n\Gamma_{1-\mu}$ is the same as $(S_l - \gamma_l)/\sqrt{n} \geq (n\Gamma_{1-\mu} - \gamma_l)/\sqrt{n}$. That γ_l is increasing in l implies that the right side of this inequality is decreasing in l. This implies

$$\{\tau_n \leq nt_n\} \subset \{\tau_n' \leq nt_n\}. \tag{A.19}$$

We know from Theorem 1 that $\gamma_{\lfloor nt_n \rfloor}/n \to \Gamma_t$. This and (4.30) imply $\lim_n (n\Gamma_{1-\mu} - \gamma_{\lfloor nt_n \rfloor})/\sqrt{n} \to C\sqrt{\mathrm{var}(X_{1-\mu})}$. Now define

$$\tau' \doteq \inf\left\{ t : \bar{X}_t \geq C\sqrt{\mathrm{var}(\bar{X}_{1-\mu})} \right\}.$$

The last two displays, that $t \to e^{ct}$ is monotone increasing, $t_n \to (1-\mu)$ (see (4.32)) and Theorem 1 imply $\lim_{n\to\infty} \mathbb{P}(\{\tau_n' \leq nt_n\}) \leq \mathbb{P}(\tau' \leq (1-\mu))$. $\bar{X}$ is a stochastic integral against a Brownian motion. Therefore, if we measure time using its quadratic variation it will be a standard Brownian motion [15, Theorem 4.6, page 174]. This and [15, Equation (6.3), page 80] give

$$\mathbb{P}(\tau' \leq (1-\mu)) = \sqrt{\frac{2}{\pi}} \int_C^\infty e^{\frac{-x^2}{2}} dx.$$

The last equality and (A.19) imply (4.33). □

References

[1] Mert Akdere, Cemal Çagatay Bilgin, Ozan Gerdaneri, Ibrahim Korpeoglu, Özgür Ulusoy, and Ugur Cetintemel. A comparison of epidemic algorithms in wireless sensor networks. *Computer Communications*, 29(13):2450–2457, 2006.

[2] Patrick Billingsley. *Convergence of Probability Measures, Second Edition*. Wiley, 1999.

[3] Alexander Birman. Computing approximate blocking probabilities for a class of all-optical networks. *Selected Areas in Communications, IEEE Journal on*, 14(5):852–857, 1996.

[4] Kenneth P. Birman, Mark Hayden, Oznur Ozkasap, Zhen Xiao, Mihai Budiu, and Yaron Minsky. Bimodal multicast. *ACM Transactions on Computer Systems (TOCS)*, 17(2):41–88, 1999.

[5] Béla Bollobás, Oliver Riordan, Joel Spencer, and Gábor Tusnády. The degree sequence of a scale-free random graph process. *Random Structures & Algorithms*, 18(3):279–290, 2001.

[6] Tom Britton. Stochastic epidemic models: a survey. *Mathematical biosciences*, 225(1):24–35, 2010.

[7] Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. Rumor spreading in social networks. *Theoretical Computer Science*, 412(24):2602–2610, 2011.

[8] Edward G. Coffman Jr, Zihui Ge, Vishal Misra, and Don Towsley. Network resilience: exploring cascading failures within bgp. In *Proc. 40th Annual Allerton Conference on Communications, Computing and Control*, 2002.

[9] Daryl J. Daley and Joseph Mark Gani. *Epidemic modelling: an introduction*, volume 15. Cambridge University Press, 2001.

[10] Stewart Ethier and Thomas G. Kurtz. Markov processes. characterization and convergence. *NY: John Willey and Sons*, 9, 1986.

[11] Patrick T. Eugster, Rachid Guerraoui, Anne-Marie Kermarrec, and Laurent Massoulié. Epidemic information dissemination in distributed systems. *Computer*, 37(5):60–67, 2004.

[12] Ayalvadi Ganesh, Laurent Massoulié, and Don Towsley. The effect of network topology on the spread of epidemics. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 2, pages 1455–1466. IEEE, 2005.

[13] Joe Gani. Random-allocation and urn models. *Journal of Applied Probability*, 41:313–320, 2004.

[14] Richard Gibbens and Frank P. Kelly. Dynamic routing in fully connected networks. *IMA Journal of Mathematical Control and Information*, 7(1):77–111, 1990.

[15] Ioannis Karatzas and Steven Eugene Shreve. *Brownian motion and stochastic calculus*, volume 113. Springer, 1991.

[16] Richard Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vocking. Randomized rumor spreading. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 565–574. IEEE, 2000.

[17] Jochen Mundinger, Richard Weber, and Gideon Weiss. Optimal scheduling of peer-to-peer file dissemination. *Journal of Scheduling*, 11(2):105–120, 2008.

[18] Öznur Özkasap, Mine Çağlar, Şule Yazıcı, and Selda Küçükçifçi. An analytical framework for self-organizing peer-to-peer anti-entropy algorithms. *Performance Evaluation*, 67(3):141–159, 2010.

[19] Dongyu Qiu and Rayadurgam Srikant. Modeling and performance analysis of bittorrent-like peer-to-peer networks. *ACM SIGCOMM Computer Communication Review*, 34(4):367–378, 2004.

[20] Sujay Sanghavi, Bruce Hajek, and Laurent Massoulié. Gossiping with multiple messages. *Information Theory, IEEE Transactions on*, 53(12):4640–4654, 2007.

[21] Henry C. Tuckwell and Ruth J. Williams. Some properties of a simple stochastic epidemic model of sir type. *Mathematical biosciences*, 208(1):76–97, 2007.

[22] Di Warren and Eugene Seneta. Peaks and eulerian numbers in a random sequence. *Journal of applied probability*, pages 101–114, 1996.

[23] Max A. Woodbury. On a probability distribution. *The Annals of Mathematical Statistics*, 20(2):311–313, 1949.

[24] Shouhuai Xu, Wenlian Lu, and Li Xu. Push-and pull-based epidemic spreading in networks: Thresholds and deeper insights. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 7(3):32, 2012.