

Network Connectivity–Information Freshness Tradeoff in Information Dissemination Over Networks

Arunabh Srivastava¹, *Graduate Student Member, IEEE*, and Sennur Ulukus², *Fellow, IEEE*

Abstract—We consider a gossip network consisting of a source generating updates and n nodes connected according to a given graph structure. The source keeps updates of a process, that might be generated or observed, and shares them with the gossiping network. The nodes in the network communicate with their neighbors and disseminate these version updates using a push-style gossip strategy. We use the version age metric to quantify the timeliness of information at the nodes. We first find an upper bound for the average version age for a set of nodes in a general network. Using this, we find the average version age scaling of a node in several network graph structures, such as two-dimensional grids, generalized rings and hyper-cubes. Prior to our work, it was known that when n nodes are connected on a ring the version age scales as $O(n^{\frac{1}{2}})$, and when they are connected on a fully-connected graph the version age scales as $O(\log n)$. Ours is the first work to show an age scaling result for a connectivity structure other than the ring and the fully-connected network, which constitute the two extremes of network connectivity. Our work helps fill the gap between these two extremes by analyzing a large variety of graphs with intermediate connectivity, thus providing insight into the relationship between the connectivity structure of the network and the version age, and uncovering a network connectivity–information freshness tradeoff.

Index Terms—Gossip networks, version age of information, graph theory, grid networks.

I. INTRODUCTION

WITH the roll-out of 5G communication, and a focus on decentralized wireless communication systems in 6G, including seamless machine-machine and human-machine interactions for time-critical tasks, timeliness has become an important metric to consider. This is of importance in many upcoming applications, such as drone swarms, networks of self-driving cars, smart manufacturing systems, and remote medical procedures.

It is now known that latency alone is not sufficient to characterize timeliness [1], and new metrics are needed. One such metric is age of information [2], [3], [4]. Age

of information is a well-accepted metric when studying the freshness of information in a wireless network [5]. Several related freshness metrics are age of incorrect information [6], age of synchronization [7], binary freshness metric [8], and version age of information [9], [10], [11].

Gossip, which is a fast and distributed way of sharing information between nodes across large networks, has been studied extensively in computer science. The first paper to study gossip algorithms is [12]. Since then, many works [13], [14], [15], [16], [17], [18], [19], have studied gossip networks from the context of disseminating information or performing a network-wide operation as fast as possible. In this paper, we consider gossip in networks as a means to disseminate information between nodes in the network, so that each node follows the update at a source node as closely as possible. In many real-time applications, timely delivery of information is crucial, but it may not be necessary for all information to reach all the nodes simultaneously or as quickly as physically possible. For example, in a network of self-driving cars, it is important for cars to be aware of the location, speed or other attributes of neighboring cars, whereas information about an accident that takes place far away is less relevant. Gossiping is particularly suitable for such applications. Gossiping is also widely used in database systems to keep data fresh, for example, in Amazon DynamoDB [20] and Apache Cassandra [21]. In such systems, it is critical for databases to store the latest information available, which is achieved using gossiping among the databases.

In this paper, we use the version age of information metric to quantify the freshness of information in the network. This metric is useful when packets carry their associated version numbers instead of a generation timestamp, which is required for the traditional age of information metric. In simple words, the version age of information of a node in a network is the number of versions by which the node lags behind with respect to the source node, which has the latest version of the update. Thus, the version age metric does not change unless the source receives or generates an update, unlike the age of information metric, which increases with time at a unit rate. Consequently, unlike the age of information metric, there is no penalty for not transmitting information unless there is an update at the source. This property makes version age useful for processes where information evolves according to a counting process.

Gossip was used with version age as a metric for the first time in [9], which characterizes the version age of information for general networks with Poisson update times and describes a set of recursive equations that can be used to calculate the

Received 28 May 2024; revised 12 February 2025; accepted 4 June 2025. Date of publication 10 September 2025; date of current version 23 April 2026. An earlier versions of this paper was presented in part at Allerton Conference, September 2023 [DOI: 10.1109/Allerton58177.2023.10313495] and in part IEEE MILCOM, October 2023 [DOI: 10.1109/MILCOM58377.2023.10356227]. (Corresponding author: Sennur Ulukus.)

The authors are with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742 USA (e-mail: arunabh@umd.edu; ulukus@umd.edu).

Communicated by E. Jorswieck, Associate Editor for Communications.

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TIT.2025.3608634>.

Digital Object Identifier 10.1109/TIT.2025.3608634

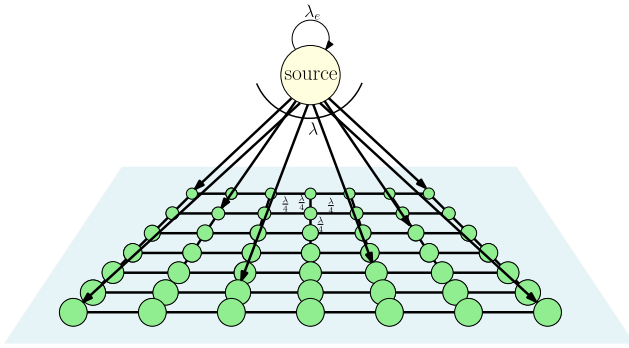


Fig. 1. A gossip network where the node in light yellow is the source generating updates and sending them to a network of nodes connected in a two-dimensional grid. There are no boundaries in the grid, and all nodes have four neighbors.

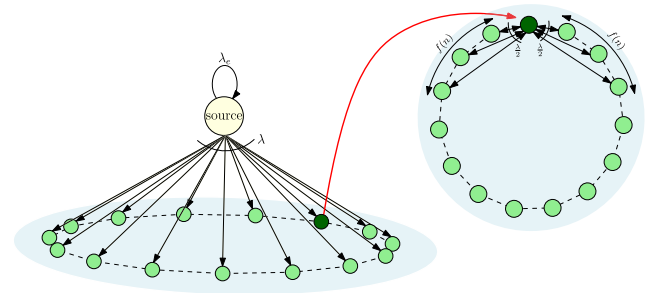


Fig. 2. A gossip network where the gossiping nodes form a generalized ring network. The source node updates itself at rate λ_e , and disseminates information to nodes arranged in a ring at rate λ . Each node communicates with $f(n)$ nodes on each of its side, thus communicating with $2f(n)$ nodes in total.

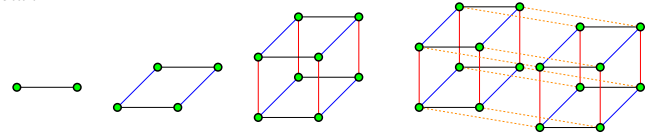


Fig. 3. The evolution of a unit hypercube network, from one dimension to four dimensions.

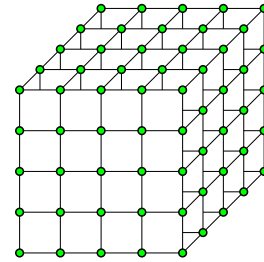


Fig. 4. A three-dimensional hypercube network with side length five.

average version age of connected subsets of a network. It also shows that the average version age of a single node in a fully-connected network scales as $O(\log n)$ with the network size n . Additionally, [9] experimentally observes, and [22] mathematically shows, that for a ring network, the version age scales as $O(\sqrt{n})$. [23] studies a network with a timestamping adversary, which can change the timestamps of the updates and fool the nodes into accepting an older version of the update. [24] studies the metric in the case where there are jamming adversaries. [25], [26] study the version age of information in a non-Poisson update setting. [27] considers version age in an age-sensing multiple access channel. [28] considers opportunistic gossiping protocols that achieve $O(1)$ scaling for version age in distributed multiple access channels. [29] studies the distributions of version age and its moments. [30] studies a gossip network consisting of energy harvesting sensors with the version age metric. [31] considers a gossip network where nodes decide the new update received using majority rule, and introduces a new error metric similar to the binary freshness metric.

In the following sections, we extend the analysis of version age in gossip networks to include several structured graph models. In order to do so, we modify the recursive equations in [9] and provide an upper bound and a lower bound for the average version age of a set in terms of the neighbors of a set or the neighboring edges of a set. Using these upper bound recursive equations, and lower bounds on the number of incoming edges to a set, we find an upper bound for the average version age of a single node for various graphs.

In this work, we consider the following graph structures, and obtain the following results:

- 1) A two-dimensional grid graph of n nodes with general sides k and m such that $n = km$ (see Fig. 1): We find that if $k = \Omega(\sqrt{m})$, then the version age scales as $O((km)^{\frac{1}{3}})$, i.e., as $O(n^{\frac{1}{3}})$. On the other hand, if $k = o(\sqrt{m})$, then the version age scales as $O(km^{\frac{1}{2}})$, i.e., as $O(n^{\frac{1}{2}})$.
- 2) A generalized ring network, where nodes are placed in a ring structure, and each node is connected to $f(n)$ nodes on either side (see Fig. 2): We find that the version age scales as $O(\log f(n) + \sqrt{\frac{n}{f(n)}})$.

- 3) A unit hypercube network in dimension $\log n$ (see Fig. 3): We find that the version age scales as $O(\log n \log \log n)$.
- 4) A general hypercube in fixed d -dimensional space, where each side goes to infinity (see Fig. 4): We perform simulations and see that the version age scales as $O(n^{\frac{1}{d+1}})$. We provide conjectures to support this claim.

The rest of the paper is organized as follows: The system model and the version age metric are introduced in Section II. A table of notations is given in Table I for quick reference and easy readability. Section III shows how we can use recursive equations to find upper and lower bounds of the version age of a set of nodes in the network. Section IV, Section VI and Section VII analyze the grid network, the generalized ring network, and the unit hypercube network, respectively, and find upper bounds for the version age of a representative single node these networks. Section VIII presents remarks and future directions. Section IX concludes the paper.

II. SYSTEM MODEL AND THE VERSION AGE METRIC

We consider a wireless network with a single source node that generates or observes updates of a particular process. This source updates itself following a Poisson process with rate λ_e independently of all other processes in the network. The network also includes a gossip network, denoted by $\mathcal{V}$, that consists of n nodes. The source node shares updates with this gossip network following a Poisson process with a total rate λ , giving each node in the network an equal chance of being

TABLE I
NOTATION AND DEFINITIONS

Notation	Definition
n	number of nodes in the gossiping network
$f(n)$	number of neighbors on one side of a node in a generalized ring network
d	fixed dimension in which a general hypercube network exists
λ_e	rate of source self-update Poisson process
$\mathcal{V}$	set of all nodes in the gossiping network
λ	rate of Poisson process with which each node shares information with its neighbors
$N_0(t)$	timestamp of the update with the source (node 0) at time t
$N_i(t)$	timestamp of the update with node i at time t
$X_i(t)$	version age of node i at time t
$X_S(t)$	version age of a set $\mathcal{S}$ of nodes at time t
v_S	long-term average version age of a set $\mathcal{S}$ of nodes
λ_{ij}	rate of Poisson process associated with sending information from node i to node j
$\lambda_i(\mathcal{S})$	total rate of updates going from node i to set $\mathcal{S}$ of nodes
$\mathcal{N}(\mathcal{S})$	number of neighboring nodes of set $\mathcal{S}$ of nodes
$\mathcal{E}(\mathcal{S})$	number of incoming edges to set $\mathcal{S}$ of nodes
$\mathcal{E}_S(i)$	number of incoming edges to set $\mathcal{S}$ of nodes from node i
$\bar{\mathcal{E}}(\mathcal{S})$	number of inner edges of set $\mathcal{S}$ of nodes
γ	Euler-Mascheroni Constant
$h(i)$	sum of the binary digits of an integer i

updated. Alternatively, this can be understood as the source sending updates to each node following a Poisson process of rate $\frac{1}{n}$, independent of all other processes going on in the network.

To quantify the freshness of update versions at the nodes in the gossip network, we use the version age metric. We start by defining the counting process associated with the updates at the source as $N_0(t)$. For node i in the gossip network, we define the counting process associated with the updates as $N_i(t)$. Next, we define the version age of node i as $X_i(t) = N_0(t) - N_i(t)$. Hence, the version age of any node in the gossip network is the number of versions the particular node is behind with respect to the source node. Similarly, we define the version age of a connected subset $\mathcal{S}$ of the gossip network as $X_S(t) = \min_{j \in \mathcal{S}} X_j(t)$. Note that $X_S(t)$ is the smallest version age in the subset of nodes in $\mathcal{S}$. Finally, we define the limiting average version age of set $\mathcal{S}$ as $v_S = \lim_{t \rightarrow \infty} \mathbb{E}[X_S(t)]$.

The version age of a node evolves as follows: If the source node updates itself, then the version age of every node in the gossip network increases by 1. If the source node sends an update to a node, then the node's version age reduces to 0. If a node i sends its version of the update to node j , then j updates itself if it has an older version than the one sent by i . Otherwise, it keeps its own version and rejects i 's version.

We define the rate of information flow from node i to node j as λ_{ij} which is the rate of the Poisson update process from i to j . We say that node i is a neighboring node of set $\mathcal{S}$ if $\lambda_{ij} > 0$ for some $j \in \mathcal{S}$. We define the total rate of information

flow into the set $\mathcal{S}$ from neighboring node i as $\lambda_i(\mathcal{S})$,

$$\lambda_i(\mathcal{S}) = \begin{cases} \sum_{j \in \mathcal{S}} \lambda_{ij}, & i \notin \mathcal{S} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Similarly, $\lambda_0(\mathcal{S})$ is the total rate of information flow from the source node into the set $\mathcal{S}$. We define $\mathcal{N}(\mathcal{S})$ to be the set of nodes which are neighboring nodes to the set $\mathcal{S}$. Finally, we define an incoming edge to set $\mathcal{S}$ as an edge that emanates from a neighboring node of set $\mathcal{S}$ and ends at a node in $\mathcal{S}$, and the set of all incoming edges to $\mathcal{S}$ is $\mathcal{E}(\mathcal{S})$.

With these definitions, we are now equipped to use the recursive equations for general networks proposed in [9]. We will also use these in Section III to find upper and lower bounds for v_S by modifying the recursive equations. In later sections, we use these to find upper bounds for the version age for the following different network topologies.

A. Description of the Two-Dimensional Grid Network

The network of gossiping nodes is arranged in a two-dimensional rectangular grid structure, with there being m nodes in the longer side, and k nodes in the shorter side. Each node in the grid has four neighbors, in the sense that there are no boundaries in the network, i.e., a node that is seemingly at the boundary in Fig. 1 is connected in a wrap-around fashion to the nodes on the opposite side of the row or column as shown in Fig. 5. Each node has a total gossiping rate of λ which it divides into four equal parts to gossip with

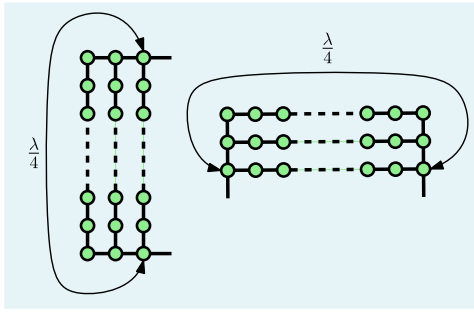


Fig. 5. All nodes have four connections. Nodes that are seemingly at the boundary in Fig. 1 are connected in a wrap-around fashion to the nodes on the opposite side of the row or column.

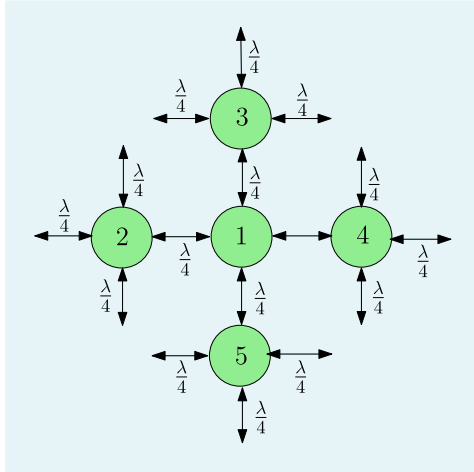


Fig. 6. Each node updates its four neighbors with an update rate of $\frac{\lambda}{4}$. Total update rate per node is λ .

its neighbors. Hence, each node in the grid gossips with its four neighbors in a push-style gossiping protocol as a rate $\frac{\lambda}{4}$ Poisson process, as shown in Fig. 6, independent of all other processes in the network.

B. Description of the Generalized Ring Network

Each node in the generalized ring network gossips with its neighbors, which are the nearest $2f(n)$ nodes, i.e., it gossips with $f(n)$ nodes on each side. It sends its version of the update to each neighboring node as a rate $\frac{\lambda}{2f(n)}$ Poisson process, resulting in a total rate λ for dissemination of information. This is shown in detail on the right hand side in Fig. 2.

C. Description of the Unit Hypercube Network

The nodes in the unit hypercube network are arranged in the shape of an m -dimensional hypercube, thus having $n = 2^m$ nodes in total. According to the properties of a hypercube, each node gossips with m neighbors as a rate $\frac{\lambda}{m}$ Poisson process. That is, each of the n nodes in the network gossips with $m = \log n$ neighbors. We analyze this network as the dimension of the unit hypercube m goes to infinity (hence, the number of nodes n goes to infinity). In Fig. 3, we show how the sets evolve as m increases from 1 to 4, and $n = 2^m$ correspondingly increases from 2 to 16.

D. Description of a General d -Dimensional Hypercube Network

The nodes in a general hypercube network are arranged in the shape of a hypercube in d dimensions, where $d > 2$. The hypercube has side length m , unlike the previous case, where each side was unit length. Hence, there are $n = m^d$ nodes in the network. We assume that this network has wrapped-around edges like the two-dimensional grid network. Hence, each node gossips with $2m$ neighbors as a rate $\frac{\lambda}{2m}$ Poisson process. We analyze this network as the length of the side m goes to infinity, and the dimension of the hypercube d is constant. In Fig. 4, we show a three dimensional hypercube ($d = 3$) with side length five ($m = 5$) with $n = m^d = 125$ nodes. Note that a $d = 2$ dimensional hypercube is a square two-dimensional grid, and a $d = 1$ dimensional hypercube is a line. A line is a ring that is cut in the middle.

III. THE RECURSIVE EQUATIONS AND UPPER/LOWER BOUNDS

In [9], Yates developed a recursive formula to find the version age of a connected set $\mathcal{S}$ in a general network as,

$$v_{\mathcal{S}} = \frac{\lambda_e + \sum_{i \in \mathcal{N}(\mathcal{S})} \lambda_i(\mathcal{S}) v_{\mathcal{S} \cup \{i\}}}{\lambda_0(\mathcal{S}) + \sum_{i \in \mathcal{N}(\mathcal{S})} \lambda_i(\mathcal{S})}. \quad (2)$$

This recursive formula expresses the version age of a subset $\mathcal{S}$ as a linear combination of the version ages of subsets that are obtained by adding one more neighboring node to this subset, i.e., $\mathcal{S} \cup \{i\}$, for all possible i . For a general network, the number of such equations obtained is exponential in the number of nodes, and is difficult to solve in closed-form.

We want to find the version age of a typical node j , $v_{\{j\}}$, in certain structured networks such as grids, generalized rings, and hypercubes. From the symmetry of the wrapped-around networks we consider, all nodes in the two-dimensional grid network and the d -dimensional general hypercube will have the same version age. Each node in the generalized ring network and the unit hypercube network will also have the same version age, since the networks are symmetric with respect to each node. Towards that end, we need to start with a single node j , i.e., $\mathcal{S} = \{j\}$ in (2), and keep adding neighboring nodes i until the entire network is exhausted. We know that the version age of the entire network $v_{\mathcal{V}}$ is $v_{\mathcal{V}} = \frac{\lambda_e}{\lambda}$. In principle, working backwards, we can find the version age of node j , $v_{\{j\}}$, by solving all of the linear equations obtained through the recursion in (2). To achieve that, we need to identify and exploit simplifications/symmetries in these equations.

References [9] and [22] were able to find the version ages of fully-connected and ring networks, respectively, by developing relatively simple bounds, as in those cases, the age of a subset depends only on the size of the subset due to the symmetry of the networks. However, in the case of a more complex network, subsets of the same size may have widely varying ages. Consider for instance a subset of 9 nodes in a two-dimensional grid network: those 9 nodes could be on a line, versus on a 3×3 square, versus on an L shape with 6 nodes on the longer arm of L and 4 nodes on the shorter arm. All these subsets of 9 nodes, even in a symmetric wrapped-around grid network, will have different version ages.

To tackle this problem, we find a general upper bound (Lemma 1) and a general lower bound (Lemma 2) for $v_{\mathcal{S}}$ in

a general network. In later sections, we show how to apply these lemmas to find upper bounds for the complex networks discussed in Section II.

The recursion in (2) provides a method to find the version age of any connected subgraph of a gossip network. Even for the simplest graphs, such as rings and fully-connected networks, the application of this recursion is tedious. For more complex graphs, such as the ones we consider in this paper, use of symmetry is restricted and direct application of the recursion is challenging since the number of sets grows quickly. In order to simplify the application of the recursion, the following lemma modifies the recursion to find bounds for sets only based on the one-expanded set that has the highest version age as opposed to all one-expanded sets. This enables us to use the geometry of the network and restrict the number of sets we need to handle to find a bound for the set in question.

Lemma 1: In any general gossip network, the following upper bound holds for a subset of nodes $\mathcal{S}$,

$$v_{\mathcal{S}} \leq \frac{\lambda_e + |\mathcal{N}(\mathcal{S})| \cdot \min_{i \in \mathcal{N}(\mathcal{S})} \lambda_i(\mathcal{S}) \cdot \max_{i \in \mathcal{N}(\mathcal{S})} v_{\mathcal{S} \cup \{i\}}}{\lambda_0(\mathcal{S}) + |\mathcal{N}(\mathcal{S})| \cdot \min_{i \in \mathcal{N}(\mathcal{S})} \lambda_i(\mathcal{S})}. \quad (3)$$

Proof: We start by rearranging (2) as follows,

$$\lambda_e = \lambda_0(\mathcal{S})v_{\mathcal{S}} + \sum_{i \in \mathcal{N}(\mathcal{S})} \lambda_i(\mathcal{S})(v_{\mathcal{S}} - v_{\mathcal{S} \cup \{i\}}). \quad (4)$$

Now, we lower bound the sum on the right hand side as,

$$\lambda_e \geq \lambda_0(\mathcal{S})v_{\mathcal{S}} + |\mathcal{N}(\mathcal{S})| \min_{i \in \mathcal{N}(\mathcal{S})} \lambda_i(\mathcal{S})(v_{\mathcal{S}} - v_{\mathcal{S} \cup \{i\}}) \quad (5)$$

$$\geq \lambda_0(\mathcal{S})v_{\mathcal{S}} + |\mathcal{N}(\mathcal{S})| \min_{i \in \mathcal{N}(\mathcal{S})} \lambda_i(\mathcal{S}) \min_{i \in \mathcal{N}(\mathcal{S})} (v_{\mathcal{S}} - v_{\mathcal{S} \cup \{i\}}) \quad (6)$$

$$= \lambda_0(\mathcal{S})v_{\mathcal{S}} + |\mathcal{N}(\mathcal{S})| \min_{i \in \mathcal{N}(\mathcal{S})} \lambda_i(\mathcal{S})(v_{\mathcal{S}} - \max_{i \in \mathcal{N}(\mathcal{S})} v_{\mathcal{S} \cup \{i\}}). \quad (7)$$

Rearranging (7) gives the desired result. ■

In a similar way, we can also prove the lower bound, as shown in the following lemma.

Lemma 2: In any general gossip network, the following lower bound holds for a subset of nodes $\mathcal{S}$,

$$v_{\mathcal{S}} \geq \frac{\lambda_e + |\mathcal{N}(\mathcal{S})| \cdot \max_{i \in \mathcal{N}(\mathcal{S})} \lambda_i(\mathcal{S}) \cdot \min_{i \in \mathcal{N}(\mathcal{S})} v_{\mathcal{S} \cup \{i\}}}{\lambda_0(\mathcal{S}) + |\mathcal{N}(\mathcal{S})| \cdot \max_{i \in \mathcal{N}(\mathcal{S})} \lambda_i(\mathcal{S})}. \quad (8)$$

Proof: We start with (4), and this time, we write an upper bound for the right hand side,

$$\lambda_e \leq \lambda_0(\mathcal{S})v_{\mathcal{S}} + |\mathcal{N}(\mathcal{S})| \max_{i \in \mathcal{N}(\mathcal{S})} \lambda_i(\mathcal{S})(v_{\mathcal{S}} - v_{\mathcal{S} \cup \{i\}}) \quad (9)$$

$$\leq \lambda_0(\mathcal{S})v_{\mathcal{S}} + |\mathcal{N}(\mathcal{S})| \max_{i \in \mathcal{N}(\mathcal{S})} \lambda_i(\mathcal{S}) \max_{i \in \mathcal{N}(\mathcal{S})} (v_{\mathcal{S}} - v_{\mathcal{S} \cup \{i\}}) \quad (10)$$

$$= \lambda_0(\mathcal{S})v_{\mathcal{S}} + |\mathcal{N}(\mathcal{S})| \max_{i \in \mathcal{N}(\mathcal{S})} \lambda_i(\mathcal{S})(v_{\mathcal{S}} - \min_{i \in \mathcal{N}(\mathcal{S})} v_{\mathcal{S} \cup \{i\}}). \quad (11)$$

Rearranging (11) gives the desired result. ■

IV. VERSION AGE IN AN $m \times k$ GRID

In this section, we find an upper bound for the version age of a single node in the two-dimensional grid network, described in Section II. We note that the average version age of each node in the grid will be the same due to the symmetry of the network. In order to calculate the upper bound, we will modify the recursive equations of [9]. We assume that m is the longer side of the grid and k is the shorter side of the grid.

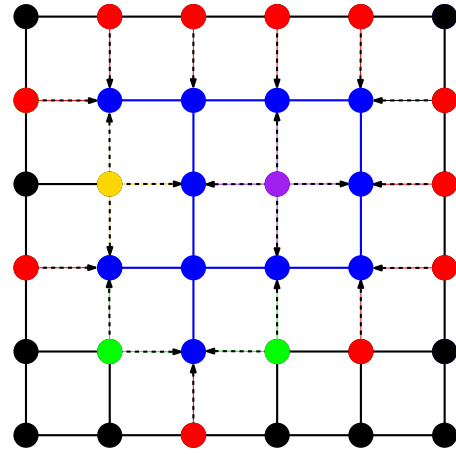


Fig. 7. An illustrative example for sets in (15), (16), (17) and (18). The set $\mathcal{S}$ is colored blue. Nodes of type $\mathcal{A}$ are colored red (they update into $\mathcal{S}$ with one link), nodes of type $\mathcal{B}$ are colored green (update into $\mathcal{S}$ with two links), nodes of type $\mathcal{C}$ colored yellow (update into $\mathcal{S}$ with three links) and nodes of type $\mathcal{D}$ colored purple (update into $\mathcal{S}$ with four links).

In the following lemma, we find an upper bound for $v_{\mathcal{S}}$ in terms of the number of incoming edges, instead of the number of neighbors of $\mathcal{S}$. Then, we find a lower bound for the number of incoming edges for subsets of the grid network, and write an upper bound for $v_{\mathcal{S}}$ in terms of the number of nodes in $\mathcal{S}$.

Lemma 3: In the grid network, consider a connected subgraph $\mathcal{S}$. Suppose $|\mathcal{S}| = j$ is such that $j \leq \frac{k^2}{4}$. Then, we have,

$$v_{\mathcal{S}} \leq \frac{\frac{2\lambda_e}{\lambda} + \sqrt{j} \max_{i \in \mathcal{N}(\mathcal{S})} v_{\mathcal{S} \cup \{i\}}}{\frac{j}{mk} + \sqrt{j}}. \quad (12)$$

Further, if $\frac{k^2}{4} + 1 \leq j \leq mk - \frac{k^2}{4}$, then, we have,

$$v_{\mathcal{S}} \leq \frac{\frac{2\lambda_e}{\lambda} + k \max_{i \in \mathcal{N}(\mathcal{S})} v_{\mathcal{S} \cup \{i\}}}{\frac{j}{mk} + k}. \quad (13)$$

Proof: We start with (4),

$$\lambda_e = \lambda_0(\mathcal{S})v_{\mathcal{S}} + \sum_{i \in \mathcal{N}(\mathcal{S})} \lambda_i(\mathcal{S})(v_{\mathcal{S}} - v_{\mathcal{S} \cup \{i\}}). \quad (14)$$

Now, we define a function $\mathcal{E}_{\mathcal{S}}(i) = \sum_{a \in \mathcal{S}} \mathbb{I}(\lambda_{ia} > 0)$, where $i \in \mathcal{N}(\mathcal{S})$ and $\mathbb{I}(\cdot)$ is the indicator function. $\mathcal{E}_{\mathcal{S}}(i)$ is the number of incident edges on set $\mathcal{S}$ that emanate from node i . Then, we can partition $\mathcal{N}(\mathcal{S})$ as follows,

$$\mathcal{A} = \{i \in \mathcal{N}(\mathcal{S}) : \mathcal{E}_{\mathcal{S}}(i) = 1\}, \quad (15)$$

$$\mathcal{B} = \{i \in \mathcal{N}(\mathcal{S}) : \mathcal{E}_{\mathcal{S}}(i) = 2\}, \quad (16)$$

$$\mathcal{C} = \{i \in \mathcal{N}(\mathcal{S}) : \mathcal{E}_{\mathcal{S}}(i) = 3\}, \quad (17)$$

$$\mathcal{D} = \{i \in \mathcal{N}(\mathcal{S}) : \mathcal{E}_{\mathcal{S}}(i) = 4\}. \quad (18)$$

That is, $\mathcal{A}$ is the set of neighbors of $\mathcal{S}$ that update into the set $\mathcal{S}$ with a single link (single edge), $\mathcal{B}$ is the set of neighbors of $\mathcal{S}$ that update into the set $\mathcal{S}$ with two links (two edges), and similarly, $\mathcal{C}$ with three links (three edges), and $\mathcal{D}$ with four links (four edges). An illustrative example is shown in Fig. 7. Using these sets, we bound (14) as,

$$\begin{aligned} \lambda_e &= \lambda_0(\mathcal{S})v_{\mathcal{S}} + \sum_{i \in \mathcal{A}} \frac{\lambda}{4}(v_{\mathcal{S}} - v_{\mathcal{S} \cup \{i\}}) + \sum_{i \in \mathcal{B}} \frac{\lambda}{2}(v_{\mathcal{S}} - v_{\mathcal{S} \cup \{i\}}) \\ &\quad + \sum_{i \in \mathcal{C}} \frac{3\lambda}{4}(v_{\mathcal{S}} - v_{\mathcal{S} \cup \{i\}}) + \sum_{i \in \mathcal{D}} \lambda(v_{\mathcal{S}} - v_{\mathcal{S} \cup \{i\}}) \end{aligned} \quad (19)$$

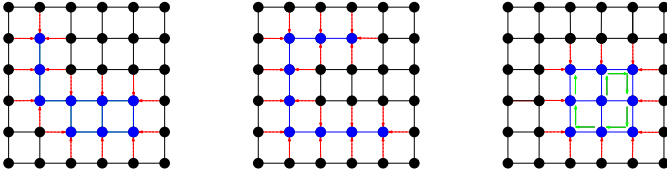


Fig. 8. An illustration of the lower bound: $|\mathcal{E}(\mathcal{S})| \geq 2\lceil 2\sqrt{j} \rceil$. The number of nodes in $\mathcal{S}$ (blue colored) is $j = 9$, thus the lower bound is $|\mathcal{E}(\mathcal{S})| \geq 12$. Different types of sets with 9 nodes have widely different number of incoming edges: The set on the left has 16 incoming edges (red arrows), the set in the middle has 20 incoming edges, and the set on the right has 12 incoming edges. The set on the right is a *spiral*, which is the set with the least number of incoming edges; the green lines show how the spiral is formed starting from the center node.

$$\begin{aligned} &\geq \lambda_0(\mathcal{S})v_S + |\mathcal{A}|\frac{\lambda}{4}\min_{i \in \mathcal{A}}(v_S - v_{S \cup \{i\}}) \\ &\quad + |\mathcal{B}|\frac{\lambda}{2}\min_{i \in \mathcal{B}}(v_S - v_{S \cup \{i\}}) + |\mathcal{C}|\frac{3\lambda}{4}\min_{i \in \mathcal{C}}(v_S - v_{S \cup \{i\}}) \\ &\quad + |\mathcal{D}|\lambda\min_{i \in \mathcal{D}}(v_S - v_{S \cup \{i\}}) \end{aligned} \quad (20)$$

$$\begin{aligned} &\geq \lambda_0(\mathcal{S})v_S + |\mathcal{A}|\frac{\lambda}{4}\min_{i \in \mathcal{N}(\mathcal{S})}(v_S - v_{S \cup \{i\}}) \\ &\quad + |\mathcal{B}|\frac{\lambda}{2}\min_{i \in \mathcal{N}(\mathcal{S})}(v_S - v_{S \cup \{i\}}) \\ &\quad + |\mathcal{C}|\frac{3\lambda}{4}\min_{i \in \mathcal{N}(\mathcal{S})}(v_S - v_{S \cup \{i\}}) \\ &\quad + |\mathcal{D}|\lambda\min_{i \in \mathcal{N}(\mathcal{S})}(v_S - v_{S \cup \{i\}}) \end{aligned} \quad (21)$$

$$\begin{aligned} &= \lambda_0(\mathcal{S})v_S + \frac{\lambda}{4}(|\mathcal{A}| + 2|\mathcal{B}| + 3|\mathcal{C}| + 4|\mathcal{D}|) \\ &\quad \times \min_{i \in \mathcal{N}(\mathcal{S})}(v_S - v_{S \cup \{i\}}) \end{aligned} \quad (22)$$

$$\begin{aligned} &= \lambda_0(\mathcal{S})v_S + \frac{\lambda}{4}(|\mathcal{A}| + 2|\mathcal{B}| + 3|\mathcal{C}| + 4|\mathcal{D}|) \\ &\quad \times (v_S - \max_{i \in \mathcal{N}(\mathcal{S})} v_{S \cup \{i\}}). \end{aligned} \quad (23)$$

Now, we note that $|\mathcal{E}(\mathcal{S})| = |\mathcal{A}| + 2|\mathcal{B}| + 3|\mathcal{C}| + 4|\mathcal{D}|$, where $|\mathcal{E}(\mathcal{S})|$ is the total number of incoming edges into the set $\mathcal{S}$. Thus, (23) states that for the two-dimensional grid, we have

$$\lambda_e \geq \lambda_0(\mathcal{S})v_S + \frac{\lambda}{4}|\mathcal{E}(\mathcal{S})|(v_S - \max_{i \in \mathcal{N}(\mathcal{S})} v_{S \cup \{i\}}). \quad (24)$$

Next, we use a result from [32] and [33] to provide a lower bound for the number of incoming edges into the set $\mathcal{S}$, i.e., $|\mathcal{E}(\mathcal{S})|$, to further lower bound (24). According to [32], on an infinite two-dimensional grid, of all the connected subsets with a fixed size j , the subset that has the minimum number of incoming edges into the subset is the *spiral*. The number of incoming edges into a spiral of j nodes is given by $2\lceil 2\sqrt{j} \rceil$. This bound has also been used in [33] in the analysis of gossip algorithms for max-consensus on grids. Hence, we have the following lower bound for $|\mathcal{E}(\mathcal{S})|$:

$$|\mathcal{E}(\mathcal{S})| \geq 2\lceil 2\sqrt{j} \rceil \geq 4\lfloor \sqrt{j} \rfloor \geq 2\sqrt{j}. \quad (25)$$

An illustrative example for this bound is shown in Fig. 8.

However, our grid network is a finite network wrapped-around at the boundaries. Hence, the effects of the boundaries affect this bound once j becomes large enough in comparison to the network size n . To see this, consider the sets on the grid given in Fig. 9. Both of the sets have $j = 15$ nodes.

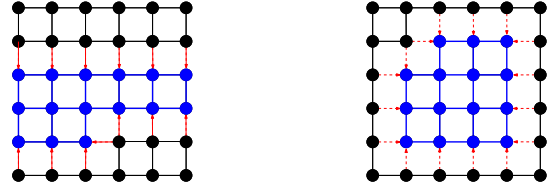


Fig. 9. Comparison of the number of incoming edges of two sets with 15 nodes on a 6×6 grid network. Due to boundary constraints, the set on the left has less incoming edges than the spiral set on the right.

Thus, the infinite-network lower bound given for these sets is $|\mathcal{E}(\mathcal{S})| \geq 2\lceil 2\sqrt{j} \rceil = 16$. We note that the right set has 16 incoming edges. The right set, in fact, is a spiral. Note that the right set does not touch (i.e., make use of) the finite boundaries of the grid network. The left set, on the other hand, makes use of the boundaries in order to cut out the incoming edges on the extreme right and extreme left of the set. Hence, the left set has 14 incoming edges, which is smaller than the infinite-network lower bound of 16. Had the grid been infinite, the left set would have had 17 incoming edges, one more than the spiral, satisfying the result in [32]. We also observe that the set on the left has a lower bound on the number of incoming edges given by $2k$. Hence, we conclude that up to a certain size of the set, the spiral will have a lower number of incoming edges, and for sets larger than that we will have a smaller lower bound as in the set on the left in Fig. 9. The exact inflection point is given by comparing the two lower bounds,

$$2k \geq 2\lceil 2\sqrt{j} \rceil. \quad (26)$$

This is satisfied for $j \leq \frac{k^2}{4}$. Hence, we conclude that $|\mathcal{E}(\mathcal{S})|$ is bounded below by $2\lceil 2\sqrt{j} \rceil$, and therefore, by $2\sqrt{j}$ from (25) by getting rid of the ceiling and floor functions, for j up to $j = \frac{k^2}{4} - 1$ and by $2k$ for j between $\frac{k^2}{4}$ and $mk - \frac{k^2}{4}$. This is because we have assumed that $m \geq k$, hence the set will first be a spiral of size up to $\frac{k^2}{4}$.

Thus, for $j \leq \frac{k^2}{4}$, substituting the first lower bound into (24), we obtain

$$\lambda_e \geq \lambda_0(\mathcal{S})v_S + \frac{\lambda}{4}2\sqrt{j}(v_S - \max_{i \in \mathcal{N}(\mathcal{S})} v_{S \cup \{i\}}). \quad (27)$$

Inserting $\lambda_0(\mathcal{S}) = \frac{\lambda j}{mk}$ and rearranging (27), we have

$$v_S \leq \frac{\lambda_e + \lambda \frac{\sqrt{j}}{2} \max_{i \in \mathcal{N}(\mathcal{S})} v_{S \cup \{i\}}}{\frac{\lambda j}{mk} + \lambda \frac{\sqrt{j}}{2}} \quad (28)$$

$$= \frac{\frac{\lambda_e}{\lambda} + \frac{\sqrt{j}}{2} \max_{i \in \mathcal{N}(\mathcal{S})} v_{S \cup \{i\}}}{\frac{j}{mk} + \frac{\sqrt{j}}{2}} \quad (29)$$

$$= \frac{\frac{2\lambda_e}{\lambda} + \sqrt{j} \max_{i \in \mathcal{N}(\mathcal{S})} v_{S \cup \{i\}}}{\frac{2j}{mk} + \sqrt{j}} \quad (30)$$

$$\leq \frac{\frac{2\lambda_e}{\lambda} + \sqrt{j} \max_{i \in \mathcal{N}(\mathcal{S})} v_{S \cup \{i\}}}{\frac{j}{mk} + \sqrt{j}}. \quad (31)$$

Now, similarly, for the range $\frac{k^2}{4} + 1 \leq j \leq mk - \frac{k^2}{4}$, we obtain

$$v_S \leq \frac{\lambda_e + \lambda \frac{k}{2} \max_{i \in \mathcal{N}(\mathcal{S})} v_{S \cup \{i\}}}{\frac{\lambda j}{mk} + \frac{\lambda}{2}k} \quad (32)$$

$$= \frac{\frac{\lambda_e}{\lambda} + \frac{k}{2} \max_{i \in \mathcal{N}(\mathcal{S})} v_{\mathcal{S} \cup \{i\}}}{\frac{j}{mk} + \frac{k}{2}} \quad (33)$$

$$= \frac{\frac{2\lambda_e}{\lambda} + k \max_{i \in \mathcal{N}(\mathcal{S})} v_{\mathcal{S} \cup \{i\}}}{\frac{2j}{mk} + k} \quad (34)$$

$$\leq \frac{\frac{2\lambda_e}{\lambda} + k \max_{i \in \mathcal{N}(\mathcal{S})} v_{\mathcal{S} \cup \{i\}}}{\frac{j}{mk} + k}, \quad (35)$$

which completes the proof. ■

This bound does not work for $j > mk - \frac{k^2}{4}$, since when j increases beyond this, we have a new type of set with the least number of incoming edges on set $\mathcal{S}$. This set $\mathcal{S}$ is equivalent to the whole grid minus a spiral shaped portion which has nodes that are not in $\mathcal{S}$. The spiral has the minimum number of incoming edges, and hence the same can be said about the complement set of nodes. This results in set $\mathcal{S}$ having the least number of incoming edges. The bound for the case $j > mk - \frac{k^2}{4}$ can be concluded from,

$$2[2\sqrt{mk-j}] < 2k, \quad (36)$$

where $|\mathcal{S}| = j$. We can now bound the age of sets with sizes between $mk - \frac{k^2}{4}$ and mk and get

$$v_{\mathcal{S}} \leq \frac{\frac{\lambda_e}{\lambda} + \lfloor \sqrt{mk-j} \rfloor \max_{i \in \mathcal{N}(\mathcal{S})} v_{\mathcal{S} \cup \{i\}}}{\frac{j}{mk} + \lfloor \sqrt{mk-j} \rfloor} \quad (37)$$

by a simple modification in Lemma 3.

Now, we state and prove the first main result of this paper in Theorem 1 below, which uses (12), (13) and (37) to find the version age scaling of a single node in the grid network.

Theorem 1: The version age of a single user in the grid network scales as $v_1 = O(\sqrt{m})$ if $k = o(\sqrt{m})$, and otherwise, the version age scales as $v_1 = O((mk)^{\frac{1}{3}})$.

Proof: In order to find the version age of a single user, we use (12) and (13) in Lemma 3 recursively, and obtain an analytic expression. The core idea we use to get recursive upper bounds using Lemma 3 is as follows: Start with a single node as the set $\mathcal{S}$. We can start with any node in the network, since each node will have the same average version age due to the symmetry of the network. Let us call the age of this set v_1 . Then, using Lemma 3, we get an upper bound on v_1 in terms of the age of a set with the maximum version age which can be constructed by adding one neighboring node to $\mathcal{S}$. The set which satisfies these conditions need not be known. We call the age of this set v_2 . In an inductive fashion, suppose we have $|\mathcal{S}| = j$ with age v_j , then we use Lemma 3 to get an upper bound by constructing a set which includes $\mathcal{S}$ and a neighboring node of $\mathcal{S}$, such that the one-larger set has the highest version age of all the possible one-larger sets constructed by adding neighboring nodes. We call the age of this set v_{j+1} . Since the upper bound is formed by independently bounding $|\mathcal{E}(\mathcal{S})|$ and $v_{\mathcal{S} \cup \{i\}}$, and the bound on $|\mathcal{E}(\mathcal{S})|$ is universal given the number of nodes in the set, we get the required recursion. Following this way of constructing one-larger sets, we eventually cover the entire grid, for which we know that the version age is exactly $\frac{\lambda_e}{\lambda}$. Hence, we recursively use these upper bounds to find an upper bound for the version age of a single node in the grid network. Thus, in essence, we

analyze a recursion of the following form, to find a bound on v_1 , by starting with $v_n = \frac{\lambda_e}{\lambda}$,

$$v_j \leq \begin{cases} \frac{\frac{2\lambda_e}{\lambda} + \sqrt{j}v_{j+1}}{\frac{j}{mk} + \sqrt{j}}, & j \leq \frac{k^2}{4}, \\ \frac{\frac{2\lambda_e}{\lambda} + kv_{j+1}}{\frac{j}{mk} + k}, & \frac{k^2}{4} + 1 \leq j \leq mk - \frac{k^2}{4}, \\ \frac{\frac{\lambda_e}{\lambda} + \lfloor \sqrt{mk-j} \rfloor v_{j+1}}{\frac{j}{mk} + \lfloor \sqrt{mk-j} \rfloor}, & mk - \frac{k^2}{4} < j \leq mk. \end{cases} \quad (38)$$

Now, we write the complete recursion in a way similar to the one found in [34, (13)-(16)]. This sum contains $n+1$ terms, which we divide into three parts, corresponding to the terms numbered between $\{1, \dots, \frac{k^2}{4}\}$, $\{\frac{k^2}{4} + 1, \dots, mk - \frac{k^2}{4}\}$ and $\{mk - \frac{k^2}{4} + 1, \dots, mk + 1\}$. We call the sum of terms in each of these parts X , Y and Z , respectively. First, we write a bound for the sum of terms making up X ,

$$X \leq \frac{2\lambda_e}{\lambda} \left(\frac{1}{1 + \frac{1}{mk}} \right) \left(1 + \sum_{i=1}^{\frac{k^2}{4}} \prod_{j=1}^i \frac{\sqrt{j}}{\sqrt{j+1} + \frac{j}{mk}} \right) \quad (39)$$

$$\leq \frac{2\lambda_e}{\lambda} \left(1 + \sum_{i=1}^{\frac{k^2}{4}} \prod_{j=1}^i \frac{\sqrt{j}}{\sqrt{j+1} + \frac{j}{mk}} \right). \quad (40)$$

Next, to find the order of the right hand side of (40), we proceed as follows,

$$X \leq \frac{2\lambda_e}{\lambda} \left(1 + \sum_{i=1}^{\frac{k^2}{4}} \prod_{j=1}^i \frac{1}{\left(1 + \frac{1}{j}\right)^{\frac{1}{2}} + \frac{\sqrt{j}}{mk}} \right) \quad (41)$$

$$\leq \frac{2\lambda_e}{\lambda} \left(1 + \sum_{i=1}^{\frac{k^2}{4}} \prod_{j=1}^i \frac{1}{1 + \frac{1}{2j} - \frac{1}{8j^2} + \frac{\sqrt{j}}{mk}} \right), \quad (42)$$

where we used $(1+x)^{\frac{1}{2}} \approx 1 + \frac{1}{2}x - \frac{1}{8}x^2$, for small x . Now, we define,

$$a_i = \prod_{j=1}^i \frac{1}{1 + \frac{1}{2j} - \frac{1}{8j^2} + \frac{\sqrt{j}}{mk}}. \quad (43)$$

By taking log of both sides, and using $\log(1+x) \approx x$, for small x , we obtain,

$$-\log a_i = \sum_{j=1}^i \log \left(1 + \frac{1}{2j} - \frac{1}{8j^2} + \frac{\sqrt{j}}{mk} \right) \quad (44)$$

$$\approx \sum_{j=1}^i \left(\frac{1}{2j} - \frac{1}{8j^2} + \frac{\sqrt{j}}{mk} \right) \quad (45)$$

$$\approx \frac{1}{2}(\log i + \gamma) - \delta + \frac{2i^{\frac{3}{2}}}{3mk}, \quad (46)$$

where $\gamma \approx 0.577$ is the Euler-Mascheroni constant, δ is a finite positive constant upper bounded by $\frac{\pi^2}{48}$ since $\sum_j \frac{1}{j^2} \rightarrow \frac{\pi^2}{6}$,

and the third term is bounded by the first term of the Euler-MacLaurin series expansion. Substituting (46) in (40),

$$X \leq \frac{2\lambda_e}{\lambda} \left(1 + \alpha \sum_{i=1}^{\frac{k^2}{4}} e^{-\frac{1}{2} \log i - \frac{2i^{\frac{3}{2}}}{3mk}} \right) \quad (47)$$

$$= \frac{2\lambda_e}{\lambda} \left(1 + \alpha \sum_{i=1}^{\frac{k^2}{4}} \frac{1}{i^{\frac{1}{2}}} e^{-\frac{2i^{\frac{3}{2}}}{3mk}} \right) \quad (48)$$

$$\leq \frac{2\lambda_e}{\lambda} \left(1 + \alpha \sum_{i=1}^{mk} \frac{1}{i^{\frac{1}{2}}} e^{-\frac{2i^{\frac{3}{2}}}{3mk}} \right), \quad (49)$$

where α considers the constants multiplied to the summation.

Next, we define a function h ,

$$h(x) = \frac{1}{x^{\frac{1}{2}}} e^{-\frac{2}{3}x^{\frac{3}{2}}}, \quad (50)$$

and then, write the Riemann sum with step size $(mk)^{-\frac{2}{3}}$,

$$\sum_{i=1}^{mk} \frac{1}{(mk)^{\frac{2}{3}}} h\left(\frac{i}{(mk)^{\frac{2}{3}}}\right) = \int_0^{\infty} \frac{1}{t^{\frac{1}{2}}} e^{-\frac{2}{3}t^{\frac{3}{2}}} dt = \beta < \infty. \quad (51)$$

Hence, we continue as follows,

$$\sum_{i=1}^{mk} \frac{1}{(mk)^{\frac{2}{3}}} h\left(\frac{i}{(mk)^{\frac{2}{3}}}\right) = \frac{1}{(mk)^{\frac{1}{3}}} \sum_{i=1}^{mk} \frac{1}{i^{\frac{1}{2}}} e^{-\frac{2i^{\frac{3}{2}}}{3mk}} = \beta. \quad (52)$$

Finally, continuing from (49), we have,

$$X \leq \frac{2\lambda_e}{\lambda} \left(1 + \beta'(mk)^{\frac{1}{3}} \right), \quad (53)$$

concluding that $X = O\left((mk)^{\frac{1}{3}}\right)$.

Next, we write the bound for the sum of terms making up Y ,

$$Y \leq \frac{2\lambda_e}{\lambda} \left(\prod_{j=1}^{\frac{k^2}{4}} \frac{\sqrt{j}}{\frac{j}{mk} + \sqrt{j+1}} \right) \left(\sum_{i=\frac{k^2}{4}+1}^{mk-\frac{k^2}{4}} \prod_{j=\frac{k^2}{4}+1}^i \frac{1}{1 + \frac{j}{mk^2}} \right) \quad (54)$$

$$= \frac{2\lambda_e}{\lambda} \left(\prod_{j=1}^{\frac{k^2}{4}} \frac{\sqrt{j}}{\frac{j}{mk} + \sqrt{j+1}} \right) \left(\prod_{j=1}^{\frac{k^2}{4}} \frac{1}{1 + \frac{j}{mk^2}} \right)^{-1} \times \left(\sum_{i=\frac{k^2}{4}+1}^{mk-\frac{k^2}{4}} \prod_{j=1}^i \frac{1}{1 + \frac{j}{mk^2}} \right). \quad (55)$$

The first product here is just $a_{\frac{k^2}{4}}$ from (43), and we use this substitution,

$$Y \leq \frac{2\lambda_e}{\lambda} \frac{2}{k} e^{-\frac{k^2}{12m}} \left(\prod_{j=1}^{\frac{k^2}{4}} \frac{1}{1 + \frac{j}{mk^2}} \right)^{-1} \left(\sum_{i=\frac{k^2}{4}+1}^{mk-\frac{k^2}{4}} \prod_{j=1}^i \frac{1}{1 + \frac{j}{mk^2}} \right). \quad (56)$$

The second product and the product in the third term can be substituted from [34], hence,

$$Y \leq \frac{2\lambda_e}{\lambda} \frac{2}{k} e^{-\frac{k^2}{12m}} e^{\frac{k^2}{16m}} \left(\sum_{i=\frac{k^2}{4}+1}^{mk-\frac{k^2}{4}} e^{-\frac{i^2}{2mk^2}} \right) \quad (57)$$

$$= \frac{2\lambda_e}{\lambda} \frac{2}{k} e^{-\frac{k^2}{48m}} \left(\sum_{i=\frac{k^2}{4}+1}^{mk-\frac{k^2}{4}} e^{-\frac{i^2}{2mk^2}} \right). \quad (58)$$

Using the method in [34], we use step size $\frac{1}{k\sqrt{m}}$, and obtain,

$$Y \leq \frac{2\lambda_e}{\lambda} \frac{2}{k} e^{-\frac{k^2}{48m}} k \sqrt{m} \left(\frac{1}{k\sqrt{m}} \sum_{i=\frac{k^2}{4}+1}^{mk-\frac{k^2}{4}} e^{-\frac{i^2}{2mk^2}} \right). \quad (59)$$

Now, when converting the summation in (59) to a Riemann integral, the upper limit in the summation corresponds to ∞ in the upper limit of the integral, since the upper limit is given by $\lim_{k \rightarrow \infty} \frac{mk-\frac{k^2}{4}}{k\sqrt{m}} = \lim_{k \rightarrow \infty} \sqrt{m} - \frac{k}{4\sqrt{m}} = \infty$. The lower limit can similarly be found to be $\lim_{k \rightarrow \infty} \frac{\frac{k^2}{4}}{k\sqrt{m}} = \frac{k}{4\sqrt{m}}$, and may go to ∞ or to a constant depending on the relation between k and m . If $m = o(k^2)$, then the lower limit goes to infinity, although more slowly than the upper limit. Hence, the integral goes to 0 in this case. On the other hand, if $k = o(\sqrt{m})$, then $e^{-\frac{k^2}{48m}} \rightarrow C$, and we get,

$$Y \leq \frac{4\lambda_e}{\lambda} C \sqrt{m} \int_0^{\infty} e^{-x^2} dx = \sqrt{\frac{\pi}{2}} \frac{4\lambda_e}{\lambda} C \sqrt{m}, \quad (60)$$

where C is a constant. If $k = \sqrt{m}$, then $\sqrt{m} = (mk)^{\frac{1}{3}}$.

Finally, we bound the sum of terms making up Z ,

$$Z = \frac{\lambda_e}{\lambda} + \frac{\lambda_e}{\lambda} \times \left(1 + \sum_{i=mk-\frac{k^2}{4}}^{mk-2} \prod_{j=mk-\frac{k^2}{4}}^i \frac{\lfloor \sqrt{mk-j} \rfloor}{\frac{j}{mk} + \lfloor \sqrt{mk-j-1} \rfloor} \right) \times \prod_{j=1}^{\frac{k^2}{4}} \frac{\sqrt{j}}{\frac{j}{mk} + \sqrt{j+1}} \frac{1}{1 - \frac{k}{4m} + \frac{\sqrt{k}}{2}} \times \left(\prod_{j=\frac{k^2}{4}+1}^{mk-\frac{k^2}{4}-1} \frac{1}{1 + \frac{j}{mk^2}} \right) \quad (61)$$

$$\leq \frac{\lambda_e}{\lambda} + \frac{\lambda_e}{\lambda} \left(1 + \sum_{i=mk-\frac{k^2}{4}}^{mk-2} \prod_{a=mk-i}^{\frac{k^2}{4}} \frac{\lfloor \sqrt{a} \rfloor}{\frac{mk-a+1}{mk} + \lfloor \sqrt{a-1} \rfloor} \right) \times \prod_{j=1}^{\frac{k^2}{4}} \frac{\sqrt{j}}{\sqrt{j+1}} \times \frac{1}{\frac{\sqrt{k}}{2}} \quad (62)$$

$$\leq \frac{\lambda_e}{\lambda} + \frac{\lambda_e}{\lambda} \frac{1}{\frac{\sqrt{k}}{2}} \frac{2}{k} \left(1 + \sum_{i=mk-\frac{k^2}{4}}^{mk-2} \prod_{a=mk-i}^{\frac{k^2}{4}} \frac{\lfloor \sqrt{a} \rfloor}{\lfloor \sqrt{a-1} \rfloor} \right) \quad (63)$$

$$\leq \frac{\lambda_e}{\lambda} + \frac{\lambda_e}{\lambda} \left(1 + \frac{1}{\frac{\sqrt{k}}{2}} \sum_{i=mk-\frac{k^2}{4}}^{mk-2} \frac{1}{\lfloor \sqrt{mk-i-1} \rfloor} \right) \quad (64)$$

$$= \frac{\lambda_e}{\lambda} + \frac{\lambda_e}{\lambda} \left(1 + \frac{1}{\frac{\sqrt{k}}{2}} \sum_{l=1}^{\frac{k^2}{4}-1} \frac{1}{\lfloor \sqrt{l} \rfloor} \right) \quad (65)$$

$$\leq \frac{\lambda_e}{\lambda} + \frac{\lambda_e}{\lambda} \left(1 + \frac{1}{\frac{\sqrt{k}}{2}} \left(1 + \sum_{l=2}^{\frac{k^2}{4}-1} \frac{1}{\sqrt{l}-1} \right) \right) \quad (66)$$

$$\leq \frac{\lambda_e}{\lambda} + \frac{\lambda_e}{\lambda} \left(1 + \frac{1}{\frac{\sqrt{k}}{2}} \left(1 + \sum_{l=2}^{\frac{k^2}{4}-1} \frac{4}{\sqrt{l}} \right) \right) \quad (67)$$

$$\leq \frac{\lambda_e}{\lambda} + \frac{\lambda_e}{\lambda} \left(1 + \frac{1}{\frac{\sqrt{k}}{2}} 4k \right) \quad (68)$$

$$= \frac{\lambda_e}{\lambda} (2 + 8\sqrt{k}). \quad (69)$$

Hence, using (53), (60) and (69), we write,

$$v_1 \leq X + Y + Z \quad (70)$$

$$\leq \frac{\lambda_e}{\lambda} \left(2 + \beta'(mk)^{\frac{1}{3}} + 2\sqrt{2\pi}e^{-\frac{k^2}{48m}}\sqrt{m} + 8\sqrt{k} \right). \quad (71)$$

Finally, we conclude that if $k = o(\sqrt{m})$, then $v_1 = O(\sqrt{m})$, otherwise, $v_1 = O((mk)^{\frac{1}{3}})$. ■

Remark 1: We found in Theorem 1 that in a two-dimensional grid network, the version age scaling depends on the relation between the number of nodes in each side of the grid. If $k = o(\sqrt{m})$, then the network behaves like a ring network, its age growing as $O(n^{\frac{1}{3}})$. On the other hand, if $m = o(k^2)$, then the version age grows slowly as $O(n^{\frac{1}{3}})$. Hence, we have a sharp cutoff, $m = k^2$, and drastically different age scaling on either side of this cutoff. To give explicit examples: As intuitively expected, a $2 \times \frac{n}{2}$ grid behaves more like a line (i.e., ring) and has an average version age of $O(n^{\frac{1}{3}})$, while a $\sqrt{n} \times \sqrt{n}$ square grid behaves like a true grid and has an average version age of $O(n^{\frac{1}{3}})$. More interestingly, and somewhat less intuitively, an $n^{\frac{1}{5}} \times n^{\frac{4}{5}}$ grid behaves like a line (i.e., ring) while a $n^{\frac{2}{5}} \times n^{\frac{3}{5}}$ grid behaves like a true grid, even though in both cases both dimensions of the grid grow unboundedly with n .

V. VERSION AGE IN A GENERALIZED RING

In this section, we find an upper bound for the version age of a single node in the generalized ring network, described in Section II. We note that the average version age of each node in the network will be the same due to the symmetry of the network. In order to calculate the upper bound, we modify the recursive equations of [9] given in (2), analogous to the method described in Section IV, but specialized to the topology of the generalized ring this time in the following lemma.

Lemma 4: For any connected subset $\mathcal{S}$ of the generalized ring network, we have,

$$v_{\mathcal{S}} \leq \frac{\frac{\lambda_e}{\lambda} + \frac{|\mathcal{E}(\mathcal{S})|}{2f(n)} \max_{i \in \mathcal{N}(\mathcal{S})} v_{\mathcal{S} \cup \{i\}}}{\frac{|\mathcal{S}|}{n} + \frac{|\mathcal{E}(\mathcal{S})|}{2f(n)}}. \quad (72)$$

Proof: We start with (4) in a similar way to the proof in Lemma 3,

$$\lambda_e = \lambda_0(\mathcal{S})v_{\mathcal{S}} + \sum_{i \in \mathcal{N}(\mathcal{S})} \lambda_i(\mathcal{S})(v_{\mathcal{S}} - v_{\mathcal{S} \cup \{i\}}), \quad (73)$$

and partition $\mathcal{N}(\mathcal{S})$ into $2f(n)$ sets according to the number of incoming nodes into $\mathcal{S}$ from any $i \in \mathcal{N}(\mathcal{S})$,

$$\mathcal{A}_j = \{i \in \mathcal{N}(\mathcal{S}) : \mathcal{E}_{\mathcal{S}}(i) = j\}, \quad (74)$$

where $1 \leq j \leq 2f(n)$. Now, we proceed from (73) as,

$$\lambda_e = \lambda_0(\mathcal{S})v_{\mathcal{S}} + \sum_{j=1}^{2f(n)} \sum_{i \in \mathcal{A}_j} \lambda_i(\mathcal{S})(v_{\mathcal{S}} - v_{\mathcal{S} \cup \{i\}}) \quad (75)$$

$$\geq \lambda_0(\mathcal{S})v_{\mathcal{S}} + \sum_{j=1}^{2f(n)} |\mathcal{A}_j| \min_{i \in \mathcal{A}_j} \lambda_i(\mathcal{S})(v_{\mathcal{S}} - v_{\mathcal{S} \cup \{i\}}) \quad (76)$$

$$\geq \lambda_0(\mathcal{S})v_{\mathcal{S}} + \sum_{j=1}^{2f(n)} |\mathcal{A}_j| \min_{i \in \mathcal{A}_j} \lambda_i(\mathcal{S}) \min_{i \in \mathcal{A}_j} (v_{\mathcal{S}} - v_{\mathcal{S} \cup \{i\}}) \quad (77)$$

$$= \lambda_0(\mathcal{S})v_{\mathcal{S}} + \sum_{j=1}^{2f(n)} |\mathcal{A}_j| \frac{j\lambda}{2f(n)} (v_{\mathcal{S}} - \max_{i \in \mathcal{A}_j} v_{\mathcal{S} \cup \{i\}}) \quad (78)$$

$$\geq \lambda_0(\mathcal{S})v_{\mathcal{S}} + \sum_{j=1}^{2f(n)} |\mathcal{A}_j| \frac{j\lambda}{2f(n)} (v_{\mathcal{S}} - \max_{i \in \mathcal{N}(\mathcal{S})} v_{\mathcal{S} \cup \{i\}}) \quad (79)$$

$$= \lambda_0(\mathcal{S})v_{\mathcal{S}} + |\mathcal{E}(\mathcal{S})| \frac{\lambda}{2f(n)} (v_{\mathcal{S}} - \max_{i \in \mathcal{N}(\mathcal{S})} v_{\mathcal{S} \cup \{i\}}), \quad (80)$$

where $|\mathcal{E}(\mathcal{S})| = \sum_{j=1}^{2f(n)} j|\mathcal{A}_j|$ is the total number of incoming edges into set $\mathcal{S}$. Rearranging this with the substitution $\lambda_0(\mathcal{S}) = \frac{|\mathcal{S}|}{n}$ proves the lemma. ■

Next, in order to use Lemma 4, we need to identify a lower bound for $|\mathcal{E}(\mathcal{S})|$ in the case where the number of nodes in $\mathcal{S}$ is fixed. To this end, in the following lemma, we find the set of j nodes that has the least number of incoming edges.

Lemma 5: Given all connected subsets $\mathcal{S}$ such that $|\mathcal{S}| = e$, the set which has the minimum number of incoming edges is the continuous set of e nodes.

Proof: We define edges that start at a node in $\mathcal{S}$ and end at a node in $\mathcal{S}$ as inner edges. Let the set of all inner edges in set $\mathcal{S}$ be denoted by $\bar{\mathcal{E}}(\mathcal{S})$. We note that the total number of edges incident on nodes in set $\mathcal{S}$ is $2ef(n)$, and each edge can only be an inner edge or an incoming edge. Hence, $2|\bar{\mathcal{E}}(\mathcal{S})| + |\mathcal{E}(\mathcal{S})| = 2ef(n)$. Thus, showing that the set of continuous nodes has the minimum number of incoming edges is the same as showing that it has the maximum number of inner edges. We will show that $|\bar{\mathcal{E}}(\mathcal{S})|$ is the highest in the continuous set of e nodes.

Let the set of e continuous nodes be $\mathcal{S}_1$, and choose any other connected set of e nodes and call it $\mathcal{S}_2$. Next, label each node in both sets as $1_{\mathcal{S}_1}, 2_{\mathcal{S}_1}, \dots, e_{\mathcal{S}_1}$ and $1_{\mathcal{S}_2}, 2_{\mathcal{S}_2}, \dots, e_{\mathcal{S}_2}$, respectively. The labels start at the node at one end of the set and end at the other end, covering each node in the order of their position. Now, we compare each $i_{\mathcal{S}_1}$ and $i_{\mathcal{S}_2}$. We know that both nodes have a total of $2f(n)$ neighbors.

First, we consider the case where $e \leq f(n)$. In this case, $i_{\mathcal{S}_1}$ has all the other nodes in $\mathcal{S}_1$ as a neighbor. Hence, it has $e - 1$ inner edges associated with it, and this is the highest achievable. $i_{\mathcal{S}_2}$ may not have all nodes in $\mathcal{S}_2$ as neighbors, and hence has at most the same number of inner edges as $i_{\mathcal{S}_1}$. This is true for each consequent node. Hence, adding the number of inner edges of each node in both sets, we see in this case that $\mathcal{S}_1$ has more inner edges than $\mathcal{S}_2$.

Next, we consider the case where $f(n) < e \leq 2f(n)$. Suppose the number of nodes in the set is $f(n) + k$, then for $i < k$ and

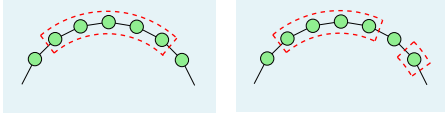


Fig. 10. On the left, we have a continuous set of 5 nodes, marked in red, as described in Lemma 5. On the right we have a set of 5 nodes that is not continuous.

$i > e - k$, i_{S_1} shares an inner edge with $f(n)$ nodes on one side and $i - 1$ and $e - i$ neighbors respectively on the other side, which is the highest possible for its position. If $k \leq i \leq e - k$, then all nodes in the set that can share an inner edge with i_{S_1} do so, which again is the highest possible number. Hence, adding the number of inner edges of each node in both sets, we see in this case that S_1 has more inner edges than S_2 .

Next, we consider the case where $2f(n) < e < n - 2f(n)$. If $i < f(n)$, then all $f(n)$ neighbors on one side share an inner edge with i_{S_1} , and all $i - 1$ neighbors on the other side also share an inner edge with i_{S_1} , which leads to i_{S_1} having the highest possible number of inner edges as it shares an inner edge with all possible nodes in its position. Hence, i_{S_2} cannot have more inner edges than i_{S_1} . Due to symmetry, this is also true for $n - f(n) < i$. Finally, if $f(n) \leq i \leq n - f(n)$, then all $2f(n)$ neighbors of i_{S_1} share an inner edge with it. Once again, i_{S_2} has at most the same number of inner edges. Hence, in this case, adding up the number of inner edges in order for both sets, we conclude that S_1 has more inner edges than S_2 .

Finally, if $e \geq n - 2f(n)$, then $\mathcal{V} \setminus \mathcal{S}$ has the same number of inner edges as $\mathcal{S}$. Hence, following the first and second cases, the continuous set has most inner edges.

From the above four cases, we conclude that the set of continuous nodes has the highest number of inner edges, and hence, the least number of incoming edges. This is shown for a simple example of $|\mathcal{S}| = 5$ nodes in Fig. 10. ■

Now, we state and prove the second main result of this paper in Theorem 2 below. This theorem utilizes Lemma 4 and Lemma 5 to calculate an upper bound for the version age of a single node in the generalized ring network.

Theorem 2: The version age of a single user in the generalized ring network scales as,

$$v_1 = O\left(\log f(n) + \frac{\sqrt{n}}{\sqrt{f(n)}}\right). \quad (81)$$

Proof: First, using Lemma 5, we write the exact lower bounds for incoming edges by counting the number of incoming edges of the sets of continuous nodes. We have three formulae, corresponding to three regions. The regions and the corresponding bounds for incoming edges are:

- 1) $j \leq f(n) : |\mathcal{E}(\mathcal{S})| \geq 2jf(n) - j(j-1)$.
- 2) $f(n) < j < n - f(n) : |\mathcal{E}(\mathcal{S})| \geq f(n)(f(n) + 1)$.
- 3) $n - f(n) \leq j : |\mathcal{E}(\mathcal{S})| \geq 2(n-j)f(n) - (n-j)(n-j-1)$.

We obtain the first bound by counting the number of inner edges for each node, which is $j - 1$, and then subtracting it from the total number of neighbors $2f(n)$. Then, the number of incoming edges for each node is $2f(n) - j - 1$. Since there are j nodes in total, the number of incoming edges into set $\mathcal{S}$ is given by $j(2f(n) - (j - 1))$. We can carry out a similar calculation to count the number of edges in the third case. In the second case, we simply calculate the number of incoming edges. The nearest neighbors on each

side has $f(n)$ incoming edges, the second nearest neighbor has $f(n) - 1$ incoming edges, and so on. Hence, the total number of incoming edges is given by $2 \times \frac{f(n)(f(n)+1)}{2} = f(n)(f(n)+1)$.

Now, the recursion can be written using Lemma 4. Following the method used to write the recursive equations in Theorem 1, we define $v_1, v_2, \dots, v_n = \frac{\lambda_e}{\lambda}$, where v_1 is the age of a single node, and each v_j is the age of a set constructed by adding a neighboring node that maximizes the version age among all one-larger sets starting from the set with age v_{j-1} . Thus, we analyze the following recursion to find v_1 , starting at v_n ,

$$v_j \leq \begin{cases} \frac{\frac{\lambda_e}{\lambda} + \frac{2jf(n)-j(j-1)}{2f(n)}v_{j+1}}{\frac{j}{n} + \frac{2jf(n)-j(j-1)}{2f(n)}}, & j \leq f(n), \\ \frac{\frac{\lambda_e}{\lambda} + \frac{f(n)+1}{2}v_{j+1}}{\frac{j}{n} + \frac{f(n)+1}{2}}, & f(n) < j < n - f(n), \\ \frac{\frac{\lambda_e}{\lambda} + \frac{2(n-j)f(n)-(n-j)(n-j-1)}{2f(n)}v_{j+1}}{\frac{j}{n} + \frac{2(n-j)f(n)-(n-j)(n-j-1)}{2f(n)}}, & n - f(n) \leq j. \end{cases} \quad (82)$$

Now, we write the complete recursion as is done in Theorem 1, and get a sum containing $n+1$ terms. We divide the sum into three parts corresponding to the terms numbered between $\{1, \dots, f(n)\}$, $\{f(n)+1, \dots, n-f(n)-1\}$ and $\{n-f(n), \dots, n+1\}$. We call the sum of terms in each of these parts X , Y , and Z , respectively.

The upper bound for the sum of terms in X can be found as follows,

$$X = \frac{\lambda_e}{\lambda} \left(\frac{1}{1 + \frac{1}{n}} \right) \left(1 + \sum_{i=1}^{f(n)} \prod_{j=1}^i \frac{\frac{2jf(n)-j(j-1)}{2f(n)}}{\frac{j+1}{n} + \frac{2(j+1)f(n)-j(j+1)}{2f(n)}} \right) \quad (83)$$

$$\leq \frac{\lambda_e}{\lambda} \left(1 + \sum_{i=1}^{f(n)} \prod_{j=1}^i \frac{\frac{2jf(n)-j(j-1)}{2f(n)}}{\frac{2(j+1)f(n)-j(j+1)}{2f(n)}} \right) \quad (84)$$

$$= \frac{\lambda_e}{\lambda} \left(1 + \sum_{i=1}^{f(n)} \prod_{j=1}^i \frac{(2f(n) - (j-1))}{2f(n) - j} \cdot \frac{j}{j+1} \right) \quad (85)$$

$$= \frac{\lambda_e}{\lambda} \left(1 + \sum_{i=1}^{f(n)} \frac{1}{i+1} \frac{2f(n)}{2f(n) - i} \right) \quad (86)$$

$$\leq \frac{\lambda_e}{\lambda} \left(1 + \sum_{i=1}^{f(n)} \frac{1}{i} \frac{2f(n)}{2f(n) - i} \right) \quad (87)$$

$$\leq \frac{\lambda_e}{\lambda} \left(2 + \sum_{l=1}^{2f(n)-1} \frac{1}{l} \right) \quad (88)$$

$$\leq \frac{\lambda_e}{\lambda} (2 + \log 2 + \log f(n) + \gamma), \quad (89)$$

where $\gamma \approx 0.577$ is the Euler-Mascheroni constant.

The upper bound for the sum of terms in Y can be found as follows,

$$Y = K \frac{\lambda_e}{\lambda} \left(\frac{1}{\frac{f(n)+1}{n} + \frac{f(n)+1}{2}} \right) \left(\frac{f(n)+1}{2} \right)$$

$$\times \left(1 + \sum_{i=f(n)+2}^{n-f(n)} \prod_{j=f(n)+2}^i \frac{1}{1 + \frac{2j}{n(f(n)+1)}} \right), \quad (90)$$

where

$$K = \prod_{j=1}^{f(n)} \frac{\frac{2jf(n)-j(j-1)}{2f(n)}}{\frac{j+1}{n} + \frac{2(j+1)f(n)-j(j+1)}{2f(n)}} \quad (91)$$

$$\leq \prod_{j=1}^{f(n)} \frac{j}{j+1} \frac{2f(n)-(j-1)}{2f(n)-j} \quad (92)$$

$$= \frac{2}{f(n)+1}. \quad (93)$$

Substituting this into (90), we get,

$$Y \leq \frac{\lambda_e}{\lambda} \frac{2}{f(n)+1} \frac{2}{f(n)+1} \frac{f(n)+1}{2} \times \left(1 + \sum_{i=f(n)+2}^{n-f(n)} \prod_{j=f(n)+2}^i \frac{1}{1 + \frac{2j}{n(f(n)+1)}} \right) \quad (94)$$

$$= \frac{\lambda_e}{\lambda} \frac{2}{f(n)+1} \frac{2}{f(n)+1} \frac{f(n)+1}{2} \left(1 + \prod_{j=1}^{f(n)+2} \left(1 + \frac{2j}{n(f(n)+1)} \right) \sum_{i=f(n)+2}^{n-f(n)} \prod_{j=1}^i \frac{1}{1 + \frac{2j}{n(f(n)+1)}} \right) \quad (95)$$

$$\leq \frac{\lambda_e}{\lambda} \frac{2}{f(n)+1} \left(1 + \prod_{j=1}^{f(n)+2} \left(1 + \frac{2j}{n(f(n)+1)} \right) \sum_{i=1}^{n-f(n)} \prod_{j=1}^i \frac{1}{1 + \frac{2j}{n(f(n)+1)}} \right). \quad (96)$$

Next, we take log of the i th product term in the sum of products term,

$$a_i = \prod_{j=1}^i \frac{1}{1 + \frac{2j}{n(f(n)+1)}}, \quad (97)$$

which gives us,

$$-\log a_i = \sum_{j=1}^i \frac{2j}{n(f(n)+1)} \quad (98)$$

$$= \frac{i(i+1)}{n(f(n)+1)}, \quad (99)$$

which further gives us,

$$a_i = e^{-\frac{i(i+1)}{n(f(n)+1)}}. \quad (100)$$

In a similar way, we can conclude that,

$$\prod_{j=1}^{f(n)+2} \left(1 + \frac{2j}{n(f(n)+1)} \right) \approx e^{\frac{f(n)}{n}}. \quad (101)$$

Substituting this into (96), gives,

$$Y \leq \frac{\lambda_e}{\lambda} \frac{2}{f(n)+1} \left(1 + e^{\frac{f(n)}{n}} \sum_{i=1}^{n-f(n)} e^{-\frac{i(i+1)}{n(f(n)+1)}} \right) \quad (102)$$

$$\leq \frac{\lambda_e}{\lambda} \frac{2}{f(n)+1} \left(1 + e^{\frac{f(n)}{n}} \sum_{i=1}^n e^{-\frac{i(i+1)}{n(f(n)+1)}} \right) \quad (103)$$

$$\leq \frac{\lambda_e}{\lambda} \frac{2}{f(n)+1} \left(1 + e^{\frac{f(n)}{n}} \sum_{i=1}^n e^{-\frac{i^2}{n(f(n)+1)}} \right). \quad (104)$$

Now, if $f(n) = o(n)$, then $e^{\frac{f(n)}{n}} \rightarrow 1$, and if $f(n) = \theta(n)$, then $e^{\frac{f(n)}{n}} = C$, where C is a constant. Next, we convert the Riemann sum associated with the summation term in (104) into a definite integral, and find its exact value. In order to do so, we use step size $\frac{1}{\sqrt{n(f(n)+1)}}$,

$$\frac{1}{\sqrt{n(f(n)+1)}} \sum_{i=1}^n e^{-\frac{i^2}{n(f(n)+1)}} = \int_0^\infty e^{-t^2} dt = \frac{\sqrt{\pi}}{2}, \quad (105)$$

as $n \rightarrow \infty$ and $f(n) = o(n)$, and the step size tending to 0. On the other hand, if $f(n) = \theta(n)$, then the above integral has lower limit 0 and upper limit a constant, thus resulting in,

$$\frac{1}{\sqrt{n(f(n)+1)}} \sum_{i=1}^n e^{-\frac{i^2}{n(f(n)+1)}} = L, \quad (106)$$

where L is a constant. Using this, we obtain,

$$\sum_{i=1}^n e^{-\frac{i^2}{n(f(n)+1)}} = L \sqrt{n(f(n)+1)} \quad (107)$$

$$\leq \frac{\sqrt{\pi}}{2} \sqrt{n(f(n)+1)}, \quad (108)$$

when $f(n) = \theta(n)$, and

$$\sum_{i=1}^n e^{-\frac{i^2}{n(f(n)+1)}} = \frac{\sqrt{\pi}}{2} \sqrt{n(f(n)+1)}, \quad (109)$$

when $f(n) = o(n)$. Substituting this back in (104), we get,

$$Y \leq \frac{\lambda_e}{\lambda} \frac{2}{f(n)+1} \left(1 + \frac{\sqrt{\pi}}{2} \sqrt{n(f(n)+1)} \right) \quad (110)$$

$$\approx \sqrt{\pi} \frac{\lambda_e}{\lambda} \frac{\sqrt{n}}{(f(n)+1)^{\frac{1}{2}}} \quad (111)$$

$$\leq \sqrt{\pi} \frac{\lambda_e}{\lambda} \frac{\sqrt{n}}{\sqrt{f(n)}}. \quad (112)$$

The upper bound for the sum of terms in Z can be found as follows,

$$Z = 2 \frac{\lambda_e}{\lambda} + \frac{\lambda_e}{\lambda} \frac{f(n)+1}{2} \prod_{j=1}^{f(n)+1} \frac{\frac{2jf(n)-j(j-1)}{f(n)}}{\frac{j+1}{n} + \frac{2(j+1)f(n)-j(j+1)}{f(n)}} \times \prod_{j=f(n)+2}^{n-f(n)} \frac{1}{1 + \frac{j}{n(f(n)+1)}} \frac{1}{1 + \frac{n-f(n)}{n}} \times \left(1 + \sum_{i=n-f(n)}^{n-2} \prod_{j=n-f(n)}^i \frac{\frac{(n-j)f(n)-(n-j)(n-j-1)}{f(n)}}{\frac{j+1}{n} + \frac{(n-j-1)f(n)-(n-j-1)(n-j-2)}{f(n)}} \right) \quad (113)$$

$$\leq 3 \frac{\lambda_e}{\lambda} + \frac{\lambda_e}{\lambda} \sum_{i=n-f(n)}^{n-2} \prod_{j=n-f(n)}^i \frac{n-j}{n-j-1} \frac{f(n)-(n-j-1)}{f(n)-(n-j-2)} \quad (114)$$

$$\leq 3 \frac{\lambda_e}{\lambda} + \frac{\lambda_e}{\lambda} \sum_{i=n-f(n)}^{n-2} \frac{1}{i} \quad (115)$$

$$\leq 3\frac{\lambda_e}{\lambda} + \frac{\lambda_e}{\lambda} \log f(n). \quad (116)$$

In order to go from (113) to (114), we approximate the first product in the first line of (113) following the calculation of K done earlier. We drop the second product in the first line since it is smaller than 1, and do the regular upper bounding in the second line.

Finally, summing the terms in the three ranges, we obtain,

$$v_1 \leq X + Y + Z \quad (117)$$

$$\leq \frac{\lambda_e}{\lambda} (2 + \log 2 + \log f(n) + \gamma) + \sqrt{\pi} \frac{\lambda_e}{\lambda} \frac{\sqrt{n}}{\sqrt{f(n)}} + 3\frac{\lambda_e}{\lambda} + \frac{\lambda_e}{\lambda} \log f(n) \quad (118)$$

$$= \frac{\lambda_e}{\lambda} (5 + \log 2 + 2 \log f(n) + \gamma) + \sqrt{\pi} \frac{\lambda_e}{\lambda} \frac{\sqrt{n}}{\sqrt{f(n)}}, \quad (119)$$

giving us the desired result. ■

A. Special Cases of the Generalized Ring Network

In this subsection, we look at some special cases of the generalized ring networks, some of which have been analyzed in earlier papers and some of which have remained open problems.

1) *Bi-Directional Ring*: We consider the bi-directional ring network in [22], [34], and it is easy to see that $f(n) = 1$. Hence, in this case, we have

$$v_1 \leq \sqrt{\pi} \frac{\lambda_e}{\lambda} \sqrt{n} \quad (120)$$

$$\approx 1.711 \frac{\lambda_e}{\lambda} \sqrt{n}, \quad (121)$$

which is in accordance with the result proved in [22].

2) *Fully-Connected Network*: In this case, it is easy to see that $f(n) = \frac{n-1}{2}$. Substituting this in our equation, we get,

$$v_1 \leq \frac{\lambda_e}{\lambda} (2 + \log(n-1)), \quad (122)$$

which is in accordance with [9], since the exact version age scaling is $\theta(\log n)$.

3) *Ring With Fixed Number of Connected Neighbors*: Suppose each node in the ring has $2d$ neighbors, where d is a constant. Then, we can again conclude that,

$$v_1 \leq \sqrt{\pi} \frac{\lambda_e}{\lambda} \frac{\sqrt{n}}{d^{\frac{1}{2}}}. \quad (123)$$

Hence, in this case, $v_1 = O(\sqrt{n})$. Thus, being connected to one neighbor on each side or a constant number d neighbors on each side yields the same age scaling on a ring.

4) $f(n) = n^\alpha$, $\alpha < 1$: This set of functions cover functions over a vast range between $f(n) = 1$ and $f(n) = \theta(n)$. We see that the version age in this case scales as a polynomial as,

$$v_1 \leq \sqrt{\pi} \frac{\lambda_e}{\lambda} n^{\frac{1-\alpha}{2}}. \quad (124)$$

For instance, if $\alpha = \frac{1}{2}$, i.e., each node is connected to $\sqrt{n}$ neighbors on each side on a ring, then the age of a node increases slower than in the case of a regular ring, as $n^{\frac{1}{4}}$.

5) $\frac{n}{\log^2 n} \leq f(n) < n$: The version age of a single node grows logarithmically for this set of functions. From [9], we know that for a fully-connected network, the version age scales as $\theta(\log n)$. Since the networks for these $f(n)$ have smaller number of connections, the version age of a single node in these networks is larger. Hence, a lower bound for the version age is also $\log n$. Hence, in this case, we can conclude that the version age of a single user scales as $\theta(\log n)$.

Remark 2: In Subsection V-A, we consider the version age scaling for $f(n) = n^\alpha$, and find that the scaling is $O\left(n^{\frac{1-\alpha}{2}}\right)$ as $n \rightarrow \infty$. Hence, the rational function dominates the logarithm in (81). However, as α increases, the rational function grows increasingly slowly and dominates the logarithm only at very high values of n . In Table II, we show the number of nodes in the network required for the rational term to become 10 times the size of the logarithm. Clearly, this is very high when $\alpha \geq 0.6$. Hence, for all practical purposes, if $f(n) \geq n^{0.6}$, then the version age scaling of a single node in the generalized ring network is logarithmic.

VI. VERSION AGE IN A m -DIMENSIONAL HYPERCUBE

In this section, we find an upper bound for the version age of a single node in a unit m -dimensional hypercube, described in Section II (see Fig. 3). Once again, due to the symmetry of the network, the version age of each node in the unit hypercube network will be the same. In order to follow the proof of Theorem 1, and consequently Lemma 3, to find an upper bound for the version age of a single node, we need to find a lower bound for the number of incoming edges. The exact upper bound for the number of inner edges of an induced subgraph of fixed size in a unit hypercube was found in [35]. Let $h(i)$ be the sum of the binary digits of an integer i . Then, the maximum number of inner edges of an induced subgraph $\mathcal{S}$ of size j of the hypercube is given by,

$$|\bar{\mathcal{E}}(\mathcal{S})|_{\max} = \sum_{i=0}^{j-1} h(i). \quad (125)$$

Hence, we can find a lower bound for the number of incoming edges using this. We can see that the total number of edges emanating from each node is m . Hence, the total number of edges emanating from set $\mathcal{S}$ is given by,

$$|\mathcal{E}(\mathcal{S})| = mj - 2 \sum_{i=0}^{j-1} h(i). \quad (126)$$

Now, we cannot find the exact usable formula for $h(i)$ for each i . Hence, we need to find an upper bound for the value of $h(i)$. This upper bound of $h(i)$ was found in [36], and is given for $j > 1$ as,

$$\sum_{i=1}^{j-1} h(i) \leq \sum_{i=1}^{\lceil \log_2(j-1) \rceil} \frac{1}{2} \left\lfloor \frac{j-1}{2^i} + 1 \right\rfloor 2^i. \quad (127)$$

Now, we further upper bound it as follows,

$$\sum_{i=1}^{j-1} h(i) \leq \left(\sum_{i=1}^{\lceil \log_2(j-1) \rceil} \frac{1}{2} \left\lfloor \frac{j-1}{2^i} \right\rfloor 2^i \right) + j \quad (128)$$

TABLE II

IN (119), THERE ARE TWO TERMS IN THE UPPER BOUND: $\log f(n)$ AND $\sqrt{n}/\sqrt{f(n)}$. IN THIS TABLE, WE LIST THE VALUES OF n AT WHICH THE RATIONAL FUNCTION TERM BEGINS TO DOMINATE THE $\log$

α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Order of v_1	$n^{0.45}$	$n^{0.4}$	$n^{0.35}$	$n^{0.3}$	$n^{0.25}$	$n^{0.2}$	$n^{0.15}$	$n^{0.1}$	$n^{0.05}$
n	0	942	24180	955318	1.22×10^8	1.64×10^{11}	3.33×10^{16}	3.9×10^{27}	2.74×10^{63}

$$\leq \left(\frac{1}{2} \sum_{i=1}^{\lceil \log_2(j-1) \rceil} \left(\frac{j-1}{2^i} \right) 2^i \right) + j \quad (129)$$

$$= \left(\frac{1}{2} \sum_{i=1}^{\lceil \log_2(j-1) \rceil} (j-1) \right) + j \quad (130)$$

$$\leq \frac{1}{2} j \lceil \log_2(j) \rceil + j \quad (131)$$

$$\approx \frac{1}{2} j \lceil \log_2(j) \rceil. \quad (132)$$

We can suppress the linear term when compared to the first term as $j \rightarrow \infty$. Hence, we conclude that $|\mathcal{E}(\mathcal{S})| \geq mj - j \lceil \log_2(j) \rceil$. However, when $j > 2^{m-1}$, the bound simplifies to $|\mathcal{E}(\mathcal{S})| \geq 0$, which is true, but impacts our recursive equations. We use the bound $|\mathcal{E}(\mathcal{S})| \geq m(n-j) - (n-j) \lceil \log_2(n-j) \rceil$ in this case using the fact that the set with $n-j$ nodes that has the least number of incoming edges has a complement set of j nodes that has the least number of incoming edges among all j sized sets. We now have the following lemma, which makes use of these lower bounds for the number of incoming edges to find upper bounds for the version age of a subset $\mathcal{S}$ of nodes in the unit hypercube network.

Lemma 6: Suppose $\mathcal{S}$ is an induced subgraph of the hypercube graph and $|\mathcal{S}| = j$, $1 \leq j \leq 2^{m-1}$, then we have,

$$v_{\mathcal{S}} \leq \frac{\frac{\lambda_e}{\lambda} + \frac{j(m - \lceil \log_2(j) \rceil)}{m}}{\frac{j}{2^m} + \frac{j(m - \lceil \log_2(j) \rceil)}{m}} \max_{i \in \mathcal{S}} v_{\mathcal{S} \cup \{i\}}, \quad (133)$$

and if $j > 2^{m-1}$, then we have,

$$v_{\mathcal{S}} \leq \frac{\frac{\lambda_e}{\lambda} + \frac{(n-j)(m - \lceil \log_2(n-j) \rceil)}{m}}{\frac{j}{2^m} + \frac{(n-j)(m - \lceil \log_2(n-j) \rceil)}{m}} \max_{i \in \mathcal{S}} v_{\mathcal{S} \cup \{i\}}. \quad (134)$$

The proof of this lemma follows in a similar way to the proof of Lemma 3, hence is omitted.

Next, we state and prove the third main result of this paper in Theorem 3 below, which makes use of Lemma 6 to find an upper bound for the version age of a single node in the unit hypercube network.

Theorem 3: The version age of a single user in the unit hypercube graph grows as $v_1 = O(\log n \log \log n)$.

Proof: The recursion can be written using Lemma 6. Following the method used in Theorem 1, we again define $v_1, v_2, \dots, v_n = \frac{\lambda_e}{\lambda}$, where v_1 is the age of a single node, and each v_j is the age of a set constructed by adding a neighboring node that maximizes the version age among all one-larger sets starting from the set with age v_{j-1} . Thus, we analyze

the following recursion to find v_1 , starting at v_n ,

$$v_j \leq \begin{cases} \frac{\frac{\lambda_e}{\lambda} + \frac{j(m - \lceil \log_2(j) \rceil)}{m}}{\frac{j}{2^m} + \frac{j(m - \lceil \log_2(j) \rceil)}{m}} v_j, & j \leq 2^{m-1}, \\ \frac{\frac{\lambda_e}{\lambda} + \frac{(n-j)(m - \lceil \log_2(n-j) \rceil)}{m}}{\frac{j}{2^m} + \frac{(n-j)(m - \lceil \log_2(n-j) \rceil)}{m}} v_j, & 2^{m-1} < j. \end{cases} \quad (135)$$

Now, we write the complete recursion as is done in Theorem 1, and get a sum containing $n+1$ terms. We divide the sum into two parts corresponding to the terms numbered between $\{1, \dots, 2^{m-1}\}$ and $\{2^{m-1} + 1, \dots, n+1\}$. We call the sum of terms in each of these parts X and Y , respectively. We first find an upper bound for the sum of terms X ,

$$X \leq \frac{\lambda_e}{\lambda} \left(\frac{1}{1 + \frac{1}{2^m}} \right) \times \left(1 + \sum_{i=1}^{2^{m-1}-1} \prod_{j=1}^i \frac{\frac{j(m - \lceil \log_2(j) \rceil)}{m}}{\frac{j+1}{2^m} + \frac{(j+1)(m - \lceil \log_2(j+1) \rceil)}{m}} \right) \quad (136)$$

$$\leq \frac{\lambda_e}{\lambda} \left(1 + \sum_{i=1}^{2^{m-1}-1} \prod_{j=1}^i \frac{\frac{j(m - \lceil \log_2(j) \rceil)}{m}}{\frac{j+1}{2^m} + \frac{(j+1)(m - \lceil \log_2(j+1) \rceil)}{m}} \right) \quad (137)$$

$$\leq \frac{\lambda_e}{\lambda} \left(1 + \sum_{i=1}^{2^{m-1}-1} \prod_{j=1}^i \frac{\frac{j(m - \lceil \log_2(j) \rceil)}{m}}{\frac{(j+1)(m - \lceil \log_2(j+1) \rceil)}{m}} \right) \quad (138)$$

$$\leq \frac{\lambda_e}{\lambda} \left(2 + \sum_{i=1}^{2^{m-1}-2} \prod_{j=1}^i \frac{j}{j+1} \frac{m - \lceil \log_2(j) \rceil}{m - \lceil \log_2(j+1) \rceil} \right) \quad (139)$$

$$= \frac{\lambda_e}{\lambda} \left(2 + \sum_{i=1}^{2^{m-1}-2} \frac{1}{i+1} \frac{m}{m - \lceil \log_2(i+1) \rceil} \right) \quad (140)$$

$$= \frac{\lambda_e}{\lambda} \left(2 + \sum_{i=2}^{2^{m-1}-1} \frac{1}{i} \frac{m}{m - \lceil \log_2(i) \rceil} \right) \quad (141)$$

$$\leq \frac{\lambda_e}{\lambda} \left(2 + \sum_{l=1}^{m-2} \sum_{i=2^l}^{2^{l+1}-1} \frac{1}{i} \frac{m}{m-l} \right) \quad (142)$$

$$= \frac{\lambda_e}{\lambda} \left(2 + \sum_{l=1}^{m-2} \frac{m}{m-l} \sum_{i=2^l}^{2^{l+1}-1} \frac{1}{i} \right) \quad (143)$$

$$= \frac{\lambda_e}{\lambda} \left(2 + \log 2 \sum_{l=1}^{m-2} \frac{m}{m-l} (\log(2^{l+1}-1) - l) \right) \quad (144)$$

$$\leq \frac{\lambda_e}{\lambda} \left(2 + \log 2 \sum_{l=1}^{m-2} \frac{m}{m-l} ((l+1) - l) \right) \quad (145)$$

$$\leq \frac{\lambda_e}{\lambda} \left(2 + \log 2 \sum_{l=1}^{m-2} \frac{m}{m-l} \right) \quad (146)$$

$$\leq \frac{\lambda_e}{\lambda} (2 + \log(2)m \log m). \quad (147)$$

Next, we find an upper bound for the sum of terms Y ,

$$\begin{aligned} Y &\leq \frac{\lambda_e}{\lambda} \left(\prod_{j=1}^{2^{m-1}-1} \frac{\frac{j(m-\lceil \log_2(j) \rceil)}{m}}{\frac{j+1}{2^m} + \frac{(j+1)(m-\lceil \log_2(j+1) \rceil)}{m}} \right) \\ &\quad \times \frac{2^{m-1}}{m} \frac{1}{\frac{2^{m-1}+1}{2^m} + \frac{2^{m-1}+1}{m}} \\ &\quad \times \left(1 + \sum_{i=2^{m-1}+1}^{2^m-1} \prod_{j=2^{m-1}+1}^i \frac{\frac{(n-j)(m-\lceil \log_2(n-j) \rceil)}{m}}{\frac{j+1}{2^m} + \frac{(n-j-1)(m-\lceil \log_2(n-j-1) \rceil)}{m}} \right) \end{aligned} \quad (148)$$

$$\begin{aligned} &\leq \frac{\lambda_e}{\lambda} \left(\prod_{j=1}^{2^{m-1}-1} \frac{\frac{j(m-\lceil \log_2(j) \rceil)}{m}}{\frac{(j+1)(m-\lceil \log_2(j+1) \rceil)}{m}} \right) \\ &\quad \left(1 + \sum_{i=2^{m-1}+1}^{2^m-1} \prod_{j=2^{m-1}+1}^i \frac{\frac{(n-j)(m-\lceil \log_2(n-j) \rceil)}{m}}{\frac{(n-j-1)(m-\lceil \log_2(n-j-1) \rceil)}{m}} \right) \end{aligned} \quad (149)$$

$$\begin{aligned} &\leq \frac{\lambda_e}{\lambda} \frac{m}{2^{m-1}} \left(1 + \sum_{i=2^{m-1}+1}^{2^m-1} \prod_{j=2^{m-1}+1}^i \frac{n-j}{n-j-1} \frac{m-\lceil \log_2(n-j) \rceil}{m-\lceil \log_2(n-j-1) \rceil} \right) \end{aligned} \quad (150)$$

$$\begin{aligned} &= \frac{\lambda_e}{\lambda} \left(1 + \frac{m}{2^{m-1}} \sum_{i=2^{m-1}+1}^{2^m-1} \frac{n-i}{2^{m-1}-2} (m-\lceil \log_2(n-i) \rceil) \right) \end{aligned} \quad (151)$$

$$\begin{aligned} &= \frac{\lambda_e}{\lambda} \left(1 + \frac{m}{(2^{m-1})(2^{m-1}-2)} \right. \\ &\quad \left. \times \sum_{l=1}^{2^{m-1}-1} (2^{m-1}-l)(m-\lceil \log_2(2^{m-1}-l) \rceil) \right) \end{aligned} \quad (152)$$

$$\begin{aligned} &= \frac{\lambda_e}{\lambda} \left(1 + \frac{m}{(2^{m-1})(2^{m-1}-2)} \sum_{a=1}^{2^{m-1}-1} a(m-\lceil \log_2(a) \rceil) \right) \end{aligned} \quad (153)$$

$$\begin{aligned} &\leq \frac{\lambda_e}{\lambda} \left(1 + \frac{m}{(2^{m-1})(2^{m-1}-2)} \right. \\ &\quad \left. \times \left(3m + \sum_{b=1}^{m-2} \sum_{a=2^{b+1}}^{2^{b+1}} a(m-\lceil \log_2(a) \rceil) \right) \right) \end{aligned} \quad (154)$$

$$\begin{aligned} &= \frac{\lambda_e}{\lambda} \left(1 + \frac{m}{(2^{m-1})(2^{m-1}-2)} \right. \\ &\quad \left. \times \left(3m + \sum_{b=1}^{m-2} (m-(b+1)) \sum_{a=2^{b+1}}^{2^{b+1}} a \right) \right) \end{aligned} \quad (155)$$

$$\begin{aligned} &\approx \frac{\lambda_e}{\lambda} \left(1 + \frac{m}{(2^{m-1})(2^{m-1}-2)} \right. \\ &\quad \left. \times \left(3m + 3 \sum_{b=1}^{m-2} (m-(b+1)) 2^{2b-1} \right) \right) \end{aligned} \quad (156)$$

$$\begin{aligned} &\leq \frac{\lambda_e}{\lambda} \left(1 + \frac{m}{(2^{m-1})(2^{m-1}-2)} \right. \\ &\quad \left. \times \left(3m + 3 \sum_{t=1}^{m-2} t 2^{2m-2t-2} \right) \right) \end{aligned} \quad (157)$$

$$\begin{aligned} &= \frac{\lambda_e}{\lambda} \left(1 + \frac{m}{(2^{m-1})(2^{m-1}-2)} \right. \\ &\quad \left. \times \left(3m + 3 \cdot 2^{2m} \sum_{t=1}^{m-2} \frac{t}{2^{2t+2}} \right) \right) \end{aligned} \quad (158)$$

$$\begin{aligned} &= \frac{\lambda_e}{\lambda} \left(1 + \frac{m}{(2^{m-1})(2^{m-1}-2)} \right. \\ &\quad \left. \times \left(3m + 3 \cdot 2^{2m} \left(\frac{1}{9} \frac{1}{4^{m-1}} (8 + 4^m - 12m) \right) \right) \right) \end{aligned} \quad (159)$$

$$\approx \frac{\lambda_e}{\lambda} \left(1 + \frac{16}{3} m \right). \quad (160)$$

Finally, we use X and Y to upper bound v_1 , and also substitute $m = \log_2 n$,

$$v_1 \leq X + Y \quad (161)$$

$$\leq \frac{\lambda_e}{\lambda} \left(3 + \frac{16}{3} m + \log(2)m \log m \right) \quad (162)$$

$$\begin{aligned} &= \frac{\lambda_e}{\lambda} \left(3 + \frac{16}{3} \log_2(n) + \log(2) \log_2(n) \log_2 \log_2(n) \right) \end{aligned} \quad (163)$$

$$\leq \frac{\lambda_e}{\lambda} C \log n \log \log n, \quad (164)$$

where C is a constant. ■

VII. DISCUSSION AND REMARKS

In this section, we discuss implications and possible extensions of our results via a sequence of remarks. We start with three general remarks, and then present our remarks grouped by each network structure in the following subsections.

A. General Remarks

Remark 3: We note that there are several works in the literature which consider gossip networks with the version age metric in different settings. These include: adversarial models such as timestamping [23] and jamming [24], hierarchical structures such as a network with a community structure [22], extended gossip algorithms such as file slicing and network coding [37], and opportunistic gossiping in multiple access and decentralized networks [27], [28]. All these papers use either the ring network or the fully-connected network as the network model. With the results of the present paper, extensions of the results in the existing literature to the generalized networks, such as grids, generalized rings and hypercubes, are possible.

Remark 4: We analyze the networks in the wrapped-around setting due to multiple reasons. The first reason lies in

its inherent symmetry and the simplification it provides for bounding calculations. The other reasons are specific to the networks. In the grid network, boundary effects are negligible because the number of wrapped-around edges is $\theta(\sqrt{n})$, while the total number of edges is $\theta(n)$. Additionally, it has been shown in [24] that in a one-dimensional line network, the average version age of a node can be both upper and lower bounded by multiples of the average version age of a node in a ring network (note that a ring network is a wrapped-around version of a line network). A similar argument applies here: The version age of a grid without wrapped-around edges can be bounded above and below by multiples of the version age in a grid with wrapped-around edges. This is further discussed in Remark 7. The unit hypercube graph without wrapped-around edges can be viewed as a unit hypercube graph with wrapped-around edges, where each edge operates at half the rate (since each node is connected to its neighbor twice, once with the inner edge and once with a wrapped-around edge), due to the independence of thinned Poisson processes. This allows us to apply the same calculations as done in Section VII and derive a similar upper bound. Finally, for general hypercube networks, while we have a conjecture, a reasoning approach similar to that of the grid network applies.

Remark 5: Here, we analyze static gossiping protocols, which are known for their simplicity, which is a major advantage, as they are used in large decentralized networks, such as internet of things networks. Static gossip algorithms are not structured, they do not involve or use feedback (i.e., not sequentially optimized), the forwarding protocol is time- and observation-independent (i.e., not dynamically evolving), they do not require coordination among the nodes or a global scheduler. On the other hand, a dynamic gossiping protocol, which affords nodes sequential decision-making ability, would allow nodes to make decisions dynamically as to when and which neighbors to send updates. While a dynamic decision-making framework based gossiping algorithm is outside the scope of this paper, it constitutes an excellent future direction in this area of research. It would be interesting to analyze how such frameworks affect the long-term average version age of a gossiping network, and compare it with that of simple gossiping protocols such as the push-only or the push-pull protocol.

B. Remarks on the Grid Network

Remark 6: In (51), $\beta = (\frac{2}{3})^{2/3} \Gamma(\frac{1}{3})$, where $\Gamma(\cdot)$ is the gamma function. Thus, $\beta' = e^{-\frac{2}{3}} \times e^{\pi^2/48} \times (\frac{2}{3})^{2/3} \Gamma(\frac{1}{3}) \approx 1.882$. Hence, if $m = o(k^2)$, then as $n \rightarrow \infty$, the second term in (71) dominates, and the version age of a single node is upper bounded as,

$$v_1 \leq 3.764 \frac{\lambda_e}{\lambda} n^{\frac{1}{3}}. \quad (165)$$

On the other hand, if $k = o(\sqrt{m})$, then the third term dominates in (71). The constant multiplier depends on the relation between k and m in this case.

Remark 7: In a square two dimensional grid, $k = m$, and hence the version age scaling is given by $v_1 = O(n^{\frac{1}{3}})$. Moreover, if a square two dimensional grid is such that there are no wrap-around edges, then the version age still has the same order. This is because, the spiral still is the best up to a point, and then the next best set makes use of three sides

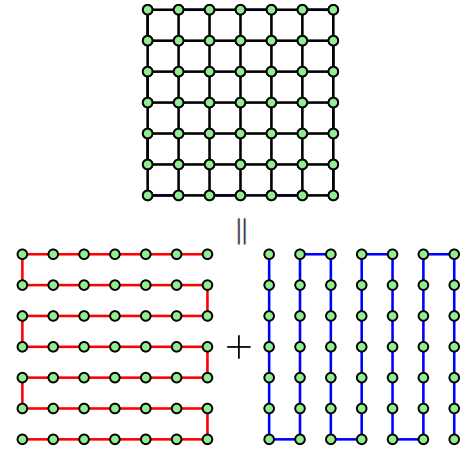


Fig. 11. The grid can be thought of as two conjoined line networks, which reduces the version age from $O(n^{\frac{1}{2}})$ to $O(n^{\frac{1}{3}})$.

of the grid, leaving only one side for incoming edges. In this new case, the number of incoming edges is lower bounded by $\sqrt{n}$. Hence, $|\mathcal{E}(\mathcal{S})|$ is bounded below by $\sqrt{j}$, until we reach the inverse spiral set. Thus, the version age still is bounded by $Kn^{\frac{1}{3}}$. Therefore, the order does not change in this case. Hypothetically, this is because the deletion of the wrapped around edges does not change the order of the diameter of the network, which is still $\theta(\sqrt{n})$.

Remark 8: It is well known that conjoining two networks disseminating the same information speeds up the information diffusion [38]. For example, suppose that there is a group of friends living in the same city, who meet each other regularly. They want to be up-to-date about the happenings in each other's lives. Then, we see that they will receive updates about their friends faster if they were connected to each other on a social media platform and also met each other in-person, rather than doing just one of the two. This is an interesting perspective with which we can also look at the grid network as a conjoined network of two lines (i.e., two rings) as shown in Fig. 11, which improves the version age from $O(n^{\frac{1}{2}})$ to $O(n^{\frac{1}{3}})$. In Fig. 11, we view the grid in black as a conjoined network of a red line and a blue line.

Remark 9: Reference [22] considers gossip networks with a community structure. It considers a system where a source node receives an update and disseminates it to cluster heads, which in turn disseminate this information to cluster nodes under them. It is shown in [22] that having a community structure improves the version age for several community structures. These community (i.e., cluster) structures include the disconnected network, the ring network and the fully-connected network. Following Section III, we can upper bound the recursion found in [22, Thm. 1] as,

$$v_S \leq \frac{\lambda_e + \lambda_c(\mathcal{S})v_c}{\lambda_c(\mathcal{S}) + |\mathcal{N}(\mathcal{S})|} \min_{i \in \mathcal{N}(\mathcal{S})} \lambda_i(\mathcal{S}) + \frac{|\mathcal{N}(\mathcal{S})| \min_{i \in \mathcal{N}(\mathcal{S})} \lambda_i(\mathcal{S}) \max_{i \in \mathcal{N}(\mathcal{S})} v_{S \cup \{i\}}}{\lambda_c(\mathcal{S}) + |\mathcal{N}(\mathcal{S})| \min_{i \in \mathcal{N}(\mathcal{S})} \lambda_i(\mathcal{S})}, \quad (166)$$

where there are x communities each with y nodes such that $n = xy$ and $v_c = x\lambda_e/\lambda_s$. Then, assuming $\lambda_c = \lambda$ and that clusters are square grid networks as we have defined in this paper, we can upper bound v_1 in one cluster of the community as,

$$v_1 \leq v_c(1 - a_y) + 2\beta' \frac{\lambda_e}{\lambda} y^{\frac{1}{3}}, \quad (167)$$

where a_y is as defined in (43). Now, as $n \rightarrow \infty$, $a_y \rightarrow 0$. Hence, we have,

$$v_1 \leq x \frac{\lambda_e}{\lambda_s} + 2\beta' \frac{\lambda_e}{\lambda} y^{\frac{1}{3}}. \quad (168)$$

Choosing $x = n^{\frac{1}{4}}$ and $y = n^{\frac{3}{4}}$, by adding community structure, we can improve the version age from $O(n^{\frac{1}{3}})$ to $O(n^{\frac{1}{4}})$.

C. Remarks on the Generalized Ring

Remark 10: Extremal animals have been a topic of study in graph theory for a long time [32]. These are connected subgraphs with minimum or maximum number of neighboring nodes, edges or faces in a graph. In this context, Lemma 5 finds the minimal edge animal for the generalized ring network.

Remark 11: In Section IV, it was shown that a two-dimensional grid has version age scaling of $O(n^{\frac{1}{3}})$. Each node in the grid network has 4 neighbors. However, in order to achieve a version age scaling of $O(n^{\frac{1}{3}})$ in the generalized ring network, we need $f(n) = n^{\frac{1}{3}}$, i.e. each node needs $2n^{\frac{1}{3}}$ neighbors. One way to explain this difference in the requirement for connectivity to achieve the same version age scaling is the following: According to Remark 8, we can view the grid as a ring network with n connections which are not local in nature. Hence, although the number of connections in a grid network is much fewer compared to the generalized ring network, the version age scaling is the same. This shows that the geometry of a network can affect the version age significantly, and having few connections between nodes far away is better than having relatively dense connections which are local in nature.

D. Remarks on the Unit Hypercube

Remark 12: We note that the version age scaling in this case is very close to the fully-connected network. In order for the version age of a generalized ring network to have the same scaling, we need $f(n) = \frac{n}{(\log n \log \log n)^2}$ connections on each side. This graph has diameter $(\log n \log \log n)^2$. This is much higher than the diameter of the unit hypercube, which is $\log n$. Moreover, each node in the generalized ring network has $\frac{2n}{(\log n \log \log n)^2}$ neighbors. This is very large when compared to a node in the unit hypercube network, where each node has $\log_2 n$ neighbors. Hence, we notice that the geometry of a graph is very important while determining the scaling of the average version age.

Remark 13: In Fig. 15, the simulated version age of the unit hypercube follows the logarithm closely, when compared to the upper bound found in Section VI. Moreover, in order for the upper bound to dominate the logarithm, i.e., be 10 times larger than the logarithm, we need $e^{e^{10}}$ nodes, which implies that the version age of a single node in the hypercube is logarithmic for all practical purposes.

E. Remarks on General Hypercubes

Remark 14: In this remark, we consider the problem of finding the version age of a single node in a general hypercube graph as defined in Section II. It is clear that due to the symmetry of the network, the version ages of any two nodes in the graph are equal. In order to write an upper bound recursion for this problem, we need to find a lower bound for $|\mathcal{E}(S)|$.

For an infinite grid graph $\mathbb{Z}^d$, it is found in [39] that the number of inner edges $|\bar{\mathcal{E}}(S)|$ is upper bounded as follows (assuming $|\mathcal{S}| = j$),

$$|\bar{\mathcal{E}}(S)| \leq \lfloor jd \left(1 - j^{-\frac{1}{d}}\right) \rfloor \quad (169)$$

$$\leq jd - \lfloor dj^{\frac{d-1}{d}} \rfloor \quad (170)$$

$$\leq jd - d \lfloor j^{\frac{d-1}{d}} \rfloor, \quad (171)$$

and since the total number of edges is given by $2jd$, the number of incoming edges can be lower bounded as,

$$|\mathcal{E}(S)| \geq 2d \lfloor j^{\frac{d-1}{d}} \rfloor. \quad (172)$$

This bound is an extension to the one found for the two-dimensional grid that we considered in Section IV. The bound in (172) is true for the case where the grid graph is infinite. However, in our case, the graph is finite (and wrapped-around at the boundaries). Hence, we need to consider the boundary effects, which will lead to different lower bounds for the number of incoming edges depending on how many boundary edges are used to decrease the number of incoming edges. Moreover, each set can have at most $2d$ faces. Hence, we conjecture that the number of incoming edges is greater than $j^{\frac{d-1}{d}}$ in the finite d -dimensional grid, up to $j = \frac{n}{2}$. For $j > \frac{n}{2}$, we can see that the complement set of the best set for $n - j$ is the set with the least number of incoming edges, and hence the bound $j^{\frac{d-1}{d}}$ works in this case as well.

Now, we can write the upper bound recursive equations following the proof of Lemma 3, and construct sets $v_1, v_2, \dots, v_n = \frac{\lambda_e}{\lambda}$ as in Theorem 1 to get the following recursion,

$$v_j \leq \begin{cases} \frac{2d\lambda_e}{\lambda} + j^{\frac{d-1}{d}} v_j, & j \leq \frac{n}{2}, \\ \frac{j}{n} + j^{\frac{d-1}{d}}, & \frac{n}{2} < j. \end{cases} \quad (173)$$

We then write out the complete recursion as done in Theorem 1 and get $n+1$ terms. We divide these terms into two parts, corresponding to the ranges $\{1, \dots, \frac{n}{2}\}$ and $\{\frac{n}{2} + 1, \dots, n+1\}$. We have two sum of terms corresponding to these ranges for j , say X and Y . We find the upper bound for X as follows,

$$X \leq \frac{2d\lambda_e}{\lambda} \left(\frac{1}{1 + \frac{1}{n}} \right) \left(1 + \sum_{i=1}^{\frac{n}{2}} \prod_{j=1}^i \frac{j^{\frac{d-1}{d}}}{(j+1)^{\frac{d-1}{d}} + \frac{j+1}{n}} \right) \quad (174)$$

$$\leq \frac{2d\lambda_e}{\lambda} \left(\frac{1}{1 + \frac{1}{n}} \right) \left(1 + \sum_{i=1}^n \prod_{j=1}^i \frac{j^{\frac{d-1}{d}}}{(j+1)^{\frac{d-1}{d}} + \frac{j}{n}} \right) \quad (175)$$

$$\leq \frac{2d\lambda_e}{\lambda} \left(1 + \sum_{i=1}^n \prod_{j=1}^i \frac{j^{\frac{d-1}{d}}}{(j+1)^{\frac{d-1}{d}} + \frac{j}{n}} \right), \quad (176)$$

as $n \rightarrow \infty$. Next, we find the order of the right hand side,

$$X \leq \frac{2d\lambda_e}{\lambda} \left(1 + \sum_{i=1}^n \prod_{j=1}^i \frac{1}{\left(1 + \frac{1}{j}\right)^{\frac{d-1}{d}} + \frac{j}{n}} \right) \quad (177)$$

$$\leq \frac{2d\lambda_e}{\lambda} \left(1 + \sum_{i=1}^n \prod_{j=1}^i \frac{1}{1 + \frac{d-1}{d} \frac{1}{j} - \frac{d-1}{2d^2} \frac{1}{j^2} + \frac{j}{n}} \right), \quad (178)$$

where (178) uses the approximate binomial expansion to bound (177). Now, we define,

$$a_i = \prod_{j=1}^i \frac{1}{1 + \frac{d-1}{d} \frac{1}{j} - \frac{d-1}{2d^2} \frac{1}{j^2} + \frac{j}{n}}. \quad (179)$$

Then, we have,

$$-\log a_i = \sum_{j=1}^i \log \left(1 + \frac{d-1}{d} \frac{1}{j} - \frac{d-1}{2d^2} \frac{1}{j^2} + \frac{j}{n} \right) \quad (180)$$

$$\approx \sum_{j=1}^i \left(\frac{d-1}{d} \frac{1}{j} - \frac{d-1}{2d^2} \frac{1}{j^2} + \frac{j}{n} \right) \quad (181)$$

$$\approx \frac{d-1}{d} \log i - C + \frac{di^{\frac{d+1}{d}}}{(d+1)n}, \quad (182)$$

where C is a finite positive constant upper bounded by $(d-1)\pi^2/6d^2$, and the third term is bounded by the first term of the Euler-MacLaurin series expansion. Substituting this back in (40), we obtain,

$$X \leq \frac{2d\lambda_e}{\lambda} \left(1 + k \sum_{i=1}^n e^{-\left(\frac{d-1}{d}\right) \log i - \frac{di^{\frac{d+1}{d}}}{(d+1)n}} \right) \quad (183)$$

$$= \frac{2d\lambda_e}{\lambda} \left(1 + k \sum_{i=1}^n \frac{1}{i^{\frac{d-1}{d}} e^{-\frac{di^{\frac{d+1}{d}}}{(d+1)n}}} \right), \quad (184)$$

where $k = e^{(d-1)\pi^2/6d^2}$. We define a function f and then write the Riemann sum with step size $n^{d/(d+1)}$ and construct an integral as follows,

$$f(x) = \frac{1}{x^{\frac{d-1}{d}}} e^{-\left(\frac{d}{d+1}\right) x^{\frac{d+1}{d}}} \quad (185)$$

$$\sum_{i=1}^n \frac{1}{n^{\frac{d}{d+1}}} f\left(\frac{i}{n^{\frac{d}{d+1}}}\right) = \int_0^\infty \frac{1}{t^{\frac{d-1}{d}}} e^{-\left(\frac{d}{d+1}\right) t^{\frac{d+1}{d}}} dt = L < \infty. \quad (186)$$

Hence, we can continue as follows,

$$\sum_{i=1}^n \frac{1}{n^{\frac{d}{d+1}}} f\left(\frac{i}{n^{\frac{d}{d+1}}}\right) = \frac{1}{n^{1/(d+1)}} \sum_{i=1}^n \frac{1}{i^{\frac{d-1}{d}}} e^{-\frac{di^{\frac{d+1}{d}}}{(d+1)n}} = L. \quad (187)$$

Finally, continuing from (184), we have,

$$X \leq \frac{2d\lambda_e}{\lambda} \left(1 + k \sum_{i=1}^n \frac{1}{i^{\frac{d-1}{d}}} e^{-\frac{di^{\frac{d+1}{d}}}{(d+1)n}} \right) \quad (188)$$

$$= \frac{2d\lambda_e}{\lambda} (1 + kLn^{1/(d+1)}). \quad (189)$$

Next, we write the recursion for Y in the same way we did for Z in Theorem 1 and observe that $Y = O(\log n)$.

Hence, we conclude that $v_1 = O(n^{1/(d+1)})$, if our conjecture for the lower bound for the number of incoming edges is correct. We note that the formula $v_1 = O(n^{1/(d+1)})$ satisfies the results for $d = 0, 1, 2$, where the version age of a single node in these cases scales as $O(n)$, $O(n^{\frac{1}{2}})$ and $O(n^{\frac{1}{3}})$, respectively, in a disconnected network, a line network and a square two-dimensional grid, which are hypercube networks with $d = 0$, $d = 1$ and $d = 2$, respectively.

Remark 15: Even though these results show the improvement in version age as the connectivity increases, these results still do not describe the entire spectrum of connectivity going from a disconnected network where the version age scales as $O(n)$ all the way to the fully-connected network where the version age scales as $O(\log n)$, since the nodes in the d -dimensional grid have $2d$ neighbors, which is still a constant even though it increases with d . In contrast, in the fully-connected network, each node has $n - 1$ neighbors, which depends on the number of nodes n in the network.

Remark 16: We can carry out a similar exercise as the one carried out in Remark 9 for a d -dimensional grid and conclude that we can improve the version age from $O(n^{\frac{1}{d+1}})$ to $O(n^{\frac{1}{d+2}})$ by introducing community structure to the network.

VIII. NUMERICAL RESULTS

In this section, we provide our numerical results grouped by each network structure studied in this paper, in the following subsections.

A. Numerical Results for the Grid Network

We perform numerical simulations to find the version age of a single node in the two-dimensional grid network, and compare them to the bounds we obtained. We set $\lambda_e = 1$, $\lambda = 1$ in each case. First, we simulate the square two-dimensional grid by varying n from 100 nodes corresponding to a 10×10 grid, to 10,000 nodes corresponding to a 100×100 grid, such that the number of nodes on one edge of the grid increases by increments of 10. We plot the results in Fig. 12. The simulated version age is plotted along with the upper bound $3.764n^{\frac{1}{3}}$ we obtained in (165). We see that our calculated upper bound bounds the simulated version age. We also plot $1.8n^{\frac{1}{3}}$ and $n^{\frac{1}{3}}$ to provide closer upper and lower bounds for the simulated version age. By trying various multiples for $n^{\frac{1}{3}}$ to match with the simulated version age, we find that $1.45n^{\frac{1}{3}}$ is a close approximation to the simulated version age we have obtained.

We also simulate a rectangular network where $m = 2k$, and the larger side is varied from 20 to 100 in steps of 20. We plot the results in Fig. 13a. We see that the upper bound we found still works in this case, since we are in the regime where $m = o(k^2)$. We also plot $1.6n^{\frac{1}{3}}$ to provide a closer upper bound for the simulated version age. Finally, we simulate a network where one side of the grid is constant. We choose the values such that the constant length side is the smaller one in each case, i.e., we choose k to be constant. The number of nodes varies from 500 to 5000 with a step size of 500. We consider four values for k , specifically 2, 4, 10 and 20. We plot the results in Fig. 13b. We note that the upper bound is given

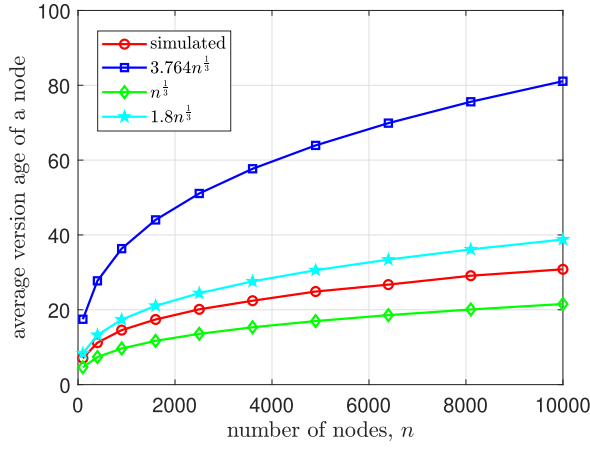
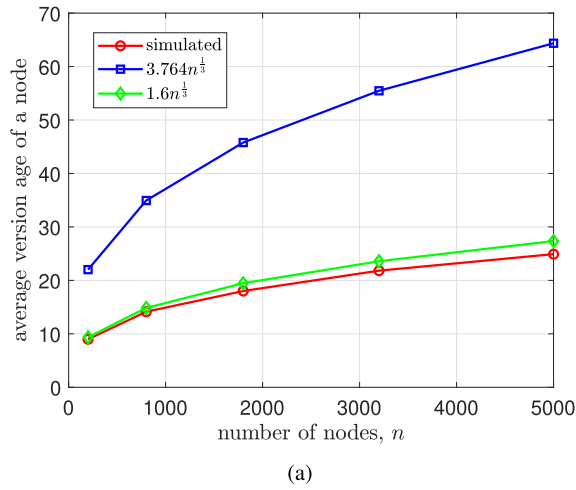
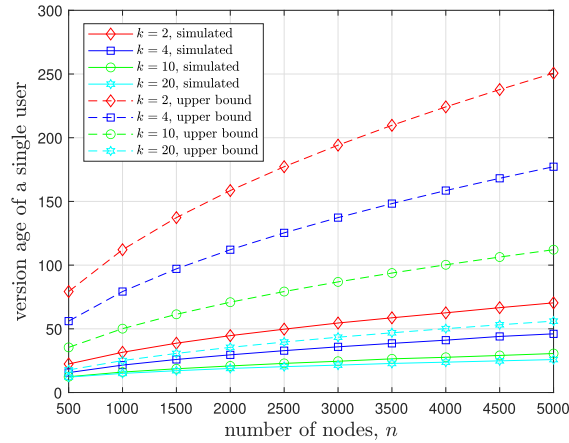


Fig. 12. Version age of a single node versus the number of nodes in a two-dimensional grid gossip network.



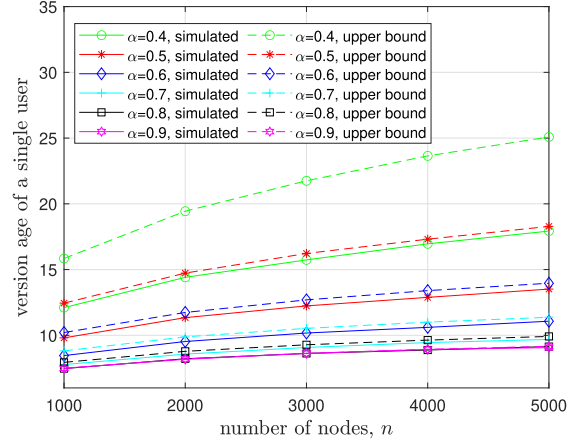
(a)



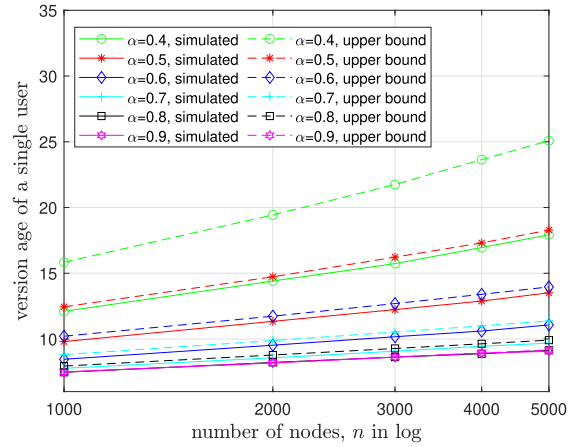
(b)

Fig. 13. (a) Variation of the version age of a single node in the grid network where one side is half the length of the other side, i.e., $m = 2k$, and the number of nodes n , i.e., $n = mk$. (b) Variation of the version age of a single node in the grid network where one side is constant, and the other side is increasing, and the number of nodes n .

by $2\sqrt{2\pi} \sqrt{\frac{n}{k}}$ in this case. Hence, as the value of k increases, the bound becomes smaller, along with the simulated version age. This is because when k is larger, the grid is more densely connected, resulting in a smaller version age.



(a)



(b)

Fig. 14. We plot the simulated version age for the generalized ring networks along with the upper bound in both linear and log scale. Here $f(n) = n^\alpha$. The left figure has a linear x-axis and the right figure has a logarithmic x-axis.

B. Numerical Results for the Generalized Ring Network

Next, we perform simulations to compare our upper bounds with the simulated version age of a single node for the generalized ring network. The upper bound for the version age of the generalized ring network depends on the number of nodes n , number of connections $2f(n)$, and the information flow rates λ_e and λ . We again choose $\lambda_e = \lambda = 1$.

We plot the version age for $f(n) = n^\alpha$ for α ranging from 0.4 to 0.9 with increments of 0.1. The number of nodes varies from 1000 to 5000 with a step size of 1000. We plot the results in Fig. 14a and Fig. 14b. In Fig. 14a, the x-axis is in a linear scale. We see that the version age decreases as α increases, which is consistent with the theoretical upper bound result. In Fig. 14b, the x-axis is in log scale. We see that the version age plots in Fig. 14b appear as straight lines, thus showing that they have approximate log scaling for low values of n , satisfying Remark 2.

We do not perform simulations for lower values of α because the functions $f(n)$ grow very slowly. For small values of α , the value of $f(n)$ might be constant, even if the number of nodes increases. Hence, for α between 0 and 0.3, we calculate the upper bound that we obtain from the recursive equations

TABLE III

WE ENUMERATE THE UPPER BOUNDS FOR THE VERSION AGE IN THE GENERALIZED RING NETWORK WHEN $f(n) = n^\alpha$, WHERE $\alpha = 0, 0.1, 0.2, 0.3$

n	10^4	10^5	10^6	10^7	10^8
Recursive upper bound	124.641	395.658	1252.645	3962.659	12532.474
Upper bound from (119)	183.515	566.769	1778.723	5611.261	17730.808

(a) Version age of a single node when $\alpha = 0$.

n	10^4	10^5	10^6	10^7	10^8
Recursive upper bound	102.198	280.508	886.491	2289.000	6700.205
Upper bound from (119)	132.987	332.071	1031.794	2516.117	7245.866

(b) Version age of a single node when $\alpha = 0.1$.

n	10^4	10^5	10^6	10^7	10^8
Recursive upper bound	67.926	178.614	445.018	1101.664	2805.389
Upper bound from (119)	82.213	197.497	469.331	1133.706	2851.794

(c) Version age of a single node when $\alpha = 0.2$.

n	10^4	10^5	10^6	10^7	10^8
Recursive upper bound	46.186	101.728	224.936	503.411	1121.325
Upper bound from (119)	57.45	113.806	237.864	517.252	1136.084

(d) Version age of a single node when $\alpha = 0.3$.

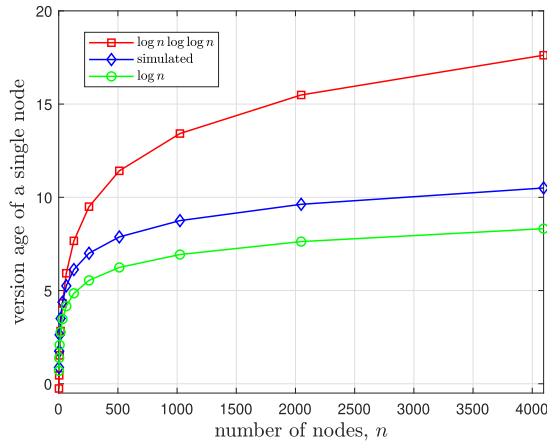


Fig. 15. Version age of a single node versus the number of nodes in a unit hypercube.

in (83), (90) and (113), and compare them to the upper bound obtained in (119). These results are summarized in Tables IIIa, IIIb, IIIc and IIId.

C. Numerical Results for the Unit Hypercube

We now simulate the unit hypercube network. We choose $\lambda_e = \lambda = 1$. We choose m ranging from 1 to 12 with step size of 1. We plot the results in Fig. 15 along with $\log n \log \log n$ and $\log n$. We notice that, for small values of n , $\log n \log \log n$ is negative. However, for larger values of n , the upper bound is satisfied. Moreover, $\log n$ is a lower bound for v_1 for all values of n that we choose. We note that the simulated version age follows the $\log n$ curve more closely than the $\log n \log \log n$ curve.

D. Numerical Results for the General Hypercube

Finally, we simulate the general hypercube network. Again, we choose $\lambda_e = \lambda = 1$. We simulate the d -dimensional grid for $d = 3, 4, 5$. For $d = 3$, we simulate for the range $m = 2$ to $m = 16$ with step size 2. For $d = 4$, we simulate for the range $m = 2$ to $m = 8$ with step size 1. For $d = 5$, we simulate for the range $m = 2$ to $m = 5$ with step size 1. We plot the results in Fig. 16a, where the plot is on a linear scale, and in Fig. 16b, where the plot is in the log scale. We see from Fig. 16a that the version age of a single node is a growing function of the number of nodes in the network, and the version age decreases as d increases, for a given number of nodes. We also plot the values in log-log scale since the version age scaling is $O(n^{\frac{1}{d+1}})$, the log-log plot will be linear with slope $\frac{1}{d+1}$. In fact, the slopes for $d = 3, 4, 5$ are 0.2478, 0.2072 and 0.1836, respectively. Hence, for $d = 3$ and 4, the slopes are consistent with the exact slope. However, for $d = 5$, the slope is a little higher than the expected 0.16667. This is because the number of nodes in the simulations is not large enough to get the exact scaling.

E. A Comparison of the Models

We now analyze the theoretical and simulation results and compare how the networks perform as n varies. When n is small, the constants significantly influence the long-term average version age, thus we rely on the simulation results for performance comparison. However, as n grows large, the impact of leading constants diminishes due to the scale of the functions, making theoretical results suitable to compare the performance of various networks.

First, we discuss the theoretical results, which accurately characterize the long-term average version age of the network when the number of nodes is large. In this case, we

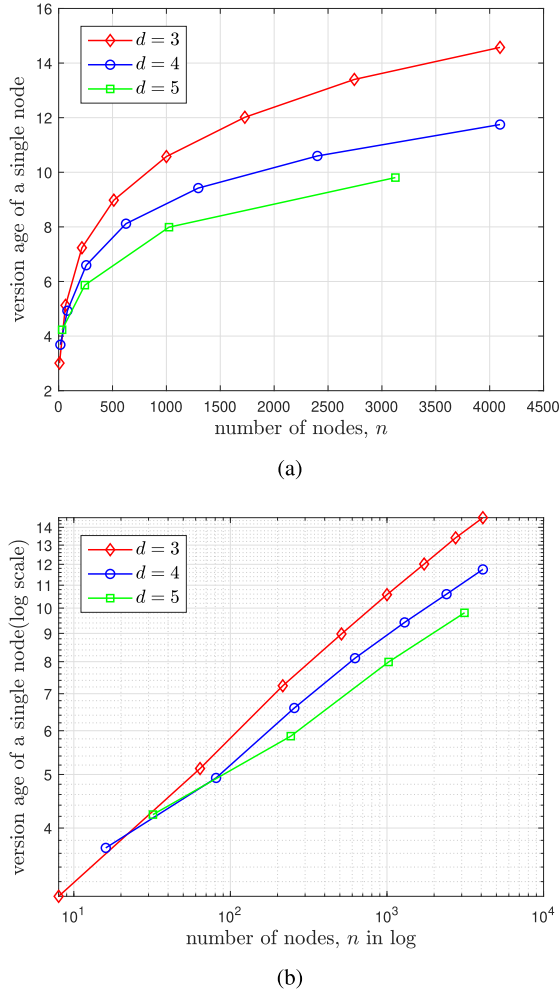


Fig. 16. Variation of the version age of a single node in a d -dimensional grid where $d = 3, 4, 5$. The left figure is in linear scale, and the right figure is in log scale.

observe that the unit hypercube network performs the best, followed by the generalized ring network with $f(n) = n^\alpha$ for $\alpha = 0.9, 0.8, 0.7$ in this order. This is then followed by the 5-dimensional general hypercube. Next, we have the 4-dimensional general hypercube and the generalized ring with $\alpha = 0.6$. This is followed by the 3-dimensional general hypercube and the generalized ring with $\alpha = 0.5$. We then have the grid network when $k = \Omega(\sqrt{m})$. Finally, we have the grid network when $k = o(\sqrt{m})$.

The relative behavior of the networks is different when the size of the network is small. In this scenario, we analyze the simulation results to compare the average version age performance of the networks. First, we notice that the generalized ring network with $\alpha = 0.9$ has lower average version age when compared to the unit hypercube network. Moreover, the generalized ring networks with $\alpha = 0.8$ and $\alpha = 0.7$ perform almost as well as the one with $\alpha = 0.9$, thus performing better than the unit hypercube network. Even though the generalized ring network with $\alpha = 0.6$ is worse than the unit hypercube network, it follows the latter closely. This reinforces the fact that when the number of nodes is small, the generalized ring networks with large α behaves like a fully-connected network.

We also observe that the 5-dimensional general hypercube network has a similar version age as the unit hypercube network. Also, the 4-dimensional general hypercube network closely follows the generalized ring network with $\alpha = 0.6$, and the 3-dimensional general hypercube network closely follows the generalized ring network with $\alpha = 0.5$.

We note that the grid networks have worse average version age performance when compared to all the other networks, even when the number of nodes in the network is small. However, we can clearly see that when $k = \Omega(\sqrt{m})$, the version age performance is better than when $k = o(\sqrt{m})$.

IX. CONCLUSION

In this work, we provided a new technique to find a tight upper bound for the version age of a connected subset of nodes in a gossiping network by modifying the recursive equations found in [9]. We found these bounds in terms of, first, the number of neighbors of the set, and then the number of neighboring edges of the set. We used this technique to analyze various graph models that are used extensively in the literature, including the two-dimensional grid graph, generalized rings with varying number of neighbors, unit hypercube graphs with varying dimension and fixed side length, and hypercube graphs in fixed dimension with varying side length. In this analysis, we found the version age of a single node in each network. We extended the results of the previously analyzed graphs, the fully-connected network and the bi-directional ring network and verified the bounds obtained in both cases. Moreover, we obtained a sequence of graphs that go from the most disconnected case, which is the bi-directional ring network to the most connected case, which is the fully-connected network, through the generalized ring network analysis. In this way, we were able to get insights into the graph properties which version age depends on.

REFERENCES

- [1] P. Popovski et al., "A perspective on time toward wireless 6G," *Proc. IEEE*, vol. 110, no. 8, pp. 1116–1146, Aug. 2022.
- [2] S. K. Kaul, R. D. Yates, and M. Gruteser, "Real-time status: How often should one update," in *Proc. IEEE INFOCOM*, Mar. 2012.
- [3] Y. Sun, I. Kadota, R. Talak, and E. Modiano, "Age of information: A new metric for information freshness," *Synth. Lectures Commun. Netw.*, vol. 12, no. 2, pp. 1–224, Dec. 2019.
- [4] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and Ş. Ulukuş, "Age of information: An introduction and survey," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1183–1210, May 2021.
- [5] S. Banerjee, S. Ulukuş, and A. Ephremides, "To re-transmit or not to re-transmit for freshness," in *Proc. 21st Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, Aug. 2023, pp. 516–523.
- [6] A. Maatouk, S. Kriouile, M. Assaad, and A. Ephremides, "The age of incorrect information: A new performance metric for status updates," *IEEE/ACM Trans. Netw.*, vol. 28, no. 5, pp. 2215–2228, Oct. 2020.
- [7] J. Zhong, R. D. Yates, and E. Soljanin, "Two freshness metrics for local cache refresh," in *Proc. IEEE ISIT*, Jun. 2018, pp. 1924–1928.
- [8] J. Cho and H. Garcia-Molina, "Effective page refresh policies for web crawlers," *ACM Trans. Database Syst.*, vol. 28, no. 4, pp. 390–426, Dec. 2003.
- [9] R. D. Yates, "The age of gossip in networks," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2021, pp. 2984–2989.
- [10] B. Abolhassani, J. Tadrus, A. Eryilmaz, and E. Yeh, "Fresh caching for dynamic content," in *Proc. IEEE INFOCOM*, May 2021.
- [11] M. Bastopcu and S. Ulukuş, "Who should Google Scholar update more often?," in *Proc. IEEE INFOCOM*, Jul. 2020.
- [12] A. Demers et al., "Epidemic algorithms for replicated database maintenance," *ACM SIGOPS Operating Syst. Rev.*, vol. 22, no. 1, pp. 8–32, Jan. 1988.

- [13] R. D. Yates, "Timely gossip," in *Proc. IEEE 22nd Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Sep. 2021, pp. 331–335.
- [14] Y. Minsky, "Spreading rumors cheaply, quickly, reliably," Ph.D. thesis, Cornell Univ., Mar.2002.
- [15] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking, "Randomized rumor spreading," in *Proc. 41st Annu. Symp. Found. Comput. Sci.*, Nov. 2000.
- [16] B. Pittel, "On spreading a rumor," *SIAM J. Appl. Math.*, vol. 47, no. 1, pp. 213–223, Feb.1987.
- [17] S. Deb, M. Medard, and C. Choute, "Algebraic gossip: A network coding approach to optimal multiple rumor mongering," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2486–2507, Jun.2006.
- [18] D. Mosk-Aoyama and D. Shah, "Information dissemination via network coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2006, pp. 1748–1752.
- [19] S. Sanghavi, B. Hajek, and L. Massoulie, "Gossiping with multiple messages," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4640–4654, Dec.2007.
- [20] G. DeCandia et al., "Dynamo: Amazon's highly available key-value store," *ACM SIGOPS Operating Syst. Rev.*, vol. 41, no. 6, pp. 205–220, 2007.
- [21] A. Lakshman and P. Malik, "Cassandra: A decentralized structured storage system," *ACM SIGOPS Operating Syst. Rev.*, vol. 44, no. 2, pp. 35–40, Apr.2010.
- [22] B. Buyukates, M. Bastopcu, and S. Ulukus, "Age of gossip in networks with community structure," in *Proc. IEEE 22nd Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Sep. 2021, pp. 326–330.
- [23] P. Kaswan and S. Ulukus, "Susceptibility of age of gossip to timestamping," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Nov. 2022, pp. 398–403.
- [24] P. Kaswan and S. Ulukus, "Age of gossip in ring networks in the presence of jamming attacks," in *Proc. 56th Asilomar Conf. Signals, Syst., Comput.*, Oct. 2022, pp. 1055–1059.
- [25] P. Kaswan and S. Ulukus, "Age of information with non-Poisson updates in cache-updating networks," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2023, pp. 957–962.
- [26] P. Kaswan and S. Ulukus, "Timely tracking of a remote dynamic source via multi-hop renewal updates," in *Proc. 62nd IEEE Conf. Decis. Control (CDC)*, Dec. 2023.
- [27] P. Mitra and S. Ulukus, "ASUMAN: Age sense updating multiple access in networks," in *Proc. 58th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2022, pp. 1–7.
- [28] P. Mitra and S. Ulukus, "Timely opportunistic gossiping in dense networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, May 2023, pp. 1–6.
- [29] M. A. Abd-Elmagid and H. S. Dhillon, "Distribution of the age of gossip in networks," *Entropy*, vol. 25, no. 2, pp. 364–395, Feb.2023.
- [30] E. Delfani and N. Pappas, "Version age-optimal cached status updates in a gossiping network with energy harvesting sensor," in *Proc. 21st Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, Aug. 2023, pp. 143–150.
- [31] M. Bastopcu, S. R. Etesami, and T. Başar, "The role of gossiping for information dissemination over networked agents," *Entropy*, vol. 26, no. 1, p. 9, Jan.2022.
- [32] F. Harary and H. Harborth, "Extremal animals," *J. Combinatorics, Inf. Syst. Sci.*, vol. 1, no. 1, pp. 1–8, Jan.1976.
- [33] A. Nowzari and M. G. Rabbat, "Improved bounds for max consensus in wireless networks," *IEEE Trans. Signal Inf. Process. over Netw.*, vol. 5, no. 2, pp. 305–319, Jun.2019.
- [34] B. Buyukates, M. Bastopcu, and S. Ulukus, "Version age of information in clustered gossip networks," *IEEE J. Sel. Areas Inf. Theory*, vol. 3, no. 1, pp. 85–97, Mar.2022.
- [35] S. Hart, "A note on the edges of the n -cube," *Discrete Math.*, vol. 14, no. 2, pp. 157–163, Apr.1976.
- [36] L. E. Bush, "An asymptotic formula for the average sum of the digits of integers," *Amer. Math. Monthly*, vol. 47, no. 3, pp. 154–156, Mar.1940.
- [37] P. Kaswan and S. Ulukus, "Timely gossiping with file slicing and network coding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2022, pp. 928–933.
- [38] O. Yagan, D. Qian, J. Zhang, and D. Cochran, "Conjoining speeds up information diffusion in overlaying social-physical networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 6, pp. 1038–1048, Jun.2013.
- [39] G. Agnarsson and K. Lauria, "Extremal subgraphs of the d -dimensional grid graph," 2013, *arXiv:1302.6517*.

Arunabh Srivastava (Graduate Student Member, IEEE) received the B.Tech. and M.Tech. degrees in electrical engineering from Indian Institute of Technology Madras in 2021. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Maryland, College Park, under the supervision of Prof. Sennur Ulukus. His research interests include timely information dissemination, gossip algorithms, networks with mobile nodes, and large language models.

Sennur Ulukus (Fellow, IEEE) received the B.S. and M.S. degrees in electrical and electronics engineering from Bilkent University and the Ph.D. degree in electrical and computer engineering from the Wireless Information Network Laboratory (WINLAB), Rutgers University.

She is currently a Distinguished University Professor and the Anthony Ephremides Professor in information sciences and systems with the Department of Electrical and Computer Engineering, University of Maryland at College Park, where she also holds a joint appointment with the Institute for Systems Research (ISR). Prior to joining UMD, she was a Senior Technical Staff Member with the AT&T Laboratories-Research. Her research interests include information theory, wireless communications, machine learning, signal processing, and networks, with recent focus on age of information, private information retrieval, blockchain consensus, quantum information, machine learning for wireless, federated learning, and large language models for communications and networks.

Dr. Ulukus is a Distinguished Scholar-Teacher of the University of Maryland. She received the 2003 IEEE Marconi Prize Paper Award in Wireless Communications, the 2019 IEEE Communications Society Best Tutorial Paper Award, the 2020 IEEE Communications Society Women in Communications Engineering (WICE) Outstanding Achievement Award, the 2020 IEEE Communications Society Technical Committee on Green Communications and Computing (TCGCC) Distinguished Technical Achievement Recognition Award, the 2024 IEEE Communications Society Communication Theory Technical Committee (CTTC) Technical Achievement Award, a 2005 NSF CAREER Award, the 2011 ISR Outstanding Systems Engineering Faculty Award, and the 2012 ECE George Corcoran Outstanding Teaching Award. She was a Distinguished Lecturer of the IEEE Information Theory Society from 2018 to 2019. She is a Senior Editor for IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING since 2020. She was an Area Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2019 to 2023, an Area Editor for IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING from 2016 to 2020, an Editor for IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS-Series on Green Communications and Networking from 2015 to 2016, an Associate Editor for IEEE TRANSACTIONS ON INFORMATION THEORY from 2007 to 2010, and an Editor for IEEE TRANSACTIONS ON COMMUNICATIONS from 2003 to 2007. She was a Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN INFORMATION THEORY in 2021, 2022, and 2023, the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS in 2008, 2015, 2021, and 2022, the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING in 2021, *Journal of Communications and Networks* in 2012, and IEEE TRANSACTIONS ON INFORMATION THEORY in 2011. She is the TPC Chair of 2021 IEEE Globecom, and the TPC Co-chair of 2024 IEEE Globecom, 2024 IEEE DySPAN, 2023 IEEE MILCOM, 2019 IEEE ITW, 2017 IEEE ISIT, 2016 IEEE Globecom, 2014 IEEE PIMRC, and 2011 IEEE CTW.