Information Mutation and Spread of Misinformation in Timely Gossip Networks

Priyanka Kaswan Sennur Ulukus
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742
pkaswan@umd.edu *ulukus@umd.edu*

***Abstract*—We consider a network of n user nodes that receives updates from a source and employs an age-based gossip protocol for faster dissemination of version updates to all nodes. When a node forwards its packet to another node, the packet information gets mutated with probability p during transmission, creating misinformation. The receiver node does not know whether an incoming packet information is different from the packet information originally at the sender node. We assume that truth prevails over misinformation, and therefore, when a receiver encounters both accurate information and misinformation corresponding to the same version, the accurate information gets chosen for storage at the node. We study the expected fraction of nodes with correct information in the network and version age at the nodes in this setting using stochastic hybrid systems (SHS) modelling and study their properties. We observe that very high or very low gossiping rates help curb misinformation, and misinformation spread is higher with moderate gossiping rates. We support our theoretical findings with simulation results which shed further light on the behavior of above quantities.**

I. Introduction

In this work, we attempt to characterize spread of misinformation in an age-based gossip network [1], where information at the source node gets updated according to a Poisson process with rate λ_e; Fig. 1. We associate a version number with each information generated at the source, such that the version number of the current information at the source gets incremented by one post each source update. The source forwards its latest version to the network of n nodes, $\mathcal{N} = \{1, \ldots, n\}$ according to a Poisson process with rate λ_s, choosing the destination node uniformly at random from $\mathcal{N}$ each time. The network nodes wish to have access to the latest possible version of information, and therefore, each node only stores the latest version of information it has received so far and gets rid of all older information packets. Further, the nodes gossip with their neighboring nodes to further improve the timeliness of information in the network, whereby each node sends updates to its neighbors according to a Poisson process with rate λ.

The source always communicates accurate information to network nodes, however, there is a possibility of information getting mutated during inter-node transmissions in the network. This could either be because the sender node is not always honest and sometimes deliberately tampers with the information before forwarding it, or the information packet could get corrupted during transmission process. An example for this problem setting could be software distribution to end users, where the software vendor always provides reliable versions

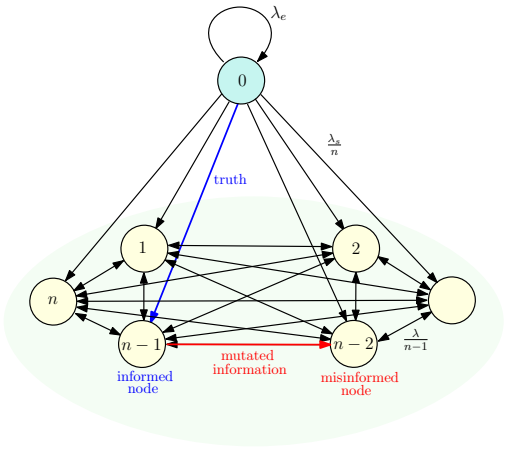


Fig. 1. Fully connected gossip network of n nodes with information probabilistically mutating into misinformation during internode gossip.

or iterations of software packages to users, however if the users instead obtain a software version from their neighboring users, they might occasionally receive an incorrectly functioning or even harmful version of the software. Other examples could be real time news dissemination in a region with truth getting mutated in inter-personal gossip, or smart sensor networks where occasionally noisy measurements are transmitted.

Network nodes want information that is both fresh and accurate. We use version age of information metric to characterize the freshness of information at nodes. If $V_s(t)$ corresponds to version number of the packet prevailing at the source at time t and $V_i(t)$ corresponds to the version number marked on the packet present at node i, then the instantaneous version age of information at node i at time t is defined as $X_i(t) = V_s(t) - V_i(t)$. Further, we use $T_i(t)$ to denote the accuracy of information at node i, with $T_i(t) = 1$ implying node i has accurate information (alternatively referred to as the truth), and $T_i(t) = 0$ implying that node i has inaccurate information (alternatively referred to as misinformation).

Whenever node i receives a packet, it compares the version number of the received packet with the version number of the packet in its possession, and if the version numbers are different, then node i discards the staler packet and keeps the fresher packet. However, node i does not prima facie know whether the piece of information it has received is the truth or not, or if it is different from sender's information or not.

Therefore, if the information already present at node i and the received information have the same version numbers, node i just keeps the information it trusts the most, say based on performance of software in the software distribution example, or measurement noise detected in smart sensor network example. If one of the packets contains accurate information (i.e., the truth), then node i would keep that packet, i.e., truth prevails over misinformation in our model. In this regard, we wish to know what fraction of nodes are misinformed in this network.

Gossip algorithms are decentralized algorithms which involve network nodes randomly contacting their neighbors to exchange packets, and traditionally, their analysis is done using the epidemic model of data spread [2] where the quantity of interest is dissemination time of a fixed message or set of messages to all nodes of the network [3]–[9]. Likewise, misinformation literature [10]–[15] also treats misinformation as a virus, such that users can become infected upon exposure, consequently turning them into spreaders. This allows for interpretation of misinformation spread as epidemic models like SIS and SIR, and the quantity of interest here is how long misinformation survives in a network. Though both gossip and misinformation spread are essentially information diffusion processes, the goals in the two cases are different, since rapid and complete spread of an update is preferred in the former.

The above works consider dissemination of static information in the network. However, with the highly dynamic nature of data sources in modern applications, network nodes are interested in the most up-to-date information at all times, which has recently motivated works in timely gossiping [1], [16]–[22]. Since our paper considers the subject of misinformation in timely gossip networks, the works that are most closely related to our paper are [20]–[22]: [20] studies the conditions under which the majority rule based estimation of time-varying binary valued source can cause incorrect source estimation or misinformation at network nodes. In [21], an adversary alters timestamps of packets, rebranding old packets as fresh and fresh packets as old, with the goal to replace circulation of fresh packets with outdated packets in the network. [22] considers a network model with two sources, such that the information obtained from one source is considered more reliable than the other source, and the network nodes prefer a reliable packet over an unreliable packet even when the former is a bit outdated with respect to the latter.

Our goal is somewhat similar to [22], as we are interested in finding what fraction of user nodes on average in the network have the truth and what is the version age of information at user nodes. However, in [22], once a packet is created at one of the sources, the packet information does not change during the packet diffusion process, whereas in the current paper, information is susceptible to mutating into misinformation during inter-node transmissions. Further, in [22], network nodes know if a particular packet has originated at the reliable source or the unreliable source, which allows nodes to consider a freshness-reliability trade-off. However, in the current work, nodes do not know whether a received information is the truth or not. In this respect, we model the information mutation problem as a stochastic hybrid system (SHS), and study the dependency of our results on various network parameters. We observe that, while very low gossip rates control the dissemination of mutated information on one hand, very high gossip rates help disseminate accurate fresh packets to all network nodes faster on the other hand. Thus, both extreme cases help curb misinformation and misinformation spread is higher with moderate gossip rates.

II. System Model and SHS Characterization

The system model consists of source node (node 0), that gets version updates with rate λ_e, and n user nodes $\mathcal{N} = \{1, \ldots, n\}$ that wish to have access to accurate and latest possible version of information. The source sends version updates to node $i \in \mathcal{N}$ on $(0, i)$ link according to a thinned Poisson process with rate $\frac{\lambda_s}{n}$. For $i, j \in \mathcal{N}$, node i sends updates to node j according to a thinned Poisson process with rate $\frac{\lambda}{n-1}$. The source possesses accurate and latest information at all times, i.e., $T_0(t) = 1$ and $X_0(t) = 0$, for all t. The source always communicates the truth to the user nodes, i.e., information does not mutate on $(0, i)$ links. However, the user nodes are not always honest. When node i sends a packet to node j on link (i, j) at time t, node i either honestly forwards the information in its possession with probability $1 - p$ or alters the contents of the packet to send misleading information with probability p, spreading misinformation in the latter case. When node j receives the packet from node i at time t, $T_j(t)$ and $X_j(t)$ are reset according to following state reset protocol:

- If $X_i(t^-) > X_j(t^-)$, node j rejects the incoming packet and $T_j(t), X_j(t)$ pair remains unchanged, since node j already has the fresher packet.
- If $X_i(t^-) < X_j(t^-)$, the incoming packet corresponds to a fresher version, hence node j replaces its packet with the incoming packet. Note that node j does not know if node i was honest or not, or the received packet is the truth or not. Hence $X_j(t) = X_i(t^-)$ and
$$T_j(t) = \begin{cases} T_i(t^-), & \text{node } i \text{ is honest} \\ 0, & \text{node } i \text{ is not honest} \end{cases}$$
- If $X_i(t^-) = X_j(t^-)$, both packets are equally fresh and $X_j(t)$ remains unchanged. Further, in our model, truth prevails over misinformation when two information packets of the same version are encountered and either of them carries the truth. The accuracy of the incoming packet is $T_i(t^-)$ if node i is honest in its communication, and 0 if node i is dishonest, while the version age of incoming packet is $X_i(t^-)$ in both cases. Hence,
$$T_j(t) = \begin{cases} \max\{T_i(t^-), T_j(t^-)\}, & \text{node } i \text{ is honest} \\ T_j(t^-), & \text{node } i \text{ is not honest} \end{cases}$$
Note that when node i communicates its information honestly to other nodes, it might lead to spread of misinformation if node i was misinformed before communication ($T_i(t^-) = 0$), i.e., honesty in communication does not imply delivery of the truth.

At time t, the fraction of users which possess the truth is

$$F(t) = \frac{T_1(t) + T_2(t) + \ldots + T_n(t)}{n} \tag{1}$$

To evaluate the expected fraction of users with truth, $F = \lim_{t\to\infty} \mathbb{E}[F(t)]$, we model the problem as an SHS. Note that node i essentially sends its packet to node j with honesty according to a thinned Poisson process with rate $(1-p)\frac{\lambda}{n-1}$ or sends a mutated copy of its packet according to a thinned Poisson process with rate $p\frac{\lambda}{n-1}$. For ease of exposition, we assume that instead of creating a mutated copy of its packet at the time of transmission, node i stores a mutated copy of its packet at all times, i.e., each node is assumed to store two packets (see Fig. 2). In Fig. 2, the green packet represents the packet actually present at the node i, having accuracy of $T_i(t)$ and version age of $X_i(t)$, and the orange packet represents a mutated copy of the green packet, hypothetically stored at the node, having accuracy of 0 (since it is a misleading packet) and version age $X_i(t)$. We are interested in finding the accuracy and version age of the green packets at all nodes. Essentially, now node i sends its green packet to node j with rate $(1-p)\frac{\lambda}{n-1}$ and its orange packet with rate $p\frac{\lambda}{n-1}$. Upon receiving a packet from node i, if node j chooses to keep the received packet as per the state reset protocol (thereby forming a new green packet at node j), then a new orange packet is also immediately created by mutating the new green packet, such that both the new packets have the same version age/number.

We now proceed to the SHS characterization, where we select the continuous state as $(\boldsymbol{T}(t), \boldsymbol{X}(t)) \in \mathbb{R}^{2n}$, where $\boldsymbol{T}(t) = [T_1(t), \ldots, T_n(t)]$ and $\boldsymbol{X}(t) = [X_1(t), \ldots, X_n(t)]$ denote the instantaneous accuracy and instantaneous version age, respectively, of the green packets stored at the n user nodes at time t. Transition (i, j, h) is said to take place when node i sends a packet to node j, with $h = 1$ depicting that node i sent its green packet (i.e., communicated its packet honestly) and $h = 0$ indicating node i sent its orange packet (i.e., communicated a misleading packet), with transition $(0, 0, 1)$ representing an update at the source. The SHS operates in a single discrete mode with the continuous state obeying the differential equation $(\dot{\boldsymbol{T}}(t), \dot{\boldsymbol{X}}(t)) = \boldsymbol{0}_{2n}$. The set of transitions for this SHS is

$$\begin{aligned}\mathcal{L} = \{(0,0,1)\} \cup \{(0,i,1) : i \in \mathcal{N}\} \cup \\ \{(i,j,h) : i,j \in \mathcal{N}, h \in \{0,1\}\}\end{aligned} \tag{2}$$

such that the transition (i, j, h) resets a state vector $(\boldsymbol{T}, \boldsymbol{X})$ at time t to $\phi_{i,j,h}(\boldsymbol{T}, \boldsymbol{X}, t) \in \mathbb{R}^{2n}$ post transition. The rates λ_{ijh} for each transition (i, j, h) are as follows,

$$\lambda_{ijh} = \begin{cases} \lambda_e, & i = 0, j = 0, h = 1 \\ \frac{\lambda_s}{n}, & i = 0, j \in \mathcal{N}, h = 1 \\ (1-p)\frac{\lambda}{n-1}, & i,j \in \mathcal{N}, h = 1 \\ p\frac{\lambda}{n-1}, & i,j \in \mathcal{N}, h = 0 \end{cases} \tag{3}$$

Next, we define some variables that would be useful in our analysis later. Given a set of nodes $A \subseteq \mathcal{N}$ and a continuous state $(\boldsymbol{T}, \boldsymbol{X})$, we define $X_A = \min_{j\in A} X_j$, with $X_A = \infty$ if $A = \varnothing$. Let $V(A) = \{j : X_j = \min_{j_1 \in A} X_{j_1}\}$ denote the set of nodes with lowest version age (note that both the green and orange packets at each node have the same version age). Consider another set of nodes B, such that $A \cap B = \varnothing$, i.e., the sets are disjoint. We define $T_{A,B}$ as follows:

- If $X_A \leq X_B$ and $1 \in \{T_j : j \in V(A)\}$, then $T_{A,B} = 1$. For all other cases, $T_{A,B} = 0$.

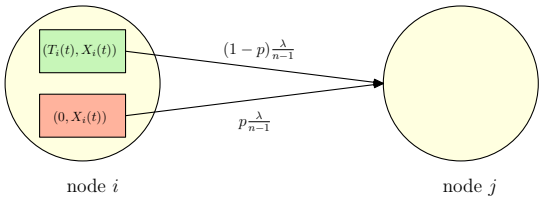


Fig. 2. The green packet depict the packet actually stored at the node i. The orange packet depicts a mutated copy of the green packet, hypothetically available at node i at all times.

Essentially, if we were to collect all the green packets from nodes in set A and all the orange packets from nodes in set B, and pick the freshest packets from this collection of green and orange packets, such that one of these freshest packets happen to have accuracy of 1, then $T_{A,B} = 1$. Clearly, if $X_A > X_B$, then all the freshest packets are orange packets, which have accuracy of 0. Further, even if $X_A \leq X_B$, no green packet is guaranteed to have the truth, since the nodes in $V(A)$ could have been misinformed by their previous senders. With these definitions, based on transition (i, j, h) at time t, the reset map to $\phi_{i,j,h}(\boldsymbol{T}, \boldsymbol{X}, t) = [T'_1, \ldots, T'_n, X'_1, \ldots, X'_n] \in \mathbb{R}^{2n}$ can be described as

$$T'_\ell = \begin{cases} 1, & i = 0, j \in \mathcal{N}, h = 1, \ell = j \\ T_{\{i,\ell\},\varnothing}, & i,j \in \mathcal{N}, h = 1, \ell = j \\ T_{\{\ell\},\{i\}}, & i,j \in \mathcal{N}, h = 0, \ell = j \\ T_\ell, & \text{otherwise} \end{cases} \tag{4}$$

and

$$X'_\ell = \begin{cases} X_\ell + 1, & i = 0, j = 0, h = 1, \ell = j \\ 0, & i = 0, j \in \mathcal{N}, h = 1, \ell = j \\ \min\{X_i, X_\ell\}, & i,j \in \mathcal{N}, h = 1, \ell = j \\ \min\{X_i, X_\ell\}, & i,j \in \mathcal{N}, h = 0, \ell = j \\ X_\ell, & \text{otherwise} \end{cases} \tag{5}$$

In the next section, we pick a series of test functions $\psi : \mathbb{R}^{2n} \times [0,\infty) \to \mathbb{R}$ that are time-invariant (consequently we will drop the third input t and write $\psi(\boldsymbol{T}, \boldsymbol{X}, t)$ as $\psi(\boldsymbol{T}, \boldsymbol{X})$), satisfying $\dot{\psi}(\boldsymbol{T}(t), \boldsymbol{X}(t)) = 0$, such that their long-term expected value $\mathbb{E}[\psi] = \lim_{t\to\infty} \mathbb{E}[\psi(\boldsymbol{T}(t), \boldsymbol{X}(t), t)]$ will be useful for analysis later. Defining $\mathbb{E}[\psi(\phi_{i,j,h})] = \lim_{t\to\infty} \mathbb{E}[\psi(\phi_{i,j,h}(\boldsymbol{T}(t), \boldsymbol{X}(t), t))]$, [23, Thm. 1] yields

$$0 = \sum_{(i,j,h)\in\mathcal{L}} (\mathbb{E}[\psi(\phi_{i,j,h})] - \mathbb{E}[\psi])\lambda_{ijh} \tag{6}$$

where the left side is set to zero due to $\frac{d\mathbb{E}[\psi(\boldsymbol{T}(t),\boldsymbol{X}(t),t)]}{dt} = 0$ at large t as the expectation stabilizes. For more details, the reader is encouraged to look at references [1], [23].

III. Misinformation and Version Age Derivations

Since the accuracy status evolution process and version age evolution process are statistically identical for all user nodes, in the following analysis, the sets A_k and B_m correspond to arbitrary sets of k and m user nodes with $A_k \cap B_m = \varnothing$. Our first test function is $\psi(\boldsymbol{T}, \boldsymbol{X}) = T_{A_k,B_m}$, which is modified upon transition (i,j,h) to $\psi(\phi_{i,j,h}(\boldsymbol{T}, \boldsymbol{X}, t)) = T'_{A_k,B_m}$ and can be characterized using (4), (5) and (6) as follows,

$$T'_{A_k,B_m} = \begin{cases} 1, & i=0, j \in A_k, h=1 \\ \mathbb{1}_{\{T_{A_k,\varnothing}=1, X_{A_k}=0\}}, & i=0, j \in B_m, h=1 \\ T_{A_k,B_{m+1}} & i \in \mathcal{N}\backslash(A_k \cup B_m), j \in A_k \cup B_m, h=0 \\ T_{A_k,B_{m+1}} & i \in \mathcal{N}\backslash(A_k \cup B_m), j \in B_m, h=1 \\ T_{A_{k+1},B_m} & i \in \mathcal{N}\backslash(A_k \cup B_m), j \in A_k, h=1 \\ T_{A_{k+1},B_{m-1}} & i \in B_m, j \in A_k, h=1 \\ T_{A_k,B_m} & \text{otherwise} \end{cases} \tag{7}$$

In (7), the effect of transition $i \in \mathcal{N}\backslash(A_k \cup B_m), j \in B_m, h = 1$ is interesting. Though node i sends its green packet to node j (as $h=1$ implies honest communication), T_{A_k,B_m} is only concerned with the orange packet at node j, which always maintains an accuracy status of 0. Hence, for the purposes of T_{A_k,B_m}, it appears as if node i sent its orange packet to potentially replace the orange packet at node j. Since $|\mathcal{N}\backslash(A_k \cup B_m)| = n-k-m$ and $|B_m| = m$, there are $(n-k-m)m$ such unique transitions, each with rate $\frac{\lambda}{n-1}$. Let us define

$$t_{k,m} = \lim_{t\to\infty} \mathbb{E}[T_{A_k,B_m}(t)], \quad c_k = \lim_{t\to\infty} \mathbb{E}[\mathbb{1}_{\{T_{A_k,\varnothing}=1,X_{A_k}=0\}}].$$

Then, using (6) with (7), we obtain

$$\begin{aligned} 0 =& (1-t_{k,m})k\frac{\lambda_s}{n} + (c_k - t_{k,m})m\frac{\lambda_s}{n} \\ &+ (t_{k,m+1} - t_{k,m})(n-k-m)(pk+m)\frac{\lambda}{n-1} \\ &+ (t_{k+1,m} - t_{k,m})(1-p)(n-k-m)k\frac{\lambda}{n-1} \\ &+ (t_{k+1,m-1} - t_{k,m})(1-p)km\frac{\lambda}{n-1} \end{aligned} \tag{8}$$

which upon rearrangement gives

$$\begin{aligned} t_{k,m} =& \\ & \frac{1}{(k+m)\frac{\lambda_s}{n} + (n-k-m)(k+m)\frac{\lambda}{n-1} + (1-p)km\frac{\lambda}{n-1}} \\ & \times \Bigg(k\frac{\lambda_s}{n} + c_k m\frac{\lambda_s}{n} + t_{k+1,m}(1-p)(n-k-m)k\frac{\lambda}{n-1} \\ & \quad + t_{k,m+1}(pk+m)(n-k-m)(n-k-m)\frac{\lambda}{n-1} \\ & \quad + t_{k+1,m-1}(1-p)km\frac{\lambda}{n-1} \Bigg) \end{aligned} \tag{9}$$

Note that since A_k and B_m are disjoint sets in (7), only

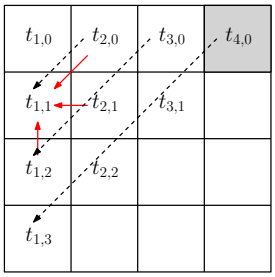


Fig. 3. Solving $t_{k,m}$ with dynamic programming for $n=4$.

$t_{k,m}$ satisfying $k \geq 0, m \geq 0, k+m \leq n$ are encountered. For example, (9) might at first give an impression that $t_{n,0}$ is dependent on $t_{n+1,0}$. However $\mathcal{N}\backslash(A_k \cup B_m) = \varnothing$ for $(k,m) = (n,0)$, hence there are no transitions of type $i \in \mathcal{N}\backslash(A_k \cup B_m), j \in B_m, h=1$ in (7), making the coefficient of $t_{n+1,0}$ zero in (9). Likewise the coefficients of $t_{n,1}$ and $t_{n+1,-1}$ also become zero, giving $t_{n,0} = 1$ from (9).

Our next test function is $\psi(\boldsymbol{T}, \boldsymbol{X}) = \mathbb{1}_{\{T_{A_k,\varnothing}=1,X_{A_k}=0\}}$, which has the following (i,j,h) transition map,

$$\mathbb{1}'_{\{T_{A_k,\varnothing}=1,X_{A_k}=0\}} = \begin{cases} 0, & i=0, j=0, h=1 \\ 1, & i=0, j \in A_k, h=1 \\ \mathbb{1}_{\{T_{A_{k+1},\varnothing}=1,X_{A_{k+1}}=0\}}, & i \in \mathcal{N}\backslash A_k, j \in A_k, h=1 \\ \mathbb{1}_{\{T_{A_k,\varnothing}=1,X_{A_k}=0\}}, & \text{otherwise} \end{cases} \tag{10}$$

Here, if there is a node $j \in A_k$ storing a green packet with accuracy 1 and version age 0, i.e., the freshest truth, then $\mathbb{1}_{\{T_{A_k,\varnothing}=1,X_{A_k}=0\}}$ will remain 1 irrespective of any type of transition, since all incoming packets to such node j will be discarded in accordance with the state reset protocol of Section II. If $\mathbb{1}_{\{T_{A_k,\varnothing}=1,X_{A_k}=0\}} = 0$, then the only way this test function can change value is if the latest truth is honestly communicated to a node in A_k (i.e., a green packet containing the latest truth is communicated), captured by the second and third cases of the reset map in (10).

The corresponding linear equation from (10) using (6) is

$$\begin{aligned} 0 =& (0-c_k)\lambda_e + (1-c_k)k\frac{\lambda_s}{n} \\ &+ (c_{k+1} - c_k)(1-p)k(n-k)\frac{\lambda}{n-1} \end{aligned} \tag{11}$$

which upon rearrangement gives

$$c_k = \frac{k\frac{\lambda_s}{n} + c_{k+1}(1-p)k(n-k)\frac{\lambda}{n-1}}{\lambda_e + k\frac{\lambda_s}{n} + (1-p)k(n-k)\frac{\lambda}{n-1}} \tag{12}$$

Since the laws of accuracy and age processes at all user nodes are the same, from (1) we get that the long-term expected fraction of nodes with truth is $F = \lim_{t\to\infty} \mathbb{E}[T_1(t)] = t_{1,0}$. Therefore, for computation of F, we first solve for c_k using (12), starting from $k=n$, with $c_n = \frac{\lambda_s}{\lambda_e+\lambda_s}$ and successively substituting for $k = n-1, \ldots, 1$ in an iterative fashion. Once we have computed all c_k, we compute $t_{k,m}$ using (9) in a dynamic programming fashion, shown in Fig. 3. Since in (9) $t_{k,m}$ depends on $t_{k+1,m}$, $t_{k,m+1}$ and $t_{k+1,m-1}$, starting with

$t_{n,0} = 1$, we solve $t_{r-s,s}$ following the order $r = n, \ldots, 1$ and $s = 0, 1, \ldots, r-1$, to reach $t_{1,0} = F$ at the end, as further guided by the arrows along the diagonals in Fig. (3).

Next, we come to test function $X_{A_k} = \min_{j \in A_k} X_j$, with $v_k = \lim_{t\to\infty} \mathbb{E}[X_{A_k}(t)]$. The (i, j, h) transition map can be written as follows,

$$X'_{A_k} = \begin{cases} X_{A_k} + 1, & i = 0, j = 0, h = 1 \\ 0, & i = 0, j \in A_k, h = 1 \\ X_{A_{k+1}}, & i \in \mathcal{N} \backslash A_k, j \in A_k, h = 1 \\ X_{A_{k+1}}, & i \in \mathcal{N} \backslash A_k, j \in A_k, h = 0 \\ X_{A_k}, & \text{otherwise} \end{cases} \tag{13}$$

Note how information mutation does not impact the version age of a packet, since the transition maps in (5) and (13) do not depend on h. The linear equation for (13) using (6) is

$$\begin{aligned} 0 =& (v_k + 1 - v_k)\lambda_e + (0 - v_k)k\frac{\lambda_s}{n} \\ &+ (v_{k+1} - v_k)(1-p)k(n-k)\frac{\lambda}{n-1} \\ &+ (v_{k+1} - v_k)pk(n-k)\frac{\lambda}{n-1} \end{aligned} \tag{14}$$

which upon rearrangement gives

$$v_k = \frac{\lambda_e + v_{k+1}k(n-k)\frac{\lambda}{n-1}}{k\frac{\lambda_s}{n} + k(n-k)\frac{\lambda}{n-1}} \tag{15}$$

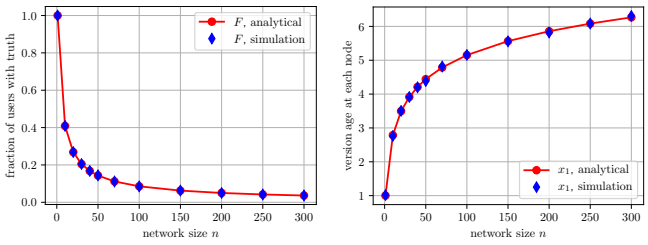
Defining $x_i = \lim_{t\to\infty} \mathbb{E}[X_i(t)]$, with $x_1 = \ldots = x_n$ due to network symmetry, similar to c_k, we can backward iterate on (15) to compute the $x_1 = v_1$.

IV. Analysis and Numerical Results

Though it is difficult to derive closed-form expressions for F (expected fraction of nodes with truth) and x_1 (expected version age at a node) from the complicated expressions found in (9), (12) and (15), we provide some interesting observations and further support our theoretical analysis with numerical results. To that end, we simulate the fully connected network model of Fig. 1 for up to a total time of 5×10^5 which we use as proxy for $t \to \infty$. We choose parameters $n = 10$, $p = 0.9$, $\lambda_e = 1$, $\lambda_s = 1$ and $\lambda = 1$, and vary one of the parameters at a time to observe their effects on F and x_1, plotting simulation points (blue dots) of $F = t_{1,0}$ on curves (red lines) obtained from equations (9), (12) and (15) in Figs. 4-8.

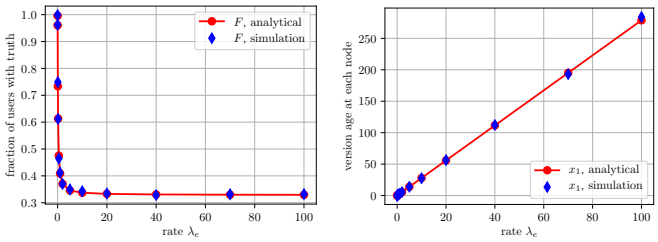
Fig. 4 shows the plots of F and x_1 with respect to the network size n. The real-time simulation points coincide with the values derived from the iterative calculations, supporting our theoretical results in Section III. Fig. 4(a) shows that $t_{1,0} = 1$ when $n = 1$, since there is no scope of information mutating in a network containing just one user node due to the absence of inter-node links. Fig. 4 suggests that a larger network size leads to more misinformation and staleness at nodes.

Next, Fig. 5 shows plots of F and x_1 with respect to λ_e. When λ_e is small, then substituting $\lambda_e \approx 0$ in (12) yields $c_k \approx 1$ recursively for $k = n, \ldots, 1$. This is because the source hardly gets updated, which causes version age to



(a) (b)

Fig. 4. Analytical (red) and simulation (blue) results compared for F and x_1 as a function of n, with parameters $\lambda_e = 1, \lambda_s = 1, \lambda = 1, p = 0.9$.



(a) (b)

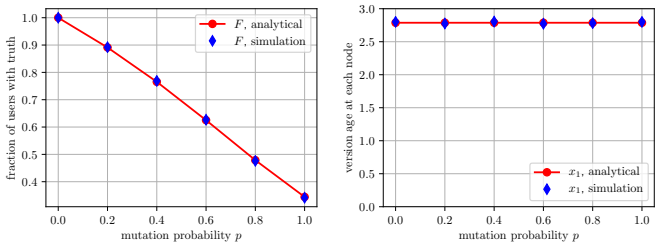
Fig. 5. F and x_1 as a function of λ_e, with $n = 10, \lambda_s = 1, \lambda = 1, p = 0.9$.

mostly remain zero at nodes, and due to the prevalence of truth over misinformation, eventually all nodes get the truth corresponding to the version present at the source, i.e., the latest truth. Consequently, by iterating over (9), as suggested in Fig. 3, we have $t_{k,m} \approx 1$ for all k, m with $k + m \leq n$.

On the other hand, when λ_e is very high, substituting $\lambda_e = \infty$ in (12) gives $c_k \approx 0$ for all k. This is because the current version number at the source is incrementing very fast, causing the versions present at the nodes to become very quickly outdated, driving x_1 very large (Fig. 5(b)) and c_k zero. $t_{k,m}$ depends on λ_e only through c_k, and as large λ_e drives c_k to zero, $t_{k,m}$ does become smaller for all k, m (can be seen by iterating in the manner suggested in Fig. 3), which suggests that faster version updates at the source causes network nodes to be more misinformed. However, unlike c_k, $t_{k,m}$ does not become zero, due to other positive terms in (9), owing to the fact that the truth prevails over misinformation, thus many nodes will still have the truth corresponding to older versions.

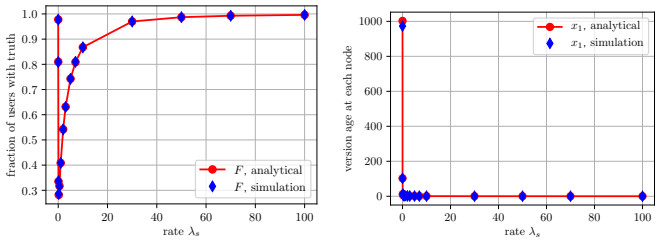
Next, if p is very low, then information hardly mutates and only versions of truth circulate in the network. This is supported by substituting $p = 0$ in (9) where $t_{k,0}$ only depends on $t_{k+1,0}$ giving $t_{k,0} = 1$ all for $k = n, \ldots, 1$ (see Fig. 6(a)). However when p is high, this does not guarantee that all nodes are misinformed, since the source constantly sends out the latest truth to the network which always gets accepted at the receiving nodes. Substituting $p = 1$ in (12) gives $c_k = \frac{k\frac{\lambda}{n}}{\lambda_s + k\frac{\lambda}{n}} > 0$, causing $t_{1,k}$ to have non-zero value for all k from (9). From (15), version age x_1 remains independent of p, which is also supported by Fig. 6(b).

The dependency of F on λ_s and λ is more interesting. When λ_s is very large, the source updates the network very fast, causing all nodes to have the latest truth at all times. Substituting $\lambda_s = \infty$ in (9), (12) and (15) gives $c_k \approx 1$,



(a) (b)

Fig. 6. F and x_1 as a function of p, with $n = 10, \lambda_e = 1, \lambda_s = 1, \lambda = 1$.



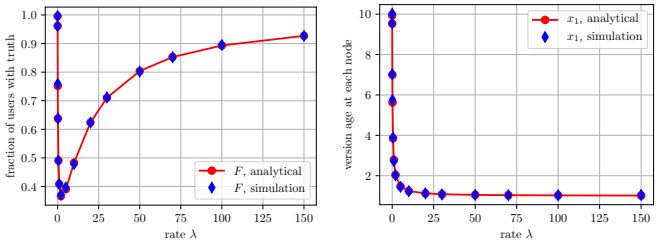
(a) (b)

Fig. 7. F and x_1 as a function of λ_s, with $n = 10, \lambda_e = 1, \lambda = 1, p = 0.9$.

$t_{k,m} \approx 1$ and $x_1 \approx 0$ (see Fig. 7). On the other hand, v_k for all k is a decreasing function of λ_s (can be inductively proved from (15) starting with $k = n$), thus when λ_s is very small, by substituting $\lambda_s = 0$ in (15) we get a large version age of $x_1 = \sum_{k=1}^{n-1} \frac{\lambda_e}{k(n-k\frac{\lambda}{n-1})} + \frac{\lambda_e}{\lambda_s} \approx \frac{\lambda_e}{\lambda_s}$ (In Fig. 7(b), the first simulation corresponds to $\lambda_s = 0.001$, for which $x_1 = \frac{\lambda_e}{\lambda_s} = \frac{1}{0.001} = 1000$). However, interestingly, since the source rarely sends any packet to the network when λ_s is close to zero, the whole network continues to gossip about the last version sent by the source to any of the network nodes, and eventually the truth corresponding to that version reaches all nodes. Substituting $\lambda_s = 0$ in (9) gives $t_{k,m} = 1$, which is also supported by the right extreme of Fig. 7(a). However, for intermediate values of λ_s, for $n \geq 2$, we get $c_k < 1$ from (12), which makes $t_{1,1} < 1$, which in turn makes $t_{1,0}$ strictly less than 1, as also supported by Fig. 7(a).

Finally, if gossiping rate λ is very large, implying that each node has high update sending capacity, then as soon as the source sends an update to a user node, the high gossiping rate allows for instant dissemination of the latest truth to all nodes of the network. This can be verified by substituting $\lambda = \infty$ in (9) which gives $t_{k,m} = 1$, and can be observed from Fig. 8(a). One would consequently expect the version age to also reduce at all nodes, however, the version age does not become zero in this case, since when the source gets a new version update, the version age at all nodes increments by one, and it continues to have a non-zero value at all nodes until the source sends a packet to some user node in the network. Substituting $\lambda = \infty$ in (15) gives $x_1 = v_1 = \ldots = v_n = \frac{\lambda_e}{\lambda_s}$, and in Fig. 8(b), version age converges to $\frac{\lambda_e}{\lambda_s} = 1$ as λ becomes large.

On the other hand, when λ is very small, implying there are negligible inter-node transmissions, then nodes do not receive mutated packets from other nodes and depend primarily on updates received from the source, which always transmits the



(a) (b)

Fig. 8. F and x_1 as a function of λ, with $n = 10, \lambda_e = 1, \lambda_s = 1, p = 0.9$.

truth. Substituting $\lambda = 0$ in (9) gives $t_{1,0} = 1$, as also observed in Fig. 8(a). Further, substituting $\lambda = 0$ in (15) gives $x_1 = \frac{n\lambda_e}{\lambda_s}$, which evaluates to 10 for the parameters in Fig. 8(b).

REFERENCES

[1] R. D. Yates. The age of gossip in networks. In *IEEE ISIT*, July 2021.

[2] N. Bailey. *The Mathematical Theory of Infectious Diseases and its Applications*. Griffin, 1975.

[3] A. J. Demers, D. H. Greene, C. H. Hauser, et al. Epidemic algorithms for replicated database maintenance. In *ACM PODC*, August 1987.

[4] Y. Minsky. *Spreading Rumors Cheaply, Quickly, and Reliably*. PhD thesis, Cornell University, March 2002.

[5] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. Randomized rumor spreading. In *FOCS*, November 2000.

[6] B. G. Pittel. On spreading a rumor. *SIAM Journal on Applied Mathematics*, 47(1):213–223, February 1987.

[7] S. Deb, M. Medard, and C. Choute. Algebraic gossip: a network coding approach to optimal multiple rumor mongering. *IEEE Transactions on Information Theory*, 52(6):2486–2507, June 2006.

[8] D. Mosk-Aoyama and D. Shah. Information dissemination via network coding. In *IEEE ISIT*, July 2006.

[9] S. Sanghavi, B. Hajek, and L. Massoulie. Gossiping with multiple messages. *IEEE Transactions on Information Theory*, 53(12):4640–4654, December 2007.

[10] L. Zhao, X. Qiu, X. Wang, and J. Wang. Rumor spreading model considering forgetting and remembering mechanisms in inhomogeneous networks. *Physica A: Statistical Mechanics and its Applications*, 392(4):987–994, February 2013.

[11] M. Nekovee, Y. Moreno, G. Bianconi, and M. Marsili. Theory of rumour spreading in complex social networks. *Physica A: Statistical Mechanics and its Applications*, 374(1):457–470, January 2007.

[12] A. Friggeri, L. Adamic, D. Eckles, and J. Cheng. Rumor cascades. In *ICWSM*, May 2014.

[13] Y. Moreno, M. Nekovee, and A. F. Pacheco. Dynamics of rumor spreading in complex networks. *Phys. Rev. E*, 69(6):066130, June 2004.

[14] F. Chierichetti, S. Lattanzi, and A. Panconesi. Rumor spreading in social networks. In *ICALP*, July 2009.

[15] D. Acemoglu, A. Ozdaglar, and A. ParandehGheibi. Spread of (mis)information in social networks. *Games and Economic Behavior*, 70(2):194–227, November 2010.

[16] B. Buyukates, M. Bastopcu, and S. Ulukus. Age of gossip in networks with community structure. In *IEEE SPAWC*, September 2021.

[17] P. Kaswan and S. Ulukus. Timely gossiping with file slicing and network coding. In *IEEE ISIT*, June 2022.

[18] P. Mitra and S. Ulukus. Timely opportunistic gossiping in dense networks. In *IEEE Infocom*, May 2023.

[19] M. Abd-Elmagid and H. Dhillon. Distribution of the age of gossip in networks. *Entropy*, 25(2):520–535, February 2023.

[20] M. Bastopcu, S. R. Etesami, and T. Başar. The dissemination of time-varying information over networked agents with gossiping. In *ISIT*, June 2022.

[21] P. Kaswan and S. Ulukus. Susceptibility of age of gossip to timestomping. In *IEEE ITW*, November 2022.

[22] P. Kaswan and S. Ulukus. Reliable and unreliable sources in age-based gossiping. In *IEEE ISIT*, June 2023.

[23] J. Hespanha. Modeling and analysis of stochastic hybrid systems. *IEE Proc. Control Theory & Applications, Special Issue on Hybrid Systems*, 153:520–535, January 2007.