Information Degradation and Misinformation in Gossip Networks

M., Arunabh Srivastava, Sennur Ulukus

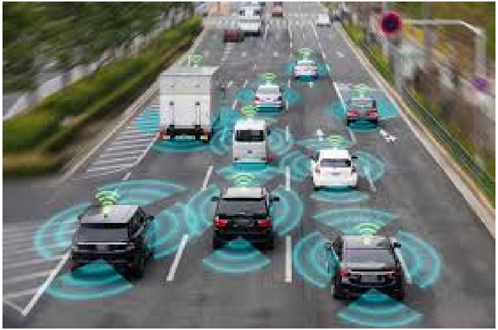
Department of Electrical and Computer Engineering
University of Maryland

November 21, 2025

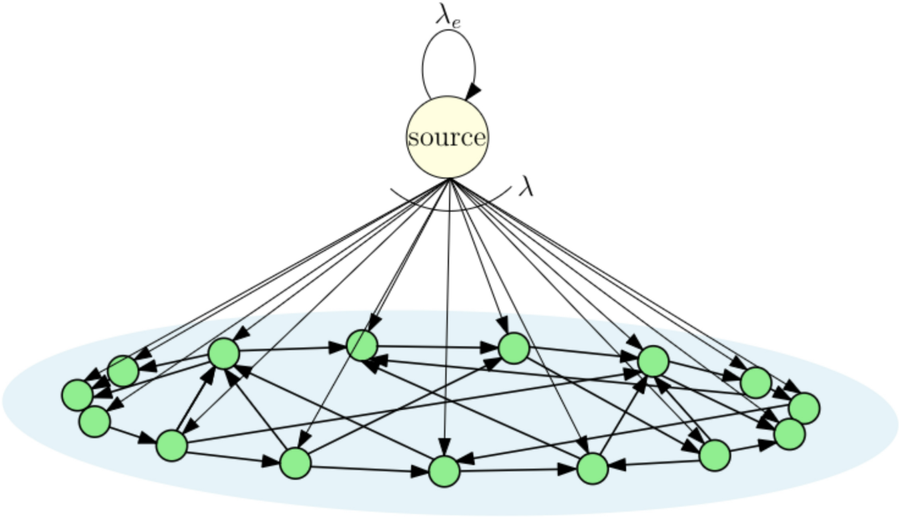
Overview

1. Gossip and Age of Information
2. Information Degradation
3. Previous Work
4. Main Results

Motivation



System Model



Model and Age of Information

- Packets are sent in the network and hold versions of some data.

Model and Age of Information

- Packets are sent in the network and hold versions of some data.
  - Think of software versions distributed to users over time

Model and Age of Information

- Packets are sent in the network and hold versions of some data.
  - Think of software versions distributed to users over time
- Nodes randomly send updates to neighbors via independent Poisson Processes

Model and Age of Information

- Packets are sent in the network and hold versions of some data.
  - Think of software versions distributed to users over time
- Nodes randomly send updates to neighbors via independent Poisson Processes
- Nodes uniformly communicate to neighbors - i.e. no preferred communication direction

Model and Age of Information

- Packets are sent in the network and hold versions of some data.
  - Think of software versions distributed to users over time
- Nodes randomly send updates to neighbors via independent Poisson Processes
- Nodes uniformly communicate to neighbors - i.e. no preferred communication direction
- Nodes always accept the newest version of the data.

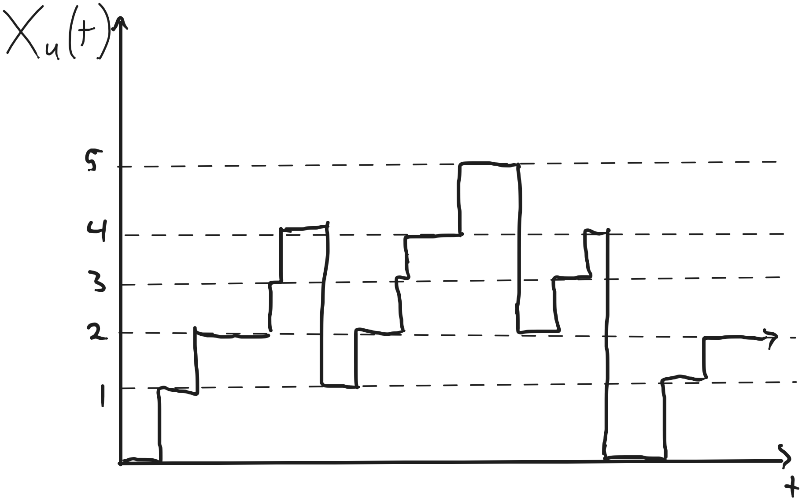
Model and Age of Information

- Packets are sent in the network and hold versions of some data.
  - Think of software versions distributed to users over time
- Nodes randomly send updates to neighbors via independent Poisson Processes
- Nodes uniformly communicate to neighbors - i.e. no preferred communication direction
- Nodes always accept the newest version of the data.
- $X_u(t)$ is the number of versions node u is behind the source n_0 at time t.

Model and Age of Information

- Packets are sent in the network and hold versions of some data.
  - Think of software versions distributed to users over time
- Nodes randomly send updates to neighbors via independent Poisson Processes
- Nodes uniformly communicate to neighbors - i.e. no preferred communication direction
- Nodes always accept the newest version of the data.
- $X_u(t)$ is the number of versions node u is behind the source n_0 at time t.
- $X_S(t)$ is the newest version available in the subset of nodes S.

AoI Example



A Model for Information Degradation

- Each node also contains a discrete variable $S_u(t) \in \{0, \ldots, k-1\}$, where we interpret each value as some measure of information quality

A Model for Information Degradation

- Each node also contains a discrete variable $S_u(t) \in \{0, \ldots, k-1\}$, where we interpret each value as some measure of information quality
- At all times, the source maintains $S_{n_0}(t) = 0$

A Model for Information Degradation

- Each node also contains a discrete variable $S_u(t) \in \{0, \ldots, k-1\}$, where we interpret each value as some measure of information quality
- At all times, the source maintains $S_{n_0}(t) = 0$
- Example: Each state the number of errors in a file - the source always contains the 'correct' file.

A Model for Information Degradation

- Each node also contains a discrete variable $S_u(t) \in \{0, \ldots, k-1\}$, where we interpret each value as some measure of information quality
- At all times, the source maintains $S_{n_0}(t) = 0$
- Example: Each state the number of errors in a file - the source always contains the 'correct' file.
- There is some Markov chain $\mathcal{M}$ on state space $\{0, \ldots, k-1\}$ which represents how the underlying data quality changes when a packet is transmitted.

A Model for Information Degradation

- Each node also contains a discrete variable $S_u(t) \in \{0, \ldots, k-1\}$, where we interpret each value as some measure of information quality
- At all times, the source maintains $S_{n_0}(t) = 0$
- Example: Each state the number of errors in a file - the source always contains the 'correct' file.
- There is some Markov chain $\mathcal{M}$ on state space $\{0, \ldots, k-1\}$ which represents how the underlying data quality changes when a packet is transmitted.
- When node u receives an update from node v at time t, its information quality is updated as:

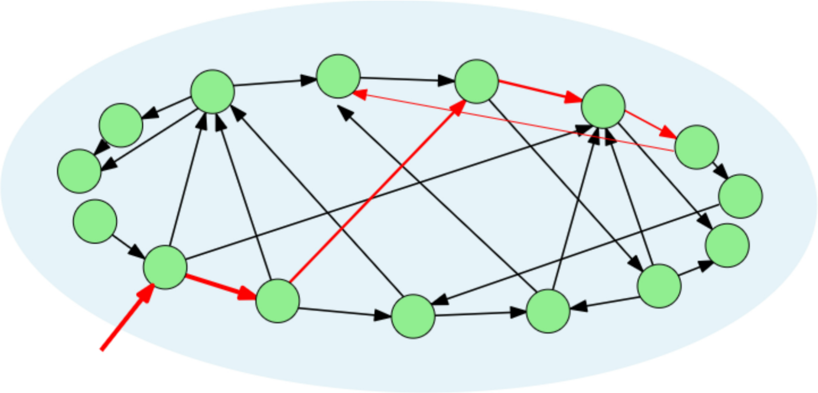
A Model for Information Degradation

- Each node also contains a discrete variable $S_u(t) \in \{0, \ldots, k-1\}$, where we interpret each value as some measure of information quality
- At all times, the source maintains $S_{n_0}(t) = 0$
- Example: Each state the number of errors in a file - the source always contains the 'correct' file.
- There is some Markov chain $\mathcal{M}$ on state space $\{0, \ldots, k-1\}$ which represents how the underlying data quality changes when a packet is transmitted.
- When node u receives an update from node v at time t, its information quality is updated as:
    - If the version age of v is less than u, $S_u(t^+)$ remains unchanged

A Model for Information Degradation

- Each node also contains a discrete variable $S_u(t) \in \{0, \dots, k-1\}$, where we interpret each value as some measure of information quality
- At all times, the source maintains $S_{n_0}(t) = 0$
- Example: Each state the number of errors in a file - the source always contains the 'correct' file.
- There is some Markov chain $\mathcal{M}$ on state space $\{0, \dots, k-1\}$ which represents how the underlying data quality changes when a packet is transmitted.
- When node u receives an update from node v at time t, its information quality is updated as:
  - If the version age of v is less than u, $S_u(t^+)$ remains unchanged
  - Else, $S_u(t^+)$ is obtained by running $\mathcal{M}$ for one step from state $S_v(t)$.

Degradation Example



Main Question

How does the interplay between the transition matrix $\mathcal{M}$ and the network topology G impact the distribution of the information quality $S_u(t)$?

Previous Combinatorial Work

- Yates (2021) showed the version age $v_G(S) = \lim_{t\to\infty} \mathbb{E} X_S(t)$ exists and can be computed combinatorially:

$$v_G(S) = \frac{\lambda_e + \sum_{i\notin S} \lambda_i(S) v_G(S \cup \{i\})}{\lambda_0(S) + \sum_{i\notin S} \lambda_i(S)}$$

Previous Combinatorial Work

- Yates (2021) showed the version age $v_G(S) = \lim_{t\to\infty} \mathbb{E}X_S(t)$ exists and can be computed combinatorially:

$$v_G(S) = \frac{\lambda_e + \sum_{i\notin S} \lambda_i(S) v_G(S \cup \{i\})}{\lambda_0(S) + \sum_{i\notin S} \lambda_i(S)}$$

- Kaswan and Ulukus (2023) studied a related misinformation model with a 2-state Markov Chain. They also obtain recurrences, which were solved numerically.

First-Passage Percolation and Hopcount

- Let $G = (V, E, w)$ be a network with random edge weights

First-Passage Percolation and Hopcount

- Let $G = (V, E, w)$ be a network with random edge weights
- For any two vertices $u, v \in V$, define the *first passage time* $T_G(u, v)$ as the shortest weighted path between u and v (found, for instance, using Dijkstra's algorithm)

First-Passage Percolation and Hopcount

- Let $G = (V, E, w)$ be a network with random edge weights
- For any two vertices $u, v \in V$, define the *first passage time* $T_G(u, v)$ as the shortest weighted path between u and v (found, for instance, using Dijkstra's algorithm)
- The *hopcount* from u to v, written $H_G(u, v)$, is the number of edges in the path realizing $T_G(u, v)$.

A General Distribution Characterization

- The AoI of u is essentially the time it takes for a packet to travel from the source to u

A General Distribution Characterization

- The AoI of u is essentially the time it takes for a packet to travel from the source to u
- This travel time is captured by the language of first-passage percolation, which is the (random) distance between two points in a graph with random edge weights.

A General Distribution Characterization

- The AoI of u is essentially the time it takes for a packet to travel from the source to u
- This travel time is captured by the language of first-passage percolation, which is the (random) distance between two points in a graph with random edge weights.

Theorem (M. and Michelen 2024)

There exists a static 'dual graph' $\bar{G}$ with random edge weights such that for any node u, for large enough t the version age of u at time t has distribution given by

$$X_u(t) = \mathrm{Pois}(\lambda_e T(\bar{u}, \bar{n}_0))$$

Where $T(\bar{a}, \bar{b})$ is the minimum random distance between $\bar{a}$ and $\bar{b}$ in $\bar{G}$

First Result: Distributional Degradation

- Let $\mathcal{M}(r)$ be the distribution on the state space of $\mathcal{M}$ induced by running $\mathcal{M}$ for r steps, starting in state 0.

First Result: Distributional Degradation

- Let $\mathcal{M}(r)$ be the distribution on the state space of $\mathcal{M}$ induced by running $\mathcal{M}$ for r steps, starting in state 0.

Theorem

For every time t and node i, the quality of information of u at time t has distribution given by,

$$S_u(t) = \mathcal{M}(H_{\bar{G}}(u, n_0))$$

Where $\bar{G}$ is the dual graph of G.

Proof Outline

- Couple the AoI of u to a modified process involving $\bar{G}$.

Proof Outline

- Couple the AoI of u to a modified process involving $\bar{G}$.
- Notice that the path achieving $T_{\bar{G}}(u, n_0)$ is unique a.s.

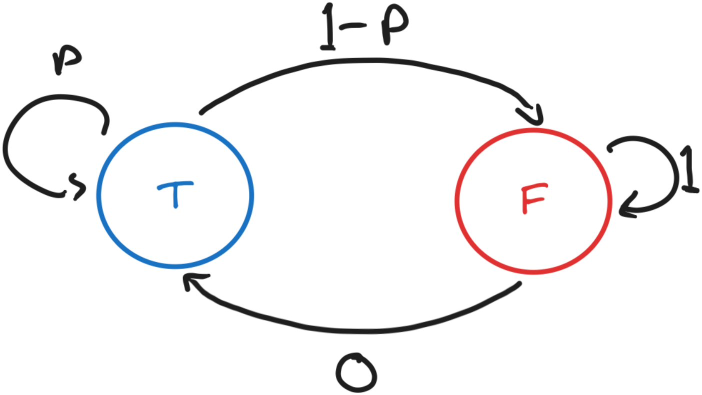
Proof Outline

- Couple the AoI of u to a modified process involving $\bar{G}$.
- Notice that the path achieving $T_{\bar{G}}(u, n_0)$ is unique a.s.
- When a packet traverses an edge on this path, the chain $\mathcal{M}$ mutates the information contained in the packet.

Proof Outline

- Couple the AoI of u to a modified process involving $\bar{G}$.
- Notice that the path achieving $T_{\bar{G}}(u, n_0)$ is unique a.s.
- When a packet traverses an edge on this path, the chain $\mathcal{M}$ mutates the information contained in the packet.
- This mutation happens once for every edge in the path, which is exactly the hopcount $H_{\bar{G}}(u, n_0)$.

Specializing to two States



Cycle and Complete Network

Theorem

The average proportion of nodes containing 'true' packets in the complete network scales as $p^{\Theta(\log n)}$.

Proof Outline:

Cycle and Complete Network

Theorem

The average proportion of nodes containing 'true' packets in the complete network scales as $p^{\Theta(\log n)}$.

Theorem

The average proportion of nodes containing 'true' packets in the cyclic network scales as $p^{\Theta(\sqrt{n})}$.

Proof Outline:

Cycle and Complete Network

Theorem

The average proportion of nodes containing 'true' packets in the complete network scales as $p^{\Theta(\log n)}$.

Theorem

The average proportion of nodes containing 'true' packets in the cyclic network scales as $p^{\Theta(\sqrt{n})}$.

Proof Outline:

- The Markov chain $\mathcal{M}$ absorbs into state F with probability $1 - p^k$ after k steps

Cycle and Complete Network

Theorem

The average proportion of nodes containing 'true' packets in the complete network scales as $p^{\Theta(\log n)}$*.*

Theorem

The average proportion of nodes containing 'true' packets in the cyclic network scales as $p^{\Theta(\sqrt{n})}$*.*

Proof Outline:

- The Markov chain $\mathcal{M}$ absorbs into state F with probability $1 - p^k$ after k steps
- Therefore any packet traversing a path γ decays to F exponentially quickly

Cycle and Complete Network

Theorem

The average proportion of nodes containing 'true' packets in the complete network scales as $p^{\Theta(\log n)}$.

Theorem

The average proportion of nodes containing 'true' packets in the cyclic network scales as $p^{\Theta(\sqrt{n})}$.

Proof Outline:

- The Markov chain $\mathcal{M}$ absorbs into state F with probability $1 - p^k$ after k steps
- Therefore any packet traversing a path γ decays to F exponentially quickly
- Compute the hopcount for K_n and C_n

Cycle and Complete Network

Theorem

The average proportion of nodes containing 'true' packets in the complete network scales as $p^{\Theta(\log n)}$*.*

Theorem

The average proportion of nodes containing 'true' packets in the cyclic network scales as $p^{\Theta(\sqrt{n})}$*.*

Proof Outline:

- The Markov chain $\mathcal{M}$ absorbs into state F with probability $1 - p^k$ after k steps
- Therefore any packet traversing a path γ decays to F exponentially quickly
- Compute the hopcount for K_n and C_n
- Plug in the hopcount for each network into this decay and average.

Comments and Open Questions

- The last two theorems are an easy corollary of hopcount analysis for K_n and C_n.
  Q: Is it true that $H_G(u, n_0) = \Theta(T_G(u, n_0))$ for any gossip network G?

Comments and Open Questions

- The last two theorems are an easy corollary of hopcount analysis for K_n and C_n.
  Q: Is it true that $H_G(u, n_0) = \Theta(T_G(u, n_0))$ for any gossip network G?
- **Q:** Are there other chains $\mathcal{M}$ that admit reasonable analysis?

Comments and Open Questions

- The last two theorems are an easy corollary of hopcount analysis for K_n and C_n.
  Q: Is it true that $H_G(u, n_0) = \Theta(T_G(u, n_0))$ for any gossip network G?
- **Q:** Are there other chains $\mathcal{M}$ that admit reasonable analysis?
- Our model plays nice with the FPP interpretation of AoI.
  Q: Analyze other models for information mutation, with more delicate interaction with the first-passage times.