Age of Information with Unreliable Transmissions in Multi-Source Multi-Hop Status Update Systems

Shahab Farazi†, Andrew G. Klein‡ and D. Richard Brown III†
†Worcester Polytechnic Institute, 100 Institute Rd., Worcester, MA 01609, Email: {sfarazi,drb}@wpi.edu
‡Western Washington University, 516 High St., Bellingham, WA 98225, Email: andy.klein@wwu.edu

Abstract—This paper studies an “age of information” problem in multi-hop networks with slotted transmissions and packet loss, where each node is both a source and monitor of information. Nodes take turns broadcasting their information to other nodes while also maintaining tables of updates for the information received from other nodes. It is assumed each link has a fixed error probability independent of other links. We present an algorithm that generates schedules for any network with a connected topology. A closed-form expression for the average number of time slots to update all tables are derived as a function of fundamental graph properties.

Index Terms—Age of information, multi-source, multi-hop, packetized communications, explicit contention, transmission error.

I. INTRODUCTION

Information freshness is of critical importance in a variety of networked monitoring and control systems such as intelligent vehicular systems, channel state feedback, and environmental monitoring as well as applications such as financial trading and online learning. A recent line of research has considered information freshness from a fundamental perspective under an Age of Information (AoI) metric first proposed in [1] and further studied in [2]–[30]. The main idea is that there are one or more sources of information along with one or more monitors. A source generates timestamped status updates which are received by a monitor after some delay. The “age of information” is defined as the difference between the current time and the timestamp of the most recent status update at the monitor.

This paper studies AoI in a general multi-source, multi-hop, time-slotted network setting with explicit contention in the sense that all delays between sources and monitors are due to explicit channel uses by other nodes in the network. Each node in the system is both a source and monitor of information. Since the only assumption on the network is that it is connected, some nodes in the network also serve as relays to facilitate multi-hop dissemination of information between nodes that are not directly connected. Using a graph theoretical approach, this paper builds on our prior results in [27]–[29] by generalizing our age of information analysis to account for unsuccessful transmissions of status updates. The main contributions of this paper are (i) the development of an explicit algorithm for constructing status update dissemination schedules for every connected network topology, and (ii) deriving a closed-form expression for the average number of time slots to update all statuses throughout the network for the schedules generated by the algorithm.

II. SYSTEM MODEL

Consider a wireless network modeled by an undirected graph $G = (V, E)$. The vertex set $V$ represents the nodes and the edge set $E$ represents the channels between the nodes in the network. Two vertices $i, j \in V$ are adjacent if edge $e_{i,j}$ is in set $E$. Equivalently, there exists a channel between nodes $i$ and $j$; as such, any wireless transmission broadcast from node $i$ is received at all adjacent nodes. Here, we only consider networks with a connected topology, i.e., there exists a path between any two distinct vertices $i, j \in V$. Each node $i \in V$ can generate samples of a local random process $H_i(t)$ at any time $t$. In addition to information on the status of its own process, every node in the network is also interested in updates of the status of the remaining $N - 1$ processes in the network. We denote the status of process $H_i(t)$ from the perspective of node $j$ at time $t$ by $H_{i,j}^{(j)}(t)$. So, at any time each of the $N$ nodes keeps a table of its most recently obtained status updates of each of the $N$ processes, giving a total of $N^2$ parameters throughout the network. Out of the $N$ parameters at each node, one is obtained locally by direct observation, and $N - 1$ are obtained by indirect observation from the status updates disseminated by other nodes. Overall, there exist $N$ directly and $N^2 - N$ indirectly obtained parameters throughout the network. We assume transmissions of status update packets with a fixed length of one unit of time. Each packet includes information about the one process that is being transmitted and a time stamp indicating the time that the information was generated. For $i \in V$, a packet transmitted by node $i$ is successfully received by node $j$ with probability $1 - \epsilon$, where $0 \leq \epsilon < 1$ and $j \in N_i(i) \Leftrightarrow e_{i,j} \in E$. Packet transmissions over any given channel are assumed to be independent of all of the other channels. Also, we assume that the transmitter knows whether the transmission to each of its neighbors is successful.

The following definitions formalize the notation and age metrics considered in this paper. First, we review some common graph theoretic concepts. We use the notation $d(i, j)$ to denote the shortest path length between vertices $i$ and $j$. Recall that a set $S \subset V$ of vertices in a graph is called a dominating set if every vertex not in $S$ is adjacent to a vertex in
S [31]. A minimum connected dominating set (MCDS) $S \subseteq V$ is a dominating set with the properties that (i) the subgraph induced by $S$, $G[S]$ is connected and (ii) $S$ has the smallest cardinality among all connected dominating sets of $G$. The cardinality of any MCDS is called the connected domination number of $G$ and denoted by $\gamma_c$. In general, the MCDS is not unique [32], [33].

**Definition 1** (Pseudo-leaf vertex). We refer to a vertex as a pseudo-leaf if it is not a member of any MCDS. That is $i \in V$ is a pseudo-leaf if $i \notin U$ where the $S_m \subseteq V$ for $m = \{1, 2, \ldots M\}$ represent all $M$ possible MCDS’s of $G$ and

$$U \triangleq S_1 \cup S_2 \cup \ldots \cup S_M.$$  

(1)

Further, we refer to the set of all pseudo-leaf vertices of $G$ by

$$\mathcal{L} \triangleq V - \mathcal{U}.$$  

(2)

Under this definition, every true leaf (i.e., every vertex with degree one) is also a pseudo-leaf. An example illustrating the notion of pseudo-leaf vertices and MCDS’s is shown in Fig. 1.

**Definition 2** (Age). Assume the most recent status update of the $H_i$ process received at node $j$ was timestamped at time $t_i$. The age of status update $H_i^{(j)}$ at time $t \geq t_i$ is defined as $\Delta_i^{(j)}(t) \triangleq t - t_i$ for $j \neq i$.

Since each node is assumed to have zero-delay access to the status of its local process, we have $\Delta_i^{(j)}(t) = 0$ for any $i \in V$ and $t$. To capture the timeliness of all of the $N^2 - N$ indirectly-obtained status update parameters throughout the network, we define the instantaneous peak age of information metric in the following.

**Definition 3** (Instantaneous peak age). For any $t \geq \bar{t}$, the instantaneous peak age is defined as

$$\Delta_{\text{peak}}(t) \triangleq \max \Delta(t)$$  

(3)

Note that $t$ is fixed here and the maximum is computed over the $N^2 - N$ elements of the vector $\Delta(t)$. Similarly, we define the instantaneous average age at any point $t \geq \bar{t}$ below.

We refer to a schedule as an ordered sequence of transmitting nodes and the corresponding status update parameter that they disseminate in each time slot.

### III. Schedule Design for Status Update Dissemination

In this section we provide an algorithm that generates schedules for refreshing all of the status update parameters throughout the network with any arbitrary topology. Observe that in the following, “Depth-First Search($G[S]$, $i$)” describes an ordered list of vertices generated by performing a depth-first search of the graph induced by $S$ where the search starts at root vertex $i$. Basically, any node in a MCDS keeps transmitting the status update that it is disseminating until all of its one-hop neighbors in the network receive the status update at least once.

**Algorithm 1: Schedule design to disseminate status updates throughout the network.

Step I:** initialize time, $t \leftarrow -1$.

**Step II:** for node $i = 1 : N$ do

1. * if $\exists$ MCDS $\bar{S}$ s.t. $i \in \bar{S}$ then
   * $S \leftarrow \bar{S}$.

2. else
   * $S \leftarrow S \cup \{i\}$, for any MCDS $S \subseteq V$.

end

* $S_{\text{sorted}} = \text{Depth-First Search}(G[S], i)$

* for $k = 1 : |S_{\text{sorted}}|$ do

1. * $j = S_{\text{sorted}}(k)$,

2. * $t' \leftarrow t$,

   while $\exists$ at least one $k \in N_1(j)$ s.t. $k$ has not received a packet during interval $[t', t]$ do

   * node $j$ transmits $H_i^{(j)}(t')$,

   * $t \leftarrow t + 1$.

end

end

**Step III:** repeat from Step II.

Observe that the schedule generated by Algorithm 1 is periodic in the absence of packet errors; however, when packet errors are present it is not periodic in general. In the next section we evaluate the expected number of time slots for the schedule generated by Algorithm 1 to update all of the statuses throughout the network, for a given network topology.

### IV. Age of Information Analysis

It was shown in [27] that in the absence of errors, the number of time slots required to update all of the statuses throughout the network is lower bounded by

$$T \geq T^* = N\gamma_c + |L|.$$  

(4)

However when the packets may not be received reliably, the parameter $T$ is a random variable that is a function of the graph parameters and the error probability. Theorem 1 represents a closed-form expression on the expected value of $T$ achieved by the schedules generated with Algorithm 1.
Theorem 1. The average number of time slots required to refresh all of the status update tables throughout the network for the schedules generated by Algorithm 1 is

\[ T = \sum_{i=1}^{N} \sum_{m=1}^{\gamma_c+1} \Xi_{i,m}, \]  

where

\[ \Xi_{i,m} = (1-\epsilon)^{J_{i,m}} + \sum_{n=1}^{J_{i,m}} \left( \frac{J_{i,m}}{n} \right) \left( \frac{1-(1-\epsilon^n)^2}{1-\epsilon^n} \right), \]  

where \( J_{i,m} \) is the number of nodes that get updated by the \( m^{th} \) node in the MCDS that disseminates the \( H_i \) process throughout the network.

Proof: Without loss of generality, consider dissemination of the \( H_i \) process throughout the network for \( i \in V \). For \( \epsilon = 0 \) and from (4), the process \( H_i \) should be disseminated for \( \gamma_c+1 \in \mathcal{L} \) times by the nodes in the sorted MCDS in the schedule generated by Algorithm 1. Denote the index of this set of sorted nodes by \( m = 1, 2, \ldots, \gamma_c+1 \in \mathcal{L} \), and the number of new nodes that get updated by the node with index \( m \) by \( J_{i,m} \). By new nodes we mean that if for node \( k \in V \) is connected to both nodes with indices \( m \) and \( m' > m \), node \( k \) is not considered in the set of new nodes that get updated by the node with index \( m' \). Denote the number of transmissions by the node with index \( m \) to update its \( J_{i,m} \) neighbors by \( k_{i,m} \). For \( \ell \in \{1, 2, 3, \ldots \} \) we have

\[ \Pr\{k_{i,m} \geq \ell\} = 1 - \sum_{n=1}^{J_{i,m}} \Pr\{k_{i,m} = n\}, \]

\[ = \sum_{n=1}^{J_{i,m}} \left( \frac{J_{i,m}}{n} \right) \epsilon^{n(\ell-1)}. \]  

Considering (7), we can write

\[ \Pr\{k_{i,m} = 1\} = (1-\epsilon)^{J_{i,m}}, \]

\[ \Pr\{k_{i,m} = \ell\} = 1 - \left( \sum_{n=1}^{J_{i,m}} \Pr\{k_{i,m} = n\} \right) - \Pr\{k_{i,m} \geq \ell+1\}, \]

\[ = \sum_{n=1}^{J_{i,m}} \left( \frac{J_{i,m}}{n} \right) \epsilon^{n(\ell-1)} - \sum_{n=1}^{J_{i,m}} \left( \frac{J_{i,m}}{n} \right) \epsilon^{n\ell}, \]

\[ = \sum_{n=1}^{J_{i,m}} \left( \frac{J_{i,m}}{n} \right) \epsilon^{n\ell} \left( \frac{1-\epsilon^n}{\epsilon^n} \right), \]

for \( \ell \in \{2, 3, \ldots \} \). From (8a)-(8b) we can write

\[ E[k_{i,m}] = \sum_{\ell=1}^{\infty} \ell \Pr\{k_{i,m} = \ell\}, \]

\[ = (1-\epsilon)^{J_{i,m}} - \sum_{n=1}^{J_{i,m}} \left( \frac{J_{i,m}}{n} \right) (1-\epsilon^n) \]

\[ + \sum_{\ell=1}^{\infty} \ell \sum_{n=1}^{J_{i,m}} \left( \frac{J_{i,m}}{n} \right) \epsilon^{n\ell} \left( \frac{1-\epsilon^n}{\epsilon^n} \right), \]

From Theorem 1, observe that there always exists at least one status update parameter that on average needs to wait for \( T \) time slots before it gets refreshed. Hence, \( T \) lower bounds the average instantaneous peak age of information.

V. NUMERICAL RESULTS

This section presents numerical examples to illustrate the achieved peak age of information. For fully-connected \( K_N \) and ring \( C_N \) networks Fig 2 represents the achieved \( T \) versus error probability \( \epsilon \in \{0, 0.01, \ldots, 0.15\} \) for different number of nodes \( N \in \{3, 4, \ldots, 10\} \). The results show that for ring networks, the average number of time slots required to refresh all tables scales linearly with \( \epsilon \). Also, for \( \epsilon = 0 \) the average number of time slots \( T \) equals \( T^* \), where \( T^* = N^2 - N \) for \( K_N \) and \( T^* = N^2 - 2N \) for \( C_N \).
VI. CONCLUSION

Will be provided in the final version.

REFERENCES


