ABSTRACT

Title of dissertation:	ENHANCEMENT AND ROBUSTNESS OF LARGE TIMELY GOSSIP NETWORKS
	Priyanka Kaswan, Doctor of Philosophy, 2024
Dissertation directed by:	Professor Sennur Ulukus Department of Electrical and Computer Engineering

In this thesis, we explore the subject of fast dissemination of real-time data from a source to multiple users, critical for time-sensitive applications such as autonomous driving sytems, internet of things (IoT), augmented reality (AR), virtual reality (VR), and real-time content sharing in social networks, that feature dense interconnected networks of devices and humans. In face of network resource limitations and increasingly dynamic data generated by various sources in these networks, it is imperative that all nodes within these networks have timely information, i.e., the latest possible updates about the source nodes at all times for seamless functioning of these networks. Although it might seem straightforward to transmit changing data at high speed to all users, practical challenges such as limited bandwidth, server servicing speed, and intermittent connectivity hinder this approach. Therefore, to achieve the goal of timeliness, measured by metrics such as the age of information, this thesis leverages gossip algorithms, which are decentralized algorithms whose popularity stems from their efficiency and scalablility for information dissemination in such constrained and uncertain networks. This thesis aims to deepen our un-

derstanding of the capabilities and limitations of timely gossip networks of various topologies in managing large volumes of dynamic data. As importantly, this thesis explores the unique threats and vulnerabilities these next-generation networks face in the evolving landscape of 5G and 6G technologies.

We open our analysis of this subject by first exploring efficient strategies for timely dissemination of time-varying data files from a source to a user via a simple network of parallel relays. We find that consolidating file update rates to the minimum number of relays improves timeliness, contrary to using all relays for each file. By solving an auxiliary single-cache problem and adapting its solution to our multi-cache network, we provide a sub-optimal solution, such that the upper bound on its gap to the optimal policy is independent of the number of files.

Next, we explore complex network topologies and examine the resilience of gossip networks as a function of their connectivity to jamming attacks. We analyze the average version age in the presence of $\tilde{n}$ jammers for both connectivity-constrained ring and connectivity-rich fully connected topologies of n nodes. Our findings reveal that a ring network is robust against up to $\sqrt{n}$ jammers, while a fully connected network withstands $n \log n$ jammers, showing that higher connectivity enhances resilience to jamming. To maximize age deterioration in the network, the jammers should attack in a manner that consolidates all remaining unjammed links into a dense cluster of the fewest possible nodes, leaving a higher number of nodes isolated.

Next, we uncover a new type of attack in age-based networks, called timestomping, where an adversary manipulates timestamps of information packets, causing nodes to discard fresh packets for stale ones. We show that in fully connected networks, a single infected node can increase the expected age from $O(\log n)$ to $O(n)$, highlighting how full connectivity can expedite adversarial impacts. Conversely, in unidirectional ring networks with sparse inter-node connectivity, we find that the adversarial impact on node age scaling is confined by its distance from the adversary, maintaining an age scaling of $O(\sqrt{n})$ for a significant fraction of the network.

Then, we demonstrate how the age-specific nature of file exchange protocols also makes gossip networks susceptible to the propagation of misinformation. We consider networks where packets can potentially get mutated during inter-node gossiping, creating misinformation. Nodes prefer latest versions of information, however, when a receiving node encounters both accurate information and misinformation for the same version, we consider two models: one where truth prevails over misinformation and another where misinformation prevails over truth. Using stochastic hybrid systems (SHS) modeling, we examine the expected fraction of nodes with correct information and the version age. We show that higher or lower gossiping rates effectively reduce misinformation when truth prevails, whereas moderate rates increase its spread. Conversely, misinformation prevalence rises with increased gossiping under the misinformation-prevailing scenario.

Then, we consider the balance between freshness and reliability of information in an age-based gossip network, where two sources (reliable and unreliable) disseminate updates to n network nodes. Nodes wish to have fresh information, however, they have preference for packets that originate at the reliable source and are willing to sacrifice their version age of information by up to G versions to switch from an unreliable packet to a reliable packet. We show that increasing G reduces unreliable packets but raises the network version age, revealing a freshness-reliability trade-off.

Next, we develop a theory of timeliness for non-Poisson updating and study cache-aided networks where the inter-update times on the links are not necessarily exponentially distributed. We characterize the expressions for instantaneous age and version age in arbitrary networks, then derive their closed form expressions in case of tree networks, where they exhibit an additive structure. Finally, we analyze age of information in networks where update processes on the links become sparse as network size increases, noting that in symmetric fully connected networks, expected age scales as $O(\log n)$.

Then, we study a system where a group of users, interested in closely tracking a time-varying event and maintaining their expected version ages below a threshold, choose between either preferably relying on gossip from their neighbors or directly subscribing to a server publishing about the event, to meet the timeliness requirements. The server wishes to maximize its profit by boosting subscriptions from users and minimizing event sampling frequency to reduce costs, setting up a Stackelberg game between the server and the users. We analyze equilibrium strategies in both directed and undirected networks, finding that well-connected networks have fewer subscribers since well-connected users dissuade their multiple neighboring nodes from subscribing.

Next, we consider a gossip network of n users hosting a library of files, such that each file is initially present at exactly one node, designated as the file source. The source gets updated with newer versions of the file according to an arbitrary distribution in real-time, and the other users in the network wish to acquire the latest possible version of the file. We present a class of gossip protocols that achieve $O(1)$ age at a typical node in a single-file system and $O(n)$ age at a typical node for a given file in an n-file system. We show that file slicing and network coding based protocols fall under the presented class of protocols.

Finally, we further explore timestomping attacks in a simplified communication model, where a source attempts to minimize the age of a user, but due to a power constraint, the source can only transmit updates directly to the user for a fraction of timeslots over a fixed time horizon. A cache node, which can afford more frequent transmissions, lies in between the source and the user, however the communication link between the cache and the user is under attack by a timestomping adversary. We formulate this adversarial cache updating problem as an online learning problem and study the achievable competitive ratios for this problem.

Enhancement and Robustness of Large Timely Gossip Networks

by

Priyanka Kaswan

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2024

Advisory Committee:
Professor Sennur Ulukus, Chair/Advisor
Professor Anthony Ephremides
Professor Behtash Babadi
Professor Dmitry Dolgopyat
Professor Richard J. La

Dedication

To my parents, Balbir Singh and Taramani, for their unwavering trust and support.

Acknowledgments

I am immensely thankful to my advisor, Professor Sennur Ulukus, for numerous reasons. First and foremost, I want to express my gratitude for her belief in my potential and for giving me the opportunity to pursue a PhD under her guidance. Throughout the five years of my PhD, she has been my mentor and counselor. Given the distance from my parents in India, she has also been like a parent, supporting me during difficult times. I am deeply grateful for her motivation and encouragement, which have driven me to always do my best. She was always available whenever I needed her support, including late-night research discussions, once up till 2 AM when we were thrilled about a new result I got. She has been a wonderful role model, and witnessing her progress and achievements has inspired me to take on new challenges with confidence. I would not have discovered my potential without her guidance and encouragement.

I would like to thank Professors Anthony Ephremides, Behtash Babadi, Dmitry Dolgopyat, and Richard J. La for serving on my dissertation committee and offering their valuable feedback. I am especially grateful to Professor Anthony Ephremides for his insightful comments on my proposal and career mentoring, and to Professor Behtash Babadi for his generous support and advice throughout my PhD. I am thankful to all the professors I have interacted with over the past years at UMD, particularly Professors Nuno Martins and Dmitry Dolgopyat, who taught me some of the most interesting and challenging courses and generously and patiently spent time addressing my questions during office hours. I deeply appreciate the support, appreciation, and encouragement from Professor André Tits, in his capacity as Faculty Advisor to the ECEGSA student body. I also gratefully acknowledge Dr. Mohammad Marzban from Qualcomm Technologies Inc., San Diego, USA for his appreciation and support in my academic journey, and Professor Tamer Başar and Professor Seyed Rasoul Etesami from UIUC for our fruitful research collaborations.

I am sincerely thankful to all my lab mates at UMD for creating a happy and supportive work environment. Special thanks to Purbesh Mitra, Sagnik Bhattacharya, Matin Mortaheb, Melih Bastopcu, Sajani Vithana, Baturalp Buyukates, Batuhan Arasli, Ajaykrishnan Nageswaran, Brian Kim, Cemil Vahapoglu, Arunabh Srivastava, Shreya Meel, Subhankar Banerjee, Lei Hu, Ismail Cosandal, Stavros Mitrolaris, Rustu Karakaya, Mohamed Nomeir, Alptug Aytekin, Sahan Liyanaarachchi, and Mustafa Doger. I am deeply grateful to Sajani Vithana and Melih Bastopcu for their invaluable mentorship and guidance throughout my PhD.

I am thankful to my house cat, Fidel, who was a constant companion and played a significant role in helping me complete my assignments and research. I am also grateful to my roommate, Nicholas Torre, who has been a constant support throughout my PhD journey, standing by me through both the difficult and joyous phases. I am deeply grateful to my labmate Purbesh Mitra, with whom I have shared a profound friendship during my time at UMD. His intelligence, inspirational attitude, and camaraderie have been invaluable, and I owe much of my enthusiasm and eagerness to progress in my career to this wonderful person. I am also very thankful to all my friends at UMD—Sagnik Bhattacharya, Yekanth Chalumuri, José Hannan, Joseph Westra, Meghan Lendhe, Vishal Ishwar Dodamani, Franklin L. Nouketcha, Arunabh Srivastava, Shwetank Srivastava and Isabella Romeyn Scribner—for creating a warm and homely environment. Special thanks to the ECE staff members, particularly Emily Irwin, for her constant kindness and willingness to help. There are countless others who have added color and encouragement to my life, and I thank them all.

Finally, I would like to thank my parents, Balbir Singh Kaswan and Taramani Choudhary. Their unwavering support has enabled me to achieve so many firsts in my extended family and friends circle: the first to earn an engineering degree, clear the IIT entrance exam, pursue a doctorate, travel abroad and live in a foreign country all by myself. I understand the sacrifices they made and the mindset shifts they underwent to trust my decisions and allow me to grow into an independent woman. I also want to acknowledge my loving younger, though way taller, brother, Ashish Kaswan, who has always appreciated my strengths, even though he enjoys teasing me regularly.

Thank you all for being a part of this incredible journey.

Table of Contents

List of Figures x

List of Tables xvi

1 Introduction 1

2 Timely Cache Updating in Parallel Multi-Relay Networks 15
2.1 Introduction 15
2.2 System Model and Freshness Function 19
2.2.1 Freshness for File i in the Single-Cache Model 20
2.2.2 Freshness of File i in the Multi-Cache Model 24
2.3 Structure of the Optimal Policy 28
2.4 The Optimization of Binary Freshness in Parallel Relay Networks 37
2.5 The SHS Method to Characterize Timeliness in Parallel Cache Systems 40
2.5.1 The Characterization of Binary Freshness by Using the SHS Method 40
2.5.2 The Characterization of Version Age by Using the SHS Method 43
2.6 The Optimization of Version Age in Parallel Relay Networks 45
2.6.1 Optimization of Version Age in Serially Connected Cache Systems 46
2.6.2 Optimization of Version Age in Parallel Connected Cache Systems 49
2.7 Numerical Results 52
2.8 Conclusion 58

3 How Robust are Timely Gossip Networks to Jamming Attacks? 59
3.1 Introduction 59
3.2 Preliminaries 62
3.3 Version Age in a Line Network 64
3.4 Bounds on Age of Line Network 67
3.4.1 Lower Bound 67
3.4.2 Upper Bound 68
3.5 Jammer Positioning for Age Deterioration in Ring Network 69

3.6 Average Age of Ring Network with Jammers 73
3.6.1 Lower Bound on System Age 73
3.6.2 Upper Bound on System Age 74
3.7 Fully Connected Network 75
3.7.1 Greedy Approach 82
3.8 Numerical Results 93
3.9 Conclusion 97

4 Timestomping Vulnerability of Age-Sensitive Gossip Networks 99
4.1 Introduction 99
4.2 System Model and SHS Characterization 103
4.3 Fully Connected Network 106
4.3.1 FCN Case 1: $p > 0, q = 1$ 110
4.3.2 FCN Case 2: $p = 0, q \leq 1$ 112
4.3.3 FCN Case 3: $p > 0, q < 1$ 113
4.4 MITM Attack on Fully Connected Network 116
4.5 Unidirectinal Ring Network 119
4.5.1 URN Case 1: $p > 0, q = 1$ 126
4.5.2 URN Case 2: $p = 0, q \leq 1$ 127
4.5.3 URN Case 3: $p > 0, q < 1$ 128
4.5.4 Key Insights 130
4.6 Numerical Results 133
4.7 Conclusion 138

5 Misinformation Spread in Gossip Networks: The Influence of Transmission Mutations 140
5.1 Introduction 140
5.2 System Model and SHS Characterization When Truth Prevails Over Misinformation 145
5.3 Misinformation and Version Age Derivations When Truth Prevails Over Misinformation 151
5.4 Misinformation and Version Age Derivations When Misinformation Prevails Over Truth 156
5.5 Numerical Analysis and Simulation Results 163
5.5.1 Analysis When Truth Prevails Over Misinformation 163
5.5.2 Analysis When Misinformation Prevails Over Truth 168
5.6 Conclusion 172

6 Sacrificing Freshness for Reliable Information in Age-Based Gossiping 173
6.1 Introduction 173
6.2 System Model and SHS Characterization 177
6.3 Reliability and Version Age Analysis 180
6.4 Numerical Results 209
6.5 Conclusion 213

7 Timeliness in Cache-Aided Networks with Non-Poisson Updating 215
7.1 Introduction . . . 215
7.2 Model and Notations for Age Analysis . . . 222
7.3 Age in Networks with Tree Structure . . . 225
7.3.1 Two-Hop Network . . . 225
7.3.2 Multi-Hop Network . . . 229
7.4 Version Age with Multi-Hop Renewal Updates . . . 233
7.5 Model and Notations for Version Age Analysis . . . 237
7.6 Version Age in Networks with Tree Structure . . . 240
7.6.1 One-Hop Network . . . 240
7.6.2 Two-Hop Network . . . 242
7.6.3 Multi-Hop Network . . . 245
7.7 Age of information in Networks with Rare Inter-node Updates . . . 247
7.7.1 Investigating Fully Connected Networks Via Sparse Updates in Large Networks . . . 249
7.8 Numerical Results . . . 250
7.9 Conclusion . . . 256

8 Strategic Profit Generation in Age-Based Systems 258
8.1 Introduction . . . 258
8.2 System Model and Age Analysis . . . 261
8.2.1 Average Age Analysis at the Server . . . 262
8.2.2 Game Formulation . . . 266
8.3 Directed Networks . . . 268
8.3.1 Directed Line Networks . . . 268
8.3.2 Directed Tree Networks . . . 273
8.3.3 Star Networks . . . 274
8.4 Undirected Networks . . . 281
8.4.1 Bidirectional Line Network . . . 281
8.4.2 Fully-Connected Network . . . 289
8.4.3 Undirected Networks in Higher Dimensions . . . 293
8.5 Numerical and Simulation Results . . . 298
8.5.1 Directed Networks . . . 298
8.5.2 Undirected Networks . . . 301
8.6 Conclusion and Future Directions . . . 303

9 Timely Gossiping with File Slicing and Network Coding 305
9.1 Introduction . . . 305
9.2 Gossiping with a Single File . . . 306
9.2.1 An Achievable Gossiping Protocol: INTERLEAVE . . . 311
9.3 Gossiping with Multiple Files . . . 313
9.3.1 An Achievable Gossiping Protocol: RLC . . . 316
9.4 Numerical Results . . . 317
9.5 Conclusion . . . 319

10 Age-Based Cache Updating Under Timestomping 320
10.1 Introduction 320
10.2 System Model and Problem Formulation 323
10.3 Algorithms and Analysis 326
10.3.1 Average Age for Policy $\breve{\pi}$ 336
10.3.2 Upper Bound on the Offline Optimal Policy 340
10.4 Conclusion 344

11 Conclusions 346

Bibliography 351

List of Figures

1.1 Evolution of age of information at cache and user node in a 2-hop cache-enabled network. 3
1.2 Two example gossip networks consisting of $n = 6$ nodes. 5

2.1 System models for (a) a single-cache system, and (b) a parallel multi-cache system. 17
2.2 Freshness function as a function of time (a) at the cache, and (b) at the user. 20
2.3 Finding optimal cache rates in each route for single file by reshuffling while keeping total cache rate constant. 28
2.4 Shuffling user rates for improving freshness. (a) Freshness of both files improve (file 2 only in route 2). (b) In upper branch, freshness of file 1 decreases and of file 2 increases (file 2 only in route 2). In lower branch, freshness of file 1 increases and of file 2 decreases (file 1 only in route 2). 33
2.5 Update rates (a) for a single-cache system, and (b) for a parallel multi-cache system considered in Lemma 2.5. 49
2.6 Total user rates and total cache rates for the uniform update rate policies (a) without consolidation, (b) with consolidation, (c) total user rates and total cache rates obtained from the auxiliary solution. Route allocations for files (d) for the uniform update policy without consolidation, (e) for the uniform update policy with consolidation, and (f) for our proposed strategy (files with blue rates in single routes). Freshness obtained for the uniform update policy (g) with and without consolidating update rates into single cache, (h) for the single-cache and parallel cache systems. 53
2.7 Total user rates and total cache rates obtained from the auxiliary solution (a) with binary freshness, and (b) with version age. (c) Route allocations for files (files with blue rates in single routes) with version age, and (d) version age for the single-cache and parallel cache systems. 57

3.1 (a) Ring network of n nodes. (b) Fully connected network of n nodes. 60
3.2 Line network model with n_0 nodes. 64

3.3 $S_{j,k}$ (pink blocks) and $\bar{S}_{j,k} = S_{2i-j-k+2,k}$ (blue blocks) positioned symmetrically about the dotted line between nodes i and $i+1$. 65
3.4 (a) A line network of size n_0. (b) A ring network of size n_0. Compared to (a), an extra link is introduced between end nodes bringing radial symmetry. 67
3.5 Jammer position on line network (a) most favorable, (b) most harmful. 71
3.6 Jammer positions on a ring (a) most favorable, (b) most harmful. . . 72
3.7 (a) A new link (dotted line) is being added between node i and node j, such that it increases the number of incoming link to the set $S \cup \{i\}$ (in orange background) by one. (b) Node i of degree d, connected with d nodes of degree 1. 77
3.8 A network of n isolated nodes, where all unique network configurations along with their corresponding values of $\sum_{i=1}^{6} \Delta_i$ are presented when $\bar{n}$ links are added to this network, such that, $\bar{n} = 1$ in (a), $\bar{n} = 2$ in (b), (c), and $\bar{n} = 3$ in (d), (e), (f), (g), (h). Links get consolidated and number of nodes engaged decreases from left to right in each row. 80
3.9 $k = 4$ links are added to a network size $k = 4$ mini-FC and $n - k = 5$ isolated nodes. (a) All k links are placed such that they connect node $k+1$ to all nodes of mini-FC, thereby creating a network of size $k+1 = 5$ mini-FC and $n-k = 4$ isolated nodes. The degree of node $k+1$ is $d = k$. (b) The additional k links engage multiple nodes, such that their sum of degrees $d_1 + d_2 + d_3 + d_4 = 7 \geq k$. 84
3.10 (a) A network with $\bar{n} = 6$ links, consisting of one size $k = 4$ mini-FC and 4 isolated nodes. (b) A network with $\bar{n} = 6$ links, consisting of $\bar{m} = 2$ clusters of mini-FCs each of size $\bar{k} = 3$ and 2 isolated nodes. . 89
3.11 Average age in ring network with $n^{0.3}$ jammers 92
3.12 Average age in ring network with $n^{0.8}$ jammers. 93
3.13 Blue dots represent the average age of a network of $n = 6$ nodes for all 8480 network configurations obtained by placing $\bar{n} = \binom{k}{2}, k \in \{1, \ldots, n\}$ links. The red dots represent the average age when the links are added in accordance with the greedy approach. The red dots coincide with the highest blue points for all $\bar{n}$, indicating that the greedy approach, in fact, provides the optimal placement of links that results in maximum possible average age in a network of 6 nodes. 94
3.14 Average age in fully connected network with $n \log n$ jammers placed in accordance with the greedy method. 95
3.15 The blue line corresponds to the total number of nodes n in the network, and the red line corresponds to the number of nodes k that are left as part of the mini-FC when $n \log n$ jammers are placed on fully connected network in accordance with the greedy method. . . . 96
3.16 Average age in fully connected network with $n^{1.8}$ jammers placed in accordance with the greedy method. 96

4.1 Fully connected network of n nodes with an infected node. 104

4.2 Age as function of ϵ when timestamps of outgoing packets are changed to $t-\epsilon$ instead of t by the timestomping adversary, for $n = 50$ nodes, $\lambda = 1$, $p = 1$. 115
4.3 MITM attack on fully connected network of n nodes. 116
4.4 Unidirectional ring comprising $n-1$ regular nodes and an infected node. 119
4.5 Line network transformations of unidirectional ring. 125
4.6 Node capture attack on fully connected network with $p = 0.5$ and $q = 1$. (a) Linear-linear, (b) linear-log (c) log-log plot of expected age at various nodes as afunction versus network size n, demonstrating that expected age increases linearly with network size when $q = 1$. . 132
4.7 Node capture attack on fully connected network with $p = 0$ and $q = 0.5$. (a) Linear-linear, (b) linear-log (c) log-log plot of expected age as a function versus network size n, demonstrating that expected age scales as $O(\log n)$. 133
4.8 Node capture attack on fully connected network with $p = 0.5$ and $q = 0.5$. (a) Linear-linear, (b) linear-log (c) log-log plot of expected age as a function versus network size n, demonstrating that expected age scales as $O(\log n)$. 134
4.9 MITM attack on fully connected network of n nodes. 136
4.10 Node capture attack on unidirectional ring with $p = 0.5$ and $q = 1$. . 136
4.11 Node capture attack on unidirectional ring with $p = 0$ and $q = 0.5$. . 137
4.12 Node capture attack on unidirectional ring with $p = 0.5$ and $q = 0.5$. 137

5.1 Fully-connected gossip network of n nodes with information probabilistically mutating into misinformation during inter-node gossip. . . 141
5.2 The green packet depict the packet actually stored at the node i. The orange packet depicts a mutated copy of the green packet, hypothetically available at node i at all times. 147
5.3 Solving $t_{k,m}$ with dynamic programming for $n = 4$. 154
5.4 Approach to computing (a) $r_{k,g}$ (b) $d_{k,g}$. 161
5.5 Analytical (red) and simulation (blue) results compared for F and x_1 as a function of n, with parameters $\lambda_e = 1, \lambda_s = 1, \lambda = 1, p = 0.9$. . . 164
5.6 F and x_1 as a function of λ_e, with $n = 10, \lambda_s = 1, \lambda = 1, p = 0.9$. . . . 165
5.7 F and x_1 as a function of p, with $n = 10, \lambda_e = 1, \lambda_s = 1, \lambda = 1$. 166
5.8 F and x_1 as a function of λ_s, with $n = 10, \lambda_e = 1, \lambda = 1, p = 0.9$. . . . 167
5.9 F and x_1 as a function of λ, with $n = 10, \lambda_e = 1, \lambda_s = 1, p = 0.9$. . . . 168
5.10 Analytical (red) and simulation (blue) results compared for F and x_1 as a function of (a) n, (b) λ_e, (c) p, (d) λ_s, (e) λ, with the fixed parameters chosen from $\lambda_e = 1$, $\lambda_s = 1$, $\lambda = 1$, $n = 10$, and $p = 0.9$. . 169

6.1 System with reliable (R) and unreliable (U) sources observing an event (E) and gossip network of 6 nodes, depicting an inter-node transition from node 5 (possessing unreliable packet) to node 4 (possessing reliable packet). Node 4 accepts the incoming packet only if this action results in a decrease in version age at node 4 by at least G versions. 174
6.2 Approach to computing (a) $c_{k,g}$ and (b) $d_{k,g}$. 183
6.3 Analytical and simulation results as a function of G for (a) expected fraction of users with unreliable information F, (b) expected version age x_1, (c) trade-off between F and x_1 induced by G. 210
6.4 (a) F and (b) x_1 as a function of network size n, with $\lambda_E = 2$, $\lambda_U = 5$, $\lambda_R = 1$, $\lambda = 0.1$ and $G = 10$. 211
6.5 F and (b) x_1 as a function of gossiping rate λ, with $\lambda_E = 2$, $\lambda_U = 5$, $\lambda_R = 0.1$, $n = 50$ and $G = 10$. 211
6.6 F and (b) x_1 as a function of reliable source update rate λ_R, with $\lambda_E = 2$, $\lambda_U = 50$, $\lambda = 0.1$, $n = 50$ and $G = 10$. 212
6.7 F and (b) x_1 as a function of unreliable source update rate λ_U, with $\lambda_E = 2$, $\lambda_R = 1$, $\lambda = 0.1$, $n = 50$ and $G = 10$. 212
6.8 F and (b) x_1 as a function of event update rate λ_E, with $\lambda_U = 5$, $\lambda_R = 1$, $\lambda = 0.1$, $n = 50$ and $G = 10$. 213

7.1 Multi-hop multi-cast tree network. 216
7.2 (a) One-hop model, consisting of source (node 0) and user (node 1). (b) Red dots show time instants of packet arrival at user from source. 217
7.3 (a) Two-hop model. (b) Superposition of two renewal processes, $N^{(0,1)}(t)$ and $N^{(0,1)}(t)$. 218
7.4 Evolution of instantaneous age of information $X_1(t)$ and $X_2(t)$, at node 1 and 2, respectively, till time t in a two-hop network. The red dots represent the time instants when updates arrive at node 2 from node 1, while the blue dots represent the time instants when updates arrive at node 1 from node 0. The instantaneous age at node 2 at time t is composed of two terms: the top part is $A^{(1,2)}(t)$, which arises from the $N^{(1,2)}(t)$ renewal process on the incident link $(1, 2)$, and the bottom part is $A^{(0,1)}(t - A^{(1,2)}(t))$, which depends on both the renewal processes $N^{(0,1)}(t)$ and $N^{(1,2)}(t)$. Hence, age evolution at node 2 after the red dot at time instant $t - A^{(1,2)}(t)$ cannot be characterized independently of the time before the red dot, since the height of the blue shaded triangle depends on the past. Likewise, age at the last blue dot time instant $t - A^{(0,1)}(t)$ is determined by time before the blue dot, highlighting correlations between different inter-renewal intervals. Notice how $X_1(t) \leq X_2(t)$, since packets arrive at node 2 from the single preceding node 1. 219
7.5 n-hop model. 229
7.6 Time segments $\Delta_i(t)$ for $i \in \{1, \ldots, n\}$. 230
7.7 Multi-hop multi-cast tree network with version updates at source. . . 233

7.8 One-hop model, where the sources gets updates according to a Poisson process with rate λ_s and updates arrive at node 1 according to a renewal process with typical inter-renewal period Y. 234
7.9 Superposition of renewal processes, $N^{(0,0)}(t)$ and $N^{(0,1)}(t)$, in the one-hop model. 235
7.10 Superposition of source renewal process $N^{(0,0)}(t)$ on the two renewal processes, $N^{(0,1)}(t)$ and $N^{(0,1)}(t)$ in the two-hop model. 236
7.11 Time segments $\Delta_i(t)$ for $i \in \{1, \dots, n\}$ in n-hop model. The points marked by red represent version updates at the source. 245
7.12 Fully connected network of n nodes, where node i transmits update packets according to $N_i(t)$ renewal process. 248
7.13 The dotted blue curve shows the expected age $\mathbb{E}[X_n(T)]$ in n-hop network with $Y^{(i,i+1)} \sim \mathcal{U}_{[0,2]}$. The solid blue curve shows the expected version age $\mathbb{E}[X_n(T)]$ in n-hop network with $Y^{(i,i+1)} \sim \mathcal{U}_{[0,2]}$ and $Y^{(0,0)} \sim Pareto(3, \frac{1}{3})$. The dotted red and solid red curves depict the expected age and expected version age, respectively, when all renewal processes are approximated as Poisson processes with the same distributional mean of update intervals on all links. 252
7.14 The dotted blue curve shows the expected age $\mathbb{E}[X_n(T)]$ in 4-hop network with $Y^{(i,i+1)} \sim \mathcal{U}_{[1-\sqrt{3v},1+\sqrt{3v}]}$. The solid blue curve shows the expected version age $\mathbb{E}[X_n(T)]$ in 4-hop network with $Y^{(i,i+1)} \sim \mathcal{U}_{[1-\sqrt{3v},1+\sqrt{3v}]}$ and $Y^{(0,0)} \sim Pareto(3, \frac{1}{3})$. 253
7.15 Expected version age $\mathbb{E}[X_n(T)]$ in 3-hop network with $Y^{(0,0)} \sim Pareto(3, m)$, $Y^{(0,1)} \sim Rayleigh(1)$, $Y^{(1,2)} \sim \chi^2(1)$ and $Y^{(2,3)} \sim Beta(2, 3)$, as plotted against different values of $\mathbb{E}[Y^{(0,0)}]$ obtained by varying m. 254
7.16 (a) The blue curve shows the expected age at a node 1 in a symmetric fully connected network where all update processes have inter-update times distributed as $\mathcal{U}_{[0,1]}$. The red curve shows the predicted expected $\mathbb{E}[Y] \times \left(\sum_{i=1}^{n} \frac{1}{k}\right)$ for large n in a fully connected network. (b) The gap $\mathbb{E}[Y] \times \left(\sum_{i=1}^{n} \frac{1}{k}\right) - \mathbb{E}[X_1(T)]$ decreases as n becomes large. . 256

8.1 System model consisting of a server and user nodes. The event process is updated with probability p_e and the server gets updates with rate β.259
8.2 Directed line network where the users are subscribing to the server's information in every K users. 269
8.3 (a) F_S and (b) K values at different β. 271
8.4 Directed tree network with $K = 2$ levels. 273
8.5 Equilibrium subscription strategies on star network as functions of β. 275
8.6 An illustration of a typical two-way line network. 282
8.7 $S_{j,h}$ (pink blocks) and $\bar{S}_{j,h} = S_{2i-h-1,2i-j-1}$ (blue blocks) positioned symmetrically about the dotted line between nodes i and $i+1$. 283
8.8 Sets $\{j_1, \dots, m-h_1\}$ with $j_1 = 2$ and $h_1 = 2$, have the same expected age irrespective of the value of m. 285
8.9 A typical fully-connected network with m subscribers. 289
8.10 $\mathcal{S}_2$ with half of the population as subscriber. 293

8.11 Each node in $\mathcal{S}_m$ is surrounded by 2^m nodes. 294
8.12 Age evolution in one-way line network when $L = 10$, $p = 0.2$, $\beta = 0.6$, and $p_e = 0.3$. For all users to be AC-stable, users in $5\mathbb{N}$ will have to subscribe. 299
8.13 The comparison of the server's optimal β^*, subscription rate F_S, and the server's optimal cost function for the line, tree, and star networks when $L = 10$, $p = 0.2$, $\beta = 0.6$, and $p_e = 0.3$. 300
8.14 The expected age of nodes in a two-way (bidirectional) line network with $m = \{7, 14\}$. 300
8.15 The expected age of a node in a fully-connected network of $n = 10$. . 301
8.16 The comparison of the server's optimal β^*, subscription rate F_S, and the server's optimal cost function for two way-line network and fully connected network. 302
8.17 The comparison of the server's optimal β^*, subscription rate F_S, and the server's optimal cost function in m-dimensional network. 302

9.1 Sample version age evolution at typical node i, with parameters $c = 3$, $\ell = 4$. The red curve corresponds to Y_t^i which remains above $2c = 6$, and increases by $c = 3$ whenever the node is unsuccessful in receiving the update in the preceding cycle. The blue curve corresponds to X_t^i which drops to lower values in the minislot after it gathers all pieces of a particular file version. 307
9.2 Version age versus network size n with INTERLEAVE protocol. . . . 317
9.3 Version age versus network size n with RLC with PUSH mechanism. 318
9.4 (Version age)/n versus n with RLC with PUSH mechanism. 318

10.1 System model for caching system. 321
10.2 Suggested deterministic algorithm, dividing timelines into $T_1 + 1$ sections of equal size. $T_1 = 3$, $T_2 = 6$, $T = 24$. 326
10.3 Optimal deterministic algorithm, dividing timelines into $T_2 + 1$ sections of equal size. $T_1 = 2$, $T_2 = 4$, $T = 27$. 335
10.4 A policy $\breve{\pi}_1$ to find an upper bound on the offline optimal policy, dividing timelines into $T_2 + 1$ sections of equal size. $T_1 = 2$, $T_2 = 4$, $T = 27$. 341

List of Tables

3.1 $R_d/\frac{\lambda_s}{\lambda}n$ values (rounded down to nearest 2 decimal point number) for $d = 1, \ldots, 22$. 79
3.2 Minimum age reduction with 3 node engagement and 7 additional links. 86
3.3 Minimum age reduction with 2 node engagement and 23 additional links. 87

7.1 Distributions, means, variances, and contributions to age. 251
7.2 Mapping of distributions to links. 251

CHAPTER 1

Introduction

Next-generation wireless networks are anticipated to be characterized by hyperconnectivity between humans and machines. With the advent of 5G and 6G, we foresee the proliferation of time-sensitive technologies, such as driverless cars, smart homes, and dense social networks, where a massive number of devices (alternatively, users or nodes) are connected to each other, all of which demand real-time interaction for their efficient functioning. For instance, in autonomous driving systems, timely communication with nearby connected devices—such as other cars, sensors, roadside infrastructure, and smartphones—is crucial for accurately performing driving actions and avoiding accidents. Similarly, in smart factory environments, a collaborative system involving humans, robots, drones, and cameras is essential for effectively performing manufacturing tasks.

In these systems, a source generates time-varying information that is of critical interest to a user or a group of users. Users aim to track this time-varying information as closely as possible in real-time to achieve their objectives. While the problem might seem straightforward—merely transmitting the changing data at high speed from the source to all users—practical applications face significant hurdles in disseminating the realistically enormous volumes of dynamic data to masses of users. These include limited bandwidth, servicing speed at servers, and intermittent or uncertain connectivity between users, resulting in delays and outdated information being delivered to users, which is unacceptable in time-sensitive applications. Consequently, it is crucial to optimize communication parameters such that minimum staleness is maintained in the system despite these limitations.

To achieve this, we first need a metric that effectively captures the staleness of information in time-sensitive applications. One such metric, proposed in the literature, is the age of information (AoI) [1–4]. For the latest information packet present at a user node at time t that has the generation time of $u(t)$ at the source, the instantaneous AoI $\Delta(t)$ at the user is defined as $\Delta(t) = t - u(t)$. This simple metric essentially indicates how long ago a user's current packet was generated at the source. The AoI of a user increases at a unit rate as time progresses, until it receives a new packet from the source with a different generation time; see Fig. 1.1. Ideally, a user would want $\Delta(t) = 0$ at all times, but this is unattainable due to the limitations mentioned earlier. Therefore, it is desired to keep the AoI as low as possible. Since AoI is a time-dependent quantity, most literary works focus on optimizing either the time-average age, or peak-age, or some other statistical property of the age process at the user. Over an interval $[0, T]$ with large T, the average age is defined as

$$\Delta = \limsup_{T\to\infty} \frac{1}{T}\int_0^T \Delta(t)dt. \tag{1.1}$$

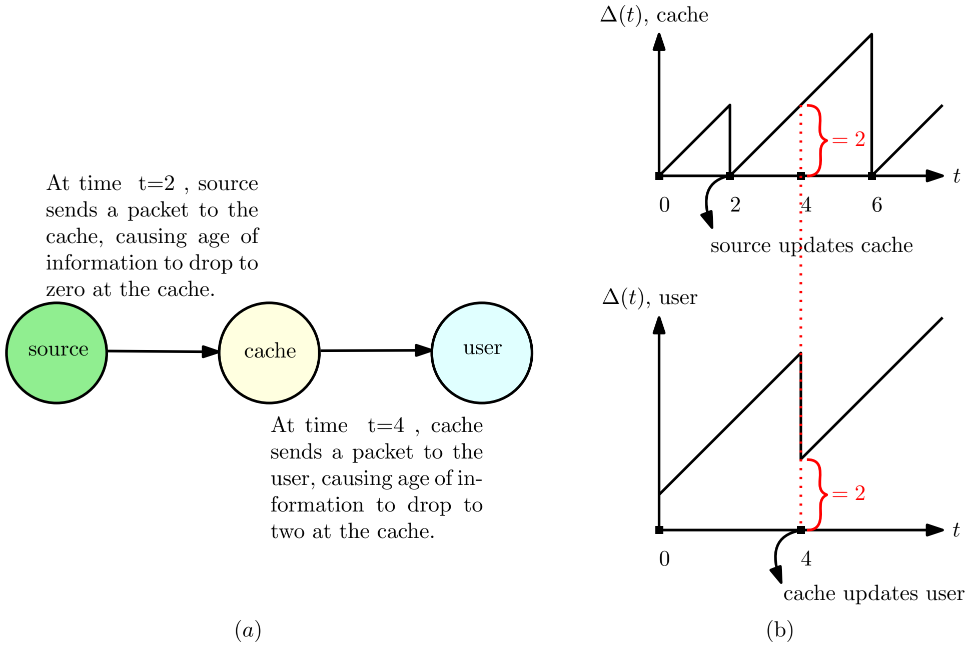
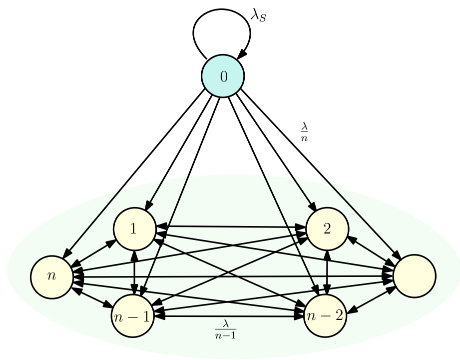


Figure 1.1: Evolution of age of information at cache and user node in a 2-hop cache-enabled network.

Graphically, the time-average age is the area under the saw-tooth curve in Fig. 1.1, normalized by the interval of observation. In addition to age of information, other timeliness metrics commonly used in the literature are version age of information [5,6], binary freshness [7–9] and age of incorrect information [10], which become more relevant when the information at the source only gets updated at certain times, say defined by a renewal point process, and stays constant at the source in the inter-update periods. The mathematical notation corresponding to these metrics varies in different chapters of this thesis due to the nature of diverse problem settings, and their real-time evolution is discussed in more detail in the later chapters.

The idea of AoI was first introduced in [11] in the context of vehicular networks and was generalized in the context of communication systems in [12]. Before delving into the details of AoI formulation and its applications, it is essential to understand why AoI is a necessary metric and why traditional metrics like delay and throughput, which have been extensively studied [13–17], fall short in time-sensitive applications. [12] illustrates the novelty and relevance of AoI through a practical example. Consider a vehicle where sensor measurements generated by various sensors are aggregated into a status update message and queued while they wait to be serviced by the car radio and transmitted to other cars. If the car radio interface receives a high volume of update messages rapidly (high throughput), thereby keeping the server always busy, it causes the queue to be backlogged with status update messages, leading to messages getting stale or outdated by the time of their delivery. Conversely, to minimize the waiting time of packets in the queue (low delay), one approach is to send packets infrequently, thereby keeping the queue empty. However, in this case, the other cars do not receive updates from this vehicle frequently enough, causing them to have outdated information about this vehicle. Clearly, traditional metrics such as throughput and delay struggle to accurately capture the semantic property of the timeliness of updates, thus motivating the need for timeliness metrics such as the age of information. Recently, numerous studies have explored age-optimal policies in a wide range of contexts, such as, queueing networks [18–21], energy harvesting systems [22–29], web crawling [30, 31], scheduling problems [32–44], UAV systems [45, 46], remote estimation [5, 7, 47–66], gossip networks [9, 67, 68], and so on.

In this thesis, we focus on the age of information in the context of a gossip network (see e.g., Fig. 1.2), which sometimes is called, the *age of gossip*. In large networks featuring asynchronous contacts between nodes, it is often not feasible



(a) Fully connected network. (b) Ring network.

Figure 1.2: Two example gossip networks consisting of $n = 6$ nodes.

for a source node to directly send updates to all nodes in a timely manner due to the large network size, uncertain network dynamics and bandwidth limitation at the source. To overcome these limitations, simple scalable decentralized protocols such as gossip protocols have been proposed in the literature for expediting the dissemination of updates.

Gossip algorithms are decentralized algorithms where two nodes of the network randomly come into contact and communicate exclusively based on the limited information available at the liaising nodes. As opposed to centralized protocols, where a centralized server coordinates or assists the exchange of files between nodes of the network, in gossip algorithms, the nodes are unaware of the information available at other nodes and all actions by a node are taken solely based on its local status or information obtained from its neighbors, thereby causing information to spread like a gossip/rumor. The idea of gossip protocols was first mentioned in [69] as an epidemic algorithm for clearinghouse database maintenance, and since then there have been numerous works on gossip protocols, e.g., [5, 9, 70–76]. For example, [71] shows that a single rumor can be spread to n nodes in $O(\log n)$ rounds. In [73], dissemination time of k messages in a large network of n nodes is studied for gossip protocols based on random linear coding (RLC), random message selection (RMS), and sequential dissemination. [73] shows that RLC-based protocol has superior performance and has $ck+O(\sqrt{k}\log k\log n)$ dissemination time in complete graphs. [74] further extends the result of [73] to arbitrary graphs. [75] studies the dissemination of a file in a large network of n nodes by dividing the file into k pieces. Therein, the authors propose a hybrid piece selection protocol, INTERLEAVE, which achieves a $O(k+\log n)$ total dissemination time for a file. These works consider the total dissemination time of a static message in the network as the performance metric.

However, data in realistic systems is not static; it keeps changing asynchronously over time as new information becomes available. Distributed databases such as Amazon DynamoDB [77] and Apache Cassandra [78], that use gossip protocols to keep their information fresh, employ timestamp versioning, wherein every new information is created with a timestamp value taken from the system clock. During a single gossip exchange between two nodes, the timestamps of data at both nodes are compared, and the node carrying the data with older timestamp discards its data for the fresher data of the other node. Hence, a specific information may get discarded or lost in the network before it can reach all nodes of the network. This renders the choice of total dissemination time as a performance metric inadequate, and in such networks, the age of information at the nodes may prove to be a more suitable performance metric. In age-based gossip protocols, the receiving node accepts a packet only if doing so results in a decrease in instantaneous age at the receiver.

Note that in timeliness applications, the nodes are assumed to have a cache buffer of size one, where each node will only store the latest packet it has received so far and discard all the old packets.

Gossip networks have been analyzed from an information timeliness perspective in [4, 5, 9, 76]. In the seminal work of [5], the recursive equation for expected version age analysis is derived using stochastic hybrid system (SHS) framework, using which, it is shown that in a fully connected network of n nodes with a single file, as shown in Fig. 1.2(a), the expected version age at each node is $O(\log n)$. Further, [5] observes through simulations that expected version age in ring network scales as $1.25\sqrt{n} = O(\sqrt{n}))$, which is later derived in [76] through certain assumptions and approximations. [76], additionally, performs parallel analysis for the expected version age in clustered gossip networks. [4] and [9] extend the SHS based analysis of [5] to the age of information metric and the binary freshness metric, respectively. This thesis aims to further improve the timeliness limits of these networks beyond the state-of-the-art results, and also test the robustness of these time-sensitive networks.

The overarching theme of this thesis is the enhancement and robustness of timely information dissemination in large gossip networks. On one hand, we focus on devising age-optimal policies that further enhance the timeliness and efficiency of these networks past the state-of-the-art results by investigating optimal paths for sending updates to users, implementing file slicing and network coding techniques, identifying optimal packet request renewal processes and implementing intelligent subscription strategies. On the other hand, we also show how a new design metric unlocks opportunities for new malfunctions and new attacks by adversaries that aim to disrupt the efficient functioning of these systems, by introducing staleness in the network. We investigate the unique threats that large age-based systems are vulnerable to, such as timestomping attacks, jamming attacks, and the propagation of misinformation and unreliable information, and how gossiping in such networks both acts as a shield against threats and also enables adversaries to propagate attacks effectively. In the remainder of this introduction, we describe each chapter of the dissertation in more detail.

In Chapter 2, we begin with devising age-optimal policies for simpler network topologies by studying timely data dissemination through a parallel network of multiple cache-enabled relay nodes. Here, we consider a system consisting of a server, which receives updates for N files according to independent Poisson processes, such that the goal of the server is to deliver the latest version of the files to a user through a parallel network of K cache nodes or relays. We derive analytical expressions for various information freshness metrics at the user. We observe that freshness for a file increases with increase in consolidation of rates across caches. Motivated by this observation, we first solve an auxiliary problem of a single-cache system. Then, we rework this auxiliary solution to our parallel-cache network by consolidating rates to single routes as much as possible, thereby yielding an approximate (sub-optimal) solution for the original multi-cache problem. We then provide an upper bound on the gap between the sub-optimal policy and the optimal policy, such that the gap is finite and is independent of the number of files.

With Chapter 3, we diverge to more complex network topologies, such as the connectivity-constrained ring topology and the connectivity-rich fully connected topology, and begin to examine the resilience of gossip networks as a function of their connectivity, by considering jamming attacks. We consider a system with a source which maintains the most current version of a file, and a network of n user nodes with the goal to acquire the latest version of the file. The source gets updated with newer file versions as a point process, and forwards them to the user nodes, which further forward them to their neighbors using a memoryless gossip protocol. Here, we study the average *version age* of the network in the presence of $\tilde{n}$ jammers that disrupt inter-node communications. We show that, in a ring network, the version age with gossiping is robust against up to $\sqrt{n}$ jammers, whereas a fully connected network is robust against $n \log n$ jammers, implying that jamming attacks are less effective against high-connectivity network topologies. A general theme of consolidation reappears here, different from that of Chapter 2, wherein we observe that to maximize age deterioration in the network, the jammers should attack in a manner such that all remaining unjammed links should be consolidated to a dense ball of minimum possible nodes, leaving a higher number of nodes isolated.

In Chapter 4, we uncover a new type of attack which uniquely affects these age-based networks, that we term as timestomping attack. Timestomping refers to timestamp manipulation by an adversary, which causes other network nodes to discard latest packets and hoard staler packets over time due to the deceptive timestamps. Chapter 4 considers gossip networks consisting of a source that maintains the current version of a file, n nodes that use asynchronous gossip mechanisms to disseminate fresh information in the network, and an oblivious adversary who infects the packets at a target node through data timestamp manipulation. We demonstrate how network topology capacitates an adversary to influence age scaling in a network, and show that in a fully connected network, a single infected node increases the expected age from $O(\log n)$ to $O(n)$. This demonstrates how the fully connected nature of a network can be both a benefit and a detriment for information freshness; full connectivity, while enabling fast dissemination of information, also enables fast dissipation of adversarial inputs. We then analyze the unidirectional ring network, the other end of the network connectivity spectrum, where we show that the adversarial effect on age scaling of a node is limited by its distance from the adversary, and the age scaling for a large fraction of the network continues to be $O(\sqrt{n})$, unchanged from the case with no adversary. Therefore, contrary to the jamming attacks of Chapter 3, timestomping attacks take advantage of high network connectivity to induce more staleness in the network.

In Chapter 5, we show that the age-specific nature of file exchange protocols also makes gossip networks susceptible to the propagation of misinformation. The system model differs from the previously considered fully connected networks in that when a node forwards its packet to another node, the packet information gets mutated with probability p during transmission, creating misinformation. The receiver node is assumed to be unaware of whether an incoming packet information is different from the packet information originally at the sender node. We consider two system models: In the first model, we assume that truth prevails over misinformation, and therefore, when a receiver encounters both accurate information and misinformation corresponding to the same version, the accurate information gets chosen for storage at the node. In the second model, we assume the opposite scenario, where misinformation prevails over truth. For both models, we study the expected fraction of nodes with correct information in the network and version age at the nodes using stochastic hybrid systems (SHS) modelling and study their properties. We observe that when truth prevails over misinformation, very high or very low gossiping rates help curb misinformation, and misinformation spread is higher with moderate gossiping rates. However, when misinformation prevails over truth, misinformation rises with increased inter-node gossiping.

In Chapter 6, we consider a system model with two sources, a reliable source and an unreliable source, who are responsible for disseminating updates regarding a time-varying process to an age-based gossip network of n nodes. Though nodes in this model wish to have fresh information, they have preference for packets that originate at the reliable source and are willing to sacrifice their version age of information by up to G versions to switch from an unreliable packet to a reliable packet. We study how this protocol impacts the prevalence of unreliable packets at nodes in the network and their version age. Using an SHS framework, we formulate analytical equations to characterize two quantities: expected fraction of nodes with unreliable packets and expected version age of information at network nodes. We show that as G increases, fewer nodes have unreliable packets, however, their version age increases as well, thereby inducing a freshness-reliability trade-off in the network. We additionally investigate the dependence of network reliability and freshness on various network parameters, such as inter-node update rates and the network size.

In Chapter 7, we develop a theory of timeliness for non-Poisson updating and study cache-aided networks where the inter-update times on the links are not necessarily exponentially distributed. We focus on the set of non-arithmetic distributions for inter-update times, which includes continuous probability distributions as a subset. We first characterize instantaneous age of information at each node for arbitrary networks. We then explicate the recursive equations for instantaneous age of information in multi-hop networks and use them to derive closed form expressions for expected age of information at an end-user in tree networks. We show how the expected age at each user is directly proportional to the variance of the inter-update times at all links between a user and the source, and exhibits an additive structure. We next prove analogous results for the version age of information metric, where we additionally find that the expected version age at end-users is inversely proportional to the mean update interval at the source. Finally, we study expected age of information in networks with the property that the update processes on the links become sparse for large network sizes, and remark that expected age scales as $O(\log n)$ in symmetric fully connected networks. We expect this analysis to help alleviate the over-dependence on exponential inter-update time (i.e., Poisson) updates for future work in age of information.

In Chapter 8, we consider a communication system consisting of a server that tracks and publishes updates about a time-varying data source or event, and a gossip network of user nodes interested in closely tracking the event. The users, on one hand, wish to have their expected version ages remain below a threshold, and have the option to either rely on gossip from their neighbors or subscribe to the server directly to follow updates about the event if the former option does not meet the timeliness requirements. The server, on the other hand, wishes to maximize its profit by increasing the number of subscribers and reducing costs associated with the frequent sampling of the event. We model the problem setup as a Stackelberg game between the server and the users, where the server is the leader deciding its frequency of sampling the event, and the users are the followers deciding whether to subscribe or not. We first focus on directed networks with unidirectional flow of information and obtain the optimal equilibrium strategies for all the players, and later extend to the general case of undirected networks with bidirectional flow of information on inter-node links. We observe that well-connected networks have fewer subscribers, since each well-connected user discourages several neighboring nodes from subscribing directly to the source.

Chapter 9 focuses on improving age scaling in dense networks with file slicing and network coding techniques. Here, we consider a system consisting of a large network of n users and a library of files, wherein inter-user communication is established based upon gossip mechanisms. Each file is initially present at exactly one node, which is designated as the file source. The source gets updated with newer versions of the file according to an arbitrary distribution in real time, and the other users in the network wish to acquire the latest possible version of the file. We present a class of gossip protocols that achieve $O(1)$ age at a typical node in a single-file system, a significant improvement from the conventional age-based gossiping, which achieves $O(\log n)$ age at network nodes, i.e., nodes become staler as network size grows. Similarly, we present a class of gossip protocols that achieve $O(n)$ age at a typical node for a given file in a n-file system, an improvement over the $O(n \log n)$ age achieved through conventional gossip protocols. We show that file slicing and network coding based protocols fall under the presented class of protocols.

In Chapter 10, we explore timestomping attacks in a simplified communication model that consists of a source, a cache and a user. The time horizon consists of total T time slots, such that the source, due to power constraint, transmits update packets to the user directly over only T_1 time slots and to the cache over only T_2 time slots. In this setting, we consider the presence of an oblivious adversary that fully controls the communication link between the cache and the user. The adversary manipulates the timestamps of outgoing packets from the cache to the user, with the goal of bringing staleness at the user node. At each time slot, the adversary can choose to either forward the cached packet to the user, after changing its timestamp to current time t, thereby rebranding an old packet as a fresh packet and misleading the user into accepting it, or stay idle. If the user receives update packets from both cache and source in a time slot, then the packet from the source prevails. The goal of the source is to design an algorithm to minimize the average age at the user, and the goal of the adversary is to increase the average age at the user. We formulate this problem in an online learning setting and provide a fundamental lower bound on the competitive ratio for this problem. We further propose a deterministic algorithm with a provable guarantee on its competitive ratio.

In Chapter 11, we present conclusions of this dissertation.

CHAPTER 2

Timely Cache Updating in Parallel Multi-Relay Networks

2.1 Introduction

In this chapter, we consider information timeliness in parallel relay networks. The parallel multi-cache system model can have important applications in modern vehicular networks. For example, 5G-enabled vehicular networks, wherein several self-sustaining wirelessly connected caching stations are placed to enhance vehicular network capacity, has been considered in [79]. Especially with a large number of vehicles in an area and considering in practice that these cache stations may have limited power capacities, in order to provide continuous coverage in a vehicular network, multiple caching stations can be placed such that their coverages may intersect with each other. In these intersection areas, a vehicle may have access to multiple base stations to communicate. For such a wirelessly connected parallel caching systems, in this chapter, we address how to enable timely communication with the caches that have limited communication capacities.

The works that are most closely related to the work in this chapter are [5, 7, 9, 28, 29, 43, 44, 61–66, 68]. In [43], a single-server single-cache refresh system is considered, where it is shown that an asymptotically optimal policy updates a cached file in proportion to the square root of its popularity. The work in [43] assumes constant file update durations, which is extended in [44] by considering file update durations to be dependent on the size and the age of the files. While [43,44] use the AoI metric, reference [7] uses a binary freshness metric in a caching system, and determines the optimum update rates at the user and the cache. [7] also extends the approach to a cascade sequence of cache nodes, and [61] generalizes it to the case of nodes with limited cache capacity. In this chapter, we further generalize [7] to a more complex network which is composed of parallel caches, shown in Fig. 2.1(b), where multiple cache nodes are available for relaying file packets from source to the user. Contrary to the above-mentioned works, which optimize timeliness specifically in line networks, where each node of the network has a unique predecessor node responsible for supplying it with fresh packets, in parallel networks this convenience is not at our disposal anymore, making the analysis more challenging. The stochastic hybrid system (SHS) approach has been used to characterize timeliness of the nodes in gossip networks in [5,68] with the version age metric, and in [9] with the binary freshness metric. These earlier works [5,9,68] are substantially different from this work and mostly focus on the scaling of information freshness in deeply connected gossip networks. In this chapter, in addition to presenting closed form expressions and policies with binary freshness using first principles, we also use the SHS method to set forth parallel formulations for version age in parallel relay networks. Compared to the earlier works, this is the first work that considers characterization and then optimization of the timeliness metrics in parallel caching networks.

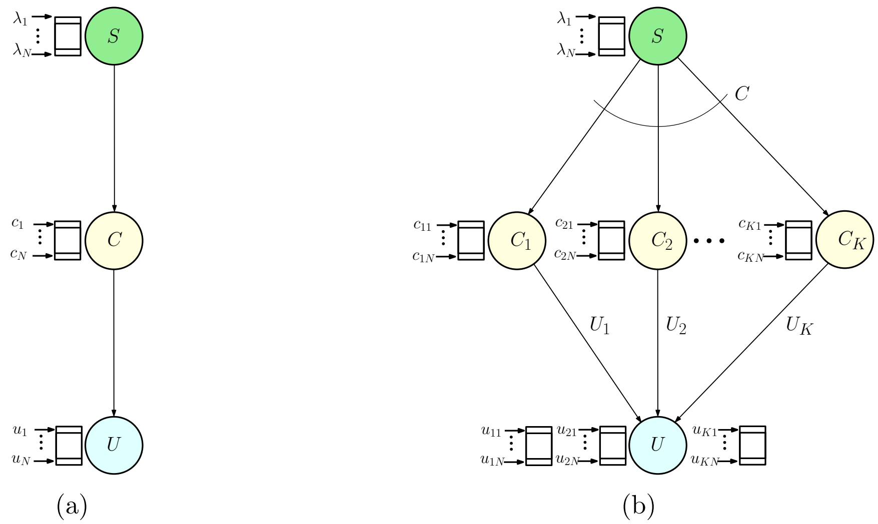


Figure 2.1: System models for (a) a single-cache system, and (b) a parallel multi-cache system.

Other related work that use caching and relaying techniques for freshness include: [62] where a tradeoff between content freshness and service latency from the aspect of mobile edge caching is studied; [63] which considers caching policies in opportunistic networks; [28] where a cache-enabled aggregator decides whether to receive a fresh update from an energy harvesting sensor or serve the request with a cached update; [64] where an optimal policy is derived when the current rate of requests for a file is dependent on both the history of requests and the freshness of the file; [65] which considers a two-hop status update system where an optimal scheduling policy is identified by a constrained Markov decision process approach; and [29] where a two-hop system with energy harvesting at source and relay nodes is considered.

The main contributions of this chapter can be summarized as follows. In this work, we consider a parallel network with multiple cache routes[1] between a source and a user; see Fig. 2.1(b). In Section 2.2, we first derive a closed-form expression for freshness at the user. Then, our goal is to maximize the overall freshness at the user subject to the constraints on the total update rates at the caches and at the user. For that, in Section 2.3, we observe from the freshness formula of the two-cache system that lop-sided distribution of rates across the routes supports higher freshness. Further, for the two-route two-file case, restricting at least one of the files to a single route maximizes the overall freshness of the system. Moreover, in a K-cache system, restricting a file to fewer routes improves the freshness. Motivated by these properties, in Section 2.4, we solve an auxiliary problem of a single-cache system and adapt its solution to our parallel cache network to obtain an approximate (sub-optimal) solution for the original problem. We provide an upper bound on the gap between the sub-optimal policy and the optimal policy. The gap is finite and is independent of the number of files. In Section 2.5, we extend our results from the binary freshness metric to the version age of information metric by using the SHS method. Then, we obtain analogous results in parallel networks with the version age of information metric. In Section 2.6, we optimize the version age metric both in serial and parallel relay networks. Finally, in Section 2.7, numerical results for both timeliness metrics show that the proposed sub-optimal policies closely approximate the optimal policy.

[1]The file transfer path from the source node to the user node via cache node k is referred to as route k in this work. Further, the words cache, cache node and relay are used interchangeably.

2.2 System Model and Freshness Function

This work is based on the system model depicted in Fig. 2.1(b). The system consists of a source, K parallel relays and a user. The source has most up-to-date versions of a library of N files. Update packets for file i arrive at the source according to a rate λ_i Poisson process. The goal is to update the user through the parallel network of K relays that have cache memories. The source in turn offers updates for file i to cache k according to a rate c_{ki} Poisson process. We assume that there is no delay or information loss in any source-cache links or cache-user links.[2] The source is subject to a total update rate constraint $\sum_{k=1}^{K}\sum_{i=1}^{N} c_{ki} \leq C$, emanating from energy or cost limitations in real-world wireless networks. The cache k sends updates for file i to the user according to a rate u_{ki} Poisson process and is subject to total update rate constraint $\sum_{i=1}^{N} u_{ki} \leq U_k$, for $k = 1, \dots, K$.

When a file is updated at the source, the stored versions of the same file at the caches and at the user become outdated. Thus, we consider an update received by the user successful, if the user receives a file version that is currently prevailing at the server. This will happen when the source updates the cache and the cache in turn updates the user before the source gets updated with a newer version. In the following subsections, we first derive freshness expression for file i in a single-cache model, and later in a multi-cache model by using a probabilistic approach.[3] For

[2]Transmission times and information loss in links considered in [80] can be assumed to be negligible if the distance between the source and the caches are small and/or the file sizes are relatively small compared to the transmission capacity as argued in [7, 81].

[3]By using the stochastic hybrid system (SHS) approach, we provide an alternative way to characterize binary freshness in Section 2.5.1.

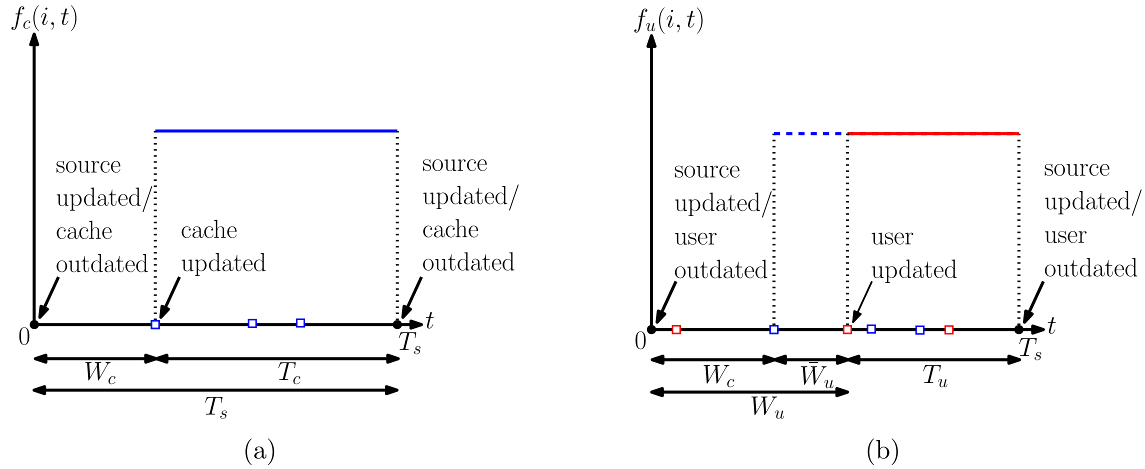


Figure 2.2: Freshness function as a function of time (a) at the cache, and (b) at the user.

simplicity, we drop subscript i from λ_i, c_{ki} and u_{ki} in the following subsection since the derivation is valid for all files (for all i).

2.2.1 Freshness for File i in the Single-Cache Model

In this subsection, we calculate the freshness expression for file i for the single-cache system shown in Fig. 2.1(a). First, we characterize the freshness at the cache. In Fig. 2.2(a), the freshness evolution at the cache is shown between two file updates at the source. We define the freshness function for file i at the cache as follows

$$f_c(i,t) = \begin{cases} 1, & \text{if file } i \text{ at the cache is fresh at time } t, \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

Let $T_s(i,j)$ denote the jth update cycle at the source, i.e., time interval between the jth and $(j+1)$th update for file i. Once the source gets updated, the cache is updated after duration $W_c(i,j)$ and it remains updated for $T_c(i,j) = T_s(i,j) - W_c(i,j)$ duration. For simplicity, we drop index i for variables $T_s(i,j)$, $T_c(i,j)$, and $W_c(i,j)$, as the results in this subsection pertain to file i. The long term average freshness of file i at the cache denoted by $F_c(i)$ is equal to

$$F_c(i) = \lim_{T \to \infty} \frac{1}{T} \int_0^T f_c(i,t) dt. \tag{2.2}$$

Let M be the number of update cycles in time duration T. Provided that the system is ergodic, similar to [7], $F_c(i)$ can be equivalently written as

$$F_c(i) = \lim_{T \to \infty} \frac{M}{T} \left(\frac{1}{M} \sum_{j=1}^{M} T_c(j) \right) = \frac{\mathbb{E}[T_c]}{\mathbb{E}[T_s]}. \tag{2.3}$$

Here, as $T_c(j)$ are independent and identically distributed (i.i.d.) over j, we drop the index j and denote $T_c(j)$ with the typical random variable T_c. Similarly, T_s and W_c denote the typical random variables for $T_s(j)$ and $W_c(j)$, respectively.

Since the files at the source are updated according to Poisson process, T_s is an exponential random variable with a typical update rate λ, we have $\mathbb{E}[T_s] = \frac{1}{\lambda}$. We define W_c' as the waiting time of receiving an update at the cache, which is exponentially distributed with rate c. Then, we have $W_c = \min\{W_c', T_s\}$, as W_c either takes a value in between 0 and T_s, or takes value T_s, i.e., $W_c = T_s$, and in this case, the cache is not updated in that cycle. Since W_c is the minimum of two exponential random variables, W_c is an exponential random variable with rate $\lambda + c$

and thus, we obtain $\mathbb{E}[W_c] = \frac{1}{\lambda+c}$. Then, we obtain $\mathbb{E}[T_c]$ as

$$\mathbb{E}[T_c] = \mathbb{E}[T_s] - \mathbb{E}[W_c] = \frac{1}{\lambda} - \frac{1}{\lambda+c} = \frac{c}{\lambda(\lambda+c)}. \tag{2.4}$$

Finally, by substituting (2.4) into (2.3), we obtain $F_c(i)$ as

$$F_c(i) = \frac{\mathbb{E}[T_c]}{\mathbb{E}[T_s]} = \frac{c}{\lambda+c}. \tag{2.5}$$

Next, we characterize the freshness at the user. Freshness evolution at the user in an update cycle is shown in Fig. 2.2(b). We define the freshness function for file i at the user as follows

$$f_u(i,t) = \begin{cases} 1, & \text{if file } i \text{ at the user is fresh at time } t, \\ 0, & \text{otherwise.} \end{cases} \tag{2.6}$$

Once file i is updated at the cache after $W_c(j)$, and the same file is updated at the user after $\bar{W}_u(j)$, file i at the user remains fresh for a time period of $T_u(j)$. Thus, the total waiting time for the user to get the freshest version of file i in the jth cycle is $W_u = W_c + \bar{W}_u$. We denote $F_u(i)$ as the long term average freshness of file i at the user which is given by

$$\begin{aligned} F_u(i) &= \lim_{T\to\infty} \frac{1}{T} \int_0^T f_u(i,t)dt \qquad (2.7) \\ &= \lim_{T\to\infty} \frac{M}{T} \left(\frac{1}{M} \sum_{j=1}^{M} T_u(j) \right) = \frac{\mathbb{E}[T_u]}{\mathbb{E}[T_s]}, \qquad (2.8) \end{aligned}$$

where T_u denotes the typical random variable for $T_u(j)$ and M is the number of update cycles in time duration T.

First, we find $\mathbb{E}[W_u] = \mathbb{E}[\bar{W}_u] + \mathbb{E}[W_c]$ by using nested expectations. Similarly, we define $\bar{W}'_u$ as the waiting time of receiving an update at the user. When the jth update arrives at the source, due to memoryless property of the exponential distribution, W'_c and $\bar{W}'_u$ are exponentially distributed with rates c and u, respectively. Hence, the distribution of $W'_u = \bar{W}'_u + W'_c$ denoted by $f_{W'_u}(x)$ is equal to the convolution of the distributions of W'_c and $\bar{W}'_u$,

$$f_{W'_u}(x) = f_{\bar{W}'_u}(x) * f_{W'_c}(x) = \frac{cu}{c-u}\left(e^{-ux} - e^{-cx}\right), \quad 0 \leq x < \infty. \tag{2.9}$$

For a given update cycle duration T_s at the source, the total waiting time $W_u = \min\{W'_u, T_s\}$ either takes a value in between 0 and T_s, or $W_u = T_s$. When $W_u = T_s$, we note that the file at the user is not updated in that cycle. Thus, we have

$$\mathbb{E}[W_u|T_s = t] = \int_0^t x f_{W_u}(x)dx + \int_t^\infty t f_{W_u}(x)dx \tag{2.10}$$

$$= \frac{cu}{c-u}\left[\frac{1-e^{-ut}}{u^2} - \frac{1-e^{-ct}}{c^2}\right]. \tag{2.11}$$

By using $\mathbb{E}[W_u] = \mathbb{E}[\mathbb{E}[W_u|T_s]]$, we obtain $\mathbb{E}[W_u]$ as

$$\mathbb{E}[W_u] = \int_0^\infty \frac{cu}{c-u}\left[\frac{1-e^{-ut}}{u^2} - \frac{1-e^{-ct}}{c^2}\right]\lambda e^{-\lambda t}dt = \frac{\lambda+c+u}{(\lambda+u)(\lambda+c)}. \tag{2.12}$$

Using (2.12), we obtain $\mathbb{E}[T_u]$ as

$$\mathbb{E}[T_u] = \mathbb{E}[T_s] - \mathbb{E}[W_u] = \frac{1}{\lambda} - \frac{\lambda + c + u}{(\lambda + u)(\lambda + c)} = \frac{uc}{\lambda(\lambda + u)(\lambda + c)}. \tag{2.13}$$

Finally, by substituting (2.13) into (2.8), we obtain $F_u(i)$ as

$$F_u(i) = \frac{\mathbb{E}[T_u]}{\mathbb{E}[T_s]} = \frac{u}{\lambda + u}\frac{c}{\lambda + c}, \tag{2.14}$$

which is equal to the freshness expression in [7]. Above, we have provided an alternative method (to [7]) to derive freshness, which will be useful in the multi-cache system next.

2.2.2 Freshness of File i in the Multi-Cache Model

In this subsection, we find the freshness expression of file i for a multi-cache system shown in Fig. 2.1(b). Again, for simplicity, we drop file index i from all variables in this subsection.

Each cache sends its updates to the user independent of other caches. After the file at the source is updated for the jth time, the file at the user becomes fresh again by the first successful update by any one of the caches. The file at cache k is updated after W'_{c_k} duration. Cache k updates the same file at the user after $\bar{W}'_{u_k}$ duration. We denote the random variable $X_k = W'_{c_k} + \bar{W}'_{u_k}$ as the total waiting time for cache k to send a successful update to the user. As W'_{c_k} and $\bar{W}'_{u_k}$ are exponentially distributed with rates c_k and u_k, respectively, similar to (2.9), we have $f_{X_k}(x) = \frac{c_k u_k}{c_k - u_k}\left(e^{-u_k x} - e^{-c_k x}\right)$ for $x \geq 0$. For a given update cycle $T_s = t$, the user is updated after W_u given by

$$W_u = \min\{t, X_1, X_2, \dots, X_K\}, \tag{2.15}$$

where $W_u = t$ denotes the case where the user is not updated in that update cycle. The complementary cumulative distribution function (ccdf) of X_k is given by

$$\mathbb{P}(X_k > x) = \begin{cases} \frac{c_k u_k}{c_k - u_k}\left(\frac{e^{-u_k x}}{u_k} - \frac{e^{-c_k x}}{c_k}\right), & x \geq 0, \\ 1, & x < 0. \end{cases} \tag{2.16}$$

Since W_u takes only positive values, $\mathbb{E}[W_u]$ can be found by integrating its ccdf as follows

$$\begin{aligned} \mathbb{E}[W_u | T_s = t] &= \int_0^\infty \mathbb{P}(W_u > x) dx & (2.17) \\ &= \int_0^t \mathbb{P}(X_1 > x)\mathbb{P}(X_2 > x) \cdots \mathbb{P}(X_K > x) dx. & (2.18) \end{aligned}$$

For ease of exposition, we define $p_v = (p_i)_{i \in [k]} \in \Pi_k\{c_k, u_k\} = V_p$, and $S_c = \sum_{k=1}^K \mathbb{1}\{p_k = c_k\}$. Then, we have

$$\mathbb{E}[W_u | T_s = t] = \frac{\prod_k c_k \prod_k u_k}{\prod_k (c_k - u_k)} \times \left(\sum_{p_v \in V_p} \frac{(-1)^{S_c}\left(1 - e^{-t\left(\sum_k p_k\right)}\right)}{\left(\sum_k p_k\right) \prod_k p_k} \right). \tag{2.19}$$

Next, we find $\mathbb{E}[W_u] = \mathbb{E}[\mathbb{E}[W_u|T_s]]$ as follows

$$\mathbb{E}[W_u] = \frac{\prod_k c_k \prod_k u_k}{\prod_k (c_k - u_k)} \left(\sum_{p_v \in V_p} \frac{(-1)^{S_c}}{\prod_k p_k} \frac{1}{\lambda + \sum_k p_k} \right). \tag{2.20}$$

As given in (2.8), the long term average freshness of file i at the user is

$$F_u(i) = \frac{\mathbb{E}[T_u]}{\mathbb{E}[T_s]} = 1 - \lambda \mathbb{E}[W_u] \tag{2.21}$$

$$= 1 - \frac{\prod_k c_k \prod_k u_k}{\prod_k (c_k - u_k)} \left(\sum_{p_v \in V_p} \frac{(-1)^{S_c}}{\prod_k p_k} \frac{1}{1 + \frac{\sum_k p_k}{\lambda}} \right). \tag{2.22}$$

Our goal is to maximize the overall freshness at the user, i.e., $\sum_{i=1}^{N} F_u(i)$, under the given total update rate constraints at the caches and at the user. We note that when $K = 1$, i.e., a single-cache system, the user freshness in (2.22) reduces to the expression in (2.14), given by

$$F_u(i) = 1 - \frac{cu}{(c-u)} \left(\frac{1}{u} \frac{1}{1 + \frac{u}{\lambda}} - \frac{1}{c} \frac{1}{1 + \frac{c}{\lambda}} \right) \tag{2.23}$$

$$= \left(\frac{u}{\lambda + u} \right) \left(\frac{c}{\lambda + c} \right). \tag{2.24}$$

When $K = 2$, i.e., a two-cache system, the user freshness in (2.22) reduces to

$$F_u(i) = 1 - \frac{(c_1 c_2)(u_1 u_2)}{(c_1 - u_1)(c_2 - u_2)} \times \left(\frac{1}{c_1 c_2} \frac{1}{1 + \frac{c_1 + c_2}{\lambda}} - \frac{1}{u_1 c_2} \frac{1}{1 + \frac{u_1 + c_2}{\lambda}} + \frac{1}{u_1 u_2} \frac{1}{1 + \frac{u_1 + u_2}{\lambda}} - \frac{1}{c_1 u_2} \frac{1}{1 + \frac{c_1 + u_2}{\lambda}} \right)$$

which can be equivalently written as

$$F_u(i) = \frac{(u_1+u_2)(c_1+c_2)}{(\lambda+u_1+u_2)(\lambda+c_1+c_2)} - \frac{\lambda}{(\lambda+u_1+u_2)(\lambda+c_1+c_2)} \times \left(\frac{u_2c_1}{\lambda+u_1+c_2} + \frac{u_1c_2}{\lambda+u_2+c_1}\right). \tag{2.25}$$

Interestingly, comparing (2.14) and (2.25), we note that freshness in a two-cache system with update rates (c_1, c_2) from the source to the caches and (u_1, u_2) from caches to the user, yields a smaller freshness than in a single-cache system with an update rate $c = c_1 + c_2$ from the source to a cache and $u = u_1 + u_2$ from the cache to the user due to the negative term in (2.25).

We observe that as λ becomes very large, $F_u(i)$ in (2.25) approaches zero, i.e., $\lim_{\lambda\to\infty} F_u(i) = 0$. This supports the intuition that it is difficult to maintain fresh information at the user when the source keeps getting newer update packets with high frequency. On the other hand, as λ approaches zero, which corresponds to the scenario where files never get updated at the source, $F_u(i)$ becomes 1, since the user needs to receive each file only once and from that point onwards, the user forever remains fresh. Further, interestingly, setting either one of the variables u_k and c_k for $k = 1, 2$ to zero transforms (2.25) into (2.24), rendering the route k bereft of any use and reducing the model to a single cache system.

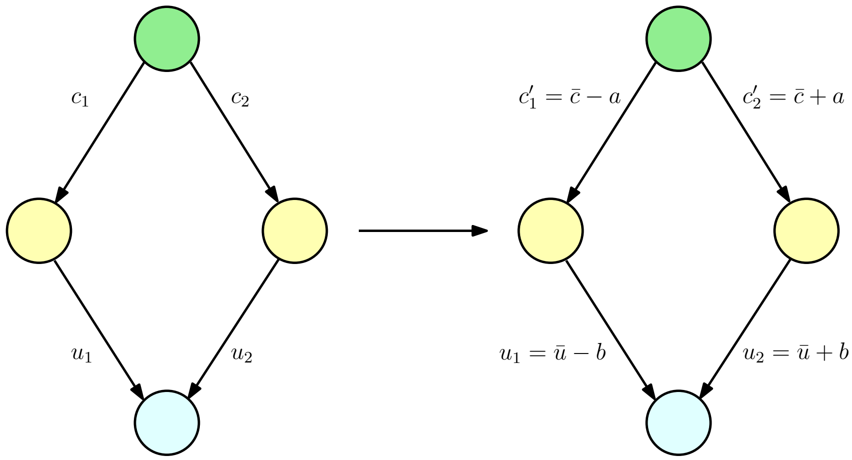


Figure 2.3: Finding optimal cache rates in each route for single file by reshuffling while keeping total cache rate constant.

2.3 Structure of the Optimal Policy

In this section, we find the optimum update rate allocation structure for the most general system with K caches and N files. First, we consider the system with $K = 2$ caches and $N = 2$ files. We denote route k as the file update path from source through cache k to the user. Again dropping file index i for the remainder of the section, let user update rates for file i be u_1 and u_2 in route 1 and route 2, respectively, also let cache update rates in route 1 and route 2 be c_1 and c_2, respectively, as shown in Fig. 2.3. We define the average update rates arriving to user as $\bar{u} = \frac{u_1+u_2}{2}$ and arriving to caches as $\bar{c} = \frac{c_1+c_2}{2}$, and deviation from these averages as $b = \frac{u_2-u_1}{2}$ and $a = \frac{c_2-c_1}{2}$. Thus, $u_1 = \bar{u} - b$, $u_2 = \bar{u} + b$, $c_1 = \bar{c} - a$, and $c_2 = \bar{c} + a$.

In the next lemma, for given user rates u_1 and u_2 (therefore, given $\bar{u}$ and b), and the total cache rate $2\bar{c}$, we find the optimal distribution of cache rates to maximize the freshness at the user, that is, we find the optimal a, a^*, in terms of b,

$\bar{u}$ and $\bar{c}$.

Lemma 2.1 *In a cache update system with $K = 2$ parallel caches, for given user rates u_1 and u_2, and the total cache rate $2\bar{c}$, the optimal cache rates are $c_1^* = \bar{c} - a^*$ and $c_2^* = \bar{c} + a^*$, where*

$$a^* = \min\left\{ b + \frac{(\bar{c}+\lambda+\bar{u})}{b(2\bar{c}+\lambda)} \left(\bar{u}(2\bar{c} + \lambda + \bar{u}) - b^2 \right.\right. \\ \left.\left. - \sqrt{(\bar{u}^2 - b^2)((2\bar{c} + \lambda + \bar{u})^2 - b^2)} \right), \bar{c} \right\}. \tag{2.26}$$

Proof: We write (2.25) equivalently as follows after inserting $\bar{u}$, $\bar{c}$, a and b,

$$F_u(i) = \frac{4\bar{c}\bar{u}}{(\lambda + 2\bar{c})(\lambda + 2\bar{u})} - \frac{\lambda}{(\lambda + 2\bar{c})(\lambda + 2\bar{u})} \times \\ \left(\frac{(\bar{u} - b)(\bar{c} + a)}{\bar{c} + \lambda + \bar{u} + b - a} + \frac{(\bar{u} + b)(\bar{c} - a)}{\bar{c} + \lambda + \bar{u} + a - b} \right). \tag{2.27}$$

Since $\bar{u}$ and $\bar{c}$ are fixed, the first term and pre-factor of the second term in (2.27) are fixed. Taking the derivative of $F_u(i)$ with respect to a yields

$$\frac{dF_u(i)}{da} = \frac{\lambda}{(\lambda + 2\bar{c})(\lambda + 2\bar{u})} \left(\frac{(b - \bar{u})}{(b - a + \bar{c} + \lambda + \bar{u})} + \frac{(b + \bar{u})}{(a - b + \bar{c} + \lambda + \bar{u})} \right. \\ \left. + \frac{(a + \bar{c})(b - \bar{u})}{(b - a + \bar{c} + \lambda + \bar{u})^2} - \frac{(b + \bar{u})(a - \bar{c})}{(a - b + \bar{c} + \lambda + \bar{u})^2} \right). \tag{2.28}$$

Setting $\frac{dF_u(i)}{da} = 0$ yields two solutions. Since a values are restricted in range $0 < a < \bar{c}$, there is only one valid solution which is the first part of the min in (2.26).

The second derivative of freshness with respect to a is given by

$$\frac{d^2F_u(i)}{da^2} =\frac{\lambda}{(\lambda+2\bar{c})(\lambda+2\bar{u})}\left(\frac{2(b-\bar{u})}{(b-a+\bar{c}+\lambda+\bar{u})^2}-\frac{2(b+\bar{u})}{(a-b+\bar{c}+\lambda+\bar{u})^2}\right.$$
$$\left.+\frac{2(a+\bar{c})(b-\bar{u})}{(b-a+\bar{c}+\lambda+\bar{u})^3}+\frac{2(b+\bar{u})(a-\bar{c})}{(a-b+\bar{c}+\lambda+\bar{u})^3}\right). \quad (2.29)$$

Again, as $0<b<\bar{u}$, the valid solution yields $\frac{\partial^2 F_u(i)}{\partial a^2}<0$, from which we conclude that a^* in (2.26) maximizes the freshness at the user.

Furthermore, given $0\leq x,y\leq 1$, the inequality $1-xy-\sqrt{(1-x^2)(1-y^2)}\geq 0$ holds true, hence, by substituting $x=b/\bar{u}$ and $y=b/(2\bar{c}+\lambda+\bar{u})$ in (2.26), we get

$$a^* =b+\frac{(\bar{c}+\lambda+\bar{u})}{b(2\bar{c}+\lambda)}\bar{u}(2\bar{c}+\lambda+\bar{u})\times\left(1-xy-\sqrt{(1-x^2)(1-y^2)}\right)\geq b. \quad (2.30)$$

Differentiating both sides with respect to b gives $\frac{\partial a^*}{\partial b}\geq 0$, and thus, a^* increases monotonically with b, till it reaches $\bar{c}$, after which a^* is equal to $\bar{c}$, yielding the result provided in (2.26). ■

Next, we define $\tilde{F}_u(i)$ as the *cache-update-rate-optimized* freshness, where for fixed u_1, u_2, we insert the optimal cache update rates c_1^* and c_2^* in (2.25). Note that $\tilde{F}_u(i)$ is a function of b, $\bar{u}$ and $\bar{c}$. In the following lemma, we show that as u_1, u_2 get more *lopsided*, i.e., as the difference (u_2-u_1) increases, cache-update-rate-optimized freshness $\tilde{F}_u(i)$ increases.

Lemma 2.2 *$\tilde{F}_u(i)$ is an increasing function of b.*

Proof: We prove the statement by showing $\frac{d\tilde{F}_u(i)}{db}>0$. Here, we consider two different cases for the value of a.

In the first case, we consider $a = a^* < \bar{c}$. Assuming $a = a^*$ in (2.27) to get $\tilde{F}_u(i)$ and using the chain rule, we have

$$\frac{d\tilde{F}_u(i)}{db} = \frac{\partial \tilde{F}_u(i)}{\partial b} + \frac{\partial \tilde{F}_u(i)}{\partial a}\frac{da}{db} = \frac{\partial \tilde{F}_u(i)}{\partial b}, \tag{2.31}$$

where (2.31) follows from the fact that $\frac{\partial \tilde{F}_u(i)}{\partial a} = 0$ when $a = a^*$. We define F_1 as follows

$$F_1 = \frac{(\bar{u} - b)(\bar{c} + a)}{(\bar{c} + \lambda + \bar{u} + b - a)} + \frac{(\bar{u} + b)(\bar{c} - a)}{(\bar{c} + \lambda + \bar{u} + a - b)}, \tag{2.32}$$

which can be equivalently written as

$$F_1 = \frac{(2\bar{u} + \bar{c} + \lambda - a)(\bar{c} + a)}{(\lambda + \bar{u} + \bar{c} + b - a)} + \frac{(2\bar{u} + \bar{c} + \lambda + a)(\bar{c} - a)}{(\lambda + \bar{u} + \bar{c} + a - b)} - 2\bar{c}. \tag{2.33}$$

Comparing (2.33) with (2.27), we note that proving $\frac{\partial \tilde{F}_u(i)}{\partial b} \geq 0$ is the same as proving $\frac{\partial F_1}{\partial b} \leq 0$,

$$\begin{aligned}\frac{\partial F_1}{\partial b} =& \big(-(\lambda + \bar{u} + \bar{c} + a - b)^2(2\bar{u} + \bar{c} + \lambda - a)(\bar{c} + a) \\ & +(\lambda + \bar{u} + \bar{c} + b - a)^2(2\bar{u} + \bar{c} + \lambda + a)(\bar{c} - a)\big) \\ & \times \frac{1}{((\lambda + \bar{u} + \bar{c})^2 - (a - b)^2)^2}.\end{aligned} \tag{2.34}$$

Denoting $A = (\lambda + \bar{u} + \bar{c})^2 + (a - b)^2$, $B = 2(\lambda + \bar{u} + \bar{c})(a - b)$, $C = 2\bar{u} + \bar{c} + \lambda$,

$D = a$ and $E = \bar{c}$, the expression for $\frac{\partial F_1}{\partial b}$ in (2.34) becomes

$$\frac{\partial F_1}{\partial b} = -\frac{2(B(CE - D^2) + AD(C - E))}{((\lambda + \bar{u} + \bar{c})^2 - (a - b)^2)^2} \leq 0, \tag{2.35}$$

where (2.35) follows from the fact that $B \geq 0$ due to (2.30), $D^2 \leq CE$ due to $D \leq C$ and $D \leq E$ as $a < \bar{c}$, $A \geq 0$ and $E \leq C$, which completes the proof of the first case.

In the second case, we consider $a = a^* = \bar{c}$. For this case, F_1 in (2.33) becomes

$$F_1 = \left(\frac{(2\bar{u} + \lambda)}{(\lambda + \bar{u} + b)} - 1\right) 2\bar{c}. \tag{2.36}$$

Then, the derivative of F_1 in (2.36) with respect to b becomes

$$\frac{\partial F_1}{\partial b} = \frac{-(2\bar{u} + \lambda)2\bar{c}}{(\lambda + \bar{u} + b)^2} \leq 0. \tag{2.37}$$

As a result, for both cases, we show that $\frac{\partial F_1}{\partial b} \leq 0$ which equivalently implies that $\frac{\partial \tilde{F}_u(i)}{\partial b} \geq 0$, and thus, $\tilde{F}_u(i)$ increases with b. ■

Lemma 2.2 implies that lopsided update rates at the user increase the freshness. Next, for a $K = 2$ cache system with $N = 2$ files, we show that we should restrict at least one of the files to a single route, that is, lopside at least one of the files to an extreme.

Lemma 2.3 *In a cache update system with $K = 2$ caches and $N = 2$ files, in the optimal policy, we need to restrict at least one file to a single route.*

Proof: Let the average rates at the caches and at the user hold values $\bar{c}_i = \frac{c_{1i}+c_{2i}}{2}$

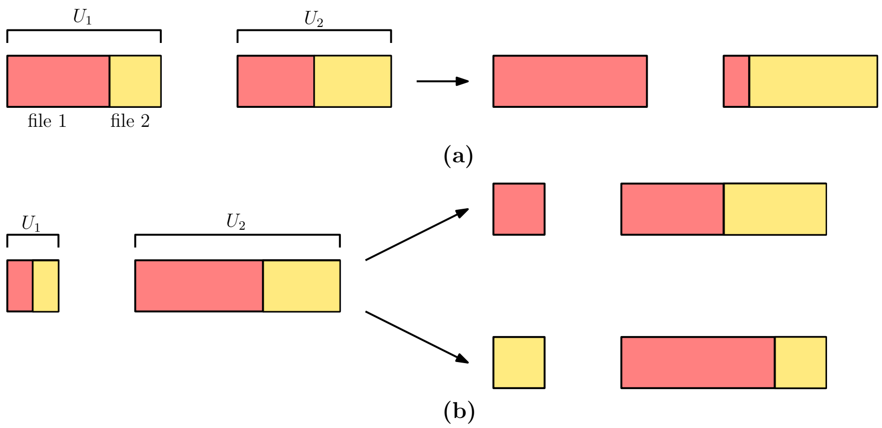


Figure 2.4: Shuffling user rates for improving freshness. (a) Freshness of both files improve (file 2 only in route 2). (b) In upper branch, freshness of file 1 decreases and of file 2 increases (file 2 only in route 2). In lower branch, freshness of file 1 increases and of file 2 decreases (file 1 only in route 2).

and $\bar{u}_i = \frac{u_{1i}+u_{2i}}{2}$ for $i = 1, 2$ which fixes total user rates and total cache rates. Similarly, we have $u_{1i} = \bar{u}_i - b_i$ and $u_{2i} = \bar{u}_i + b_i$ which satisfies the total update rate constraints $u_{11} + u_{12} = U_1$ and $u_{21} + u_{22} = U_2$. Then, we change the update rates at the user to $u'_{11} = \bar{u}_1 - b_1 - \delta_1$, $u'_{21} = \bar{u}_1 + b_1 + \delta_1$, $u'_{12} = \bar{u}_2 - b_2 + \delta_2$ and $u'_{22} = \bar{u}_2 + b_2 - \delta_2$ such that we have $|\delta_1| = |\delta_2|$, $u'_{11} + u'_{12} = U_1$, and $u'_{21} + u'_{22} = U_2$ still hold. We analyze two cases of shuffling, shown in Fig. 2.4.

In the first case, increasing b_i for one file leads to increasing b_i value for the other file as shown in Fig. 2.4(a). As distribution of user rates for both files become lopsided simultaneously, it is a win-win situation for both files. For this case, we increase b_i values of files till one file is completely in a single route. For example, in Fig. 2.4(a), the user rates for the second file (shown in yellow) are $\bar{u}_2 - b_2$ and $\bar{u}_2 + b_2$ in route 1 and route 2, respectively. Then, we increase b_2 till $\bar{u}_2 - b_2 = 0$ in route 1 and the second file is completely restricted to route 2. Such shuffling also leads to a simultaneous increase in b_1.

In the second case, increasing b_i value of one file decreases b_i value of the other file. This case is shown in Fig. 2.4(b) where both files have larger user update rates in route 2. In order to determine which file to prioritize, we compare $\frac{d\tilde{F}_u(i)}{db_i}$ for both files. If $\frac{d\tilde{F}_u(1)}{db_1} > \frac{d\tilde{F}_u(2)}{db_2}$, then we prioritize improving freshness of file 1. We observe that $\frac{d^2\tilde{F}_u(i)}{db_i^2} > 0$. Thus, the increase in freshness of file 1 is always larger than the decrease in freshness of file 2. Similarly, if $\frac{d\tilde{F}_u(2)}{db_2} > \frac{d\tilde{F}_u(1)}{db_1}$, then we increase the freshness of the second file which decreases the freshness of the first file. Thus, we need to restrict at least one file to a single route to obtain the optimum freshness.

■

Thus, for a $K = 2$ cache and $N = 2$ file system with a given set of update rates u_{11}, u_{21}, u_{12} and u_{22}, we can shuffle these rates to increase the total freshness while keeping average rates $\bar{u}_1$, $\bar{u}_2$, $\bar{c}_1$, and $\bar{c}_2$ the same. In this process, we always end up restricting one of the files to only one route. Extending this result to a $K = 2$ cache but arbitrary N files case, we iteratively choose a pair of files and increase freshness of the pair by restricting one of these files to a single route. We repeat this process until we restrict $N - 1$ files to a single route each. Thus, for a $K = 2$ cache, arbitrary N file system, only at most one file will be updated through both relays, and the remaining $N - 1$ files will settle to a single relay.

Lemma 2.4 *Freshness of a file in a K-cache system with update rates at the cache $(c_1, c_2, c_3, \ldots, c_K)$, and at the user $(u_1, u_2, u_3, \ldots, u_K)$ is smaller than the freshness in a $(K-1)$-cache system with update rates at the cache $(c_1 + c_2, c_3, \ldots, c_K)$, and at the user $(u_1 + u_2, u_3, \ldots, u_K)$.*

Proof: With the notation of Section 2.2, since freshness $F_u(i) = 1 - \lambda E[W_u]$, where $E[W_u] = \int_0^\infty E[W_u|T_s = t]\lambda e^{-\lambda t}dt$, we prove the lemma by showing $E[W_u^K|T_s = t] - E[W_u^{K-1}|T_s = t] \geq 0$, where K in W_u^K denotes the total waiting time for the user in a K-cache system.

For file i in the K-cache system, W_u in (2.15) states that

$$W_u^K = \min\{t, X_1, X_2, X_3, \ldots, X_K\}. \tag{2.38}$$

Therefore, similar to (2.18), we have

$$\begin{aligned} \mathbb{E}[W_u^K|T_s = t] &= \int_0^\infty \mathbb{P}(W_u^K > x)dx &(2.39)\\ &= \int_0^t \mathbb{P}(X_1 > x)\mathbb{P}(X_2 > x)\cdots\mathbb{P}(X_K > x)dx, &(2.40)\end{aligned}$$

where X_k is the total waiting time for cache k to send a successful update to the user which has the following distribution

$$\mathbb{P}(X_k > x) = \begin{cases} \frac{c_k e^{-u_k x} - u_k e^{-c_k x}}{(c_k - u_k)}, & x \geq 0, \\ 1, & x < 0. \end{cases} \tag{2.41}$$

For file i in $(K-1)$-cache system, we have $W_u^{K-1} = \min\{t, X_p, X_3, \ldots, X_K\}$ where X_p is the total waiting time in the route having update rate $c_1 + c_2$ from source to cache and u_1+u_2 from cache to user, backed by the following distribution

$$\mathbb{P}(X_p > x) = \begin{cases} \frac{(c_1+c_2)e^{-(u_1+u_2)x}-(u_1+u_2)e^{-(c_1+c_2)x}}{((c_1+c_2)-(u_1+u_2))}, & x \geq 0, \\ 1, & x < 0. \end{cases} \tag{2.42}$$

Then, we write $\mathbb{E}[W_u^{K-1}|T_s=t]$ as

$$\begin{aligned} \mathbb{E}[W_u^{K-1}|T_s=t] &= \int_0^\infty \mathbb{P}(W_u^{K-1} > x)dx \qquad (2.43)\\ &= \int_0^t \mathbb{P}(X_p > x)\mathbb{P}(X_3 > x)\cdots\mathbb{P}(X_K > x)dx. \qquad (2.44) \end{aligned}$$

Hence, we obtain $\mathbb{E}[W_u^K|T_s=t] - \mathbb{E}[W_u^{K-1}|T_s=t]$ as

$$\int_0^t (\mathbb{P}(X_1 > x)\mathbb{P}(X_2 > x) - \mathbb{P}(X_p > x)) \times \mathbb{P}(X_3 > x)\cdots\mathbb{P}(X_K > x)dx. \tag{2.45}$$

We prove $\mathbb{E}[W_u^K|T_s=t] - \mathbb{E}[W_u^{K-1}|T_s=t] \geq 0$ by showing $G(x) \geq 0$ for all $x \geq 0$, where $G(x) = \mathbb{P}(X_1 > x)\mathbb{P}(X_2 > x) - \mathbb{P}(X_p > x)$ is given by

$$\begin{aligned} G(x) = e^{-(c_1+c_2)x}\Bigg(&\frac{(c_1e^{(c_1-u_1)x}-u_1)(c_2e^{(c_2-u_2)x}-u_2)}{(c_1-u_1)(c_2-u_2)} \\ &-\frac{(c_1+c_2)e^{((c_1+c_2)-(u_1+u_2))x}-(u_1+u_2)}{(c_1+c_2)-(u_1+u_2)}\Bigg). \end{aligned} \tag{2.46}$$

Proving $G(x) \geq 0$ for all $x \geq 0$ is equivalent to proving $G_1(x) \geq 0$ for all $x \geq 0$, where

$$G_1(x) = \Bigg(\frac{(c_1e^{(c_1-u_1)x}-u_1)(c_2e^{(c_2-u_2)x}-u_2)}{(c_1-u_1)(c_2-u_2)}$$

$$-\frac{(c_1+c_2)e^{((c_1+c_2)-(u_1+u_2))x}-(u_1+u_2)}{(c_1+c_2)-(u_1+u_2)}\Bigg). \tag{2.47}$$

Since $G_1(0) = 0$, we prove $G_1(x) \geq 0$ for $x \geq 0$ by showing that $\frac{dG_1(x)}{dx} \geq 0$ for $x \geq 0$. Then, we obtain $\frac{dG_1(x)}{dx} = e^{((c_1+c_2)-(u_1+u_2))x}G_2(x)$ where

$$G_2(x) = \left(\frac{c_1\left(c_2 - u_2e^{-(c_2-u_2)x}\right)}{(c_2-u_2)} + \frac{c_2\left(c_1 - u_1e^{-(c_1-u_1)x}\right)}{c_1-u_1} - (c_1+c_2)\right), \tag{2.48}$$

for $x \geq 0$. Since $G_2(0) = 0$ and $\frac{dG_2(x)}{dx} = c_1u_2e^{-(c_2-u_2)x} + c_2u_1e^{-(c_1-u_1)x} \geq 0$, we show that $G_2(x) \geq 0$ for $x \geq 0$ which proves that $G(x) \geq 0$ for $x \geq 0$, and thus, completes the proof. ■

Hence, for given total update rates $\sum_{k=1}^{K} u_{ki}$ and $\sum_{k=1}^{K} c_{ki}$ for file i, the maximum freshness is obtained by concentrating the rates in a single route to the extent possible. In the next section, we provide an approximate way of finding total update rates for files and scheduling them to individual links.

2.4 The Optimization of Binary Freshness in Parallel Relay Networks

Based on the insights obtained for the optimal caching structure in Section 2.3, in this section, we provide a sub-optimal solution that finds the cache and user update rates for each file, and then, propose a way to allocate these rates to the cache routes. We first write the freshness maximization problem for our system as,

$$\max_{c_{ki},u_{ki}} \quad \sum_{i=1}^{N} F_u(i)$$

$$\text{s.t.} \quad \sum_{k=1}^{K}\sum_{i=1}^{N} c_{ki} \leq C$$
$$\sum_{i=1}^{N} u_{ki} \leq U_k, \quad k=1,\dots,K,$$
$$c_{ki} \geq 0, \quad u_{ki} \geq 0, \quad k=1,\dots,K, \quad i=1,\dots,N. \tag{2.49}$$

We note that the problem in (2.49) is not a convex optimization problem since the objective function in (2.49) is not concave. Moreover, this parallel cache problem is significantly more complex than the cascade cache problem in [7]. A Lagrangian approach as in [7] seems prohibitive as it results in highly nonlinear KKT conditions, and thus, it is difficult to derive closed form expressions for c_{ki} and u_{ki} that satisfy the KKT conditions. For this reason, we pursue an approximate solution approach utilizing the properties of the optimal solution found in Lemma 2.3.

First, we construct a single-cache problem by bringing all relays together, where the source-to-cache total update constraint is C and the relay-to-user total update constraint is $U = \sum_{k=1}^{K} U_k$. The optimal solution of this single-cache problem forms an upper bound for the optimum solution of our multi-cache problem, as it allows distributed relays to share update rate capacities. We denote this upper bound by F_{ub}.

Second, we extract a feasible solution for our multi-cache problem from the optimum solution of the constructed single-cache problem. We know from Lemma 2.4 that files need to be restricted to single routes for maximum freshness. Thus, our approximate solution takes the optimum solution of the constructed single-cache problem, and distributes the update rates in the multi-cache setting in such a way that each file is updated only through a single relay to the extent possible. Let the solution of the single-cache problem be u_i which is $u_i = \sum_{k=1}^{K} u_{ki}$. We assign the files in order of decreasing u_i to one of the routes. We start with the first route and fit fully as many files as possible, till we reach a file which will not fit completely and we make it split rates with the last route (route K). We follow this for $K-1$ routes. If a file rate u_i exceeds route capacity U_k, we first fill maximal full routes with it, then try to fit the remaining rate fully in the remaining routes. This leaves us with at most $K-1$ files that split rates between two routes. The remaining files go to route K. This approximate solution gives us a sub-optimal freshness F_{so}. Denoting the optimal freshness in our problem in (2.49) as F^*, we have

$$F_{so} < F^* < F_{ub} \tag{2.50}$$

which means $F^* - F_{so} \leq F_{ub} - F_{so}$, i.e., the gap between the sub-optimal solution and the optimal solution is bounded by the gap between the upper bound and the sub-optimal solution.

Next, we bound $F_{ub} - F_{so}$. We note that, in the sub-optimal policy, we assign at most $K-1$ files to two routes. From Lemma 2.2, freshness for file i increases when b_i increases, with minimum at $b_i = 0$ ($a^* = 0$) and maximum at $b_i = \bar{u}_i$ ($a^* = \bar{c}$). Hence, using (2.27), we find an upper bound on the maximum freshness loss ratio ρ possible for a file due to splitting as

$$\rho = \frac{F_u(i)|_{(b_i,a_i^*)=(\bar{u}_i,\bar{c}_i)} - F_u(i)|_{(b_i,a_i^*)=(0,0)}}{F_u(i)|_{(b_i,a_i^*)=(\bar{u}_i,\bar{c}_i)}} = \frac{\lambda_i}{2(\lambda_i + \bar{u}_i + \bar{c}_i)} < 0.5. \tag{2.51}$$

Since $F_{ub}(i) < 1$, the optimality gap is $F^* - F_{so} \leq \rho(K-1)F_{ub}(i) < 0.5(K-1)$. These $K-1$ files have low u_is, owing to very high or very low λ_is, as observed in [7]. In the former case, $F_{ub}(i)$ is low, while in the latter case, ρ is very low.

2.5 The SHS Method to Characterize Timeliness in Parallel Cache Systems

The stochastic hybrid system (SHS) approach has been introduced in [20] as an alternative way to characterize the timeliness in communication systems. Specifically, by using the SHS method, timeliness of the nodes in gossip networks is analyzed with the version age metric in [5, 68], and with the binary freshness metric in [9]. In this section, first, we use the SHS method to provide a different way to find the binary freshness which is derived in Section 2.2 by using a probabilistic approach. Then, we utilize the SHS method to characterize the version age in parallel relay networks.

2.5.1 The Characterization of Binary Freshness by Using the SHS Method

We define the set $\mathcal{K}$ as $\mathcal{K} \triangleq \{1, 2, \dots, K\}$ to denote cache indices. Then, we denote the collection of all the subsets of $\mathcal{K}$ with m elements as $\mathcal{K}_m$. For example, we have $\mathcal{K}_1 = \{\{1\}, \{2\}, \dots, \{K\}\}$. Next, we drop file index i in the remainder of the section and denote the freshness of cache k as $F_k(t)$ and the freshness of any set $S \subseteq \mathcal{K} \cup \{u\}$, where u denotes the index for the user, as $F_S(t) \triangleq \max_{j \in S} F_j(t)$ at time t. We define the average freshness of the set S as $F_S = \lim_{t\to\infty} \mathbb{E}[F_S(t)]$. Next, by using the [9, Thm. 1], we write the freshness of the user in K-cache system as

$$F_u = \frac{\sum_{k_1=1}^{K} u_{k_1} F_{\{u\}\cup\{k_1\}}}{\lambda + \sum_{k_1=1}^{K} u_{k_1}}. \tag{2.52}$$

Note that $F_{\{u\}\cup\{k_1\}}$ denotes the freshness of the set consisting of the user and the cache k_1, where $k_1 \in \mathcal{K}$. Then, we write $F_{\{u\}\cup\{k_1\}}$, or simply $F_{\{u,k_1\}}$, as

$$F_{\{u,k_1\}} = \frac{c_{k_1} + \sum_{k_2\in\mathcal{K}-\{k_1\}} u_{k_2} F_{\{u,k_1\}\cup\{k_2\}}}{\lambda + c_{k_1} + \sum_{k_2\in\mathcal{K}-\{k_1\}} u_{k_2}}, \quad \{k_1\} \in \mathcal{K}_1, \tag{2.53}$$

where $\mathcal{K} - \{k_1\}$ denotes the subtraction of $\{k_1\}$ from the set $\mathcal{K}$. Then, we write the freshness expressions for $F_{\{u,k_1,k_2\}}$ in terms of $F_{\{u,k_1,k2\}\cup\{k_3\}}$, and so on. If we continue to write these sequential equations for the ℓth time, i.e., $F_{\{u,k_1,k_2,\dots,k_\ell\}}$, we obtain

$$\begin{aligned} F_{\{u,k_1,k_2,\dots,k_\ell\}} =& \frac{\sum_{\bar{k}_{\ell+1}\in\{k_1,k_2,\dots,k_\ell\}} c_{\bar{k}_{\ell+1}}}{\lambda + \sum_{\bar{k}_{\ell+1}\in\{k_1,k_2,\dots,k_\ell\}} c_{\bar{k}_{\ell+1}} + \sum_{k_{\ell+1}\in\mathcal{K}-\{k_1,k_2,\dots,k_\ell\}} u_{k_{\ell+1}}} \\ &+ \frac{\sum_{k_{\ell+1}\in\mathcal{K}-\{k_1,k_2,\dots,k_\ell\}} u_{k_{\ell+1}} F_{\{u,k_1,k_2,\dots,k_\ell\}\cup\{k_{\ell+1}\}}}{\lambda+\sum_{\bar{k}_{\ell+1}\in\{k_1,k_2,\dots,k_\ell\}} c_{\bar{k}_{\ell+1}} + \sum_{k_{\ell+1}\in\mathcal{K}-\{k_1,k_2,\dots,k_\ell\}} u_{k_{\ell+1}}}, \end{aligned} \tag{2.54}$$

for $\{k_1, k_2, \dots, k_\ell\} \in \mathcal{K}_\ell$ where $\ell = 1, 2, \dots, K-1$. At the last step, we have

$$F_{\{u,1,2,\dots,K\}} = \frac{\sum_{k=1}^{K} c_k}{\lambda + \sum_{k=1}^{K} c_k}. \tag{2.55}$$

After obtaining all these freshness expressions, we start inserting the value of $F_{\{u,1,2,\dots,\ell+1\}}$ back into $F_{\{u,1,2,\dots,\ell\}}$ for $\ell = 1, 2, \dots, K-1$, and finally, we obtain the freshness of the user F_u in (2.52) by inserting the $F_{\{u,k_1\}}$ values. With the SHS method, in order to find the freshness of the user F_u in a parallel K-cache system, we need to find 2^K different freshness expressions. For example, in a $K = 2$ cache system, we find the freshness at the user F_u by using 4 equations which are given as follows

$$F_u = \frac{u_1 F_{\{u,1\}} + u_2 F_{\{u,2\}}}{\lambda + u_1 + u_2}, \tag{2.56}$$

where $F_{\{u,1\}}$ and $F_{\{u,2\}}$ are given by

$$F_{\{u,1\}} = \frac{c_1 + u_2 F_{\{u,1\}\cup\{2\}}}{\lambda + c_1 + u_2}, \qquad F_{\{u,2\}} = \frac{c_2 + u_1 F_{\{u,2\}\cup\{1\}}}{\lambda + c_2 + u_1}. \tag{2.57}$$

Then, we find $F_{\{u,1,2\}}$ as

$$F_{\{u,1,2\}} = \frac{c_1 + c_2}{\lambda + c_1 + c_2}. \tag{2.58}$$

Finally, by using $F_{\{u,1\}}$ and $F_{\{u,2\}}$ in (2.57), $F_{\{u,1,2\}}$ in (2.58), we obtain the freshness expression at the user F_u for $K = 2$ cache system as

$$\begin{aligned} F_u = {} & \frac{u_1 c_1 + u_2 c_2}{(\lambda+u_1+u_2)(\lambda+c_1+c_2)} + \frac{1}{(\lambda+u_1+u_2)(\lambda+c_1+c_2)} \times \\ & \left(\frac{u_2 c_1 (u_1+c_2)}{\lambda+u_1+c_2} + \frac{u_1 c_2 (u_2+c_1)}{\lambda+u_2+c_1} \right) \end{aligned} \tag{2.59}$$

which is the same as (2.25). Therefore, the SHS method can be used as an alternative way to characterize the binary freshness at the user in a parallel K-cache system.

In the following subsection, we use the SHS method to characterize the version age of information in a parallel K-cache system.

2.5.2 The Characterization of Version Age by Using the SHS Method

The version age of information or simply the version age has been recently introduced in [5, 66]. Here, each update at the source is denoted as a new version of the information, and thus, the version age measures how many versions a particular node is behind compared to the most current version at the source. We denote the version of the information at the source as $N_s(t)$, at cache k as $N_k(t)$, and at the user as $N_u(t)$ at time t. Then, the version age is defined at cache k as $\Delta_k(t) \triangleq N_s(t) - N_k(t)$, and at the user as $\Delta_u(t) \triangleq N_s(t) - N_u(t)$. When the information at the source is updated, the version age at the caches and at the user increases by 1, i.e., $\Delta'_k(t) = \Delta_k(t) + 1$, for all $k \in \mathcal{K}$ and $\Delta'_u(t) = \Delta_u(t) + 1$, where $\Delta'(t)$ represents the version age after the transition. When cache k gets an update from the source, its version age becomes 0, i.e., $\Delta'_k(t) = 0$, as the source has the latest version of the update. However, when the user gets an update from cache k, its version age becomes $\Delta'_u(t) = \min\{\Delta_u(t), \Delta_k(t)\}$. In other words, the user updates its information only if cache k has more recent version of the information. We define the version age of any set S as $\Delta_S(t) \triangleq \min_{j \in S} \Delta_j(t)$ at time t. In addition, we define the average version age of any set S as $\Delta_S = \lim_{t \to \infty} \mathbb{E}[\Delta_S(t)]$. Correspondingly, Δ_k and Δ_u denote the average version age at cache k and at the user. Next, by using the [5, Thm. 1], we write the version age of the user in a

K-cache system as

$$\Delta_u = \frac{\lambda + \sum_{k_1=1}^{K} u_{k_1} \Delta_{\{u\}\cup\{k_1\}}}{\sum_{k_1=1}^{K} u_{k_1}}. \tag{2.60}$$

Here, $\Delta_{\{u\}\cup\{k_1\}}$, or simply $\Delta_{\{u,k_1\}}$, denotes the version age of the set containing cache k_1 and the user. Next, we write $\Delta_{\{u,k_1\}}$ as

$$\Delta_{\{u,k_1\}} = \frac{\lambda + \sum_{k_2 \in \mathcal{K}-\{k_1\}} u_{k_2} \Delta_{\{u,k1\}\cup\{k_2\}}}{c_{k_1} + \sum_{k_2 \in \mathcal{K}-\{k_1\}} u_{k_2}}, \ \ k_1 \in \mathcal{K}_1. \tag{2.61}$$

Continuing to write sequential equations for the version age for the ℓ caches, i.e., $\Delta_{\{u,k_1,k_2,\ldots,k_\ell\}}$, we get

$$\Delta_{\{u,k_1,k_2,\ldots,k_\ell\}} = \frac{\lambda + \sum_{k_{\ell+1} \in \mathcal{K}-\{k_1,k_2,\ldots,k_\ell\}} u_{k_{\ell+1}} \Delta_{\{u,k_1,k_2,\ldots,k_\ell\}\cup\{k_{\ell+1}\}}}{\sum_{\bar{k}_{\ell+1} \in \{k_1,k_2,\ldots,k_\ell\}} c_{\bar{k}_{\ell+1}} + \sum_{k_{\ell+1} \in \mathcal{K}-\{k_1,k_2,\ldots,k_\ell\}} u_{k_{\ell+1}}}, \tag{2.62}$$

for $\{k_1, k_2, \ldots, k_\ell\} \in \mathcal{K}_\ell$ where $\ell = 1, 2, \ldots, K-1$. In the final step, we have

$$\Delta_{\{u,1,2,\ldots,K\}} = \frac{\lambda}{\sum_{k=1}^{K} c_k}. \tag{2.63}$$

After obtaining all these version age expressions, by substituting $\Delta_{\{u,1,2,\ldots,\ell+1\}}$ back into $\Delta_{\{u,1,2,\ldots,\ell\}}$ for $\ell = 1, 2, \ldots, K-1$, and substituting $\Delta_{\{u,1\}}$ in (2.61) into Δ_u in (2.60), we obtain the version age at the user. Thus, characterizing version age at the user by using the SHS method requires solving 2^K different equations sequentially.

Next, as an example, we find the version age in a $K = 2$ cache system. For this system, the version age at the user is given by

$$\Delta_u = \frac{\lambda + u_1\Delta_{\{u,1\}} + u_2\Delta_{\{u,2\}}}{u_1 + u_2}, \tag{2.64}$$

where $\Delta_{\{u,1\}}$ and $\Delta_{\{u,2\}}$ are given by

$$\Delta_{\{u,1\}} = \frac{\lambda + u_2\Delta_{\{u,1\}\cup\{2\}}}{c_1 + u_2}, \qquad \Delta_{\{u,2\}} = \frac{\lambda + u_1\Delta_{\{u,2\}\cup\{1\}}}{c_2 + u_1}. \tag{2.65}$$

Next, we find $\Delta_{\{u,1,2\}}$ as

$$\Delta_{\{u,1,2\}} = \frac{\lambda}{c_1 + c_2}. \tag{2.66}$$

Finally, we obtain the version age at the user Δ_u in (2.64) by using $\Delta_{\{u,1\}}$ and $\Delta_{\{u,2\}}$ in (2.65), and $\Delta_{\{u,1,2\}}$ in (2.66) as

$$\Delta_u = \frac{\lambda}{u_1 + u_2} + \frac{\lambda}{c_1 + c_2} + \frac{\lambda}{(u_1 + u_2)(c_1 + c_2)} \times \left(\frac{u_2c_1}{u_1 + c_2} + \frac{u_1c_2}{u_2 + c_1}\right). \tag{2.67}$$

In the next section, we optimize the update rates and select file routes that minimize the average version age of the user in a parallel relay network.

2.6 The Optimization of Version Age in Parallel Relay Networks

In this section, we consider the problem of minimizing the average version age in parallel relay networks. For this purpose, before considering the parallel relay problem, let us first consider the minimization of version age in a serially connected cache system as shown in Fig. 2.1(a). In Section 2.6.1, we first find the optimal update rates to minimize the version age for the serially connected network. We also utilize these rates for the optimization of the version age in parallel relay networks which we consider in Section 2.6.2. Here, we first show that it is better to consolidate update rates of the files to a single route, and propose an approximate solution to minimize the version age for the multi-cache system.

2.6.1 Optimization of Version Age in Serially Connected Cache Systems

By using the SHS method provided in Section 2.5.2 based on recursive use of [5, Thm. 1], we write the version age of file i at the user for the serially connected cache system as $\Delta_u(i) = \frac{\lambda_i}{u_i} + \Delta_{\{c\}\cup\{u\}}(i) = \frac{\lambda_i}{u_i} + \Delta_c(i)$, where $\Delta_c(i)$ denotes the version age of file i at the cache and $\Delta_{\{c\}\cup\{u\}}(i) = \min\{\Delta_c(i), \Delta_u(i)\} = \Delta_c(i)$ as the user obtains its information only via the cache. Then, we find $\Delta_c(i) = \frac{\lambda_i}{c_i}$. Thus, the version age of file i at the user for a single-cache system is $\Delta_u(i) = \frac{\lambda_i}{u_i} + \frac{\lambda_i}{c_i}$. We note that the version age at the user is equal to the sum of the version age at the cache $\frac{\lambda_i}{c_i}$ and the version age of the user with respect to the cache $\frac{\lambda_i}{u_i}$. Therefore, for a serially connected cache system, the version age becomes additive. Thus, for a serially connected K-cache system, the version age at the user is

$$\Delta_u(i) = \sum_{k=1}^{K} \frac{\lambda_i}{c_{ki}} + \frac{\lambda_i}{u_i}, \tag{2.68}$$

where the first term is equal to the version age at cache k. When we have a serially connected cache system, we see in (2.24) that the binary freshness at the user has a multiplicative structure whereas the version age in (2.68) has an additive structure. [4]

Next, we formulate the problem of minimizing the version age for a single-cache system with N files as follows

$$
\begin{aligned}
\min_{c_i,u_i} \quad & \sum_{i=1}^{N} \frac{\lambda_i}{u_i} + \frac{\lambda_i}{c_i} \\
\text{s.t.} \quad & \sum_{i=1}^{N} c_i \leq C, \\
& \sum_{i=1}^{N} u_i \leq U, \\
& c_i \geq 0, \quad u_i \geq 0, \quad i = 1, \ldots, N, \qquad (2.69)
\end{aligned}
$$

where u_i and c_i are the user and cache update rates, respectively, for file i in a single-cache system. This formulation highlights some key differences between binary freshness and version age. Not only is the optimization problem in (2.69) convex, but it is also separable as the optimal u_is and c_is can be found independently of each other. This is not the case with the binary freshness metric where the objective function is non-convex and the update rates found via alternating maximization

[4] If we use the traditional age of information metric for the serially connected K-cache system with dynamically changing source instead of a source which is updated according to a Poisson process, the average age of file i at the user denoted by $\hat{\Delta}_u(i)$ becomes equal to $\hat{\Delta}_u(i) = \sum_{k=1}^{K} \frac{1}{c_{ki}} + \frac{1}{u_i}$ as in [67, (43)]. We note that we can obtain $\hat{\Delta}_u(i)$ by substituting $\lambda_i = 1$ in the version age expression in (2.68). This result is intuitive as the traditional age metric increases linearly over time with a unit rate as if we have $\lambda_i = 1$ in the case of version age. Thus, the version age and the traditional age metrics are closely related.

based method might not be globally optimal [7]. We separate the problem in (2.69), and write the problem for finding the optimal cache rates as follows

$$\begin{aligned} \min_{c_i} \quad & \sum_{i=1}^{N} \frac{\lambda_i}{c_i} \\ \text{s.t.} \quad & \sum_{i=1}^{N} c_i \leq C, \\ & c_i \geq 0, \quad i = 1, \ldots, N. \end{aligned} \tag{2.70}$$

Similarly, we can write an optimization problem to determine the optimal user update rates in (2.69). The solution to the optimization problem in (2.70) is (see also [51]),

$$c_j = \frac{C\sqrt{\lambda_j}}{\sum_{i=1}^{N}\sqrt{\lambda_i}}, \qquad u_j = \frac{U\sqrt{\lambda_j}}{\sum_{i=1}^{N}\sqrt{\lambda_i}}. \tag{2.71}$$

Thus, the optimal policy for the version age in a single-cache system is a square-root policy, which means that every file gets updated proportional to the square-root of the respective file update rate at the source. [5] In contrast, the policy for the binary freshness in [7] is a threshold based policy, where the files updated frequently at the source may not get updated by the cache and by the users.

[5]Even for arbitrarily connected networks, whenever a cache is serially connected to a preceding cache, the version age becomes additive, and thus, the optimization problem becomes separable. Therefore, for these caches, independent of file update rates, or the connection types in the preceding cache layers, the optimal updating policy for the serially connected caches is to update the files that are updated in the preceding cache based on the square-root policy obtained in this work.

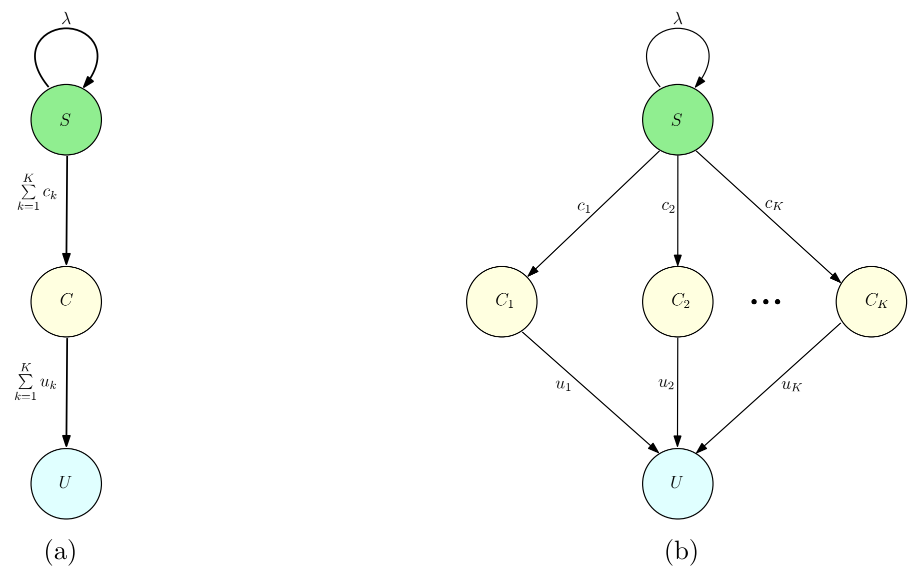


Figure 2.5: Update rates (a) for a single-cache system, and (b) for a parallel multi-cache system considered in Lemma 2.5.

2.6.2 Optimization of Version Age in Parallel Connected Cache Systems

In this section, we provide an approximate solution for the problem of minimizing the version age in a parallel connected cache system shown in Fig. 2.1(b). In the following lemma, we show that the version age of a file is smaller in the single-cache system compared to the multi-cache system when the total update rates are kept the same.

Lemma 2.5 *Version age of a file in a K-cache system with update rates at the cache $(c_1, c_2, \ldots, c_K)$, and at the user $(u_1, u_2, \ldots, u_K)$ is higher than the freshness in a single-cache system with update rates at the cache $\sum_{k=1}^{K} c_k$, and at the user $\sum_{k=1}^{K} u_k$.*

Proof: We demonstrate the caching systems considered in this lemma in Fig. 2.5, where we drop subscript i for file i in the remainder of the section for ease of exposition. We denote Δ_S^{single} and Δ_S^{multi} as the version age of a set S in a single-cache system shown in Fig. 2.5(a) and in a multi-cache system shown in Fig. 2.5(b), correspondingly. As all the rates for a file are consolidated in a single route, such that the user rate in the route is $\sum_{k=1}^{K} u_k$ and the cache rate is $\sum_{k=1}^{K} c_k$, as in Fig. 2.5(a), the version age expression at the cache is

$$\Delta_c^{single} = \frac{\lambda}{\sum_{k=1}^{K} c_k}, \tag{2.72}$$

and the version age at the user is

$$\Delta_u^{single} = \frac{\lambda}{\sum_{k=1}^{K} u_k} + \frac{\lambda}{\sum_{k=1}^{K} c_k}. \tag{2.73}$$

Next, let us consider the K-cache system where route k has user rate u_k and cache rate c_k. Then, by using the SHS method developed in Section 2.5.2 employing [5, Thm. 1] recursively, we have

$$\Delta_u^{multi} = \frac{\lambda}{\sum_{k=1}^{K} u_k} + \frac{\sum_{k=1}^{K} u_k \Delta_{\{u,c_k\}}^{multi}}{\sum_{k=1}^{K} u_k}. \tag{2.74}$$

Note that for all k,

$$\Delta_{\{u,c_k\}}^{multi} = \min\{\Delta_{c_k}^{multi}, \Delta_u^{multi}\} \tag{2.75}$$

$$\geq \min\{\Delta_{c_1}^{multi}, \Delta_{c_2}^{multi}, \dots, \Delta_{c_K}^{multi}, \Delta_u^{multi}\} = \Delta_{\{c_1,\dots,c_K,u\}}^{multi} = \Delta_c^{single}. \tag{2.76}$$

Thus, from (2.74), we have

$$\Delta_u^{multi} \geq \frac{\lambda}{\sum_{k=1}^K u_k} + \Delta_{\{c_1,\ldots,c_K,u\}}^{multi} = \frac{\lambda}{\sum_{k=1}^K u_k} + \Delta_c^{single} = \Delta_u^{single}. \tag{2.77}$$

Thus, the version age at the user is smaller when all rates are consolidated in a single route. ■

Specifically, comparing the version age for the two-cache system in (2.67) with the corresponding single-cache case having version age $\frac{\lambda}{u_1+u_2} + \frac{\lambda}{c_1+c_2}$ from (2.73), we observe that the two-cache case has an extra positive term, and hence, a higher version age.

Similar to the analysis applied to (2.25) in Section 2.3, we rewrite (2.67) in terms of $u_1 = \bar{u} - b$, $u_2 = \bar{u} + b$, $c_1 = \bar{c} - a$, and $c_2 = \bar{c} + a$. Then, for given user rates u_1 and u_2, and the total cache rate $2\bar{c}$, we obtain a_v^* that minimizes the version age as follows

$$a_v^* = \min\left\{ b + \frac{\bar{c}+\bar{u}}{2b\bar{c}} \left(\bar{u}(2\bar{c}+\bar{u}) - b^2 - \sqrt{(\bar{u}^2 - b^2)((2\bar{c}+\bar{u})^2 - b^2)} \right), \bar{c} \right\}. \tag{2.78}$$

Therefore, for given user rates u_1 and u_2, the optimal cache rates are $c_1^* = \bar{c} - a_v^*$ and $c_2^* = \bar{c} + a_v^*$. We use a_v^* in (2.78) in the simulation results provided in Section 2.7. The expression obtained for a_v^* in (2.78) is intuitive as the portion of third term dependent on a and b variables in (2.25) is similar to (2.67), with λ replaced with 0. Note that there is no negative sign in the third term in (2.67), since our goal is to minimize version age, as opposed to maximizing the binary freshness in (2.25).

Lemma 2.5 shows that instead of updating a file over multiple routes, it is better to update it over a single route with the same total update rates at the cache and at the user. Parallel to the solution method proposed for the binary freshness in Section 2.4, here, we propose an approximate solution for the version age. First, for a single-cache system with the total update rates C and $\sum_{k=1}^{K} U_k$, we find the optimal update rates at the cache and at the user based on the square-root policy obtained in Section 2.6.1. Then, we use the method employed in Section 2.4 to distribute these optimal rates across the K routes to get an approximate solution, with the objective of assigning each file to only a single route as much as possible.

Due to Lemma 2.5, the optimal solution obtained for a single-cache system lower bounds the optimal solution for the multi-cache system. Therefore, by looking at the difference between the optimum version age value for the single-cache system and the version age value as a result of the approximate solution proposed in this section, we can have an upper bound on the optimality gap for the multi-cache problem. In the following section, we provide numerical results for the binary freshness and version age metrics for multi-cache systems.

2.7 Numerical Results

In the first numerical result, we choose number of routes $K = 5$, number of files $N = 30$, total update rate at source $C = 50$ and at caches $U = 100$ where each route has $U_k = 20$. We use update arrival rates $\lambda_i = bq^i$ at the source for $i = 1, \dots, N$, where $b > 0$, $q = 0.7$, and $\sum_{i=1}^{N} \lambda_i = a$, with $a = 100$. Note that since $q < 1$, the

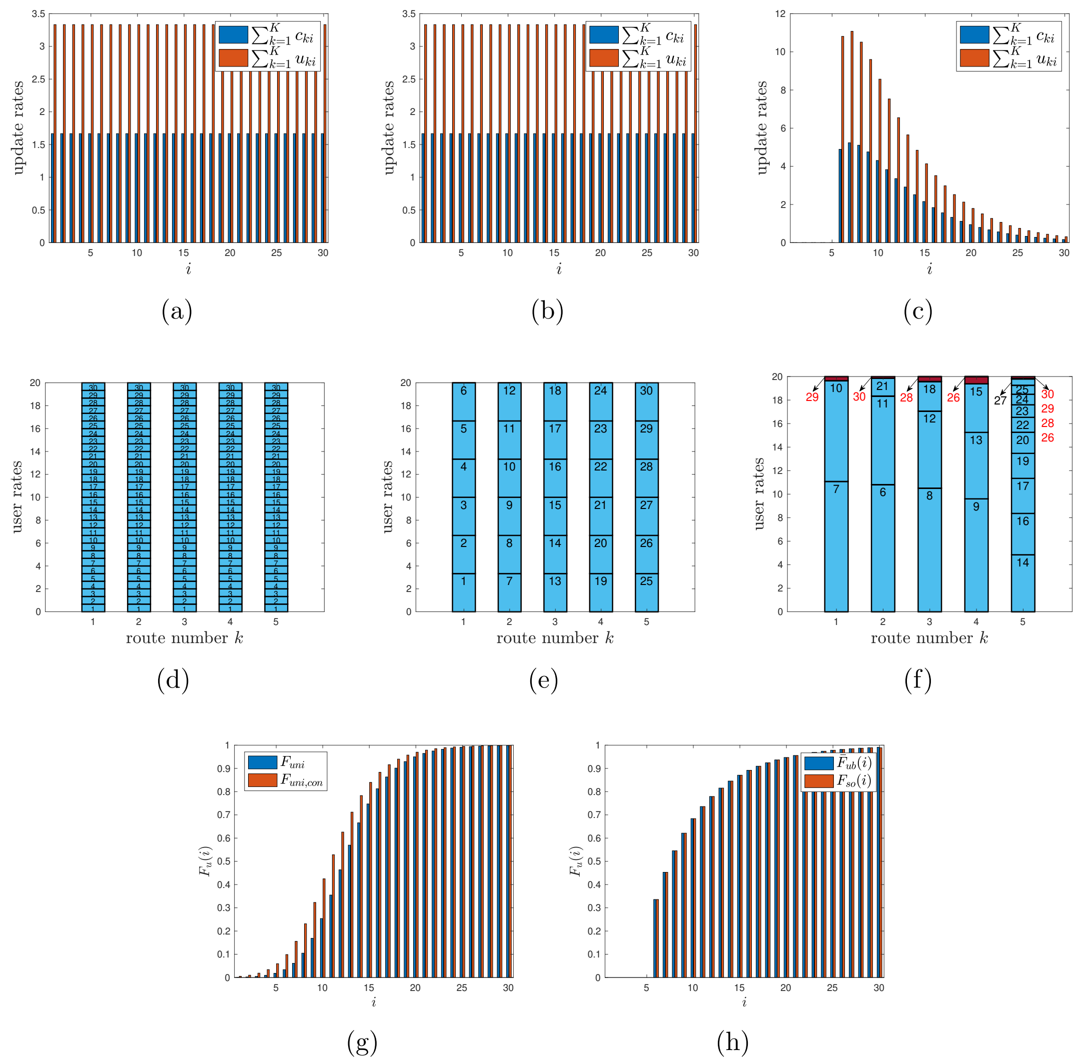


Figure 2.6: Total user rates and total cache rates for the uniform update rate policies (a) without consolidation, (b) with consolidation, (c) total user rates and total cache rates obtained from the auxiliary solution. Route allocations for files (d) for the uniform update policy without consolidation, (e) for the uniform update policy with consolidation, and (f) for our proposed strategy (files with blue rates in single routes). Freshness obtained for the uniform update policy (g) with and without consolidating update rates into single cache, (h) for the single-cache and parallel cache systems.

update arrival rates at the source λ_i decrease with the file index, i.e., $\lambda_i > \lambda_j$ for $i < j$.

In the first numerical result, we use the binary freshness metric. We apply alternating maximization approach described in [7] to solve the auxiliary single-cache problem to obtain total update rate for file i at the user $\sum_{k=1}^{K} u_{ki}$ and at the caches $\sum_{k=1}^{K} c_{ki}$ as shown in Fig. 2.6(c). The total cache update rate constraint, i.e., $\sum_{k=1}^{K}\sum_{i=1}^{N} c_{ki} \leq C$, is already satisfied by both problems. In a parallel cache system, each route has its own total update rate constraint $\sum_{i=1}^{N} u_{ki} \leq U_k$, whereas the single-cache system has only one total update rate constraint for the user, i.e., $\sum_{k=1}^{K}\sum_{i=1}^{N} u_{ki} \leq U$ where $U = \sum_{k=1}^{K} U_k$. Thus, we need to choose the user rate allocation for all files in each route as described in Section 2.4, with corresponding cache rates found by (2.26) which are shown in Fig. 2.6(f). In order to compare the performance of our proposed algorithm, we consider two different cache updating schemes with uniform update rates. For both of these schemes, we choose the update rates as $\sum_{k=1}^{K} c_{ki} = \frac{C}{N}$, $\sum_{k=1}^{K} u_{ki} = \frac{U}{N}$ which is shown in Fig. 2.6(a)-(b). In the first updating scheme called uniform updating policy without consolidation, all the files are updated by all the caches as shown in Fig. 2.6(d), and we denote the freshness obtained by this updating scheme by F_{uni}. In the second updating scheme called uniform updating policy with consolidation, we consolidate the file update rates into a single cache as shown in Fig. 2.6(e) and denote $F_{uni,con}$ as the freshness obtained by this scheme. We denote the freshness at the user for the single-cache system obtained by the method in [7] as $\bar{F}_{ub}$. We plot F_{uni}, $F_{uni,con}$ in Fig. 2.6(g) and $\bar{F}_{ub}$ and F_{so} in Fig. 2.6(h). Among these three updating schemes, the uniform updating policy without consolidation performs the worst and the total freshness obtained by this scheme is $\sum_{i=1}^{N} F_{uni}(i) = 17.8004$. As we observed from Section 2.3 that consolidating update rates into a single cache increases the freshness. With the uniform updating policy with consolidation, the freshness obtained from all files is equal to $\sum_{i=1}^{N} F_{uni,con}(i) = 19.4495$. Lastly, if we further optimize the update rates among the files and use the proposed algorithm in Section 2.4, we obtain the total freshness as $\sum_{i=1}^{N} F_{so}(i) = 21.0692$ which significantly outperforms both of the uniform updating schemes. Even though we split the update rates among different routes for $K-1=4$ files that have some of the highest freshness (files 26, 28, 29, 30), their freshness loss is negligible as shown in Fig. 2.6(c). In this system, since $\bar{F}_{ub} = 21.0718$, the total freshness loss, i.e., $\bar{F}_{ub} - F_{so}$, is equal to 0.0026, which is much smaller than the theoretical upper bound $0.5(K-1) = 2$.

To get more perspective on these freshness values, a more loose upper bound can be obtained by removing the relay-to-user update constraint rate U in the single cache problem by setting $U=\infty$, which allows each $u_i=\infty$, thereby giving $\lim_{u_i\to\infty} F_u(i) = \lim_{u_i\to\infty} \frac{u_i}{\lambda_i+u_i}\frac{c_i}{\lambda_i+c_i} = \frac{c_i}{\lambda_i+c_i}$. Denoting freshness of this system model by $F_{u,d}(i)$, the corresponding freshness maximization problem for the system with no cache is a convex optimization problem stated as

$$\begin{aligned} \max_{\{c_i\}} \quad & \sum_{i=1}^{N} F_{u,d}(i) = \sum_{i=1}^{N} \frac{c_i}{\lambda_i + c_i} \\ \text{s.t.} \quad & \sum_{i=1}^{N} c_i \leq C, \\ & c_i \geq 0, \quad i = 1, \dots, N, \end{aligned} \tag{2.79}$$

for which the optimum update rates c_is are given by $c_i = \lambda_i \left(\frac{1}{\sqrt{\theta\lambda_i}} - 1\right)^+$, where $\theta > 0$ is the Lagrange multiplier satisfying total update rate constraint, i.e., $\sum_{i=1}^{N} c_i \leq C$, and corresponding total optimum freshness is $F^*_{u,d} = 22.5096$, which is close to the total freshness obtained by our algorithm F_{so}. Note that from (2.8), for a set of 30 files, the maximum possible freshness value is 30, when $f_u(i,t) = 1$ for all i and t, which is achieved when all constraints are relaxed. In that case $u_{ki} = c_{ki} = \infty$, and as a result files are instantly delivered from source through the caches to the user. In this example, we choose λ_i as a monotonically decreasing function of i, which gives a diverse set of source update rates. Instead, if all files have the same update rate $\lambda_i = \frac{a}{N}$, $a = 100$, such that $\sum_i \lambda_i = a$ as before, we get $F_{uni,con} = F_{so} = F_{ub} = 5$, since the update rates u_i, c_i are the same for all files and all these policies support consolidation. However, $F_{uni} = 2.0072$ is lower due to lack of rate consolidation across routes as a result of the uniform updating policy without consolidation. This further supports the key insights of our work.

In the second numerical result, we consider the version age metric for a multi-cache system. We choose number of files $N = 10$, and keep the rest of the parameters the same as in the first numerical result. We repeat the process of finding the update rates of files in the first numerical result, with the difference that we use (2.71), which is the square-root policy found in Section 2.6.1, to solve the auxiliary single-cache problem. This difference is highlighted in Fig. 2.7(a) and Fig. 2.7(b). We see that in order to maximize the binary freshness, the cache and the user may not update the files that are frequently changing at the source as shown in Fig. 2.7(a), whereas in order to minimize the version age, each file is updated at the cache and at the user proportional to the square-root of file change rate at the source shown in Fig. 2.7(b). As a result, although both metrics measure the information freshness

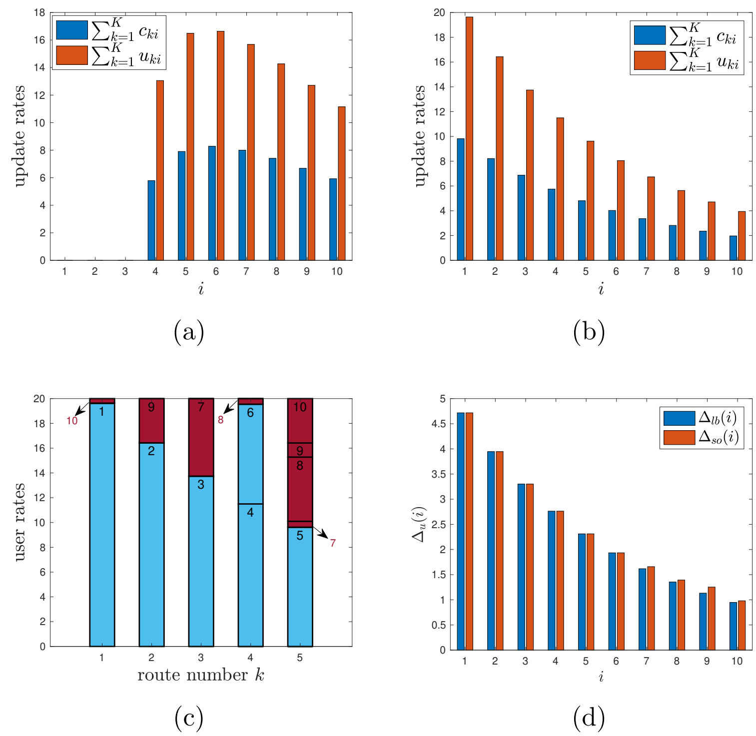


Figure 2.7: Total user rates and total cache rates obtained from the auxiliary solution (a) with binary freshness, and (b) with version age. (c) Route allocations for files (files with blue rates in single routes) with version age, and (d) version age for the single-cache and parallel cache systems.

in different ways, we see that depending on the metric selection, the optimal rate allocation policy may differ significantly. We provide the user rate allocation in Fig. 2.7(c) where 4 files (files 7, 8, 9, 10) are split across two routes. For the files that are updated across two caches, we have derived the specific formula in (2.67) which we use in this numerical result to find the corresponding version age, with corresponding cache rates found by (2.78). The single-cache system gives the lower bound for the version age denoted by $\Delta_{lb} = \sum_{i=1}^{n} \Delta_{lb}(i)$, where $\Delta_{lb}(i)$ is the version age corresponding to file i. Let the approximate policy give the sub-optimal version age $\Delta_{so} = \sum_{i=1}^{n} \Delta_{so}(i)$. We plot Δ_{lb} and Δ_{so} in Fig. 2.7(d). In this numerical result, the total version age increase, i.e., $\Delta_{so} - \Delta_{lb}$, is equal to 0.2334, which is small compared to the $\Delta_{lb} = 24.0257$ and $\Delta_{so} = 24.2591$.

2.8 Conclusion

In this chapter, we considered timeliness of files in a parallel cache network via the binary freshness and version age metrics. First, for the binary freshness metric, we derived a closed form expression for the system and observed that freshness for a file is maximum when it does not have to split its rates across multiple routes. We further analyzed that when number of files is large, most files are restricted to single routes in the optimal update policy. We used this observation to find an approximate policy for assigning rates to all files, by solving the alternate problem of a single-cache system. We further found an upper bound for the gap between the optimal policy and the approximate policy. Next, with the help of the SHS approach, we derived a closed form expression for the version age of information of the system, and determined the counterpart properties in a parallel relay network. Numerical results showed that the proposed sub-optimal policy closely approximates the optimal policy for both timeliness metrics.

CHAPTER 3

How Robust are Timely Gossip Networks to Jamming Attacks?

3.1 Introduction

In this chapter, we investigate the resilience of gossip-based information dissemination over networks used for timeliness purposes, against intentional jamming[1] [82–86] as a function of network connectivity. In particular, we consider two extremes of connectivity in symmetric gossip networks: ring connectivity and full connectivity. In both cases, a source node which keeps the latest version of a file, updates n nodes with equal update rates $\frac{\lambda}{n}$. The source itself is updated with a rate λ_s. In the case of a ring network, see Fig. 3.1(a), each node has two neighbors, and updates its neighbors with equal update rates of $\frac{\lambda}{2}$. In the case of a fully connected network, see Fig. 3.1(b), each node has $(n-1)$ neighbors, and updates its neighbors with equal update rates of $\frac{\lambda}{n-1}$. Without any jammers, in the ring network, the version

[1] In this work, only jamming of the inter-node links between network nodes, i.e., gossip links, is studied, which helps shed light on impact of degree of gossiping or network connectivity on version age of the network. Such networks could arise if communications from the source are more secure than communications in the gossiping layer. We note that jamming of both source-to-node links and gossip links gives rise to more interesting network topologies that we wish to study in future works.

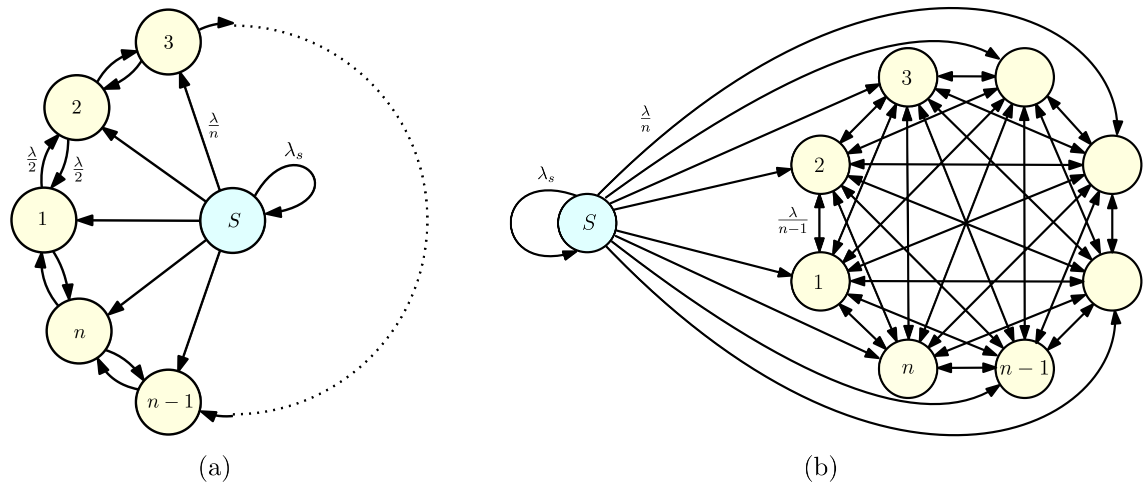


Figure 3.1: (a) Ring network of n nodes. (b) Fully connected network of n nodes.

age of a node scales as $\sqrt{n}$ with the network size n [5,76], and in the fully connected network, the version age of a node scales as $\log n$ [5]. Further, in a disconnected network, where no gossiping is possible, the version age scales as n. We address the following questions: If there are $\tilde{n}$ jammers in a gossip network, how high can they drive the version age? What are the most and least-harmful jammer configurations? How many jammers does it take to drive the version age to order n, the version age with no gossiping among the nodes? We answer these questions for ring and fully connected networks.

A jammer is a malicious entity that disrupts communication between two nodes, say by jamming the channel with noise. The jammer can also be a proxy for communication link failure, network partitioning, network congestion or information corruption during transfer, prevalent in distributed networks. Several works have characterized the effect of adversarial interference on gossip networks for the total dissemination time [87, 88].

In this chapter, we initiate a study of adversarial robustness of gossip networks from an age of information perspective. As an initial work in this direction, we focus on characterizing the impact of the number of jammers on the version age scaling of two types of gossip based networks, the ring network and the fully connected network, which represent the extremes of connectivity spectrum of networks. In the ring network, we show that when the number of jammers $\tilde{n}$ scales as a fractional power of network size n, i.e., $\tilde{n} = cn^{\alpha}$, the average version age scales with a lower bound $\Omega(\sqrt{n})$ and an upper bound $O(\sqrt{n})$ when $\alpha \in \left[0, \frac{1}{2}\right)$, and with a lower bound $\Omega(n^{\alpha})$ and an upper bound $O(n^{\alpha})$ when $\alpha \in \left[\frac{1}{2}, 1\right]$, implying the version age with gossiping is robust against up to $\sqrt{n}$ jammers in a ring network, since the version age of a ring network without any jammers scales as $\sqrt{n}$. To this purpose, we construct an alternate system model of mini-rings (see Fig. 3.6) and prove that the version age of the original model can be sandwiched between constant multiples of the version age of the alternate model. Along the way, we consider average version age in line networks (see Fig. 3.2) and prove structural results.

Then, we study the fully connected gossip network, where we derive a greedy approach to place $\tilde{n}$ jammers with the goal to maximize the age of the resultant network, see Fig. 3.9. Our greedy method involves using the $\tilde{n}$ jammers to isolate maximum possible nodes, thereby consolidating all links into a single mini-fully connected network. We show in this network that the average version age scales as $O(\log n)$ when $\tilde{n} = O(n \log n)$ and as $O(n^{\alpha-1})$, $1 < \alpha \leq 2$ when $\tilde{n} = O(n^{\alpha})$, implying that the network is robust against $n \log n$ jammers, since the version age in a fully connected network without jammers scales as $\log n$.

3.2 Preliminaries

The general system model consists of a source S, alternately denoted by node 0, which maintains the most up-to-date version of a file and a large network of n user nodes $\{1, \dots, n\}$ that wish to acquire the latest version of the file. The source is updated with newer file versions with exponential inter-update times with rate λ_s. The source forwards the current file to each user node with exponential inter-update times with rate $\frac{\lambda}{n}$. Further, each user node sends updates with exponential inter-update times with rate λ to a neighbor, chosen uniformly at random from one of its neighbors, which leads to thinning of this Poisson process into neighbor specific Poisson processes. We use λ_{ij} to represent the rate with which node i sends updates to node j. In addition, the system is faced with the presence of $\tilde{n}$ jammers which jam, i.e., cut, inter-node links, thereby disrupting any communication between the nodes connected by these links.

When two jammers try to cut the same link, they will together be considered as a single jammer. That is, each jammer in our model is assumed to cut a distinct inter-node link. At time t, if $N_i(t)$ is the latest version of a file available at user node i and $N(t)$ is the current version prevailing at the source, then the instantaneous version age at node i at time t is $\Delta_i(t) = N(t) - N_i(t)$. The long-term expected version age of node i is given by $\Delta_i = \lim_{t\to\infty} \mathbb{E}[\Delta_i(t)]$. For a set S of nodes, $\Delta_S(t) = \min_{i\in S} \Delta_i(t)$, and $\Delta_S = \lim_{t\to\infty} \mathbb{E}[\Delta_S(t)]$. The expected values Δ_S are governed by [5, Thm. 1], which we state here for sake of completeness, since we will need it multiple times later. Let $N(S)$ denote the set of updating neighbors of set of nodes S, then [5, Thm. 1] provides the following recursive equation,

$$\Delta_S = \frac{\lambda_s + \sum_{i\in N(S)}(\sum_{j\in S}\lambda_{ij})\Delta_{S\cup\{i\}}}{\sum_{j\in S}\lambda_{0j} + \sum_{i\in N(S)}(\sum_{j\in S}\lambda_{ij})} \tag{3.1}$$

where λ_{ij} is the update rate with which packets arrive from node i at node j.

In this work, we consider the two symmetric networks, the ring network in Fig. 3.1(a) and the fully connected network in Fig. 3.1(b), which represent the two extremes of network connectivity spectrum in symmetric networks. In the ring network, each user node updates each of its two nearest neighbours with exponential inter-update times with rate $\frac{\lambda}{2}$, whereas in the fully connected network, each user nodes updates every other node with exponential inter-update times with rate $\frac{\lambda}{n-1}$, such that the total update rate in both the cases is still λ for each node.

We begin by analyzing the ring network. When multiple adversaries cut inter-node communication links in this symmetric ring, the ring network is dismembered into a collection of isolated groups of nodes, where each group has the structure of a *line network* shown in Fig. 3.2. The age of nodes in each such group are no longer statistically identical, owing to the disappearance of circular symmetry. In this respect, we begin by examining the spatial variation of version age over a line network of n_0 nodes in the next section. The analysis of a line network will be instrumental in the analysis of the ring network in subsequent sections.

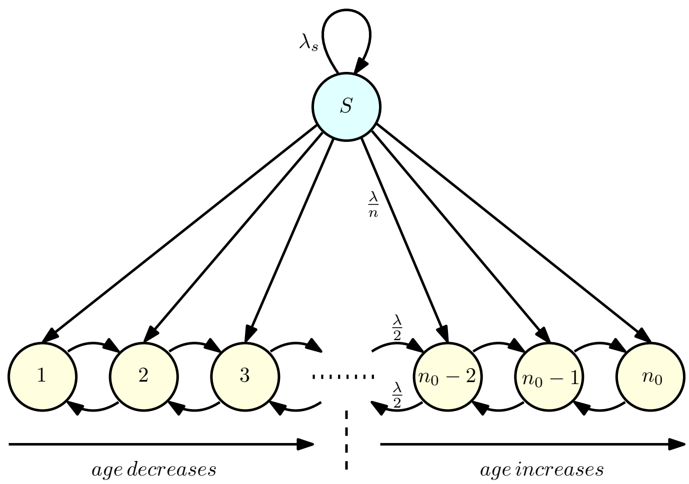


Figure 3.2: Line network model with n_0 nodes.

3.3 Version Age in a Line Network

Consider the line network of n_0 nodes as shown in Fig. 3.2. In Theorem 3.1 below, we show that the expected version age in this network is highest at the corner nodes and decreases towards the center. Intuitively, this is due to the fact that corner nodes are updated less frequently as they are connected with only one inter-node link, and thus, the exchanges towards the network corners are based on relatively staler file versions. Here, superscript $\ell(n_0)$ denotes a line network of size n_0.

Theorem 3.1 *In a line network,* $\Delta_{i+1}^{\ell(n_0)} \leq \Delta_i^{\ell(n_0)}$, $i \leq \frac{n_0}{2}$.

Proof: Fix the i in the statement of the theorem. Let $S_{j,k} = \{j, \dots, j+k-1\}$ denote a set of k contiguous nodes, beginning with node j, in a size n_0 line network, where $j+k-1 \leq n_0$, see Fig. 3.3. We use $\Delta_{j,k}^{\ell(n_0)}$ to denote $\Delta_{S_{j,k}}^{\ell(n_0)}$, i.e., replace the set with its indices. Define $\bar{S}_{j,k}$ as the mirror image of set $S_{j,k}$ about the dotted line between nodes i and $i+1$. Note that $\bar{S}_{j,k} = S_{2i-j-k+2,k}$; see Fig. 3.3 for examples.

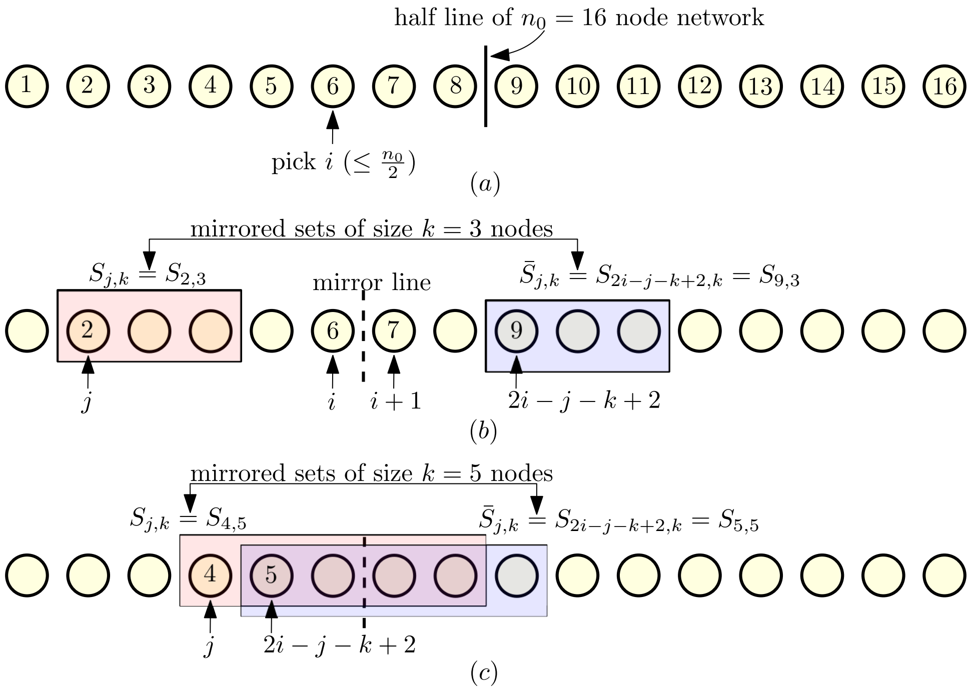


Figure 3.3: $S_{j,k}$ (pink blocks) and $\bar{S}_{j,k} = S_{2i-j-k+2,k}$ (blue blocks) positioned symmetrically about the dotted line between nodes i and $i+1$.

Similarly, we use $\bar{\Delta}_{j,k}^{\ell(n_0)}$ to denote $\Delta_{\bar{S}_{j,k}}^{\ell(n_0)}$. We will consider sets $S_{j,k}$ where majority of the elements of $S_{j,k}$ lie to the left of node i, i.e., $j \leq i + 1 - \frac{k}{2}$, and prove that $\bar{\Delta}_{j,k}^{\ell(n_0)} \leq \Delta_{j,k}^{\ell(n_0)}$. Then, taking $k = 1$ (size one set) with $j = i$ gives the desired result.

We provide a proof by induction, beginning with the case $k = 2i$, where $j = 1$ is the only value that meets the condition $j \leq i + 1 - \frac{k}{2}$. This case gives $\bar{S}_{j,k} = S_{j,k}$ and $\bar{\Delta}_{j,k}^{\ell(n_0)} = \Delta_{j,k}^{\ell(n_0)}$ which satisfies the claim. Next, we assume that the claim holds for some $k(> 1)$, i.e., $\bar{\Delta}_{j,k}^{\ell(n_0)} \leq \Delta_{j,k}^{\ell(n_0)}$ for all $j \leq i + 1 - \frac{k}{2}$, and prove it for $k - 1$. To relate version ages of size k and $k - 1$ sets, we apply (3.1) to obtain the version age of $S_{j,k-1}$

$$\Delta_{j,k-1}^{\ell(n_0)} = \begin{cases} \frac{\lambda_s + \frac{\lambda}{2}\Delta_{j,k}^{\ell(n_0)} + \frac{\lambda}{2}\Delta_{j-1,k}^{\ell(n_0)}}{\frac{(k-1)\lambda}{n}+\lambda}, & j > 1 \\ \frac{\lambda_s + \frac{\lambda}{2}\Delta_{j,k}^{\ell(n_0)}}{\frac{(k-1)\lambda}{n}+\frac{\lambda}{2}}, & j = 1 \end{cases} \tag{3.2}$$

and for its mirrored set $\bar{S}_{j,k-1}$ as

$$\bar{\Delta}_{j,k-1}^{\ell(n_0)} = \frac{\lambda_s + \frac{\lambda}{2}\bar{\Delta}_{j,k}^{\ell(n_0)} + \frac{\lambda}{2}\bar{\Delta}_{j-1,k}^{\ell(n_0)}}{\frac{(k-1)\lambda}{n} + \lambda} \tag{3.3}$$

Since $\bar{S}_{j,k-1} \subset \bar{S}_{j-1,k}$, we have $\bar{\Delta}_{j,k-1}^{\ell(n_0)} \geq \bar{\Delta}_{j-1,k}^{\ell(n_0)}$ as an extended feasible region provides a lower minimum. Using this inequality to eliminate the last term in (3.3) gives

$$\bar{\Delta}_{j,k-1}^{\ell(n_0)} \leq \frac{\lambda_s + \frac{\lambda}{2}\bar{\Delta}_{j,k}^{\ell(n_0)}}{\frac{(k-1)\lambda}{n} + \frac{\lambda}{2}} \tag{3.4}$$

We wish to prove the claim for $k-1$, i.e., $\bar{\Delta}_{j,k-1}^{\ell(n_0)} \leq \Delta_{j,k-1}^{\ell(n_0)}$ under the condition $j \leq i+1-\frac{(k-1)}{2}$. If $j = i+1-\frac{(k-1)}{2}$, then we get $j = 2i-j-k+3$, giving $\bar{S}_{j,k-1} = S_{j,k-1}$ and $\bar{\Delta}_{j,k-1}^{\ell(n_0)} = \Delta_{j,k-1}^{\ell(n_0)}$ which satisfies the claim. For other j, as j must be integer valued, $j \leq i+1-\frac{(k-1)}{2}$ is automatically implied from $j \leq i+1-\frac{k}{2}$. Consequently, since the claim is assumed to hold for k, we have $\bar{\Delta}_{j,k}^{\ell(n_0)} \leq \Delta_{j,k}^{\ell(n_0)}$ and $\bar{\Delta}_{j-1,k}^{\ell(n_0)} \leq \Delta_{j-1,k}^{\ell(n_0)}$. Now, comparing the terms in (3.2) and (3.3) for case $j > 1$, and (3.2) and (3.4) for case $j = 1$ gives $\bar{\Delta}_{j,k-1}^{\ell(n_0)} \leq \Delta_{j,k-1}^{\ell(n_0)}$, and thus, the claim holds for $k-1$. Finally, setting $k = 1$ and $j = i$, we have $\Delta_{i+1}^{\ell(n_0)} = \Delta_{i+1,1}^{\ell(n_0)} = \bar{\Delta}_{i,1}^{\ell(n_0)} \leq \Delta_{i,1}^{\ell(n_0)} = \Delta_{i}^{\ell(n_0)}$, which completes the proof. ■

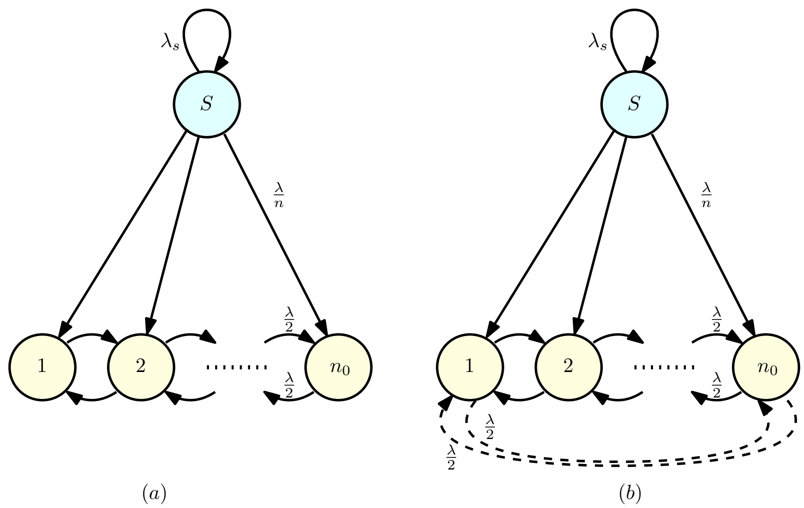


Figure 3.4: (a) A line network of size n_0. (b) A ring network of size n_0. Compared to (a), an extra link is introduced between end nodes bringing radial symmetry.

3.4 Bounds on Age of Line Network

Here, we bound $\Delta_i^{\ell(n_0)}$, age in a line network of size n_0 with $\Delta_i^{r(n_0)}$, age in a ring network of size n_0, see Fig. 3.4.

3.4.1 Lower Bound

Heuristically, due to the presence of an additional link compared to the line network, the ring network will have more age-conformed transitions, and therefore, improved age at the nodes. Mathematically, the recursive equation in (3.1) is identical in the two networks for every subset of nodes $S_{j,k} \subseteq \{2, \ldots, n_0 - 1\}$ that excludes corner nodes 1 and n_0.

Further, for subsets $S_{1,k}$, $k < n_0$, which include corner node 1, similar to (3.2) for case $j = 1$, we have

$$\Delta_{1,k}^{\ell(n_0)} = \frac{\lambda_s + \frac{\lambda}{2}\Delta_{1,k+1}^{\ell(n_0)}}{\frac{k\lambda}{n} + \frac{\lambda}{2}} \tag{3.5}$$

The corresponding equation for the ring, similar to (3.3)-(3.4), is

$$\Delta_{1,k}^{r(n_0)} = \frac{\lambda_s + \frac{\lambda}{2}\Delta_{1,k+1}^{r(n_0)} + \frac{\lambda}{2}\Delta_{n_0,k+1}^{r(n_0)}}{\frac{k\lambda}{n} + \lambda} \leq \frac{\lambda_s + \frac{\lambda}{2}\Delta_{1,k+1}^{r(n_0)}}{\frac{k\lambda}{n} + \frac{\lambda}{2}} \tag{3.6}$$

where $S_{n_0,k+1}$ refers to the set $\{n_0, 1, 2, \ldots, k\}$, and hence, $S_{1,k} \subset S_{n_0,k+1}$ gives the inequality $\Delta_{n_0,k+1}^{r(n_0)} \leq \Delta_{1,k}^{r(n_0)}$.

Further, for $k = n_0$, applying (3.1) gives $\Delta_{1,k}^{\ell(n_0)} = \Delta_{1,k}^{r(n_0)} = \frac{\lambda_s}{\frac{n_0\lambda}{n}}$, with which $\Delta_{1,k}^{\ell(n_0)}$ can be precisely computed from (3.5) and $\Delta_{1,k}^{r(n_0)}$ can be bounded from (3.6), recursively for all $k < n_0$. Now, comparing (3.5) and (3.6) gives $\Delta_{1,k}^{r(n_0)} \leq \Delta_{1,k}^{\ell(n_0)}$. By symmetry, $\Delta_{n_0+1-k,k}^{r(n_0)} \leq \Delta_{n_0+1-k,k}^{\ell(n_0)}$ for subsets which include the other corner node n_0. Thus, $\Delta_{j,k}^{r(n_0)} \leq \Delta_{j,k}^{\ell(n_0)}$, for all j, k, and $\Delta_i^{r(n_0)} \leq \Delta_i^{\ell(n_0)}$, which gives us a lower bound on $\Delta_i^{\ell(n_0)}$.

3.4.2 Upper Bound

Note that due to the radial symmetry of the ring network, $\Delta_i^{r(n_0)} = \Delta_1^{r(n_0)}$, for all i. Hence, combining results of Sections 3.3 and 3.4.1, with bilateral symmetry of line network,

$$\Delta_1^{r(n_0)} \leq \Delta_{\lceil \frac{n_0}{2} \rceil}^{\ell(n_0)} = \Delta_{\lfloor \frac{n_0}{2} \rfloor}^{\ell(n_0)} \leq \ldots\ldots \leq \Delta_{n_0-1}^{\ell(n_0)} = \Delta_2^{\ell(n_0)} \leq \Delta_{n_0}^{\ell(n_0)} = \Delta_1^{\ell(n_0)} \tag{3.7}$$

Hence, an upper bound on $\Delta_1^{\ell(n_0)}$ is an upper bound on $\Delta_i^{\ell(n_0)}$ for all i. Recursively applying (3.1), see also [76, Lem. 2],

$$\Delta_1^{r(n_0)} = \frac{\lambda_s}{\lambda}\left[\sum_{j=1}^{n_0-1}\prod_{k=1}^{j}\frac{1}{\frac{k}{n}+1} + \frac{1}{\frac{n_0}{n}}\prod_{k=1}^{n_0-1}\frac{1}{\frac{k}{n}+1}\right] \tag{3.8}$$

and

$$\begin{aligned}\Delta_1^{\ell(n_0)} =& \frac{2\lambda_s}{\lambda}\left[\sum_{j=1}^{n_0-1}\prod_{k=1}^{j}\frac{1}{\frac{k}{n/2}+1} + \frac{1}{\frac{n_0}{n/2}}\prod_{k=1}^{n_0-1}\frac{1}{\frac{k}{n/2}+1}\right] && (3.9)\\ \leq& \frac{\lambda_s}{\lambda}\left[\sum_{j=1}^{n_0-1}\prod_{k=1}^{j}\frac{1}{\frac{k}{n}+1} + \frac{1}{\frac{n_0}{n}}\prod_{k=1}^{n_0-1}\frac{1}{\frac{k}{n}+1}\right] && (3.10)\\ =& 2\Delta_1^{r(n_0)} && (3.11)\end{aligned}$$

Hence for each node i,

$$\Delta_1^{r(n_0)} \leq \Delta_i^{\ell(n_0)} \leq 2\Delta_1^{r(n_0)} \tag{3.12}$$

Since $\Delta_i^{\ell(n_0)}$ is sandwiched between constant multiples of $\Delta_1^{r(n_0)}$, they both scale similarly when n is large. Intuitively, when the number of nodes is large, the effect of one additional link in the *line* versus *ring* models becomes insignificant.

3.5 Jammer Positioning for Age Deterioration in Ring Network

To characterize the total age of the system, we define $\Delta^{\ell(n_0)} = \sum_{i=1}^{n_0}\Delta_i^{\ell(n_0)}$ and $\Delta^{r(n_0)} = \sum_{i=1}^{n_0}\Delta_i^{r(n_0)}$, the sum of version ages in size n_0 line and ring networks, respectively.

Lemma 3.1 $\Delta^{r(n_0)} - \Delta^{r(n_0+1)}$ *decreases with increase in* n_0.

Proof: We have $\Delta^{r(n_0)} = n_0\Delta_1^{r(n_0)}$ due to the radial symmetry of a ring. Hence, by obtaining expressions for $\Delta_1^{r(n_0)}$ and $\Delta_1^{r(n_0+1)}$ from (3.8), we get

$$\Delta^{r(n_0)} - \Delta^{r(n_0+1)} = n_0\Delta_1^{r(n_0)} - (n_0+1)\Delta_1^{r(n_0+1)} \tag{3.13}$$

$$= \frac{\lambda_s}{\lambda}\left[-\sum_{j=1}^{n_0}\prod_{k=1}^{j}\frac{1}{\frac{k}{n}+1}\right] \tag{3.14}$$

which decreases with increasing n_0. ■

Next, we consider the problem where a single jammer cuts a single link in the line network model of Fig. 3.2. Let the jammer cut the link between nodes m and $m+1$, effectively converting the line network of size n_0 into two smaller line networks of sizes m and $n_0 - m$, such that the total age of this resulting system, denoted by $\Delta^{\ell(n_0)(m)}$, is now the sum of the total ages of the two smaller line networks, i.e., $\Delta^{\ell(n_0)(m)} = \Delta^{\ell(m)} + \Delta^{\ell(n_0-m)}$. We will consider the mini-ring approximation to the resulting two line networks, and assume that the end points of the individual line networks are connected to form mini-rings as shown in Fig. 3.5. The age of this network with mini-rings is $\Delta^{r(n_0)(m)} = \Delta^{r(m)} + \Delta^{r(n_0-m)}$. From Section 3.4, the total ages of the actual dismembered line network and the mini-ring approximation are related as $\Delta^{r(n_0)(m)} \leq \Delta^{\ell(n_0)(m)} \leq 2\Delta^{r(n_0)(m)}$. Next, as an approximation to finding m that maximizes (worst case jammer) and minimizes (most favorable jammer) $\Delta^{\ell(n_0)(m)}$, we will find m that maximizes/minimizes $\Delta^{r(n_0)(m)}$ instead. We show in

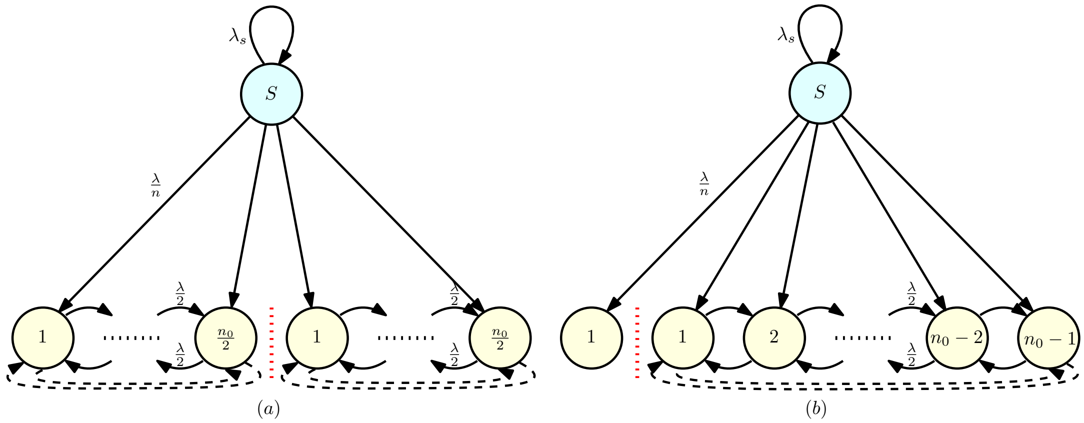


Figure 3.5: Jammer position on line network (a) most favorable, (b) most harmful.

Theorem 3.2 below that most harmful jammer cuts the link that separates the node at the corner, and the most favorable jammer cuts the center link, as shown in Fig. 3.5.

Theorem 3.2 *In a ring,* $\Delta^{r(n_0)(m+1)} \leq \Delta^{r(n_0)(m)}$, $m \leq \frac{n_0}{2}$.

Proof: We have

$$\Delta^{r(n_0)(m)} - \Delta^{r(n_0)(m+1)} = \left[\Delta^{r(m)} + \Delta^{r(n_0-m)}\right] - \left[\Delta^{r(m+1)} + \Delta^{r(n_0-m-1)}\right] \tag{3.15}$$

$$= \left[\Delta^{r(m)} - \Delta^{r(m+1)}\right] - \left[\Delta^{r(n_0-m-1)} - \Delta^{r(n_0-m)}\right] \tag{3.16}$$

$$\geq 0 \tag{3.17}$$

where the last inequality follows from Lemma 3.1, as $m < n_0 - m - 1$ since we are only considering $m \leq \frac{n_0}{2}$. ■

Consider the original ring network of n nodes in Fig. 3.1(a) in the presence of $\tilde{n}$ jammers cutting $\tilde{n}$ communication links. This results in $\tilde{n}$ isolated line networks. Let Δ^{ℓ} denote the average version age at in this dismembered ring, which is composed

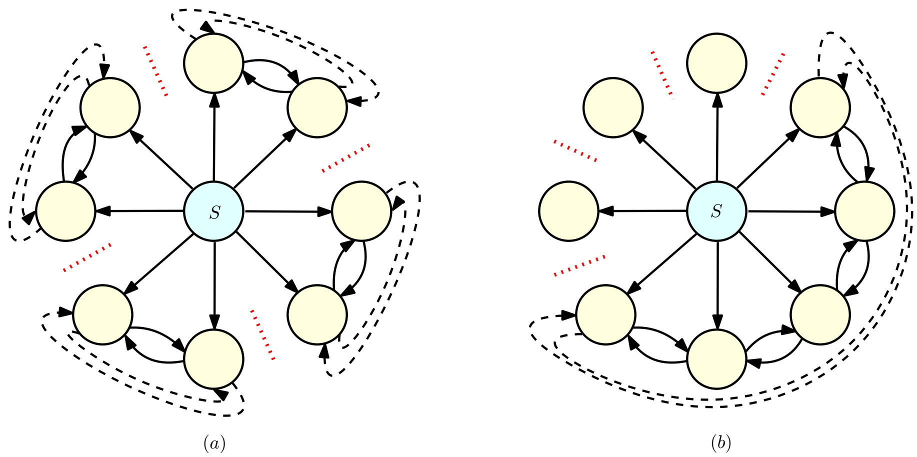


Figure 3.6: Jammer positions on a ring (a) most favorable, (b) most harmful.

of $\tilde{n}$ line networks. Consider the alternate model, where all the $\tilde{n}$ line networks are replaced by their mini-ring versions, as shown in Fig. 3.6(a), and let Δ^r denote the average age of this alternate system. Then, from Section 3.4, we know that Δ^ℓ is bounded by constant multiples of Δ^r.

From Theorem 3.2, the least harmful positioning of jammers for Δ^r is the equidistant placement around the ring, as shown in Fig. 3.6(a), since we can keep switching a jammer's position until it has equal size mini-rings on both sides. Likewise, the most detrimental positioning of jammers is if they cut adjacent links, as shown in Fig. 3.6(b), which results in $\tilde{n}-1$ isolated nodes and a single line network of $n-\tilde{n}+1$ nodes. This is because from Theorem 3.2, presence of two line networks of i_1 and i_2 nodes has lower age than presence of a single line network of i_1+i_2-1 nodes and an isolated node.

3.6 Average Age of Ring Network with Jammers

Lemma 3.2 *(a)* $\sum_{j=1}^{n_0}\left[\prod_{k=1}^{j}\frac{1}{\frac{k}{n}+1}\right] = O(\sqrt{n})$.

(b) If $n_0 = \omega(\sqrt{n})$*, then* $\sum_{j=1}^{n_0}\left[\prod_{k=1}^{j}\frac{1}{\frac{k}{n}+1}\right] = \Omega(\sqrt{n})$.

Proof: We use $\frac{x}{1+x} \leq \log(1+x) \leq x$ and $\sum_{k=1}^{j}\frac{k}{n} = \frac{j(j+1)}{2n}$ to bound $\sum_{k=1}^{j}\log\left(1+\frac{k}{n}\right)$ and then exponentiate to obtain

$$e^{-\frac{j^2}{n}} \leq \prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}} \leq e^{-\frac{j^2}{4n}} \tag{3.18}$$

Then, we sum over j and use Riemann sums to obtain

$$\int_{\frac{1}{\sqrt{n}}}^{\frac{n_0}{\sqrt{n}}} e^{-\frac{t^2}{C}}\,dt \leq \frac{1}{\sqrt{n}}\sum_{j=1}^{n_0} e^{-\frac{j^2}{Cn}} \leq \int_{0}^{\frac{n_0}{\sqrt{n}}} e^{-\frac{t^2}{C}}\,dt \tag{3.19}$$

and note that $\int_0^\infty e^{-\frac{t^2}{C}}\,dt = \frac{\sqrt{C\pi}}{2}$ for a constant $C > 0$. ■

3.6.1 Lower Bound on System Age

Theorem 3.3 *For a ring network of* n *nodes with* $\tilde{n} = cn^{\alpha}$ *jammers, where* $\alpha \in [0,1]$ *and* c *is a scaling constant,*

$$\Delta^{\ell} = \begin{cases} \Omega(n^{\alpha}), & \alpha \geq \frac{1}{2} \\ \Omega(\sqrt{n}), & \alpha < \frac{1}{2} \end{cases} \tag{3.20}$$

Proof: Based on Section 3.5, to lower bound $\Delta^{r}(\leq \Delta^{\ell})$, we consider the least detrimental model of Fig. 3.6(a), where each node is a part of $n_0 = \frac{n}{\tilde{n}} = \frac{1}{c}n^{1-\alpha}$ size mini-ring, and hence all nodes have identical average age giving $\Delta^r = \Delta_1^{r(n_0)}$.

For $\alpha < \frac{1}{2}$, using Lemma 3.2(b) in (3.8) gives

$$\Delta_1^{r(n_0)} \geq \frac{\lambda_s}{\lambda}\left[\sum_{j=1}^{n_0-1}\prod_{k=1}^{j}\frac{1}{\frac{k}{n}+1} + \frac{n}{n_0}\prod_{k=1}^{n_0}\frac{1}{\frac{k}{n}+1}\right] \geq \Omega(\sqrt{n}) \tag{3.21}$$

For $\alpha \geq \frac{1}{2}$, using (3.18) in (3.8) gives

$$\Delta_1^{r(n_0)} \geq \frac{\lambda_s}{\lambda}\left[0 + cn^{\alpha}e^{-\frac{n_0^2}{n}}\right] = \Omega(n^{\alpha}) \tag{3.22}$$

where $e^{-\frac{n_0^2}{n}} = e^{-\frac{n^{-(2\alpha-1)}}{2c^2}} \to 1$ for large n. Since Δ^{ℓ} is bounded by constant multiples of Δ^r, (3.20) follows. ■

3.6.2 Upper Bound on System Age

Theorem 3.4 *For a ring network of n nodes with $\tilde{n} = cn^{\alpha}$ jammers, where $\alpha \in (0,1)$ and c is a scaling constant,*

$$\Delta^{\ell} = \begin{cases} O(n^{\alpha}), & \alpha \geq \frac{1}{2} \\ O(\sqrt{n}), & \alpha < \frac{1}{2} \end{cases} \tag{3.23}$$

Proof: To upper bound $\Delta^r (\geq \frac{1}{2}\Delta^{\ell})$, we consider the most detrimental model of Fig. 3.6(b), which has $\tilde{n}-1$ isolated nodes and a line network of $n-\tilde{n}+1$ nodes.

Then,

$$\Delta^r = \frac{(\tilde{n}-1)\Delta_1^{r(1)} + (n-\tilde{n}+1)\Delta_1^{r(n-\tilde{n}+1)}}{n} \tag{3.24}$$

Since $\frac{\tilde{n}-1}{n} \leq \frac{\tilde{n}}{n}$ and $\frac{n-\tilde{n}+1}{n} = \frac{n(1-cn^{-(1-\alpha)}+\frac{1}{n})}{n} \leq 1$, using (3.8), (3.18) and Lemma 3.2(a) in (3.24) gives

$$\begin{aligned} \Delta^r \leq & \frac{\tilde{n}}{n}\frac{\lambda_s n}{\lambda} + \frac{n-\tilde{n}+1}{n}\frac{\lambda_s}{\lambda}\left[O(\sqrt{n}) + \frac{nce^{-\frac{(n-\tilde{n})^2}{4n}}}{n-\tilde{n}+1}\right] && (3.25) \\ \leq & \frac{\lambda_s}{\lambda}\left[cn^\alpha + O(\sqrt{n}) + c\right] && (3.26) \end{aligned}$$

completing the proof. ■

3.7 Fully Connected Network

In the previous sections, we focused on the ring network, where each node only communicates with its neighbors on both sides with rates $\frac{\lambda}{2}$, and hence, there are fewer links in the network carrying a larger load of information. In contrast, the fully connected network exhibits the other extreme of network connectivity, where every node communicates with every other node of the network where all links transmit information at a lower rate of $\frac{\lambda}{n-1}$; Fig. 3.1(b). We saw that version age scaling in a ring network is robust up to $O(\sqrt{n})$ jammers. One would expect jammers to have more success in the ring network compared to the fully connected network, since jamming any link blocks a higher number of transitions in a ring. To investigate this intuition, the first step is to find what the worst case configuration for $\tilde{n}$ jammers in a fully connected network of n nodes is, and if the robustness of age scaling against jamming is stronger than in ring networks.

To this end, we consider an alternate fully connected network, where inter-node communications happen at rate $\frac{\lambda}{n}$ instead of $\frac{\lambda}{n-1}$. Naturally, the age of this modified network, which we will henceforth simply call the fully connected network, is going to be an upper bound to the age of the *original* fully connected network, since the links now have lower update rates, translating to fewer age-minimizing updates on all the links, which can be verified from (3.1). Therefore, the robustness of age scaling of this upper bound would imply robustness of age scaling in the original network, and in the regime of large n, with $\frac{\lambda}{n-1} \approx \frac{\lambda}{n}$, expected ages in both the networks would converge to each other.

A fully connected network in the absence of any adversary has $\binom{n}{2} = \frac{n(n-1)}{2}$ links. Therefore, choosing positions of $\tilde{n}$ jammers in a fully connected network is the same as placing $\bar{n} = \binom{n}{2} - \tilde{n}$ links of rate $\frac{\lambda}{n}$ in a group of n isolated nodes, as in Fig. 3.8.

Lemma 3.3 *Consider an arbitrarily connected network of n nodes, where the long-term expected version age at node ℓ is Δ_ℓ. If a new link is introduced in this network, then the long-term expected version age at node ℓ in the resultant network $\bar{\Delta}_\ell$ satisfies $\bar{\Delta}_\ell \leq \Delta_\ell$, for all ℓ.*

Proof: Heuristically, the presence of an additional link offers the network the opportunity to have more age-conformed transitions through this link, which would lead to improved age at the nodes. Mathematically, assume that the additional link is in-

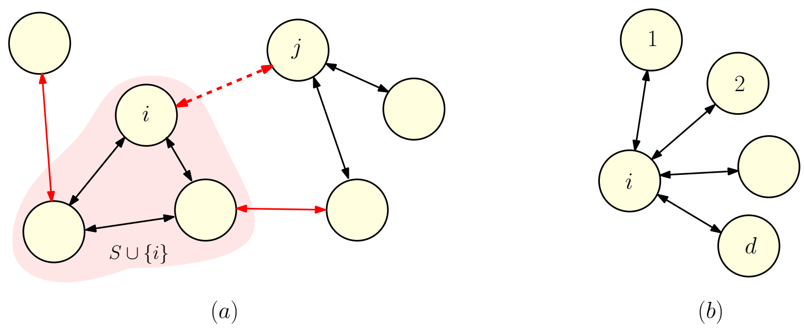


Figure 3.7: (a) A new link (dotted line) is being added between node i and node j, such that it increases the number of incoming link to the set $S \cup \{i\}$ (in orange background) by one. (b) Node i of degree d, connected with d nodes of degree 1.

troduced between the previously disconnected nodes i and j, as shown in Fig. 3.7(a), and consider a set $S_1 \subseteq \mathcal{N} \backslash \{i, j\}$. Then, for sets of the form $S = S_1 \cup \{i, j\}$, we will have $\bar{\Delta}_S = \Delta_S$, since repeated application of (3.1) yields identical set of recursive equations for both $\bar{\Delta}_S$ and Δ_S.

However, for sets of the form $S = S_1 \cup \{i\}$, we will have $\bar{\Delta}_S \leq \Delta_S$. This is because the additional link (i, j) introduces additional terms of $\lambda_{ij} \bar{\Delta}_{S \cup \{j\}}$ and λ_{ij} in the numerator and denominator, respectively, of the equation for $\bar{\Delta}_S$ obtained from applying (3.1), compared to Δ_S. Since $S \subset S \cup \{j\}$, re-employing the trick of (3.3)-(3.4), we use the inequality $\bar{\Delta}_{S \cup \{j\}} \leq \bar{\Delta}_S$ in the numerator to get rid of $\bar{\Delta}_{S \cup \{j\}}$ term and consequently $\bar{\Delta}_S \leq \Delta_S$ can be verified again by the set of recursive equations for $\bar{\Delta}_S$ and Δ_S. Similarly, we can show for sets of the form $S = S_1 \cup \{j\}$, we will have $\bar{\Delta}_S \leq \Delta_S$. Finally, a single application of (3.1) for the sets of the form $S = S_1$ yields similar equation for both $\bar{\Delta}_S$ and Δ_S, as S_1 will have same neighbors in both the networks.

Hence, for any set of nodes S, $\bar{\Delta}_S \leq \Delta_S$. The lemma follows from choosing $S = \{\ell\}$, $\ell \in \mathcal{N}$, such that there exists a path from node ℓ to either node i or node j

in the original network. If no such path exists, then $\bar{\Delta}_\ell = \Delta_\ell$, as the corresponding set of recursive equations are identical in both networks and independent of the age processes at node i or node j. $\blacksquare$

We define the degree of a node henceforth as the number of network nodes (excluding the source node) connected to it. An interesting outcome of Lemma 3.3 is that the maximum possible age at a typical node i with degree d is obtained when it is connected to d nodes of degree 1, as shown in Fig. 3.7(b). This is because, deleting any link in Fig. 3.7(b) would cause node i not to have degree d anymore, while adding more links would lower the age at the node i due to Lemma 3.3. In this case, to compute the age Δ_i at node i, note that the age processes at all the nodes in the set $\{1, 2, \ldots, d\}$ have statistically similar age processes, and therefore, without loss of generality, we will represent any subset of d_1 nodes in this set, for $d_1 \leq d$, by $\{1, \ldots, d_1\}$. In this case, $\Delta_{\{i\}\cup\{1,\ldots,d_1\}}$ can be characterized using (3.1) as

$$\Delta_{\{i\}\cup\{1,\ldots,d_1\}} = \frac{\lambda_s + \frac{(d-d_1)\lambda}{n}\Delta_{\{i\}\cup\{1,\ldots,d_1+1\}}}{\frac{(d_1+1)\lambda}{n} + \frac{(d-d_1)\lambda}{n}} \tag{3.27}$$

$$= \frac{\lambda_s + \frac{(d-d_1)\lambda}{n}\Delta_{\{i\}\cup\{1,\ldots,d_1+1\}}}{\frac{(d+1)\lambda}{n}} \tag{3.28}$$

where by iterative substitution of $d_1 = d, d-1, \ldots, 0$, finally gives

$$\Delta_i = \frac{\lambda_s}{\lambda}\frac{n}{(d+1)}\left(1 + \sum_{d_2=0}^{d-1}\prod_{d_1=0}^{d_2}\frac{d-d_1}{d+1}\right) \tag{3.29}$$

Note that if node i was isolated, meaning it had degree zero whereby it was not connected to any other network node and only received packets from the source with rate $\frac{\lambda}{n}$, then its age would be $\frac{\lambda_s}{\lambda}n$. This implies that if d links are added to an isolated node, then the age reduction at the node is lower bounded by R_d, where

$$R_d = \left[\frac{\lambda_s}{\lambda} n\right] - \left[\frac{\lambda_s}{\lambda}\frac{n}{(d+1)}\left(1 + \sum_{d_2=0}^{d-1}\prod_{d_1=0}^{d_2}\frac{d-d_1}{d+1}\right)\right] \tag{3.30}$$

$$= \frac{\lambda_s}{\lambda} n \left[1 - \frac{1}{(d+1)}\left(1 + \sum_{d_2=0}^{d-1}\prod_{d_1=0}^{d_2}\frac{d-d_1}{d+1}\right)\right] \tag{3.31}$$

Table 3.1 below lists coefficients of $\frac{\lambda_s}{\lambda}n$ in R_d values for $d = 1, \ldots, 22$, where $R_d/\frac{\lambda_s}{\lambda}n$ has been rounded down up to two decimal points, thereby giving a further lower bound.

d	1	2	3	4	5	6
$R_d/\frac{\lambda_s}{\lambda}n$	0.25	0.37	0.44	0.49	0.53	0.56

d	7	8	9	10	11	12
$R_d/\frac{\lambda_s}{\lambda}n$	0.59	0.61	0.63	0.64	0.66	0.67

d	13	14	15	16	17	18
$R_d/\frac{\lambda_s}{\lambda}n$	0.68	0.69	0.70	0.71	0.72	0.72

d	19	20	21	22
$R_d/\frac{\lambda_s}{\lambda}n$	0.73	0.74	0.74	0.75

Table 3.1: $R_d/\frac{\lambda_s}{\lambda}n$ values (rounded down to nearest 2 decimal point number) for $d = 1, \ldots, 22$.

Notice in Table 3.1, how adding the first link causes an age drop of $0.25\frac{\lambda_s}{\lambda}n$, but for larger d, adding additional links does not cause much reduction in the age of a node. Thus, adding a new link to d degree node might cause less system age drop instead of adding it to an isolated node.

Given all this machinery, we seek the configuration of $\bar{n}$ links on a group of n isolated nodes that would result in the highest total age (or average age) for the

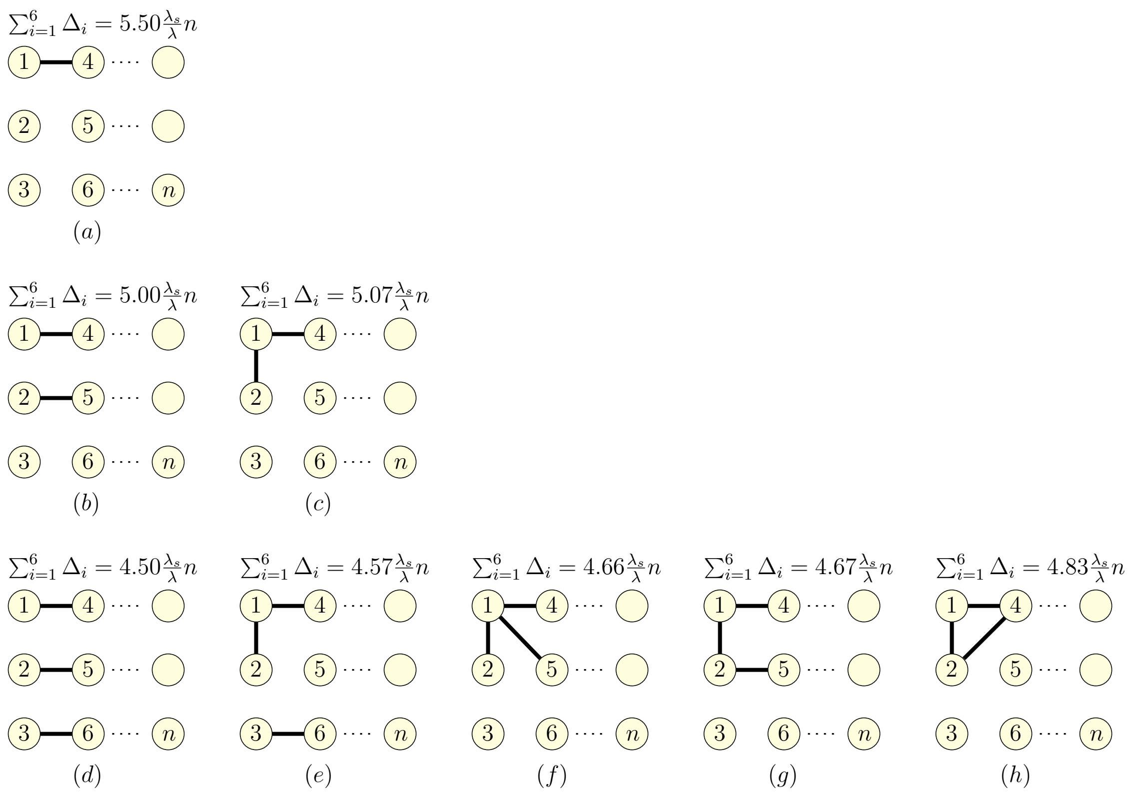


Figure 3.8: A network of n isolated nodes, where all unique network configurations along with their corresponding values of $\sum_{i=1}^{6} \Delta_i$ are presented when $\bar{n}$ links are added to this network, such that, $\bar{n} = 1$ in (a), $\bar{n} = 2$ in (b), (c), and $\bar{n} = 3$ in (d), (e), (f), (g), (h). Links get consolidated and number of nodes engaged decreases from left to right in each row.

network. Fig. 3.8 shows all possible configurations with number of links $\bar{n} = 1, 2, 3$, along with the exact evaluation of their total age for the network. Since $\bar{n} \leq 3$ links can engage with at most 6 distinct nodes, we focus on the total age of the set of nodes $\{1, 2, \dots, 6\}$, as the remaining (isolated) nodes have the same total age of $\sum_{i=7}^{n} \Delta_i = (n-6)\frac{\lambda_s}{\lambda}n$ for all configurations of Fig. 3.8. Here Δ_i, corresponding to the expected age of node i, is computed through the set of recursive equations obtained from (3.1). For example, in the case of Fig. 3.8(a), these recursive equations for nodes 1 and 4 are

$$\Delta_{\{1,4\}} = 0.5\frac{\lambda_s}{\lambda}n \quad \Rightarrow \quad \Delta_1 = \Delta_4 = 0.5\frac{\lambda_s}{\lambda}n + 0.5\Delta_{\{1,4\}} = 0.75\frac{\lambda_s}{\lambda}n \tag{3.32}$$

which give the age of nodes 1 and 4 each as $0.75\frac{\lambda_s}{\lambda}n$. For isolated nodes 2, 3, 5,

$$\Delta_2 = \Delta_3 = \Delta_5 = \Delta_6 = \frac{\lambda_s}{\lambda}n \tag{3.33}$$

Thus, the total age for $\{1, 2, \ldots, 6\}$ in this case is $(0.75 + 0.75 + 1 + 1 + 1 + 1)\frac{\lambda_s}{\lambda}n = 5.50\frac{\lambda_s}{\lambda}n$.

In each row of Fig. 3.8, all network configurations have the same number of links, such that the number of nodes engaged by these links decreases from left to right. For example in the third row, the left most configuration engages six nodes, making their degree one, and the right most network engages three nodes, fully interconnected amongst themselves, forming a mini-fully connected network (mini-FC) of size three. We observe in each row that the total age of the network increases from left to right with consolidation of the links to fewer engaged nodes, leaving higher number of nodes isolated. We observed something similar in Section 3.5 for the ring network, where the worst configuration involved consolidating all links in one single line network thereby creating multiple isolated nodes, whereas the most favorable configuration involved spreading the links more evenly around the ring.

If the network topology resulting from placing links of a set E containing $\bar{n}$ links (i.e., $|E| = \bar{n}$) is denoted by T_E, such that the expected age of node i in the resultant network is denoted by $\Delta_i(T_E)$, then the optimization problem of interest can be formally written as follows,

$$\underset{E:E\subseteq\{(i,j):i,j\in\mathcal{N},i\neq j\},|E|=\bar{n}}{\operatorname{argmax}} \quad \frac{\sum_{i=1}^{n}\Delta_i(T_E)}{n} \tag{3.34}$$

That is, we are interested in finding the optimal positions of the $\bar{n}$ links that lead to maximum age in the resultant network. Further, we are also interested in how the average age scales as a function of network size n when the available number of links $\bar{n}$ is a function of n. However, in general for n-node network with $\bar{n}$ links, there are $\binom{\binom{n}{2}}{\bar{n}} = O(n^2/\bar{n})$ different possible network configurations, and unlike a ring network where the dismembered components always exhibited line topology, very complicated topologies can arise in an arbitrarily jammed fully connected network. This makes it difficult to pick out the worst configuration due to the nature of recursive equations which may require computation of ages of up to 2^n unique sets of nodes for deriving the age at all network nodes. Since finding the optimum configuration is exponentially complex[2], next we develop a greedy approach of placing $\bar{n}$ links with the goal of maximizing the average age in the resultant network.

3.7.1 Greedy Approach

In our greedy approach, instead of placing all $\bar{n}$ links in one go, we place them in $\bar{k}$ steps such that $\binom{\bar{k}-1}{2} \leq \bar{n} \leq \binom{\bar{k}}{2}$. At step k, we have a network of $\binom{k}{2}$ links, in which we place k new links in a manner that maximizes the total age of the resultant

[2]Note that we do not have a formal proof of the computational complexity of the optimization problem in (3.34).

network of $\binom{k+1}{2}$ links. In Lemma 3.4 next, mini-FC (mini fully connected) of k nodes corresponds to set of $k \leq n$ nodes where every node receives update packets from the source with rate $\frac{\lambda}{n}$ and from each of the other $k-1$ nodes with link of rate $\frac{\lambda}{n}$, such that the k nodes are fully connected amongst themselves.

Lemma 3.4 *Consider a network of n nodes where k nodes are connected in a mini-FC and $n-k$ nodes are isolated. Given that we place k new links in this network, the resultant network will have the maximum total network age when the new k links are placed such that they connect a single isolated node to all the k nodes of the mini-FC, resulting in a network where $k+1$ nodes are connected in a mini-FC and $n-k-1$ nodes are isolated.*

Proof: In a mini-FC of k nodes, the age at each node, using (3.1), is $\frac{\lambda_s}{\lambda}\frac{n}{k}\left(\sum_{j=1}^{k}\frac{1}{j}\right)$. Then, the total age in network of n nodes with k nodes connected in mini-FC and $n-k$ nodes isolated is

$$\sum_{i=1}^{n}\Delta_i = k \times \frac{\lambda_s}{\lambda}\frac{n}{k}\left(\sum_{j=1}^{k}\frac{1}{j}\right) + (n-k) \times \frac{\lambda_s}{\lambda}n \tag{3.35}$$

$$= \frac{\lambda_s}{\lambda}\left(n\left(\sum_{j=1}^{k}\frac{1}{j}\right) + n(n-k)\right) \tag{3.36}$$

If k new links are added to this network in a manner that connects an isolated node to the k nodes of this mini-FC, as in Fig. 3.9(a), the total age of the resultant

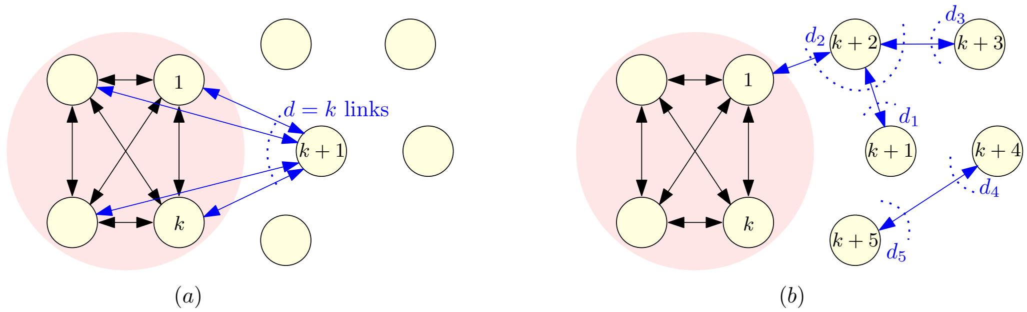


Figure 3.9: $k = 4$ links are added to a network size $k = 4$ mini-FC and $n - k = 5$ isolated nodes. (a) All k links are placed such that they connect node $k+1$ to all nodes of mini-FC, thereby creating a network of size $k+1 = 5$ mini-FC and $n-k = 4$ isolated nodes. The degree of node $k+1$ is $d = k$. (b) The additional k links engage multiple nodes, such that their sum of degrees $d_1 + d_2 + d_3 + d_4 = 7 \geq k$.

network will be

$$\sum_{i=1}^{n} \Delta_i = (k+1) \times \frac{\lambda_s}{\lambda} \frac{n}{k+1} \left(\sum_{j=1}^{k+1} \frac{1}{j} \right) + (n-k-1) \times \frac{\lambda_s}{\lambda} n \tag{3.37}$$

$$= \frac{\lambda_s}{\lambda} \left(n \left(\sum_{j=1}^{k+1} \frac{1}{j} \right) + n(n-k-1) \right) \tag{3.38}$$

As a consequence of adding these k links, the total age of the system reduces by

$$\bar{R} = \frac{\lambda_s}{\lambda} \left(n \left(\sum_{j=1}^{k} \frac{1}{j} \right) + n(n-k) \right) - \frac{\lambda_s}{\lambda} \left(n \left(\sum_{j=1}^{k+1} \frac{1}{j} \right) + n(n-k-1) \right) \tag{3.39}$$

$$= \frac{\lambda_s}{\lambda} \left(n - n \frac{1}{k+1} \right) \tag{3.40}$$

$$= \frac{\lambda_s}{\lambda} n \left(\frac{k}{k+1} \right) \tag{3.41}$$

We will now show that if the k links were instead added in any other manner, the total age of the network would drop by more than $\bar{R}$. Thus, starting with a k node mini-FC and $n-k$ isolated nodes, the worst (maximum age) configuration that can be obtained is a $k+1$ node mini-FC and $n-k-1$ isolated nodes, when k

new links are added.

Without loss of generality, let the nodes of the mini-FC correspond to the set of node indices $\{1, \ldots, k\}$. Note that no new link can be placed between any two nodes of this set, since all links in the mini-FC are already saturated. Hence, each new link will engage at least one node in the set $\{k+1, \ldots, n\}$ of previously isolated nodes. If all the k links were to engage exactly one isolated node, say node $k+1$, that would be possible only if the k links connect the single node $k+1$ to all the nodes $\{1, \ldots, k\}$ of the mini-FC, which just results in a network of $k+1$-size mini-FC and $n-k-1$ isolated nodes as described above. Hence, to prove Lemma 3.4, it suffices to show that engaging two or more nodes from the set $\{k+1, \ldots, n\}$ is not optimal.

It is easy to see that engaging four or more nodes is not optimal. This is because from Table 3.1, $R_1 = 0.25\frac{\lambda_s}{\lambda}n$, and from Lemma 3.3, R_d will be be an increasing function of d, as a higher degree node has more links which reduces the node age. Since engaging a node means making at least one link incident on it, engaging four or more nodes would give a minimum system age reduction of $4 \times R_1 = \frac{\lambda_s}{\lambda}n$. This age reduction is more than (3.41), since $\bar{R} \leq \frac{\lambda_s}{\lambda}n$.

Next, we show that engaging three nodes is not optimal. Let the three engaged nodes be $k+1$, $k+2$ and $k+3$, such that their degrees after placing the k links become d_1, d_2 and d_3. Note that $d_1 + d_2 + d_3 \geq k$, where the equality holds when each new link connects a node from set $\{k+1, k+2, k+3\}$ to a node in the set $\{1, \ldots, k\}$. The inequality comes from the fact that when a new link connects, say, node $k+1$ and node $k+2$, it increments both degrees d_1 and d_2 by one, hence each such link gets counted twice, in degrees of both nodes, as shown in Fig. 3.9(b).

Consider the case of $k = 7$, where Table 3.2 lists all possible unique ways of satisfying $d_1 + d_2 + d_3 = 7$ and $R_{d_1} + R_{d_2} + R_{d_3}$ represents the minimum age reduction caused by turning three previously isolated nodes into nodes of degree d_1, d_2 and d_3. We can see in Table 3.2 that the system age reduction caused by engaging three nodes will be greater $\frac{\lambda_s}{\lambda}n > \bar{R}$ in all configurations, when $k = 7$ and $d_1 + d_2 + d_3 = 7$. Note that the age reduction values in Table 3.2 are a lower bound to the actual system age reduction, since we are only considering a lower bound to the age reduction caused at three nodes of degree d_1, d_2 and d_3, and we have not accounted for the age reduction at nodes of the set $\{1, \dots, k\}$ due to the new links.

d_1	d_2	d_3	$(R_{d_1} + R_{d_2} + R_{d_3})/\frac{\lambda_s}{\lambda}n$
5	1	1	$0.53 + 0.25 + 0.25 = 1.03$
4	2	1	$0.49 + 0.37 + 0.25 = 1.11$
3	3	1	$0.44 + 0.44 + 0.37 = 1.13$
3	2	2	$0.44 + 0.37 + 0.37 = 1.18$

Table 3.2: Minimum age reduction with 3 node engagement and 7 additional links.

If $d_1 + d_2 + d_3 > 7$, then the age reduction caused is even higher than the values of Table 3.2, since R_d is an increasing function of d. Likewise $k > 7$ implies $d_1 + d_2 + d_3 \geq k > 7$, and hence, age reduction is again higher than $\bar{R}$. Finally, for $d_1 + d_2 + d_3 < 7$, we have numerically verified that the age reduction is always more than $\bar{R} = \frac{\lambda_s}{\lambda}n\left(\frac{k}{k+1}\right)$. For example, consider $(d_1, d_2, d_3) = (2, 1, 1)$, in which case $k \leq 4$ and $\bar{R} \leq \frac{4}{5}\frac{\lambda_s}{\lambda}n$ since $\bar{R}$ is an increasing function of k. From Table 3.1, in this case $R_{d_1} + R_{d_2} + R_{d_3} = (0.37 + 0.25 + 0.25)\frac{\lambda_s}{\lambda}n = 0.87\frac{\lambda_s}{\lambda}n \geq \frac{4}{5}\frac{\lambda_s}{\lambda}n \geq \bar{R}$.

Next, we show that engaging two nodes is not optimal. In this case Table 3.3

lists all possible pairs (d_1, d_2) satisfying $d_1+d_2=23$ and the corresponding minimum age reduction caused by two nodes of degree d_1 and d_2, where we see $R_{d_1}+R_{d_2} \geq \frac{\lambda_s}{\lambda}n > \bar{R}$ for all listed pairs (d_1, d_2). Hence, for $d_1+d_2 \geq 23$, we have a minimum system age reduction of $\frac{\lambda_s}{\lambda}n > \bar{R}$. Further, for all $d_1+d_2 < 23$, we have numerically verified for all possible pairs that the system age reduction is higher than $\bar{R}$, similar to the three node case. Observe how in Table 3.2 and Table 3.3, when links are more concentrated in d_1, the lower bound for age reduction is poorer, but as links get more evenly distributed amongst the engaged nodes, the age reduction lower bound increases, which is in line with the general worstness of link consolidation for the system age observed in ring networks in Section 3.5, and for fully connected networks for $\bar{n} \leq 3$ in Fig. 3.8.

d_1	d_2	$(R_{d_1}+R_{d_2})/\frac{\lambda_s}{\lambda}n$
22	1	$0.75+0.25=1.00$
21	2	$0.74+0.37=1.11$
20	3	$0.74+0.44=1.18$
19	4	$0.73+0.49=1.22$
18	5	$0.72+0.53=1.25$
17	6	$0.72+0.56=1.28$
16	7	$0.71+0.59=1.30$
15	8	$0.70+0.61=1.31$
14	9	$0.69+0.63=1.32$
13	10	$0.68+0.64=1.32$
12	11	$0.67+0.66=1.33$

Table 3.3: Minimum age reduction with 2 node engagement and 23 additional links.

Hence, engaging just one node yields the minimum possible age reduction, which upon adding k links to the network of size k mini-FC and $n-k$ isolated nodes results in a network of size $k+1$ mini-FC and $n-k-1$ isolated nodes. ■

Going back to our greedy approach, at step 1, we add one link to a network of n isolated nodes, that creates a size-2 mini-FC and $n-2$ isolated nodes. Next, at step 2, the greedy approach adds two links with the goal of maximizing the age of the resultant system of three links. From Lemma 3.4, we know that greedy approach will engage just one additional node, resulting in a size-3 mini-FC, as in Fig. 3.8(h). Similarly in later steps, the greedy approach will keep adding links in consolidated manner that engages no more than one node at every step, leaving majority of nodes isolated. At the last step, we just add the remaining links again to a single new node.

However, it is not clear if the greedy approach gives the worst network configuration over all possible network configurations with $\bar{n} = \binom{k+1}{2}$ links, $k \in \{1, \ldots, n-1\}$, since Lemma 3.4 only compares all networks that have at least one size-k mini FC and a total of $\binom{k+1}{2}$ links, which is a subset of networks that can be created with $\binom{k+1}{2}$ links. Though it is difficult to compare all network configurations, it would be interesting to see if link placement by the greedy approach, which creates one single mini-FC of size k, also results in higher total age than a network containing multiple mini-FCs of smaller size with the same number of links; see Fig. 3.10.

We know that the total network age for network with size k mini-FC and $n-k$ isolated nodes is given by (3.36). Let us say that the links are instead arranged into a $\bar{m}$ clusters or mini-FCs of size $\bar{k}$, $\bar{m} > 1$, such that the total numbers of links are

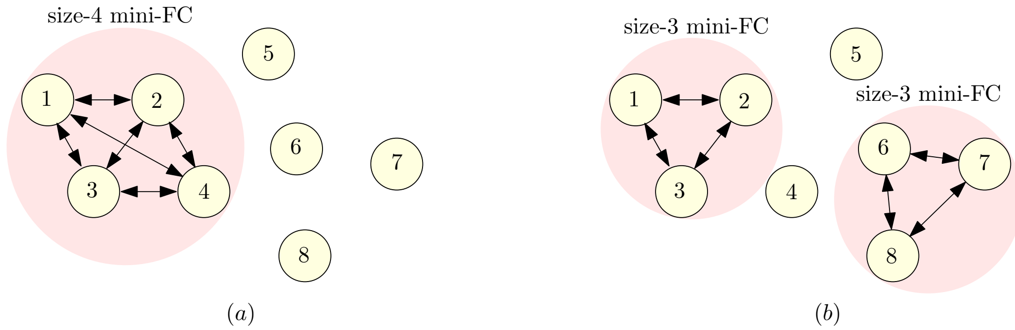


Figure 3.10: (a) A network with $\bar{n} = 6$ links, consisting of one size $k = 4$ mini-FC and 4 isolated nodes. (b) A network with $\bar{n} = 6$ links, consisting of $\bar{m} = 2$ clusters of mini-FCs each of size $\bar{k} = 3$ and 2 isolated nodes.

the same in both cases,

$$\bar{m}\binom{\bar{k}}{2} = \binom{k}{2} \quad \Rightarrow \quad \frac{k(k-1)}{\bar{k}(\bar{k}-1)} = \bar{m} > 1 \tag{3.42}$$

Since the age at each node in a mini-FC of $\bar{k}$ nodes is $\frac{\lambda_s}{\lambda}\frac{n}{\bar{k}}\left(\sum_{j=1}^{\bar{k}}\frac{1}{j}\right)$, the total network age in the second network configuration, e.g., see Fig. 3.10(b), would be

$$\begin{aligned}
&\frac{\lambda_s}{\lambda}\left(\bar{m}\bar{k}\frac{n}{\bar{k}}\left(\sum_{j=1}^{\bar{k}}\frac{1}{j}\right) + (n-\bar{m}\bar{k})n\right) \\
&= \frac{\lambda_s}{\lambda}\left(\frac{k(k-1)}{\bar{k}(\bar{k}-1)}n\left(\sum_{j=1}^{\bar{k}}\frac{1}{j}\right) + (n-\frac{k(k-1)}{\bar{k}(\bar{k}-1)}\bar{k})n\right) && (3.43)\\
&= \frac{\lambda_s}{\lambda}\left(\frac{k(k-1)}{\bar{k}(\bar{k}-1)}n\left(-\sum_{j=1}^{\bar{k}}\frac{j-1}{j}\right) + n^2\right) && (3.44)\\
&\leq \frac{\lambda_s}{\lambda}\left(n\left(-\sum_{j=1}^{k}\frac{j-1}{j}\right) + n^2\right) && (3.45)\\
&= \frac{\lambda_s}{\lambda}\left(n\left(\sum_{j=1}^{k}\frac{1}{j}\right) + (n-k)n\right) && (3.46)
\end{aligned}$$

where the inequality in (3.45) comes from $\frac{1}{\bar{k}(\bar{k}-1)}\left(\sum_{j=1}^{\bar{k}}\frac{j-1}{j}\right)$ being a decreasing function of $\bar{k}$, which in turn can be verified by comparing the expression for $\bar{k}$ and $\bar{k}+1$, and the expression in (3.46) is the same as (3.36). This shows that consolidation of links into one big mini-FC results in higher total network age instead of grouping the links into multiple smaller mini-FCs.

Next, we examine the robustness of the average age $\Delta = \frac{\sum_{i=1}^{n} \Delta_i}{n}$ of the network that results from the remaining $\binom{n}{2} - \tilde{n}$ links (after placing the $\tilde{n}$ jammers) arranged in a manner that creates the network suggested by the greedy approach.

Lemma 3.5 *For a fully connected network of n nodes with $\tilde{n}$ jammers placed such that the remaining links are consolidated in accordance with the greedy approach, the network continues to have $\Delta = O(\log n)$ so long as $\tilde{n} = O(n \log n)$.*

Proof: First, let us consider the case where $\tilde{n}$ is such that there are exactly $\bar{n} = \binom{k}{2}$ links left in the network. Then, from (3.36), the average age in network formed through greedy approach is

$$\Delta = \frac{\sum_{i=1}^{n} \Delta_i}{n} = \frac{\lambda_s}{\lambda} \left(\left(\sum_{j=1}^{k} \frac{1}{j} \right) + (n-k) \right) = O(\log k) + O(n-k) \tag{3.47}$$

where the harmonic series $\sum_{j=1}^{k} \frac{1}{j}$ exceeds the natural logarithm $\log k$ by Euler–Mascheroni constant as limit $k \to \infty$.

The age scaling in (3.47) will continue to be $O(\log n)$ if $n - k = O(\log n)$, i.e., $k = n - O(\log n)$, in which case the total number of links in the network will be

$$\binom{k}{2} = \frac{k(k-1)}{2} = \frac{(n - O(\log n))(n - O(\log n) - 1)}{2} \tag{3.48}$$

$$= \frac{n^2 - O(n \log n)}{2} = \binom{n}{2} - \tilde{n} \tag{3.49}$$

i.e., the number of jammers $\tilde{n} = O(n \log n)$.

Conversely, if $\tilde{n} = O(n \log n)$ jammers are given, then $\binom{k}{2} = \binom{n}{2} - \tilde{n}$ yields

$$k = \sqrt{n^2 - O(n \log n)} \approx n - O(\log n) \tag{3.50}$$

where we used $(1-x)^{\frac{1}{2}} \approx 1 - \frac{x}{2}$ for small x and the terms $\frac{k}{2}$, $\frac{n}{2}$ are masked under $O(n \log n)$. Substituting (3.50) in (3.47) gives $\Delta = O(\log n)$.

Next, let us say $\tilde{n}$ is such that $\binom{k}{2} < \binom{n}{2} - \tilde{n} < \binom{k}{2} + k = \binom{k+1}{2}$. Then, the age of this network can be sandwiched between the age of the network with $\binom{n}{2} - \binom{k+1}{2}$ jammers and $\binom{n}{2} - \binom{k}{2}$ jammers, both of which still have $O(n \log n)$ average age. ■

Lemma 3.5 implies that the fully connected network is robust against $n \log n$ jammers when the jammers are placed according to the network resulting from the greedy approach. That is, both the original fully connected network and the jammed fully connected network have average version age of $\log n$ so long as the number of jammers is up to $n \log n$.

Lemma 3.6 *For a fully connected network of n nodes with $\tilde{n} = cn^\alpha$ jammers, $\alpha \in (1, 2]$, placed in accordance with the greedy approach, the network has $\Delta = O(n^{\alpha-1})$.*

Proof: The proof proceeds similar to (3.50). When $\tilde{n} = cn^\alpha$, $\alpha \in (1, 2)$, then similar

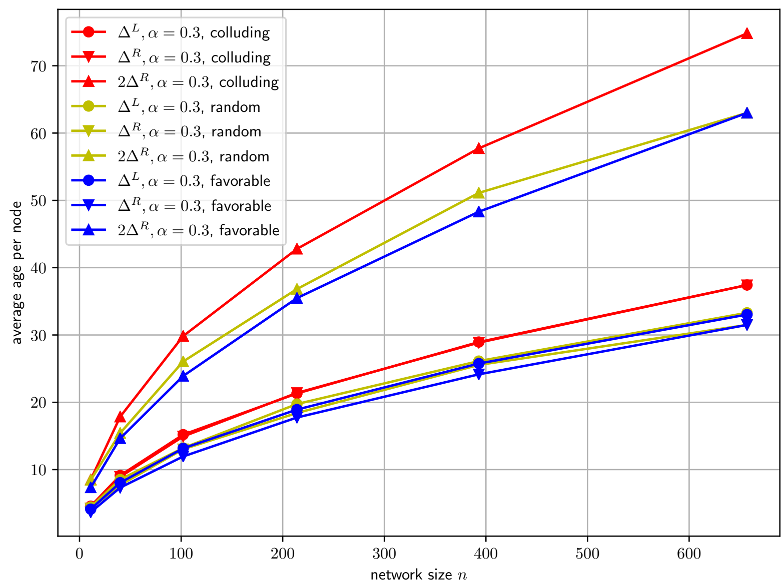


Figure 3.11: Average age in ring network with $n^{0.3}$ jammers

to (3.49), the size of the mini-FC denoted by k can be found as

$$k^2 = n^2 - O(n^\alpha) \tag{3.51}$$

where all $o(n^\alpha)$ terms are masked under the $O(n^\alpha)$ term. Since in the regime of large n, $\frac{n^\alpha}{n^2}$ converges to zero, taking square root and using Taylor approximation, we obtain

$$k \approx n - O(n^{\alpha-1}) \tag{3.52}$$

which, upon subsitution in (3.47) gives $\Delta = O(n^{\alpha-1})$. If instead $\tilde{n} = cn^2$ with $c \in (0, \frac{1}{2})$, then in the regime of large n, we have $\frac{k^2}{2} = \frac{n^2}{2} - cn^2$, which gives $k = n\sqrt{1-2c}$. Hence, the second term in (3.47) is $O(n)$, thereby completing the proof. ■

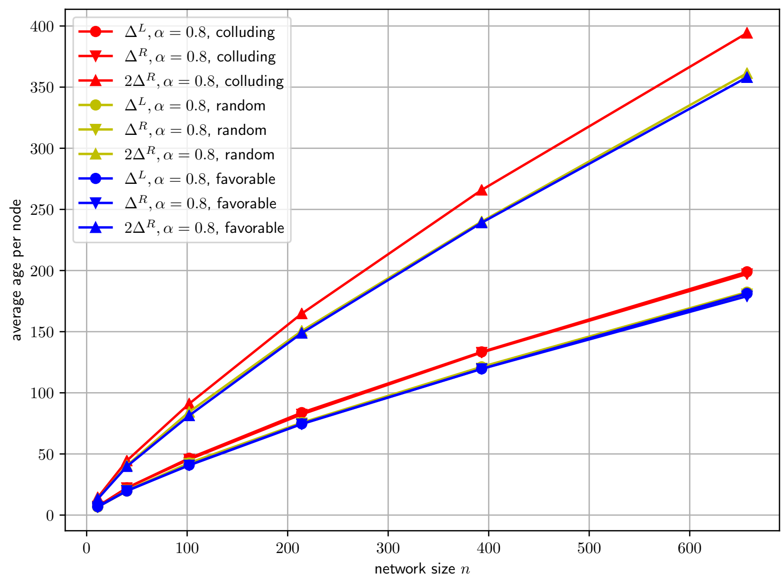


Figure 3.12: Average age in ring network with $n^{0.8}$ jammers.

3.8 Numerical Results

We first validate the aforementioned bounds in the case of ring networks by performing real-time gossip protocol simulations for both dismembered ring and its altered mini-ring model for three cases of jammer positions, a) most harmful or colluding jammer positions, b) random jammer positions, and c) least harmful or favorable jammer positions. For $\tilde{n} = n^\alpha$ and $\frac{\lambda_s}{\lambda} = 1$, Fig. 3.11 shows the average age plots as a function of the network size n for $\alpha = 0.3$. Since the number of jammers $\tilde{n}$ must be an integer, we choose $\tilde{n} = \lfloor n^{0.3} \rfloor$ and accordingly pick $n = \lceil \tilde{n}^{1/0.3} \rceil$, $\tilde{n} \in \mathbb{N}$ on horizontal axis to have $\tilde{n}$ increase consistently with n. In Fig. 3.11, we observe that Δ^ℓ is closely approximated by its lower bound Δ^r in all three cases and both fall between $\sqrt{\frac{\pi}{2}}\sqrt{n} + e^{-\frac{n^{(1-2\times 0.3)}}{2}} n^{0.3}$ of (3.21) and $n^{0.3} + \sqrt{\frac{\pi}{2}}\sqrt{n}$ of (3.25). Notice that the (yellow) graph for the case of random jammer placements is irregular, since we

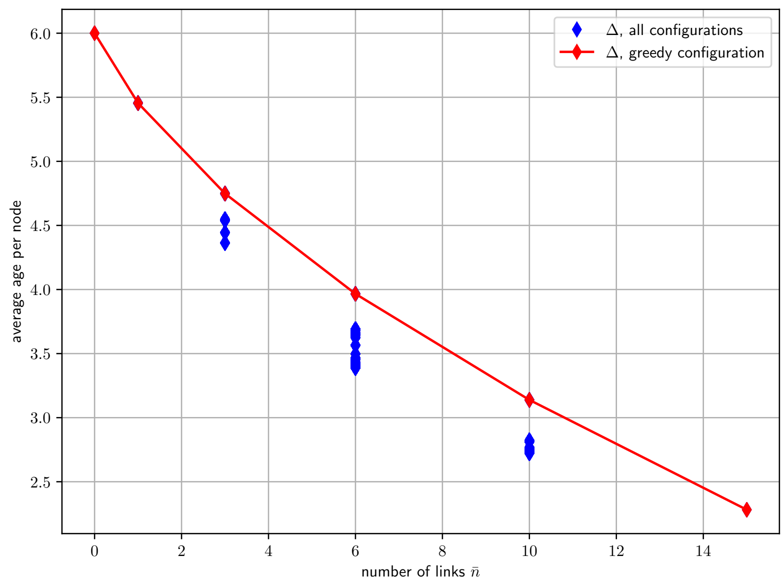


Figure 3.13: Blue dots represent the average age of a network of $n = 6$ nodes for all 8480 network configurations obtained by placing $\bar{n} = \binom{k}{2}, k \in \{1, \ldots, n\}$ links. The red dots represent the average age when the links are added in accordance with the greedy approach. The red dots coincide with the highest blue points for all $\bar{n}$, indicating that the greedy approach, in fact, provides the optimal placement of links that results in maximum possible average age in a network of 6 nodes.

have not chosen any persistent pattern for the choice of jammer positions, unlike the colluding and favorable positions cases. Nevertheless, the age in the former always falls between the latter two extremes. We further note that, in the case of $\alpha = 0.8$ plotted in Fig. 3.12, where the age values increase steeply owing to the presence of larger number of jammers, the graphs look almost linear because as α approaches 1, n^α begins to appear linear. Here again, Δ^ℓ and Δ^r almost overlap, and both fall between $n^{0.8}$ of (3.22) and $n^{0.8} + \sqrt{\frac{\pi}{2}}\sqrt{n}$ of (3.25).

Next, we consider the case of fully connected networks. In Fig. 3.13, the blue points correspond to the average age of network of $n = 6$ nodes for all $\sum_{k=1}^{n} \binom{\binom{n}{2}}{\binom{k}{2}} =$ 8480 network configurations that are possible with $\bar{n}$ links, where we have picked $\bar{n} = \binom{k}{2}$, $k \in \{1, \ldots, n\}$. The red line corresponds to the average age when the

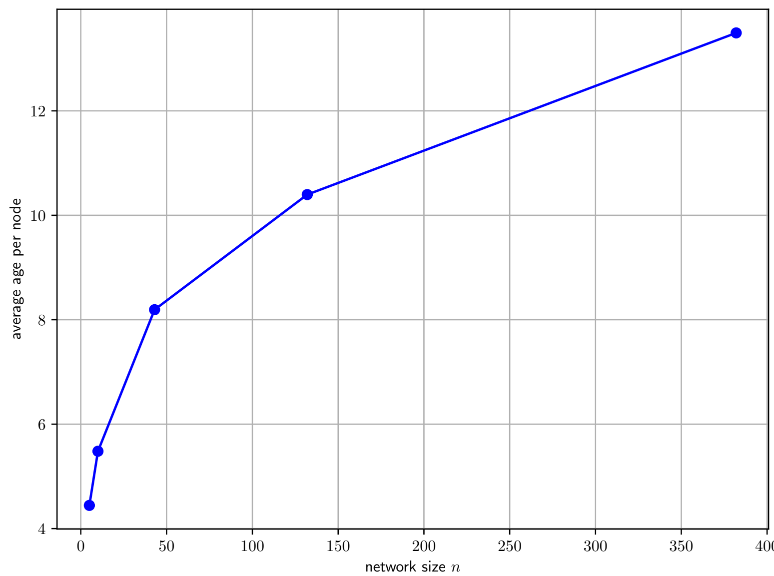


Figure 3.14: Average age in fully connected network with $n \log n$ jammers placed in accordance with the greedy method.

links are placed according to the greedy strategy, which outperforms all other link placements in for a network of 6 nodes, which suggests that greedy approach may be the optimal approach for placing $\bar{n}$ links for maximum age deterioration.

Fig. 3.14 shows the average plot as a function of the network size n when the number of jammers is $\tilde{n} = n \log n$, placed in accordance with the greedy strategy. The number of available links $\bar{n} = \binom{n}{2} - \tilde{n}$ will be of the form $\binom{k}{2} + c$, $c \in \{0, 1, \ldots, k-1\}$, and we pick values of n for which $c = 0$ to have age increase consistently with n, since the graph increases consistently only for fixed values of c. We observe that the graph increases as $2 \log n$ with $k \approx \sqrt{n^2 - 2n \log n} \approx n - \log n$, as predicted in Lemma 3.5.

Fig. 3.15 shows the plot of k and n in red and blue line, respectively, where k and n are related as $\binom{k}{2} + c = \binom{n}{2} - n \log n$, $c \in \{0, 1, \ldots, k-1\}$, i.e., k is the largest size of the mini-FC with $n \log n$ jammers. We observe that the difference between

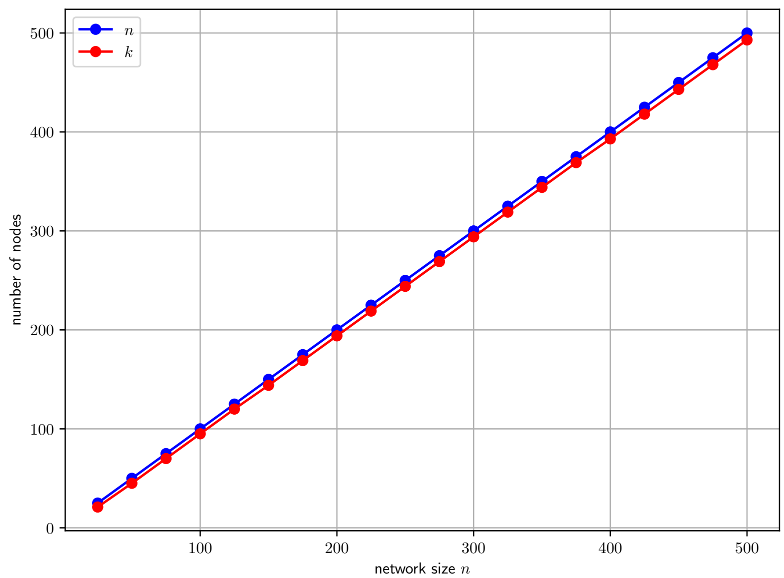


Figure 3.15: The blue line corresponds to the total number of nodes n in the network, and the red line corresponds to the number of nodes k that are left as part of the mini-FC when $n \log n$ jammers are placed on fully connected network in accordance with the greedy method.

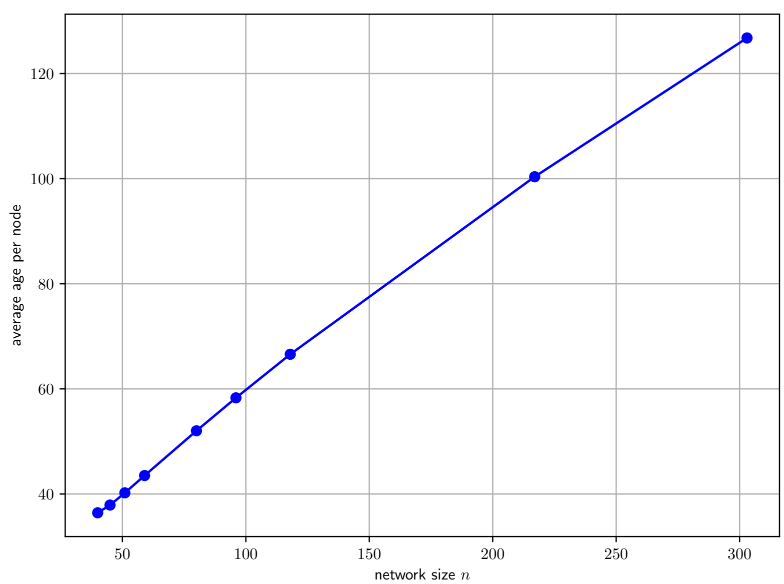


Figure 3.16: Average age in fully connected network with $n^{1.8}$ jammers placed in accordance with the greedy method.

the two lines is very small, since they only differ by a $\log n$ term.

Finally, Fig. 3.16 shows the average age plot as a function of the network size

n when the number of jammers is $\tilde{n} = n^{1.8}$, choosing values of n that give $c = 0$ as before. The average age increases steeply owing to the presence of a larger number of jammers, and the graph looks almost linear because as α approaches 1, n^α begins to appear linear.

3.9 Conclusion

In this chapter, we first studied the effects of jamming attacks on the average age of a ring network. We showed that when the number of jammers $\tilde{n}$ scales as a fractional power of the network size n, i.e., $\tilde{n} = cn^\alpha$, the average version age scales as $\sqrt{n}$ when $\alpha \in \left[0, \frac{1}{2}\right)$, and as n^α when $\alpha \in \left[\frac{1}{2}, 1\right]$, implying that the version age with gossiping is robust against up to $\sqrt{n}$ jammers in a ring network, since version age of a ring network without any jammers scales as $\sqrt{n}$. Along the way, we studied average version age in line networks as well. We then studied the fully connected gossip network, where we derived a greedy approach to place $\tilde{n}$ jammers with the goal to maximize the age of the resultant network, which involved using the $\tilde{n}$ jammers to isolate the maximum possible nodes, thereby consolidating all links into a single mini-fully connected network. We showed in this network that the average version age scales as $O(\log n)$ when $\tilde{n} = O(n \log n)$, and as $O(n^{\alpha-1})$, $1 < \alpha \leq 2$, when $\tilde{n} = O(n^\alpha)$, implying that the network is robust against $n \log n$ jammers, since version age of a fully connected network without jammers scales as $\log n$. Our work shows that connectivity improves resilience against jamming attacks and preserves timeliness of disseminated information; while the ring network (the lowest end of connectivity) is able to neutralize up to $\sqrt{n}$ jammers, the fully connected network (the highest end of connectivity) is able to neutralize up to $n \log n$ jammers, in an n-user gossip network.

CHAPTER 4

Timestomping Vulnerability of Age-Sensitive Gossip Networks

4.1 Introduction

Sensor networks generally have limited resources, which prevents them from implementing traditional computer security techniques, making them vulnerable to adversarial attacks. Uncertain dynamics of such networks often force them to rely on decentralized gossip protocols for information dissemination. In this work, we consider the presence of an adversary in a gossip network, who corrupts the gossip operation by manipulating the timestamps of some data packets flowing in the network, a technique known as *timestomping* [89], with the goal of bringing about staleness and inefficiency to the network. A timestomping attack can be launched in many ways. For instance, a malicious insider node can deviate from the gossip protocol and inject old packets by rebranding them as fresh packets via timestamp manipulation, while maintaining the gossiping frequency to evade suspicion. Other methods include *meddler in the middle* (MITM) attacks, where the adversary inserts its node undetected between two nodes and manipulates communication, and *eclipse* attacks where the adversary manipulates the target node by redirecting its inbound and outbound links away from legitimate neighboring nodes to adversary controlled nodes, thereby isolating the node from the rest of the network, as encountered in gossip based blockchain networks.

As per the notation followed throughout this chapter, given $U_i(t)$ as the timestamp of the packet with node i at time t, the instantaneous age of information is given by $X_i(t) = t - U_i(t)$. The nodes wish to have access to the most up-to-date information at all times, and therefore, are prompted to decrease $X_i(t)$ by fetching packets with more recent timestamps, e.g., with higher $U_i(t)$, therefore a malicious entity, through manipulation of timestamps of packets, can replace circulation of fresh packets with outdated packets in the network.

Timestomping is often used by malware authors as an anti-forensics technique to make files blend in with the rest of the system. In this work, an adversary uses timestomping with the goal of worsening the expected age in the network. Consider two nodes, A and B, that randomly come in contact to exchange information and consider the presence of adversary at node A capable of altering timestamps of all incoming and outgoing files. If node A is outdated compared to node B, the adversary would be inclined to increase the timestamp of an outgoing packet from node A to make it appear fresher so as to misguide node B into discarding its packet in favor of a staler packet, and also, decrease the timestamp of an incoming packet from node B so as to avoid its acceptance at node A. Conversely, if node A is more up-to-date than node B, the adversary would reduce timestamps of outgoing packets and increase timestamps of incoming packets to make node B reject fresher files and node A accept staler files. More the manipulated timestamps digress away from their true value, higher are the chances of error in deciding which packet should be discarded, since this decision is based on a comparison of timestamps. At time t, the maximum error is caused when file timestamp is either changed to the current time t or the earliest time 0. Thus, we consider an adversary, who, for each packet, makes the decision of changing its timestamp to either t or 0. The adversary is *oblivious* in that it does not look into a packet and see its actual timestamp[1] . Thus, the adversary changes the timestamp to either t or 0 probabilistically.

In this chapter, we consider gossip networks where an adversary captures a node and manipulates the timestamps of packets coming into and going out of the node, and demonstrate how network topology capacitates an adversary to influence age scaling in a network. We show that in a fully connected network (FCN), see Fig. 4.1, one infected node can single-handedly suppress the availability of fresh information in a large network of n users employing a gossip protocol, and increase the expected age from $O(\log n)$ found in [4] to $O(n)$. In addition, we show that the optimal action for the adversary is to always increase the timestamp of every outgoing packet to t and decrease the timestamp of every incoming packet to 0, in effect, preventing all incoming files from being accepted by the infected node and actively persuading other nodes to always accept outgoing packets from the infected node. Further, we show that if the the infected node is allowed to accept even a small fraction of incoming packets from the network, then a large network can curb

[1]Even if the adversary sees the timestamp marked on a packet prior to altering it, the adversary does not know the actual timestamp of the packets. The adversary cannot tell if the timestamp marked on an encountered packet is the actual timestamp generated at the source, or a manipulated timestamp in case the packet passed through the adversarial node previously, which makes it difficult for the adversary to base its decision on these timestamps.

the spread of stale files coming from the infected node by effectively lowering the infected node's age. These observations show how the fully connected nature of a network can be both a benefit and a detriment for network staleness. Additionally, we show that if the malicious node contacts only one other node instead of all nodes of the network, see Fig. 4.3, the system age can still be degraded to $O(n)$, which highlights how little an effort is needed on the part of the adversary to bring down the freshness of the entire network.

We then switch to the other end of the network connectivity spectrum and analyze the unidirectional ring network (URN), see Fig. 4.4. As opposed to the fully connected network, where each node uniformly at random contacts all nodes of the network, in unidirectional ring each node devotes all its resources to establish contact with just one other node. Here we show that the adversarial effect on age scaling of a node is limited by its distance from the adversary, and the age scaling for a larger fraction of the network still continues to be $O(\sqrt{n})$ irrespective of adversarial policy. Note that in a unidirectional (or bidirectional) ring with no adversary, the age scaling of a node follows $O(\sqrt{n})$ for a network size n. We show that the optimal action for the adversary here again is to always increase the timestamp of every outgoing packet to t and to decrease the timestamp of every incoming packet to 0, similar to the fully connected network.

In the remainder of the chapter, we frequently make use of the following four asymptotic notations: $O(\cdot)$, $o(\cdot)$, $\Omega(\cdot)$ and $\omega(\cdot)$ for big–O, little–o, big–Omega and little–omega, respectively. If $f(n)$ and $g(n)$ are functions that map positive integers to positive real numbers, then

- $f(n) = O(g(n))$ iff there exists a positive integer n_0 and a positive constant c such that $f(n) \leq cg(n)$ for all $n \geq n_0$.

- $f(n) = o(g(n))$ iff for any positive constant c, there exists a positive integer n_0 such that $f(n) < cg(n)$ for all $n \geq n_0$, i.e., $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$.

- $f(n) = \Omega(g(n))$ iff there exists a positive integer n_0 and a positive constant c such that $f(n) \geq cg(n)$ for all $n \geq n_0$.

- $f(n) = \omega(g(n))$ iff for any positive constant c, there exists a positive integer n_0 such that $f(n) > cg(n)$ for all $n \geq n_0$, i.e., $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$.

For example, $f(n) = 2n^2 + 3n$ is $O(n^2)$, $O(n^3)$, $o(n^3)$, $\Omega(n)$, $\Omega(n^2)$, $\Omega(1)$, $\omega(n)$ and $\omega(1)$. For $f(\cdot)$ and $g(\cdot)$ to be of the same order, $f(n)$ should be both $O(g(n))$ and $\Omega(g(n))$, for instance, $f(n)$ is exactly order n^2, since constant multiples of n^2 serve as both lower and upper bounds to $f(n)$.

4.2 System Model and SHS Characterization

We first study a fully connected network (FCN), shown in Fig. 4.1, which comprises a source and n user nodes $\mathcal{N} = \{1, \ldots, n\}$. The source, alternatively referred to as node 0, is assumed to always posses the latest file packet and consequently has zero age at all times. The nodes wish to acquire the most up-to-date file to lower their average age from the source, who updates each user node as a Poisson process with rate $\frac{\lambda}{n}$. Further, a user node i randomly sends its current packet to a user node j according to a Poisson process with rate $\lambda_{ij} = \frac{\lambda}{n-1}$. Thus, all nodes send out

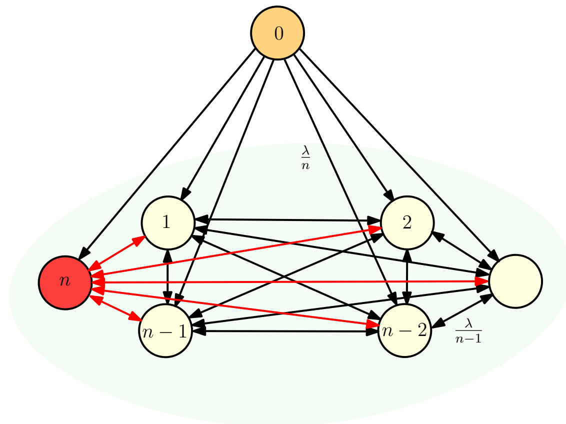


Figure 4.1: Fully connected network of n nodes with an infected node.

updates after exponential inter-update times with a total rate λ. Let $U_j(t)$ denote the timestamp marked on the file stored at node j. Then, at the receiving node j, the claimed timestamp of the incoming packet is compared with $U_j(t)$ to determine which packet should be kept. Note that a node always accepts an update from the source which generates update packets with the current timestamp t.

We assume that the highest index node, node n, is under attack by an adversary that manipulates the timestamps of all incoming and outgoing packets of node n. In the following sections, we alternately refer to node n as the infected node or adversarial node. For the outgoing packets, the adversary chooses to increase the timestamp (to current time t) with probability p and decrease the timestamp to 0 with probability $1-p$. Similarly, the adversary increases and decreases the timestamp of incoming packets with probability $1-q$ and q, respectively. We will refer to the nodes in set $\mathcal{N}_R = \{1, \ldots, n-1\}$ as *regular* nodes and node n as the *infected* node. We assume that the infected node always accepts packets from the source like other nodes, delivered to it with rate $\frac{\lambda}{n}$, which helps the adversary evade suspicion of malicious activity by maintaining a remote contemporary relevance of the contents of its manipulated packets.

We denote the long-term average age at node i by v_i, where $v_i = \lim_{t\to\infty} \mathbb{E}[X_i(t)]$, and wish to study its extent of deterioration through timestomping. Note that the actual instantaneous age at node i is $X_i(t) = t - \bar{U}_i(t)$, where $\bar{U}_i(t)$ indicates the true packet generation time, which can be different from the claimed timestamp $U_i(t)$ if the file timestamp has been tampered with. For a set of nodes S at time t, let $X_{N(S)}(t)$ indicate the actual instantaneous age of the node claiming to possess the most recent timestamped packet in set S, i.e., $X_{N(S)}(t) = X_{\arg\max_{j\in S} U_j(t)}(t)$. We define $v_S = \lim_{t\to\infty} \mathbb{E}[X_{N(S)}(t)]$. Here we would like to point out that in a network without adversary where all files are marked with true timestamps, $X_{N(S)}(t)$ reduces to $X_S(t) = \min_{j\in S} X_j(t)$ defined in [4], since the node with the highest timestamp will also have the lowest age in the set S.

Reference [4] demonstrates how stochastic hybrid system (SHS) models yield linear equations useful for deriving long-term average age at nodes in a gossip network of n users with any arbitrary topology. Due to the presence of a timestomping adversary, we choose the continuous state for our SHS model as $(\boldsymbol{X}(t), \boldsymbol{U}(t)) \in \mathbb{R}^{2n}$, where $\boldsymbol{X}(t) = [X_1(t), \ldots, X_n(t)]$ denotes the instantaneous ages at the n nodes and $\boldsymbol{U}(t) = [U_1(t), \ldots, U_n(t)]$ denotes the timestamps marked on the packets at the n nodes at time t. The convenience of the SHS based age characterization follows from the presence of a single discrete mode with trivial stochastic differential equation $(\dot{\boldsymbol{X}}(t), \dot{\boldsymbol{U}}(t)) = (\mathbf{1}_n, \mathbf{0}_n)$, where the age at each node grows at unit rate when there is no update transfer, since the timestamps of the node packets do not change between such transitions. Consider a test function $\psi : \mathbb{R}^{2n} \times [0,\infty) \to \mathbb{R}$ that is time-invariant, i.e., its partial derivative with respect to t is $\frac{\partial \psi(\boldsymbol{X},\boldsymbol{U},t)}{\partial t} = 0$, such that we are interested in finding its long-term expected value $h_\psi = \lim_{t\to\infty} \mathbb{E}[\psi(\boldsymbol{X}(t),\boldsymbol{U}(t),t)]$. Since the test function only depends on the continuous state values $(\boldsymbol{X},\boldsymbol{U})$ and is time-invariant, for simplicity, we will drop the third input t and write $\psi(\boldsymbol{X},\boldsymbol{U},t)$ as $\psi(\boldsymbol{X},\boldsymbol{U})$, which we assume to flow according to the differential equation $\dot{\psi}(\boldsymbol{X}(t),\boldsymbol{U}(t)) = 1$. Let $\mathcal{L}$ correspond to the set of directed edges (i,j), such that node i sends updates to node j on this edge according to a Poisson process of rate λ_{ij}, with this transition resetting the state $(\boldsymbol{X},\boldsymbol{U})$ at time t to $\phi_{i,j}(\boldsymbol{X},\boldsymbol{U},t) \in \mathbb{R}^{2n}$ post transition. Defining $h_{\psi\phi}(i,j) = \lim_{t\to\infty} \mathbb{E}[\psi(\phi_{i,j}(\boldsymbol{X}(t),\boldsymbol{U}(t),t))]$, [90, Thm. 1] yields

$$0 = 1 + \sum_{(i,j)\in\mathcal{L}} \lambda_{ij}(h_{\psi\phi}(i,j) - h_\psi) \tag{4.1}$$

which is similar to derivations in [4], where the left side becomes 0 as expectations stabilize. We will use this equation repeatedly by defining a series of time-invariant test functions appropriate for our analysis. For more details, the reader is encouraged to consult references [90] and [4].

4.3 Fully Connected Network

Note that packets arriving at infected node n from a node $i \in \mathcal{N}_R$ with rate $\frac{\lambda}{n-1}$ Poisson process are accepted (or discarded) with probability $1-q$ (or q) when the adversary changes timestamp of incoming packet to t (or 0) to make it appear fresh (or stale). This is equivalent to packets arriving at node n from node i with thinned Poisson process with rate $\lambda_{in} = \frac{(1-q)\lambda}{n-1}$ such that these packets are always accepted. The remaining packets are always discarded and have no effect on age dynamics of the system. Similarly, as the outgoing packets from the infected node n are accepted at node $i \in \mathcal{N}_R$ with probability p, this is equivalent to node n sending packets with timestamp t to node i with a thinned Poisson process of rate $\lambda_{ni} = \frac{p\lambda}{n-1}$ such that these packets are always accepted. The rates λ_{ij} for each transition (i, j) are

$$\lambda_{ij} = \begin{cases} \frac{\lambda}{n}, & i = 0, j \in \mathcal{N} \\ \frac{\lambda}{n-1}, & i, j \in \mathcal{N}_R \\ \frac{p\lambda}{n-1}, & i = n, j \in \mathcal{N}_R \\ \frac{(1-q)\lambda}{n-1}, & i \in \mathcal{N}_R, j = n \end{cases} \tag{4.2}$$

Thus, based on transition (i, j) at time t, the reset map $\phi_{i,j}(\boldsymbol{X}, \boldsymbol{U}, t) = [X'_1, \ldots, X'_n, U'_1, \ldots, U'_n]$ can be described by

$$U'_\ell = \begin{cases} t, & i = 0, j \in \mathcal{N}, \ell = j \\ \max\{U_i, U_\ell\}, & i, j \in \mathcal{N}_R, \ell = j \\ t, & i = n, j \in \mathcal{N}_R, \ell = j \\ t, & i \in \mathcal{N}_R, j = n, \ell = j \\ U_\ell, & \text{otherwise} \end{cases} \tag{4.3}$$

and

$$X'_\ell = \begin{cases} 0, & i = 0, j \in \mathcal{N}, \ell = j \\ X_{N(\{i,\ell\})}, & i, j \in \mathcal{N}_R, \ell = j \\ X_n, & i = n, j \in \mathcal{N}_R, \ell = j \\ X_i, & i \in \mathcal{N}_R, j = n, \ell = j \\ X_\ell, & \text{otherwise} \end{cases} \tag{4.4}$$

Here, $X_{N(S)} = X_{\arg\max_{j \in S} U_j}$ for state $(\boldsymbol{X}, \boldsymbol{U})$ and a subset of nodes S. Since all regular nodes have statistically similar age processes, every arbitrary set S_k of k regular nodes will have the same expected age v_{S_k}, $S_k \subseteq \mathcal{N}_R$, with $v_{S_1} = v_1$. We pick our first test function to be $\psi(\boldsymbol{X}, \boldsymbol{U}) = X_{N(S_k)}$, which is modified upon transition (i, j) to $\psi(\phi_{i,j}(\boldsymbol{X}, \boldsymbol{U}, t)) = X'_{N(S_k)}$. This in turn is characterized using (4.3) and (4.4) as

$$X'_{N(S_k)} = \begin{cases} 0, & i = 0, j \in S_k, \ell = j \\ X_{N(S_k \cup \{i\})}, & i \in \mathcal{N}_R \backslash S_k, j \in S_k, \ell = j \\ X_n, & i = n, j \in S_k, \ell = j \\ X_{N(S_k)}, & \text{otherwise} \end{cases} \tag{4.5}$$

To check if this test function satisfies $\dot{\psi}(\boldsymbol{X}(t), \boldsymbol{U}(t)) = 1$, remember that $\dot{\boldsymbol{X}}(t) = \mathbf{1}_n$ and hence $X_{N(S_k)}(t) = X_{\arg\max_{j \in S_k} U_j(t)}(t)$ grows at unit rate between transitions.

Employing (4.1), this test function yields,

$$0 =1 + \frac{k\lambda}{n}(0 - v_{S_k}) + \frac{(n-k-1)k\lambda}{n-1}(v_{S_{k+1}} - v_{S_k}) + \frac{kp\lambda}{n-1}(v_n - v_{S_k}) \tag{4.6}$$

which upon rearrangment gives

$$v_{S_k} = \frac{\frac{1}{k\lambda} + \frac{n-k-1}{n-1}v_{S_{k+1}} + \frac{pv_n}{n-1}}{\frac{1}{n} + \frac{n-k-1}{n-1} + \frac{p}{n-1}} \tag{4.7}$$

Our second test function is simply $\psi(\boldsymbol{X},\boldsymbol{U}) = X_n$, i.e., the age at infected node, such that its (i,j) transition map is

$$X_n' = \begin{cases} 0, & i = 0, j = n, \ell = j \\ X_i, & i \in \mathcal{N}_R, j = n, \ell = j \\ X_n, & \text{otherwise} \end{cases} \tag{4.8}$$

which upon proceeding similarly to (4.6) and (4.7), gives

$$v_n = \frac{\frac{1}{\lambda} + (1-q)v_1}{\frac{1}{n} + (1-q)} \tag{4.9}$$

Our goal is to obtain an analytical expression for the expected age of a regular node $v_{S_1} = v_1$, by making use of (4.7) and (4.9). We now proceed to analyze how the choice of probabilities p and q by the adversary affects the expected age at regular nodes in the regime of large n.

4.3.1 FCN Case 1: $p > 0, q = 1$

These probabilities pertain to the case where the infected node blocks all incoming packets from the regular nodes, and at the same time sends out packets with timestamps manipulated to t with rate $\frac{p\lambda}{n-1}$ to each regular node, misleading them into accepting these packets.

Using $\frac{1}{n} \leq \frac{1}{n-1}$ and $p \leq 1$ in the denominator of (4.7) gives the following lower bound

$$v_{S_k} \geq \frac{\frac{1}{k\lambda} + \frac{n-k-1}{n-1} v_{S_{k+1}} + \frac{pv_n}{n-1}}{\frac{2}{n-1} + \frac{n-k-1}{n-1}} \tag{4.10}$$

Letting $y_k = v_{S_k} \frac{n-k}{n-1}$ in (4.10) gives

$$y_k \geq \frac{n-k}{n-k+1} \left(y_{k+1} + \frac{1}{k\lambda} + \frac{pv_n}{n-1} \right) \tag{4.11}$$

Starting from $y_1 = v_1$, and successively substituting for $y_2, y_3, \ldots, y_{n-1}$, we obtain

$$v_1 \geq \frac{1}{\lambda} \sum_{k=1}^{n-1} \frac{n-k}{nk} + pv_n \sum_{k=1}^{n-1} \frac{n-k}{n(n-1)} \tag{4.12}$$

$$= \frac{1}{\lambda} \sum_{k=1}^{n-1} \frac{1}{k} - \frac{1}{\lambda} \frac{n-1}{n} + \frac{pv_n}{n(n-1)} \sum_{k=1}^{n-1} k \tag{4.13}$$

$$= \Omega(\log n) + \frac{pv_n}{2} \tag{4.14}$$

since $\sum_{k=1}^{n-1} \frac{1}{k}$ grows asymptotically as $\log n$ and $0 \leq \frac{n-1}{n} \leq 1$. The v_n in the second term can in turn be obtained by substituting $q = 1$ in (4.9), giving $v_n = \frac{n}{\lambda}$. Hence,

$$v_1 \geq \Omega(\log n) + \frac{pn}{2\lambda} = \Omega(n) \tag{4.15}$$

In other words, the age at a regular node in this case scales at least as n. In fact, we next show that the age at a regular node in this case scales at most as n as well, i.e., v_1 is $O(n)$.

We add $\frac{1}{n-1} - \frac{p}{n-1} - \frac{1}{n} = \frac{1-np}{n(n-1)}$ (which is ≤ 0 for large n) to the denominator of (4.7) to get the following upper bound

$$v_{S_k} \leq \frac{\frac{1}{k\lambda} + \frac{n-k-1}{n-1} v_{S_{k+1}} + \frac{pv_n}{n-1}}{\frac{1}{n-1} + \frac{n-k-1}{n-1}} \tag{4.16}$$

where we again employ $y_k = v_{S_k} \frac{n-k}{n-1}$ which gives

$$y_k \leq y_{k+1} + \frac{1}{k\lambda} + \frac{pv_n}{n-1} \tag{4.17}$$

Here an iterative substitution for $y_2, y_3, \ldots, y_{n-1}$ with $y_1 = v_1$ yields

$$v_1 \leq \frac{1}{\lambda} \sum_{k=1}^{n-1} \frac{1}{k} + pv_n \tag{4.18}$$

Since $\sum_{k=1}^{n-1} \frac{1}{k} = O(\log n)$ and $v_n = \frac{n}{\lambda} = O(n)$, we conclude v_1 is $O(n)$.

To put this deterioration in age scaling into perspective, remember that in a fully disconnected network with no gossiping [76], expected age at each node also scales as $O(n)$, to be exact, $\frac{n}{\lambda}$, a fact that will come handy later.

4.3.2 FCN Case 2: $p = 0, q \leq 1$

In this case, the infected node accepts timestomped packets from the $n-1$ regular nodes with rate $\frac{(1-q)\lambda}{n-1}$ but effectively does not transmit any files, even when it possesses the latest file, thereby limiting its contribution, positive or negative, to the system age.

Substituting $p = 0$ in (4.7) gives

$$v_{S_k} = \frac{\frac{1}{k\lambda} + \frac{n-k-1}{n-1} v_{S_{k+1}}}{\frac{1}{n} + \frac{n-k-1}{n-1}} \tag{4.19}$$

Again defining $y_k = v_{S_k} \frac{n-k}{n-1}$, using $\frac{n-k-1}{n-1} \geq \frac{n-k-1}{n}$ in the denominator gives $y_k \leq \frac{n}{n-1}(y_{k+1} + \frac{1}{k\lambda})$. Since $\left(\frac{n}{n-1}\right)^j \leq \left(\frac{n}{n-1}\right)^{n-1}$ for $j \leq n-1$, an iterative substitution for $k = \{1, \ldots, n-1\}$ gives

$$v_1 = y_1 \leq \left(\frac{n}{n-1}\right)^{n-1} \left(\frac{1}{\lambda} \sum_{k=1}^{n-1} \frac{1}{k}\right) = O(\log n) \tag{4.20}$$

where we employed $\lim_{n\to\infty} \left(\frac{n}{n-1}\right)^{n-1} = e$ and $\sum_{k=1}^{n-1} \frac{1}{k} = O(\log n)$. Hence, such an adversary does not affect the network age scaling. Further, specifically for the case of $q < 1$, using $\frac{1}{n} \geq 0$ in (4.9) gives

$$v_n = \frac{\frac{1}{\lambda} + (1-q)v_1}{\frac{1}{n} + (1-q)} \leq \frac{1}{\lambda(1-q)} + v_1 \tag{4.21}$$

Hence v_n scales as $O(\log n)$ similar to v_1, which indicates that the deterioration of age at the infected node itself is negligible as long as it accepts even a small fraction of incoming packets. For the case of $q = 1$, when infected node does not accept any packets from the gossip process, (4.9) gives $v_n = O(n)$, since it behaves as an isolated node.

4.3.3 FCN Case 3: $p > 0, q < 1$

In this case, the adversary partially both accepts and sends timestomped packets to regular nodes. Here, we employ a combination of (4.18) and (4.21) to get

$$v_1 \leq \frac{1}{\lambda} \sum_{k=1}^{n-1} \frac{1}{k} + p\left(\frac{1}{\lambda(1-q)} + v_1\right) \tag{4.22}$$

which upon rearrangement gives

$$v_1 \leq \frac{1}{(1-p)} \left(\frac{1}{\lambda} \sum_{k=1}^{n-1} \frac{1}{k} + \frac{p}{\lambda(1-q)}\right) \tag{4.23}$$

where $\sum_{k=1}^{n-1} \frac{1}{k} = O(\log n)$. Hence, v_1 scales as $O(\log n)$ and (4.21) provides that v_n also scales as $O(\log n)$. This case, in combination with *Case 2*, highlights how the infected node fails to deteriorate the age scaling at regular nodes from $O(\log n)$ as long as the infected node receives packets from the gossip network.

Before we end this section, we remark that the maximum age deterioration in the network caused by the adversary happens when $p = 1$ and $q = 1$, i.e., a special case of *Case 1*. To see this, we first note that $O(n)$ age is achieved only when $q = 1$, as can be seen by comparing all the three cases above. Thus, we take $q = 1$. When $q = 1$, we have $v_n = \frac{n}{\lambda}$. To show the optimality of $p = 1$, we rewrite (4.7) as follows where we use v_n and $\frac{n}{\lambda}$ interchangeably

$$
\begin{aligned}
v_{S_k} =& \frac{\frac{1}{k\lambda} + \frac{n-k-1}{n-1} v_{S_{k+1}} + \frac{p v_n}{n-1}}{\frac{1}{n} + \frac{n-k-1}{n-1} + \frac{p}{n-1}} && (4.24)\\
=& \frac{\left(1 - 1 + \frac{1}{k}\right)\frac{1}{\lambda} + \frac{n-k-1}{n-1}(v_n - v_n + v_{S_{k+1}}) + \frac{p v_n}{n-1}}{\frac{1}{n} + \frac{n-k-1}{n-1} + \frac{p}{n-1}} && (4.25)\\
=& \frac{n}{\lambda}\left(\frac{\frac{1}{n} + \frac{n-k-1}{n-1} + \frac{p}{n-1}}{\frac{1}{n} + \frac{n-k-1}{n-1} + \frac{p}{n-1}}\right) + \frac{\left(-1 + \frac{1}{k}\right)\frac{1}{\lambda} + \frac{n-k-1}{n-1}(-v_n + v_{S_{k+1}})}{\frac{1}{n} + \frac{n-k-1}{n-1} + \frac{p}{n-1}} && (4.26)\\
=& \frac{n}{\lambda} - \frac{\left(1 - \frac{1}{k}\right)\frac{1}{\lambda} + \frac{n-k-1}{n-1}(v_n - v_{S_{k+1}})}{\frac{1}{n} + \frac{n-k-1}{n-1} + \frac{p}{n-1}} && (4.27)
\end{aligned}
$$

Bringing $v_n = \frac{n}{\lambda}$ to the left hand side, we obtain

$$
v_n - v_{S_k} = \frac{\left(1 - \frac{1}{k}\right)\frac{1}{\lambda} + \frac{n-k-1}{n-1}(v_n - v_{S_{k+1}})}{\frac{1}{n} + \frac{n-k-1}{n-1} + \frac{p}{n-1}} \tag{4.28}
$$

Substituting $k = n-1$ yields

$$
v_n - v_{S_{n-1}} = \frac{\left(1 - \frac{1}{n-1}\right)\frac{1}{\lambda}}{\frac{1}{n} + \frac{p}{n-1}} \tag{4.29}
$$

Thus, $v_n - v_{S_{n-1}}$ is positive and a decreasing function of p. Hence, $v_{S_{n-1}}$ is an increasing function of p. Thereafter, using (4.28), we recursively show that $v_n - v_{S_k}$, for $k = n-2, n-3, \ldots, 1$, is a decreasing function of p, hence, v_{S_k} for all k, is an increasing function of p. Therefore, $v_1 = v_{S_1}$ is an increasing function of p. Hence, the age deterioration in the network caused by the adversary increases with increasing p, and the maximum happens when $p = 1$.

Before we proceed to the next section, we would like to point out that actively sending out packets with timestamp t can cause the malfunctioning node to be easily

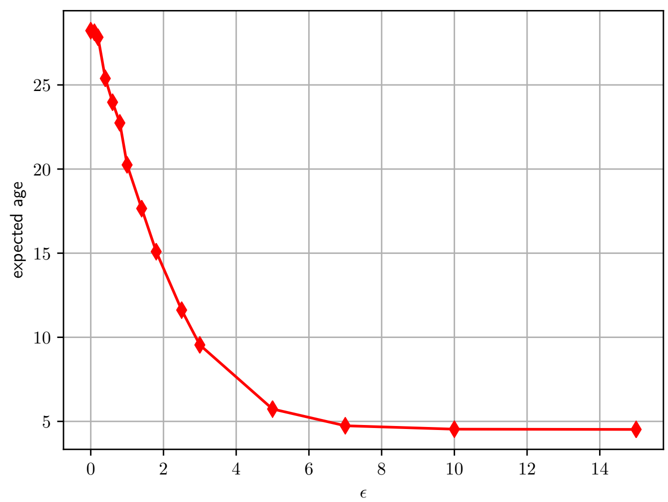


Figure 4.2: Age as function of ϵ when timestamps of outgoing packets are changed to $t-\epsilon$ instead of t by the timestomping adversary, for $n=50$ nodes, $\lambda=1$, $p=1$.

detected, since it is a probabilistically unlikely event that the timestamp of a packet from a node other than the source equals the current time t, i.e., a source-to-node and node-to-node transmissions have occurred simultaneously. However, the adversary could get around this detection mechanism by changing packet timestamps to $t-\epsilon$ by choosing a small ϵ. Fig. 4.2 shows the age at a typical network node in a fully connected network of $n = 50$ nodes, plotted as a function of ϵ, with the adversary changing the timestamp of outgoing packets to $t-\epsilon$ instead of t. Though this is analytically difficult problem, we observe from Fig. 4.2 that the impact of the timestomping attack becomes lesser with increasing ϵ, as the decreasing age improves the freshness of information at nodes. This aligns with our discussion in the introduction, emphasizing that the more the manipulated timestamps deviate from their true value, the higher the confusion in the network.

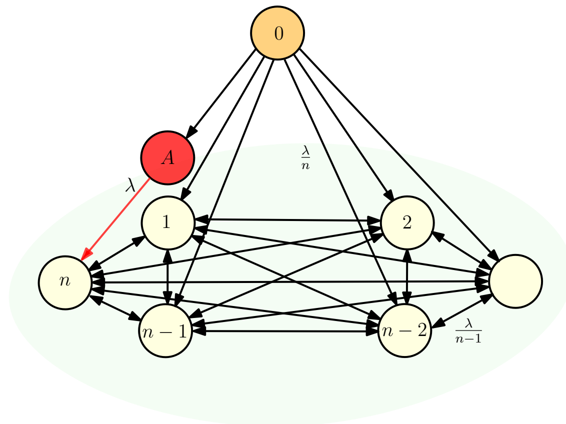


Figure 4.3: MITM attack on fully connected network of n nodes.

4.4 MITM Attack on Fully Connected Network

In previous sections, the adversarial node was in direct contact with all other nodes due to the fully connected nature of the network (relevant in context of eclipse attacks or node capture attacks), and the adversary could raise the system age to $O(n)$ using probabilities $p = 1$ and $q = 1$, effectively sending out all packets with timestamp t and not accepting any packets from the gossip process. Here, an interesting question to ask is, if the network could do better if the adversary instead had access to only one node. To this end, we consider the network model of Fig. 4.3, where the adversary, which we will refer to as node A, intercepts the updates to node n coming from the source. In turn, the adversary sends updates with its total rate λ, after changing the timestamps of every outgoing packet to the current time, only to node n (this corresponds to meddler-in-the-middle (MITM) attack on a single link).

Clearly the expected age at the adversary, denoted by v_A, scales as $O(n)$ since it is isolated from the gossip network and only receives updates from the source with rate $\frac{\lambda}{n}$. The two reset maps useful for our analysis are

$$X'_{N(S_k\cup\{n\})} = \begin{cases} 0, & i=0, j\in S_k, \ell=j \\ X_{N(S_{k+1}\cup\{n\})}, & i\in\mathcal{N}_R\backslash S_k, j\in S_k\cup\{n\}, \ell=j \\ X_A, & i=A, j=n, \ell=j \\ X_{N(S_k\cup\{n\})}, & \text{otherwise} \end{cases} \tag{4.30}$$

and

$$X'_{N(S_k)} = \begin{cases} 0, & i=0, j\in S_k, \ell=j \\ X_{N(S_{k+1})}, & i\in\mathcal{N}_R\backslash S_k, j\in S_k, \ell=j \\ X_{N(S_k\cup\{n\})}, & i=n, j\in S_k, \ell=j \\ X_{N(S_k)}, & \text{otherwise} \end{cases} \tag{4.31}$$

We claim $v_{S_k\cup\{n\}} \geq \frac{v_A}{2}$, a loose lower bound that is trivially verified with induction as follows. Invoking (4.1) regarding (4.30) for $k=n-1$ results in

$$v_{S_{n-1}\cup\{n\}} = \frac{\frac{1}{\lambda}+v_A}{\frac{n-1}{n}+1} \geq O(1)+\frac{v_A}{2} \geq \frac{v_A}{2} \tag{4.32}$$

which verifies the claim for $k=n-1$. Next, we assume that the claim holds for $k+1$, i.e., $v_{S_{k+1}\cup\{n\}} \geq \frac{v_A}{2}$, and verify for k. Invoking (4.1) regarding (4.30) for

$k \leq n-2$ and using $\frac{1}{\lambda} > 0$ in the numerator and $\frac{k}{n} \leq 1$ in the denominator gives

$$v_{S_k \cup \{n\}} = \frac{\frac{1}{\lambda} + \frac{(k+1)(n-1-k)}{n-1} v_{S_{k+1} \cup \{n\}} + v_A}{\frac{k}{n} + \frac{(k+1)(n-1-k)}{n-1} + 1} \tag{4.33}$$

$$\geq \frac{\frac{(k+1)(n-1-k)}{n-1} v_{S_{k+1} \cup \{n\}}}{\frac{(k+1)(n-1-k)}{n-1} + 2} + \frac{v_A}{\frac{(k+1)(n-1-k)}{n-1} + 2} \tag{4.34}$$

$$\geq \frac{\frac{(k+1)(n-1-k)}{n-1} \frac{v_A}{2}}{\frac{(k+1)(n-1-k)}{n-1} + 2} + \frac{\frac{2v_A}{2}}{\frac{(k+1)(n-1-k)}{n-1} + 2} \tag{4.35}$$

$$= \frac{v_A}{2} \tag{4.36}$$

Finally, we re-invoke (4.1) for (4.31) which results in

$$v_{S_k} = \frac{\frac{1}{k\lambda} + \frac{n-k-1}{n-1} v_{S_{k+1}} + \frac{v_{S_k \cup \{n\}}}{n-1}}{\frac{1}{n} + \frac{n-k-1}{n-1} + \frac{1}{n-1}} \tag{4.37}$$

Let $y_k = v_{S_k} \frac{n-k}{n-1}$, using $\frac{1}{n-1} \approx \frac{1}{n}$ for large n, (4.37) becomes

$$y_k = \frac{n-k}{n-k+1} \left(y_{k+1} + \frac{1}{k\lambda} + \frac{v_{S_k \cup \{n\}}}{n-1} \right) \tag{4.38}$$

$$\geq \frac{n-k}{n-k+1} y_{k+1} + \frac{(n-k) v_{S_k \cup \{n\}}}{(n-k+1)(n-1)} \tag{4.39}$$

Starting from $y_1 = v_1$, we substitute for $y_2, y_3, \ldots, y_{n-1}$ and use $v_{S_k \cup \{n\}} \geq \frac{v_A}{2}$ to obtain

$$v_1 \geq \frac{1}{n(n-1)} \sum_{k=1}^{n-1} (n-k) v_{S_k \cup \{n\}} \tag{4.40}$$

$$\geq \frac{v_A}{2n(n-1)} \sum_{k=1}^{n-1} (n-k) = \frac{v_A}{4} \tag{4.41}$$

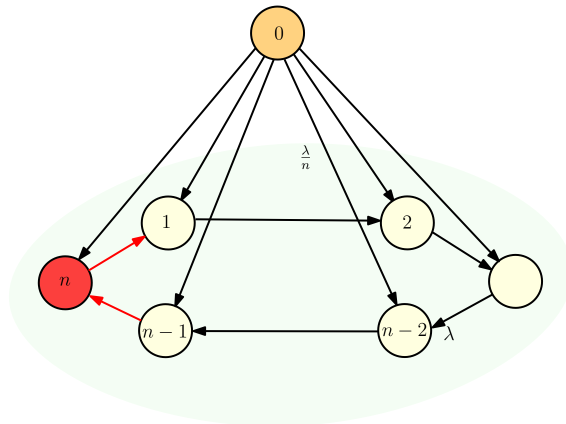


Figure 4.4: Unidirectional ring comprising $n-1$ regular nodes and an infected node.

Hence, v_1 scales at least as $O(n)$ for all regular nodes. This result is far from intuitive, for it brings home the point how an adversary, with so little an effort as sending tampered packets to just one node, can bring down the freshness of an entire large gossip network.

4.5 Unidirectinal Ring Network

We now consider the resource constrained unidirectional ring network (URN), shown in Fig. 4.4, which represents the other end of the network connectivity spectrum. This network differs from the fully connected network in the sense that each node uses its entire rate λ to transmit packets to a single node, i.e., node i only transmits to node $j = (i \bmod n) + 1$. The source node, referred to as node 0, always possesses the latest file packet, which it transmits to each user node in the set $\mathcal{N} = \{1, \ldots, n\}$ according to a Poisson process with rate $\frac{\lambda}{n}$. The nodes accept all packets coming from the source since it generates update packets with current timestamp t. We refer to node n as the *infected* node and nodes in the set $\mathcal{N}_R = \{1, \ldots, n-1\}$ as *regular* nodes. Similar to Section 4.2, the oblivious adversary increases timestamps to current time t of outgoing and incoming packets with probabilities p and $1-q$, respectively. Effectively this means that packets arrive at the infected node n from node $n-1$ according to a thinned Poisson process with rate $(1-q)\lambda$ while the infected node n sends packets to node 1 according to a thinned Poisson process with rate $p\lambda$. The rates λ_{ij} for each transition (i,j) are given as

$$\lambda_{ij} = \begin{cases} \frac{\lambda}{n}, & i=0, j \in \mathcal{N} \\ \lambda, & i \in \mathcal{N}_R \backslash \{n-1\}, j=i+1 \\ p\lambda, & i=n, j=1 \\ (1-q)\lambda, & i=n-1, j=n \end{cases} \tag{4.42}$$

Thus, based on transition (i,j) at time t, the reset map $\phi_{i,j}(\boldsymbol{X},\boldsymbol{U},t) = [X'_1,\ldots,X'_n,U'_1,\ldots,U'_n]$ can be described by

$$U'_\ell = \begin{cases} t, & i=0, j \in \mathcal{N}, \ell=j \\ \max\{U_i,U_\ell\}, & i \in \mathcal{N}_R \backslash \{n-1\}, j=i+1, \ell=j \\ t, & i=n, j=1, \ell=j \\ t, & i=n-1, j=n, \ell=j \\ U_\ell, & \text{otherwise} \end{cases} \tag{4.43}$$

and

$$X'_\ell = \begin{cases} 0, & i = 0, j \in \mathcal{N}, \ell = j \\ X_{N(\{\ell-1,\ell\})}, & i \in \mathcal{N}_R \backslash \{n-1\}, j = i+1, \ell = j \\ X_n, & i = n, j = 1, \ell = j \\ X_{n-1}, & i = n-1, j = n, \ell = j \\ X_\ell, & \text{otherwise} \end{cases} \tag{4.44}$$

Keeping in mind that for state $(\boldsymbol{X}, \boldsymbol{U})$ and subset $S = \{m-k+1, \ldots, m\}$, $X_{N(S)} = X_{\arg\max_{j \in S} U_j}$ chooses the age of the node claiming to possess the packet with the latest timestamp among the set of k contiguous nodes, we wish to find the long-term average age v_m at node m, where $v_m = \lim_{t\to\infty} \mathbb{E}[X_m(t)]$. To this purpose, for some $k \in \{1, \ldots, m-1\}$, the following test function will be useful

$$X'_{N(\{m-k+1,\ldots,m\})} = \begin{cases} 0, & i = 0, j \in S_k, \ell = j \\ X_{N(\{m-k,\ldots,m\})}, & i = m-k, j = m-k+1, \\ & \ell = j \\ X_{N(\{m-k+1,\ldots,m\})}, & \text{otherwise} \end{cases} \tag{4.45}$$

Using (4.1), (4.42) and (4.45), we get the following recursive equation

$$v_{\{m-k+1,\ldots,m\}} = \frac{1 + \lambda v_{\{m-k,\ldots,m\}}}{\lambda + \frac{k\lambda}{n}} = \frac{\frac{1}{\lambda}}{1 + \frac{k}{n}} + \frac{v_{\{m-k,\ldots,m\}}}{1 + \frac{k}{n}} \tag{4.46}$$

Note that when $k = 1$, $v_{\{m-k+1,\ldots,m\}} = v_{\{m,\ldots,m\}} = v_m$, since $X_{N(\{m\})} = X_m$. Solving together the set of equations obtained by substituting in (4.46) all values of $k \in \{1, \ldots, m-1\}$, we obtain

$$v_m = \frac{1}{\lambda} \sum_{j=1}^{m-1} \prod_{k=1}^{j} \frac{1}{1+\frac{k}{n}} + v_{\{1,\ldots,m\}} \prod_{k=1}^{m-1} \frac{1}{1+\frac{k}{n}} \tag{4.47}$$

The next test function of interest then is

$$X'_{N(\{1,\ldots,m\})} = \begin{cases} 0, & i = 0, j \in S_k, \ell = j \\ X_n, & i = n, j = 1, \ell = j \\ X_{N(\{1,\ldots,m\})}, & \text{otherwise} \end{cases} \tag{4.48}$$

which combined with (4.1) and (4.42) gives

$$v_{\{1,\ldots,m\}} = \frac{1 + p\lambda v_n}{p\lambda + \frac{m\lambda}{n}} \tag{4.49}$$

Clearly

$$\frac{1 + p\lambda v_n}{\lambda + \frac{m\lambda}{n}} \leq v_{\{1,\ldots,m\}} \leq \frac{1 + p\lambda v_n}{p(\lambda + \frac{m\lambda}{n})} \tag{4.50}$$

which coupled with (4.47) gives

$$\frac{1}{\lambda} \sum_{j=1}^{m} \prod_{k=1}^{j} \frac{1}{1+\frac{k}{n}} + p v_n \prod_{k=1}^{m} \frac{1}{1+\frac{k}{n}} \leq v_m \leq \frac{1}{p\lambda} \sum_{j=1}^{m} \prod_{k=1}^{j} \frac{1}{1+\frac{k}{n}} + v_n \prod_{k=1}^{m} \frac{1}{1+\frac{k}{n}} \tag{4.51}$$

The last useful test function is

$$X'_n = \begin{cases} 0, & i = 0, j \in S_k, \ell = j \\ X_{n-1}, & i = n-1, j = n, \ell = j \\ X_{N(\{1,\ldots,m\})}, & \text{otherwise} \end{cases} \tag{4.52}$$

which combined with (4.1) and (4.42) gives

$$v_n = \frac{1 + (1-q)\lambda v_{n-1}}{(1-q)\lambda + \frac{\lambda}{n}} = \frac{\frac{1}{\lambda}}{1 - q + \frac{1}{n}} + \frac{(1-q)v_{n-1}}{1 - q + \frac{1}{n}} \tag{4.53}$$

Before we move forward to analyze the adversarial effect on age scaling at regular nodes for different cases of probabilities p and q, we state the following two lemmas that will be helpful in our derivations. Lemma 4.1 and Lemma 4.2 below are refined and generalized versions of [76, Lemma 2]. Our results are more precise and general as they allow summations to go up to an arbitrary n_0 which is a function of n and give order-wise exact expressions.

Lemma 4.1 *Let $n_0 = \omega(\sqrt{n})$, then $\sum_{j=1}^{n_0} \left(\prod_{k=1}^{j} \frac{1}{1+\frac{k}{n}} \right)$ is both $O(\sqrt{n})$ and $\Omega(\sqrt{n})$.*

Proof: Since

$$\frac{x}{1+x} \leq \log(1+x) \leq x \tag{4.54}$$

we choose $x = \frac{k}{n}$ to get

$$\frac{\frac{k}{n}}{1+\frac{k}{n}} \leq \log\left(1+\frac{k}{n}\right) \leq \frac{k}{n} \tag{4.55}$$

$$\frac{k}{2n} \leq \frac{k}{n+k} \leq \log\left(1+\frac{k}{n}\right) \leq \frac{k}{n} \tag{4.56}$$

Summing over k gives

$$\frac{1}{2}\sum_{k=1}^{j}\frac{k}{n} \leq \sum_{k=1}^{j}\log\left(1+\frac{k}{n}\right) \leq \sum_{k=1}^{j}\frac{k}{n} \tag{4.57}$$

Using $\sum_{k=1}^{j}\frac{k}{n} = \frac{j(j+1)}{2n}$ and $\frac{j^2}{2n} \leq \frac{j(j+1)}{2n} \leq \frac{j^2}{n}$ gives the following bounds

$$\frac{j^2}{4n} \leq \sum_{k=1}^{j}\log\left(1+\frac{k}{n}\right) \leq \frac{j^2}{n} \tag{4.58}$$

Note that

$$-\log\left(\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}}\right) = \sum_{k=1}^{j}\log\left(1+\frac{k}{n}\right) \tag{4.59}$$

and therefore, raising all sides to negative exponent in (4.58) gives

$$e^{-\frac{j^2}{n}} \leq \prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}} \leq e^{-\frac{j^2}{4n}} \tag{4.60}$$

and consequently

$$\frac{1}{\sqrt{n}}\sum_{j=1}^{n_0}e^{-\frac{j^2}{n}} \leq \frac{1}{\sqrt{n}}\sum_{j=1}^{n_0}\left(\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}}\right) \leq \frac{1}{\sqrt{n}}\sum_{j=1}^{n_0}e^{-\frac{j^2}{4n}} \tag{4.61}$$

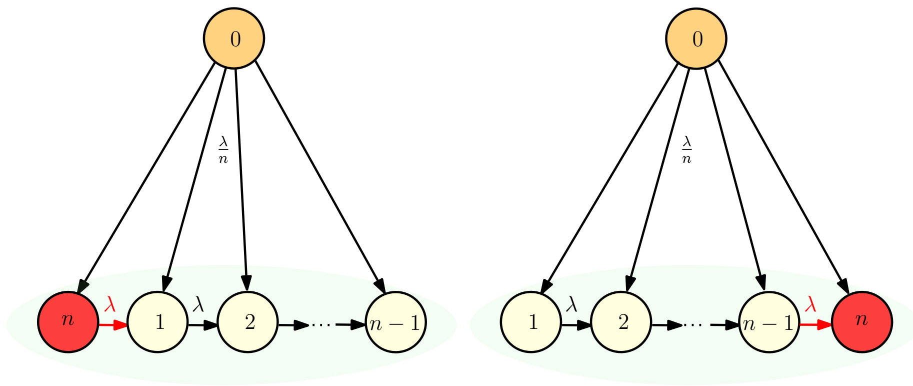


Figure 4.5: Line network transformations of unidirectional ring.

Finally, note that, for any constant C, using upper and lower Riemann sums, we get

$$\int_{\frac{1}{\sqrt{n}}}^{\frac{n_0}{\sqrt{n}}} e^{-\frac{t^2}{C}}\,dt \le \frac{1}{\sqrt{n}}\sum_{j=1}^{n_0} e^{-\frac{j^2}{Cn}} \le \int_0^{\frac{n_0}{\sqrt{n}}} e^{-\frac{t^2}{C}}\,dt \tag{4.62}$$

Hence, for $n_0 = \omega(\sqrt{n})$ and for large n,

$$\frac{1}{\sqrt{n}}\sum_{j=1}^{n_0} e^{-\frac{j^2}{Cn}} = \int_0^{\infty} e^{-\frac{t^2}{C}}\,dt = \frac{\sqrt{C\pi}}{2} \tag{4.63}$$

Thus, (4.61) results in $\sum_{j=1}^{n_0}\left(\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}}\right)$ being both $O(\sqrt{n})$ and $\Omega(\sqrt{n})$. ■

Lemma 4.2 *Let $n_0 = O(\sqrt{n})$, then $\sum_{j=1}^{n_0}\left(\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}}\right)$ is $O(\sqrt{n})$.*

The proof of Lemma 4.2 follows similarly to the proof of Lemma 4.1.

We now analyze how the choice of probabilities p and q by the adversary in unidirectional ring affect the expected age at regular nodes in the regime of large n.

4.5.1 URN Case 1: $p > 0, q = 1$

This corresponds to the case where the infected node n does not receive anything from the gossip process but sends out packets, with timestamps manipulated to current time t, to node 1 according Poisson process with rate $p\lambda$. Since the network topology under consideration is a unidirectional ring, this case essentially converts the ring network of Fig. 4.4 into the line network model shown in Fig. 4.5(a). In this case, (4.53) gives $v_n = \frac{n}{\lambda} = O(n)$, since the infected node is an isolated node that only receives updates from the source. Further, from (4.51) we have

$$\frac{1}{\lambda}\sum_{j=1}^{m}\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}} + p\frac{n}{\lambda}\prod_{k=1}^{m}\frac{1}{1+\frac{k}{n}} \leq v_m \leq \frac{1}{p\lambda}\sum_{j=1}^{m}\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}} + \frac{n}{\lambda}\prod_{k=1}^{m}\frac{1}{1+\frac{k}{n}} \tag{4.64}$$

The presence of adversary at node n breaks the symmetry in the ring and the age processes at all the regular nodes are no longer statistically similar. Therefore we analyze how expected age v_m scales at node index m, where m scales as a specific function of network size n.

URN Case 1(a): $m = O(n^{\frac{1}{2}})$. For the first term in the bounds of (4.64), we know that $\sum_{j=1}^{m}\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}}$ is $O(\sqrt{n})$ from Lemma 4.2. The second term in (4.64) has product of n and $\prod_{k=1}^{m}\frac{1}{1+\frac{k}{n}}$. From (4.60), we know $e^{-\frac{m^2}{n}} \leq \prod_{k=1}^{m}\frac{1}{1+\frac{k}{n}} \leq e^{-\frac{m^2}{4n}}$. Since $m = O(\sqrt{n})$, for any constant C, $e^{-\frac{m^2}{Cn}}$ converges to a non-zero constant for large n, making the second term $O(n)$. Hence, we conclude from the second term of (4.64) that v_m is both $O(n)$ and $\Omega(n)$, and expected age at node m scales as n.

URN Case 1(b): $m = \Omega(n^{\frac{1}{2}+\epsilon})$, $\epsilon > 0$. In this case, Lemma 4.1 provides that

$\sum_{j=1}^{m}\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}}$ is both $O(\sqrt{n})$ and $\Omega(\sqrt{n})$. Further, for any constant C, both $e^{-\frac{m^2}{Cn}}$ and $ne^{-\frac{m^2}{Cn}}$ converge to 0 as n becomes large. Hence, the first term of (4.64) provides that expected age at node m scales as $\sqrt{n}$, and thus, v_m is both $O(\sqrt{n})$ and $\Omega(\sqrt{n})$.

URN Case 1(c): $m = \Omega(n^{\frac{1}{2}+\epsilon(n)})$, $\epsilon(n) = \frac{\log\alpha+\log\log n}{2\log n}$, $\alpha > 0$. This case gives a flavor of the transitional behavior between *Case 1(a)* and *Case 1(b)* above. From (4.60), $\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}}$ scales as $e^{-\frac{j^2}{\beta n}}$, $\beta \in [1,4]$, and consequently, the expression $n\prod_{k=1}^{m}\frac{1}{1+\frac{k}{n}}$ in the second term in the upper bound and lower bound of (4.64) becomes

$$n\prod_{k=1}^{m}\frac{1}{1+\frac{k}{n}} \approx ne^{-\frac{m^2}{\beta n}} = ne^{-\frac{\alpha n\log n}{\beta n}} = nn^{-\frac{\alpha}{\beta}} = n^{1-\frac{\alpha}{\beta}} = n^{1-\gamma} \tag{4.65}$$

When $1-\gamma > \frac{1}{2}$, which is true for smaller values of α, then the expected age v_m at node index m scales as $n^{1-\gamma}$, since the first term in the bounds of (4.64) is at most $O(\sqrt{n})$.

4.5.2 URN Case 2: $p = 0, q \leq 1$

This corresponds to the case where no timestomped packets are effectively injected into the gossiping process, since all packets sent by the infected node n carry a timestamp of 0 causing them get rejected at the receiving node 1, even when the received packet has fresh content. This effectively converts the ring network of Fig. 4.4 into the line network of Fig. 4.5(b). We now analyze how absence of a contributing node before node 1 affects scaling behaviour of the expected age at other regular nodes of the network. In this case, (4.49) becomes $v_{\{1,\ldots,m\}} = \frac{n}{m\lambda}$

which transforms (4.47) into

$$v_m = \frac{1}{\lambda}\left(\sum_{j=1}^{m-1}\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}} + \frac{n}{m}\prod_{k=1}^{m-1}\frac{1}{1+\frac{k}{n}}\right) \tag{4.66}$$

We now look at expected age v_m when m scales as a specific function of network size n.

URN Case 2(a): $m = o(\sqrt{n})$. The first term in (4.66) is $O(\sqrt{n})$ from Lemma 4.2. The second term can be bounded from above and below by expressions of form $\frac{n}{m}e^{-\frac{(m-1)^2}{Cn}}$ due to (4.60), where C is some constant. In this case, in the regime of large n, $e^{-\frac{(m-1)^2}{Cn}}$ converges to $e^0 = 1$, and hence, the age scaling in this case is determined by the fraction $\frac{n}{m}$ and is at least $\omega(\sqrt{n})$. For example, if $m = n^\alpha, \alpha \in [0, \frac{1}{2})$, then v_m scales as $O(n^{1-\alpha})$ and $\Omega(n^{1-\alpha})$.

URN Case 2(b): $m = \Omega(\sqrt{n})$. Here in (4.66), Lemma 4.1 provides that $\sum_{j=1}^{m-1}\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}}$ is $O(\sqrt{n})$. Further $\frac{n}{m}e^{-\frac{(m-1)^2}{Cn}}$ is $O(\sqrt{n})$, since $e^{-\frac{(m-1)^2}{Cn}}$ converges to a constant and $\frac{n}{m}$ is $O(\sqrt{n})$.

4.5.3 URN Case 3: $p > 0, q < 1$

This corresponds to the case where the infected node both partially receives packets from node $n-1$ and partially sends packets to node 1 carrying timestamps manipulated to t. This case is particularly interesting, for it highlights how the adversary fails to deteriorate age scaling at regular nodes when it both sends and receives packets via gossiping at the same time.

We see from (4.53) that v_n is dependent on v_{n-1}, which in turn can be obtained using a combination of (4.47) and (4.49) as follows

$$v_{n-1} = \frac{1}{\lambda}\sum_{j=1}^{n-2}\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}} + v_{\{1,\dots,n-1\}}\prod_{k=1}^{n-2}\frac{1}{1+\frac{k}{n}} \tag{4.67}$$

$$= \frac{1}{\lambda}\sum_{j=1}^{n-2}\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}} + \frac{1+p\lambda v_n}{p\lambda+\frac{(n-1)\lambda}{n}}\prod_{k=1}^{n-2}\frac{1}{1+\frac{k}{n}} \tag{4.68}$$

thereby giving v_{n-1} in terms of v_n, which can be plugged back into (4.53) as

$$v_n = \frac{\frac{1}{\lambda}}{1-q+\frac{1}{n}} + \frac{(1-q)}{1-q+\frac{1}{n}} \times \left(\frac{1}{\lambda}\sum_{j=1}^{n-2}\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}} + \frac{1+p\lambda v_n}{p\lambda+\frac{(n-1)\lambda}{n}}\prod_{k=1}^{n-2}\frac{1}{1+\frac{k}{n}}\right) \tag{4.69}$$

to obtain age of v_n as

$$\begin{aligned} v_n =& \frac{1}{\lambda}.\frac{\frac{1}{1-q+\frac{1}{n}}}{1-\frac{(1-q)}{(1-q)+\frac{1}{n}}\left(\frac{p}{p+\frac{(n-1)}{n}}\prod_{k=1}^{n-2}\frac{1}{1+\frac{k}{n}}\right)} \\ &+ \frac{1}{\lambda}.\frac{\frac{(1-q)}{1-q+\frac{1}{n}}\left(\sum_{j=1}^{n-2}\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}} + \frac{1}{p+\frac{(n-1)}{n}}\prod_{k=1}^{n-2}\frac{1}{1+\frac{k}{n}}\right)}{1-\frac{(1-q)}{(1-q)+\frac{1}{n}}\left(\frac{p}{p+\frac{(n-1)}{n}}\prod_{k=1}^{n-2}\frac{1}{1+\frac{k}{n}}\right)} \end{aligned} \tag{4.70}$$

Using $\frac{(1-q)}{(1-q)+\frac{1}{n}} \leq 1$ and $\prod_{k=1}^{n-2}\frac{1}{1+\frac{k}{n}} \leq 1$, we obtain the following upper bound

$$v_n \leq \frac{1}{\lambda}.\frac{\frac{1}{1-q+\frac{1}{n}} + \left(\sum_{j=1}^{n-2}\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}} + \frac{1}{p+\frac{(n-1)}{n}}\right)}{1-\left(\frac{p}{p+\frac{(n-1)}{n}}\right)} \tag{4.71}$$

We first find the age scaling at the infected node v_n in the regime of large network size n. The bounds $\frac{n-1}{n} \geq \frac{1}{2}$ for $n \geq 2$, $\frac{1}{1-q+\frac{1}{n}} \leq \frac{1}{1-q}$, and $\sum_{j=1}^{n-2}\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}} \leq$

$\sum_{j=1}^{n}\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}}$ provide the following further simplification of the upper bound

$$v_n \leq \frac{1}{\lambda}.\frac{\frac{1}{1-q}+\left(\sum_{j=1}^{n}\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}}+\frac{1}{p+\frac{1}{2}}\right)}{1-\left(\frac{p}{p+\frac{1}{2}}\right)} \tag{4.72}$$

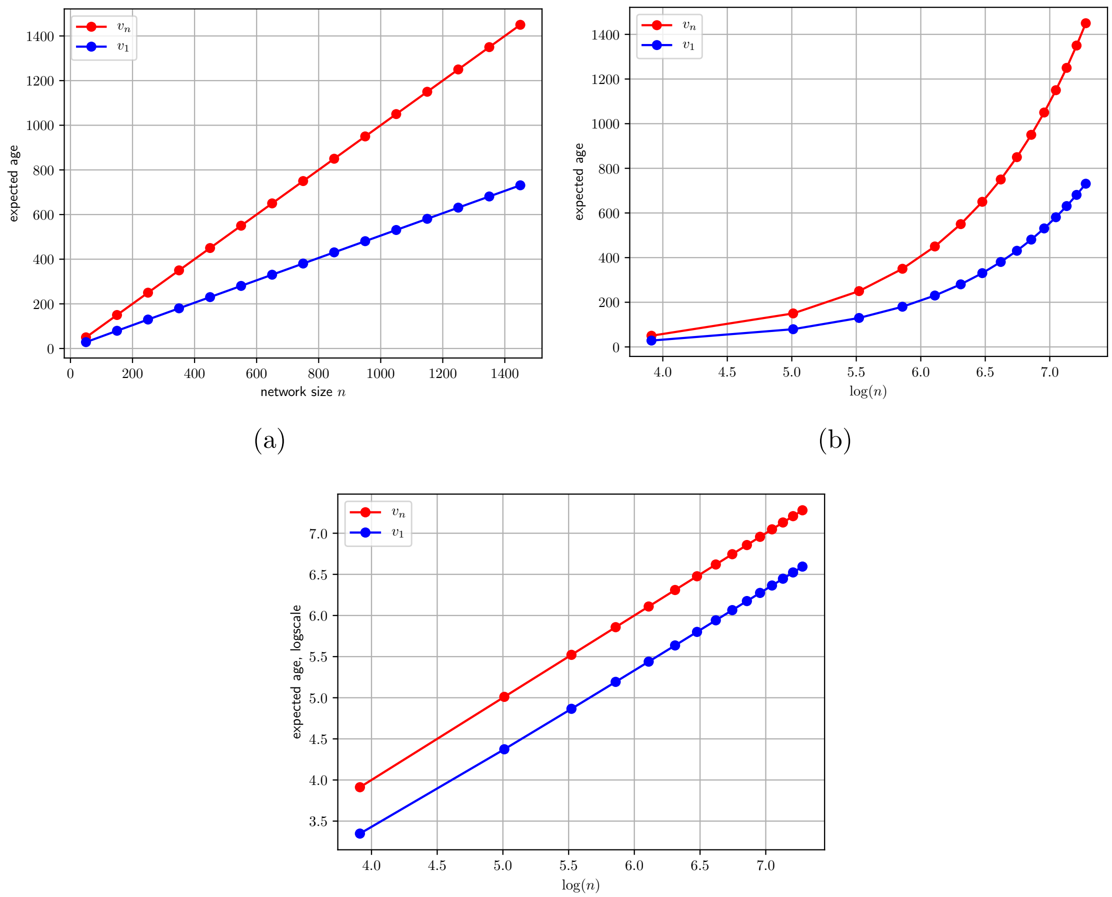
Since $q < 1$ and $p > 0$ in this case, the above mentioned bound comprises primarily constants and age is thus determined by the term $\sum_{j=1}^{n}\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}}$, which is $O(\sqrt{n})$ from Lemma 4.1. Hence, irrespective of how large q is, the age at the infected node still scales as $O(\sqrt{n})$ instead of $O(n)$. This observation is similar to what was seen in fully connected networks, where we observed that when the infected node accepts packets via gossiping, then the gossip network brings down the age of the infected node from $O(n)$ to $O(\log n)$ characteristic of fully connected networks, the analogue of which is $O(\sqrt{n})$ for the unidirectional ring.

Now, we proceed to see how age scales at a typical regular node on the ring, for which we consider individual terms in the upper and lower bound of (4.51). The first term comprises $\sum_{j=1}^{m}\prod_{k=1}^{j}\frac{1}{1+\frac{k}{n}}$ which is $O(\sqrt{n})$ irrespective of node index m from Lemmas 4.1 and 4.2. The second term contains a product of v_n and $\prod_{k=1}^{m}\frac{1}{1+\frac{k}{n}}$, where the former is $O(\sqrt{n})$ from (4.72) and the later is bounded from (4.60) by $e^{-\frac{m^2}{4n}} \leq 1$. Hence, v_m is $O(\sqrt{n})$ for each node index m.

4.5.4 Key Insights

URN Case 2 highlights an interesting attribute of the expected age in the unidirectional ring. Upon cutting one node-to-node link on the ring, while the age still scales as $O(n)$ at the first exposed node, it transitions (more gradually in *URN Case 2* than *URN Case 1*) to $O(\sqrt{n})$ until $m = o(\sqrt{n})$, after which it stays at $O(\sqrt{n})$ for the rest of the ring. Note that in the absence of any adversary or jammed link, [76] shows that age scales as $\sqrt{n}$ at all nodes of a unidirectional ring. Though node 1 in *URN Case 2* could not receive any packets from the gossip process and had to rely on occasional updates from the source, the missing link before node 1 affects the age in only the initial portion of the ring and for nodes of higher indices the age still scales as $O(\sqrt{n})$. Further, we note that v_n is $O(\sqrt{n})$ for $q < 1$ and $O(n)$ for $q = 1$ in *URN Case 2*, this can be obtained by noting that $v_{n-1} = O(\sqrt{n})$ from *URN Case 2(b)* since $n - 1 = \Omega(\sqrt{n})$, and using $\frac{1}{n} \geq 0$ in (4.53).

Finally, its worthwhile to note that the maximum age deterioration in the unidirectional ring is caused when the adversary chooses to actively send out packets with timestamp t to the following node 1 and reject all incoming packets from preceding node $n-1$, i.e., $q = 1$ and $p = 1$, a special case of *URN Case 1*, and therefore it represents the preferred attack method for the adversary in the unidirectional ring. This is because $v_n = \frac{n}{\lambda} = O(n)$ and for this case $v_{\{1,\dots,m\}}$ in (4.49) (and consequently v_m in (4.47)) turns out to be an increasing function of p, a fact that can be verified by taking first derivative of $v_{\{1,\dots,m\}}$ in (4.49) with respect to p, which justifies choice of $p = 1$. Further, in *URN Case 1*, which corresponds to $q = 1$, the expected age v_m is $O(n)$ for significantly larger portion of the ring than in *URN Case 2*, where the expected age v_m reduces form $O(n)$ to $O(\sqrt{n})$ more gradually. Finally, we note that for any meaningful age scaling deterioration, the adversary must choose between *URN Case 1* and *URN Case 2*, i.e., the infected node should either not accept packets at all but only inject packets in the gossiping process, or



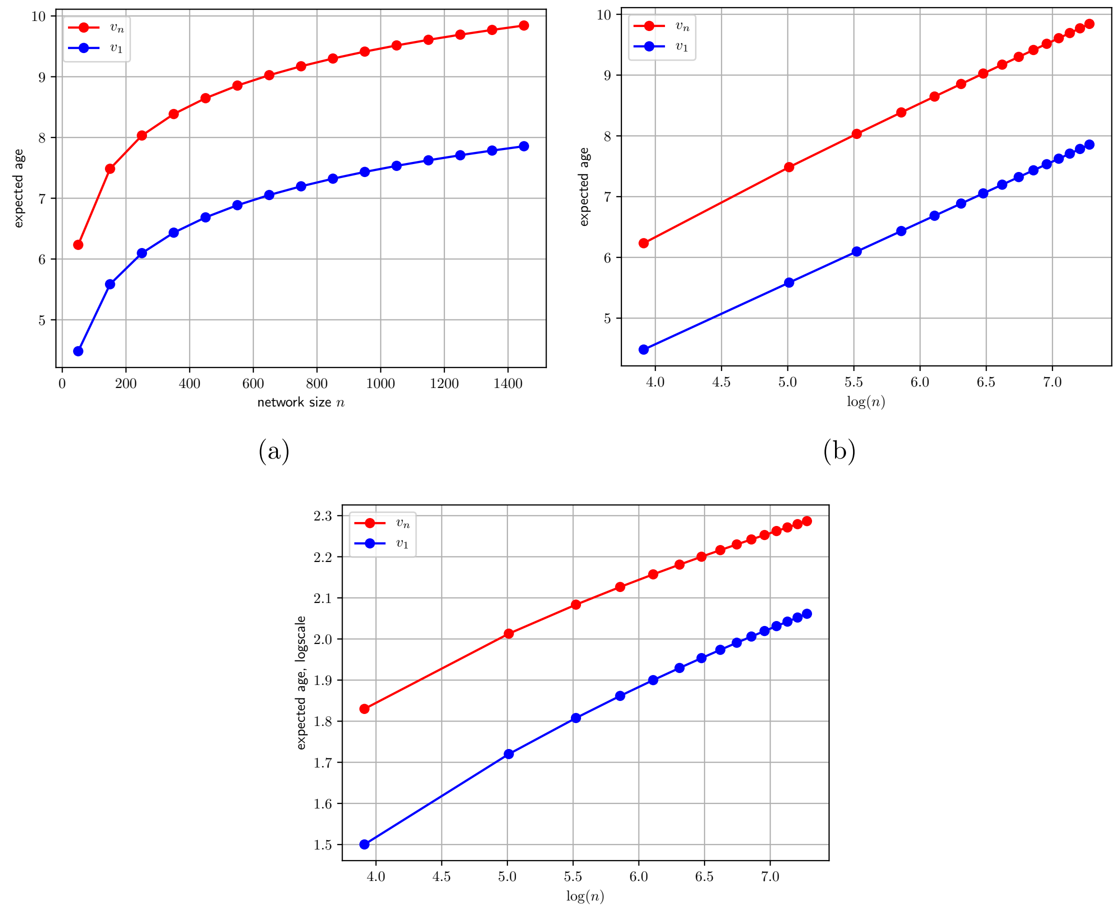
(a) (b)

(c)

Figure 4.6: Node capture attack on fully connected network with $p = 0.5$ and $q = 1$. (a) Linear-linear, (b) linear-log (c) log-log plot of expected age at various nodes as afunction versus network size n, demonstrating that expected age increases linearly with network size when $q = 1$.

not inject any packets at all and thereby limit its contribution completely. A mix of these two actions is not optimal, as seen in *URN Case 3*, where age scales as $O(\sqrt{n})$ at all nodes.[2]

[2] Another interesting network to study is the bidirectional ring network, where each node transmits packets to two other nodes with rate $\frac{\lambda}{2}$. In the absence of any adversarial actions, it is known that both unidirectional and bidirectional ring networks have exactly the same age for all n [76]. Though it would be interesting to see in the presence of adversarial actions, if the infected node still fails to propagate its effect on age scaling of a large portion of the network, the manipulated timestamps and lack of symmetry in the network due to the presence of an adversary make it particularly difficult to analyze the bidirectional ring, and we leave it as a future direction for now.

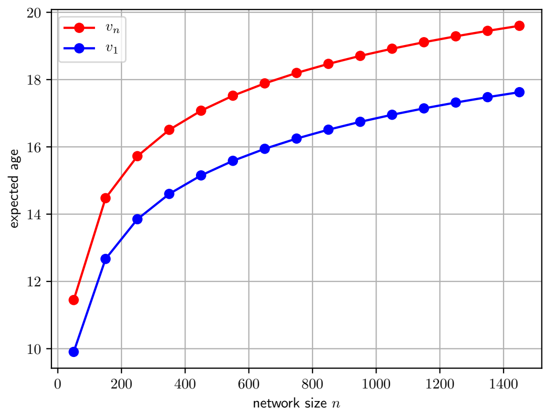


(a) (b) (c)

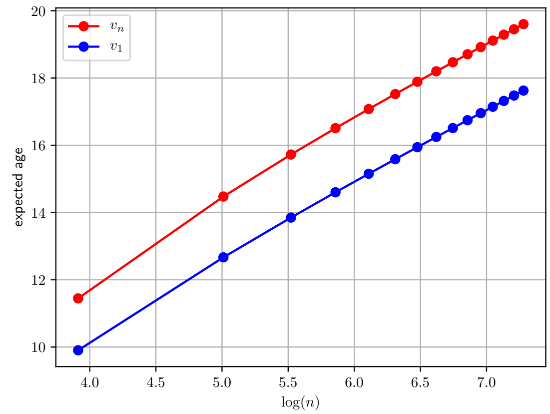
Figure 4.7: Node capture attack on fully connected network with $p = 0$ and $q = 0.5$. (a) Linear-linear, (b) linear-log (c) log-log plot of expected age as a function versus network size n, demonstrating that expected age scales as $O(\log n)$.

4.6 Numerical Results

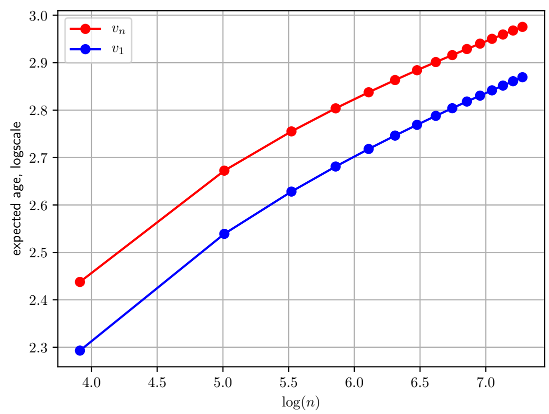
We first provide numerical results for all three cases of Section 4.3 corresponding to fully connected networks (FCNs), choosing $\lambda = 1$ for the purposes of this section. Fig. 4.6 shows the expected age at a regular node v_1 and the infected node v_n in *FCN Case 1*, employing choice of probabilities $p = 0.5$ and $q = 1$. The age at the infected node v_n in red color grows as $\frac{n}{\lambda}$ as expected from (4.9) by plugging $q = 1$.



(a)



(b)



(c)

Figure 4.8: Node capture attack on fully connected network with $p = 0.5$ and $q = 0.5$. (a) Linear-linear, (b) linear-log (c) log-log plot of expected age as a function versus network size n, demonstrating that expected age scales as $O(\log n)$.

The age at a regular node v_1 in blue color grows linearly as $O(n)$ and lies broadly between pv_n and $\frac{pv_n}{2}$, as suggested by (4.14) and (4.18).

Fig. 4.7 shows the expected ages v_1 and v_n for *FCN Case 2*, using $p = 0$ and $q = 0.5$. The red and blue curves are roughly separated by a distance of about $\frac{1}{\lambda(1-q)} \approx 2$, given our choice of $\lambda = 1$ and $q = 0.5$, as suggested in (4.21), and both scale as logarithmically, with $v_1 \leq e \log n$ as suggested in (4.20).

Finally, we plot v_1 and v_n for *FCN Case 3* in Fig. 4.8, choosing $p = 0.5$ and $q =$

0.5. Again the red and blue curves are roughly separated by a distance of $\frac{1}{\lambda(1-q)} \approx 2$ as suggested in (4.21). Both v_1 and v_n scale as $O(\log n)$, with $v_1 \approx 2(\log n + 1)$ as suggested in (4.23).

Fig. 4.9 shows the expected age at different nodes for Section 4.4 when the adversary is positioned between the source and a node in a fully connected network, i.e., MITM attack. The green line shows the lower bound $\frac{v_A}{4}$ of (4.41), where the age at a regular node v_1 lies above this lower bound. Adversary age v_A grows as $\frac{n}{\lambda} = O(n)$ by virtue of being an isolated node. Finally, though (4.36) yields a loose lower bound of $\frac{v_A}{2}$, the graph shows that the age at the node that is in contact with the adversary, v_n, closely follows adversary age v_A.

Next, we present the set of numerical results for all three cases of Section 4.5 corresponding to unidirectional ring networks (URNs). Fig. 4.10 demonstrates how expected age v_m scales with network size n for various index functions discussed in *URN Case 1.* Notice that the data points are concentrated more towards the left side of the graph. This is because we choose node index $m = \lfloor n^{0.3} \rfloor$ casusing m to change value only when $\lfloor n^{0.3} \rfloor$ changes its integer value, motivating us to choose $n = \lceil \ell^{\frac{1}{0.3}} \rceil$, $\ell \in \mathbb{N}$ to have m increase consistently with n.

Clearly, the v_n in red color grows as $\frac{n}{\lambda}$ as stated in *URN Case 1.* Next, we see that when $m = n^{0.3} = O(n^{\frac{1}{2}})$, the expected age v_m in yellow color scales almost as n since $e^{-\frac{m^2}{Cn}}$ converges to 0 for larger values of n, which is in accordance with *URN Case 1(a). URN Case 1(b)* is represented by the green curve, where $m = n^{0.8} = \Omega(n^{\frac{1}{2}+\epsilon})$ for $\forall \epsilon \in (0, 0.3]$, and the corresponding v_m scales as $1.25\sqrt{n}$ similar to a unidirectional ring network without adversary of [76]. The blue curve

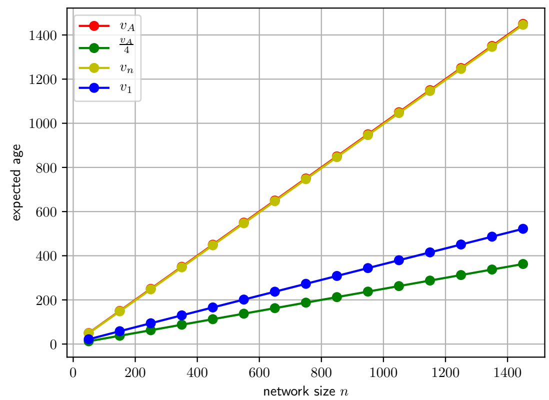


Figure 4.9: MITM attack on fully connected network of n nodes.

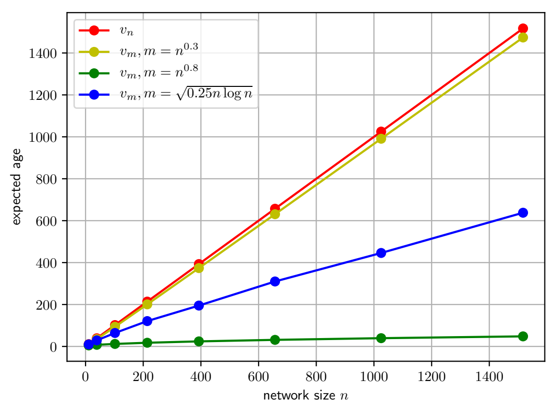


Figure 4.10: Node capture attack on unidirectional ring with $p = 0.5$ and $q = 1$.

shows *URN Case 1* as a transition case between *URN Case 1(a)* and *URN Case 1(b)*. For choice of parameters $p = 0.5$, $q = 0.5$ and $\alpha = 0.25$, v_m is dictated primarily by the second term of (4.64) to be $n^{0.875}$ with $\beta \approx 2$ as supported by (4.65).

Fig. 4.11 demonstrates how expected age v_m scales for various index functions discussed in *URN Case 2*. The blue curve represents *URN Case 2(a)* for choice of $\alpha = 0.3$, and the term $n^{0.7}$ primarily dictates expected age v_m for $m = n^{0.3}$. Likewise the yellow curve represents *URN Case 2(a)* for choice of $\alpha = 0.4$, causing v_m to dominantly follow $n^{0.6}$, which illustrates how the age scaling behaviour gradually transitions across node indices in *URN Case 2*. Lastly, the green curve exemplifies

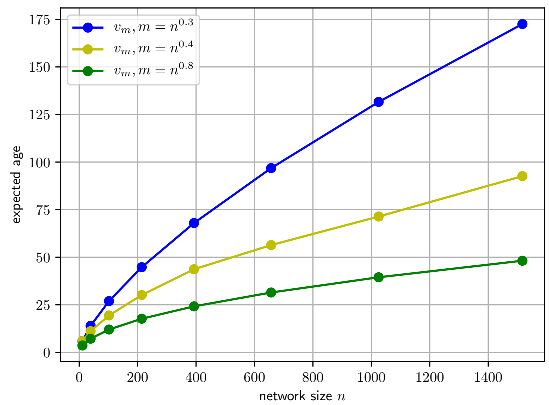


Figure 4.11: Node capture attack on unidirectional ring with $p = 0$ and $q = 0.5$.

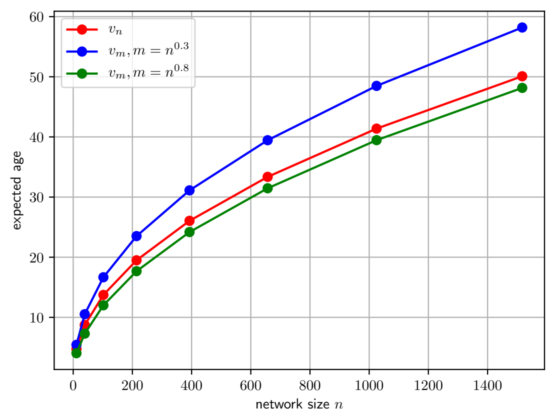


Figure 4.12: Node capture attack on unidirectional ring with $p = 0.5$ and $q = 0.5$.

how for $n^{0.8} = \Omega(\sqrt{n})$, v_m is practically unaffected by the presence of the adversary in *URN Case 2(b)* and v_m scales as approximately $1.25\sqrt{n}$ similar to a unidirectional ring network without adversary in [76].

Lastly, for *URN Case 3*, we observe in Fig. 4.12 that the expected age scales as $O(\sqrt{n})$ at all nodes of the network. In accordance with (4.53), the red curve v_n closely follows the green curve v_m for $m = n^{0.8}$, which almost scales as $1.25\sqrt{n}$ since all nodes of higher indices are almost unaffected by the adversary. Since packets are sent in a fixed direction around the ring, the blue curve representing v_m for $m = n^{0.3}$ has higher expected age than v_n at the infected node, owing to the presence of extra first term in (4.51).

4.7 Conclusion

In this chapter, we first studied the effects of timestomping attacks on the age of gossip in a large fully connected network. We showed that one infected node in such a network can increase the age at all other nodes from $O(\log n)$ to $O(n)$ through timestamp manipulation. Further, we showed that the optimal behavior for the adversary is to reset the timestamps of all outgoing packets to current time thereby disguising them as current packets and of all incoming packets to an outdated time to prevent their acceptance at the infected node. We showed that if the adversary allows the infected node to accept even a very small fraction of the incoming packets from the network, then a large network can manage to curb the spread of stale files coming from the infected node and pull the network age back to $O(\log n)$. Additionally, we showed that if an infected node contacts only a single node instead of all nodes of the network, the system age can still be degraded to $O(n)$. We then analyzed the unidirectional ring network, where we showed that the adversarial effect on age scaling of a node is limited by its distance from the adversary, and the age scaling for a large fraction of the network continues to be $O(\sqrt{n})$, unchanged from the case with no adversary. We showed that the optimal behavior for the adversary is again to reset the timestamps of all outgoing packets to current time and of all incoming packets to an outdated time. Finally, we would like to point out that between these two network extremes representing very high and very low connectivity, there is a whole gamut of networks potentially exhibiting diverse age scaling behavior, which we plan to work on as future research direction.

CHAPTER 5

Misinformation Spread in Gossip Networks: The Influence of Transmission Mutations

5.1 Introduction

In this chapter, we attempt to characterize spread of misinformation in an age-based gossip network [5], where information at the source node gets updated according to a Poisson process with rate λ_e; Fig. 5.1. We associate a version number with each information generated at the source, such that the version number of the current information at the source gets incremented by one post each source update. The source forwards its latest version to the network of n nodes, $\mathcal{N} = \{1, \dots, n\}$, according to a Poisson process with rate λ_s, choosing the destination node uniformly at random from $\mathcal{N}$ each time. The network nodes wish to have access to the latest possible version of information, and therefore, each node only stores the latest version of information it has received so far and gets rid of all older information packets. Further, the nodes gossip with their neighboring nodes to further improve the timeliness of information in the network, whereby each node sends updates to its neighbors according to a Poisson process with rate λ.

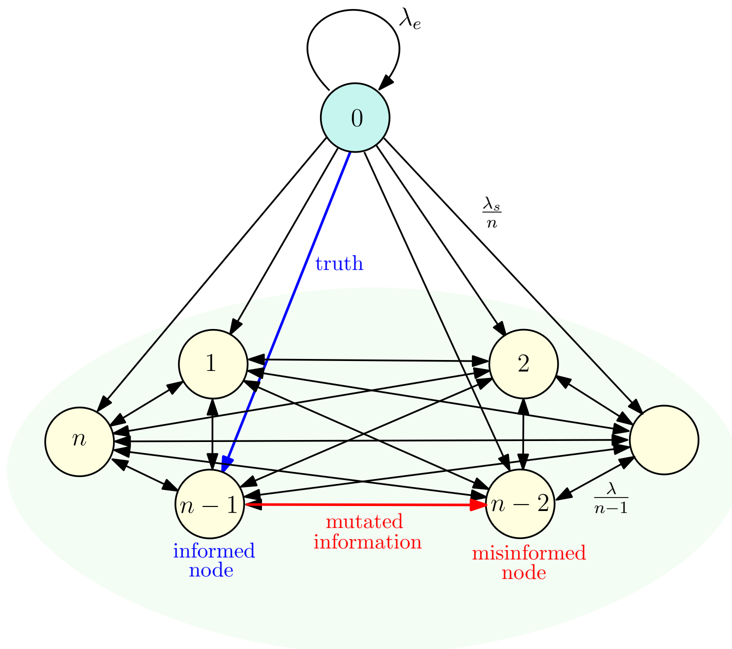


Figure 5.1: Fully-connected gossip network of n nodes with information probabilistically mutating into misinformation during inter-node gossip.

The source always communicates accurate information to network nodes, however, there is a possibility of information getting mutated during inter-node transmissions in the network. This could either be because the sender node is not always honest and sometimes deliberately tampers with the information before forwarding it, or the information packet could get corrupted during transmission process. An example for this problem setting could be software distribution to end-users, where the software vendor always provides reliable versions or iterations of software packages to users, however if the users instead obtain a software version from their neighboring users, they might occasionally receive an incorrectly functioning or even harmful version of the software. Other examples could be real-time news dissemination in a region with truth getting mutated in inter-personal gossip, or smart sensor networks where occasionally noisy measurements are transmitted.

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

Whenever node i receives a packet, it compares the version number of the received packet with the version number of the packet in its possession, and if the version numbers are different, then node i discards the staler packet and keeps the fresher packet. However, node i does not prima facie know whether the piece of information it has received is the truth or not, or if it is different from the sender's information or not. Therefore, if the information already present at node i and the received information have the same version numbers, node i just keeps the information it trusts the most, say based on performance of software in the software distribution example, or measurement noise detected in smart sensor network example. We consider two models of information exchange in this work, where in the first model, if one of the packets contains accurate information (i.e., the truth), then node i would keep that packet, i.e., truth prevails over misinformation in this model. In the second model, we assume the opposite scenario, wherein misinformation prevails over truth, i.e., between the two packets corresponding to the same version, the one with false information prevails. It was shown in [91] that false information diffuses significantly farther, faster, and deeper in social networks than the truth, since false news is more novel than true news, and people are more likely to share novel information, which serves as a motivation for our second model of information exchange. For both models, we wish to know what fraction of nodes are misinformed.

Traditionally, the analysis of gossip algorithms is done using the epidemic model of data spread [92] where the quantity of interest is dissemination time of a fixed message or set of messages to all nodes of the network [71–75,93,94]. Likewise, misinformation literature [95–100] also treats misinformation as a virus, such that users can become infected upon exposure, consequently turning them into spreaders. This allows for interpretation of misinformation spread as epidemic models like SIS and SIR, and the quantity of interest here is how long misinformation survives in a network. Though both gossip and misinformation spread are essentially information diffusion processes, the goals in the two cases are different, since rapid and complete spread of an update is preferred in the former.

Since this chapter considers the subject of misinformation in timely gossip networks, the work most closely related to this work is [101], which studies the conditions under which the majority rule based estimation of time-varying binary valued source can cause incorrect source estimation or misinformation at network nodes. Further, there is a connection to Chapter 4, where an adversary alters timestamps of packets, rebranding old packets as fresh and fresh packets as old, with the goal to replace circulation of fresh packets with outdated packets in the network, thereby corrupting the gossip process. This chapter is also related to Chapter 6, which considers a network model with two sources, such that the information obtained from one source is considered more reliable than the other source, where we study the prevalence of unreliable information in the gossip network.

Our goal is somewhat similar to Chapter 6, as we are interested in finding what fraction of user nodes on average in the network have the truth and what is the version age of information at user nodes. However, in Chapter 6, once a packet is created at one of the sources, the packet information does not change during the packet diffusion process, whereas in the current chapter, information is susceptible to mutating into misinformation during inter-node transmissions. Further, in Chapter 6, network nodes know if a particular packet has originated at the reliable source or the unreliable source, which allows nodes to consider a freshness-reliability trade-off. However, in the current work, nodes do not know whether a received information is the truth or not. In this respect, we model the information mutation problem as a stochastic hybrid system (SHS), and study the dependency of our results on various network parameters. We observe that when truth prevails over misinformation, very high or very low gossiping rates help curb misinformation, and misinformation spread is higher with moderate gossiping rates. On the other hand, when misinformation prevails, misinformation rises with increased gossiping.

5.2 System Model and SHS Characterization When Truth Prevails Over Misinformation

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

- If $X_i(t^-) > X_j(t^-)$, node j rejects the incoming packet and $T_j(t), X_j(t)$ pair remains unchanged, since node j already has the fresher packet.
- If $X_i(t^-) < X_j(t^-)$, the incoming packet corresponds to a fresher version, hence node j replaces its packet with the incoming packet. Note that node j does not know if node i was honest or not, or the received packet is the truth

or not. Hence $X_j(t) = X_i(t^-)$, and

$$T_j(t) = \begin{cases} T_i(t^-), & \text{node } i \text{ is honest,} \\ 0, & \text{node } i \text{ is not honest.} \end{cases} \tag{5.1}$$

- If $X_i(t^-) = X_j(t^-)$, both packets are equally fresh and $X_j(t)$ remains unchanged. Further, in this first model, truth prevails over misinformation when two information packets of the same version are encountered and either of them carries the truth. The accuracy of the incoming packet is $T_i(t^-)$ if node i is honest in its communication, and 0 if node i is dishonest, while the version age of incoming packet is $X_i(t^-)$ in both cases. Hence,

$$T_j(t) = \begin{cases} \max\{T_i(t^-), T_j(t^-)\}, & \text{node } i \text{ is honest,} \\ T_j(t^-), & \text{node } i \text{ is not honest.} \end{cases} \tag{5.2}$$

  Note that when node i communicates its information honestly to other nodes, it might lead to spread of misinformation if node i was misinformed before communication ($T_i(t^-) = 0$), i.e., honesty in communication does not imply delivery of the truth.

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

$$F(t) = \frac{T_1(t) + T_2(t) + \ldots + T_n(t)}{n}. \tag{5.3}$$

To evaluate the expected fraction of users with truth, $F = \lim_{t\to\infty} \mathbb{E}[F(t)]$, we model

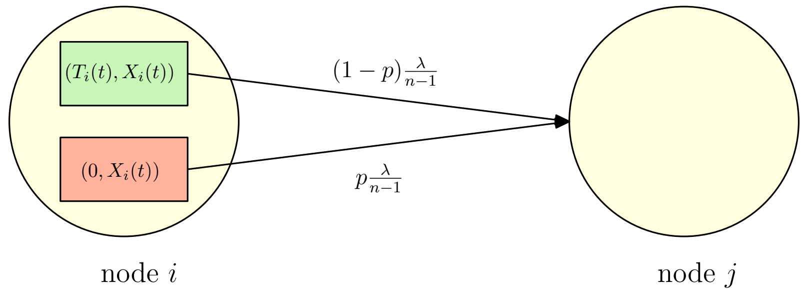


Figure 5.2: The green packet depict the packet actually stored at the node i. The orange packet depicts a mutated copy of the green packet, hypothetically available at node i at all times.

the problem as an SHS. Note that node i essentially sends its packet to node j with honesty according to a thinned Poisson process with rate $(1-p)\frac{\lambda}{n-1}$ or sends a mutated copy of its packet according to a thinned Poisson process with rate $p\frac{\lambda}{n-1}$. For ease of exposition, we assume that instead of creating a mutated copy of its packet at the time of transmission, node i stores a mutated copy of its packet at all times, i.e., each node is assumed to store two packets (see Fig. 5.2). In Fig. 5.2, the green packet represents the packet actually present at node i, having accuracy of $T_i(t)$ and version age of $X_i(t)$, and the orange packet represents a mutated copy of the green packet, hypothetically stored at the node, having accuracy of 0 (since it is a misleading packet) and version age $X_i(t)$. We are interested in finding the accuracy and version age of the green packets at all nodes. Essentially, now node i sends its green packet to node j with rate $(1-p)\frac{\lambda}{n-1}$ and its orange packet with rate $p\frac{\lambda}{n-1}$. Upon receiving a packet from node i, if node j chooses to keep the received packet as per the state reset protocol (thereby forming a new green packet at node j), then a new orange packet is also immediately created by mutating the new green packet, such that both the new packets have the same version age/number.

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

$$\mathcal{L} = \{(0,0,1)\} \cup \{(0,i,1) : i \in \mathcal{N}\} \cup \{(i,j,h) : i,j \in \mathcal{N}, h \in \{0,1\}\}, \tag{5.4}$$

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

$$\lambda_{ijh} = \begin{cases} \lambda_e, & i = 0, j = 0, h = 1 \\ \frac{\lambda_s}{n}, & i = 0, j \in \mathcal{N}, h = 1 \\ (1-p)\frac{\lambda}{n-1}, & i, j \in \mathcal{N}, h = 1 \\ p\frac{\lambda}{n-1}, & i, j \in \mathcal{N}, h = 0, \end{cases} \tag{5.5}$$

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

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

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

$$T'_\ell = \begin{cases} 1, & i = 0, j \in \mathcal{N}, h = 1, \ell = j \\ T_{\{i,\ell\},\varnothing}, & i, j \in \mathcal{N}, h = 1, \ell = j \\ T_{\{\ell\},\{i\}}, & i, j \in \mathcal{N}, h = 0, \ell = j \\ T_\ell, & \text{otherwise,} \end{cases} \tag{5.6}$$

and

$$X'_\ell = \begin{cases} X_\ell + 1, & i = 0, j = 0, h = 1, \ell = j \\ 0, & i = 0, j \in \mathcal{N}, h = 1, \ell = j \\ \min\{X_i, X_\ell\}, & i, j \in \mathcal{N}, h = 1, \ell = j \\ \min\{X_i, X_\ell\}, & i, j \in \mathcal{N}, h = 0, \ell = j \\ X_\ell, & \text{otherwise.} \end{cases} \tag{5.7}$$

In the next section, we pick a series of test functions $\psi : \mathbb{R}^{2n} \times [0, \infty) \to \mathbb{R}$ that are time-invariant (consequently we will drop the third input t and write $\psi(\boldsymbol{T}, \boldsymbol{X}, t)$ as $\psi(\boldsymbol{T}, \boldsymbol{X})$), satisfying $\dot{\psi}(\boldsymbol{T}(t), \boldsymbol{X}(t)) = 0$, such that their long-term expected value $\mathbb{E}[\psi] = \lim_{t\to\infty} \mathbb{E}[\psi(\boldsymbol{T}(t), \boldsymbol{X}(t), t)]$ will be useful for analysis later. Defining $\mathbb{E}[\psi(\phi_{i,j,h})] = \lim_{t\to\infty} \mathbb{E}[\psi(\phi_{i,j,h}(\boldsymbol{T}(t), \boldsymbol{X}(t), t))]$, [90, Thm. 1] yields

$$0 = \sum_{(i,j,h)\in\mathcal{L}} \left(\mathbb{E}[\psi(\phi_{i,j,h})] - \mathbb{E}[\psi]\right) \lambda_{ijh}, \tag{5.8}$$

where the left side is set to zero due to $\frac{d\mathbb{E}[\psi(\boldsymbol{T}(t), \boldsymbol{X}(t), t)]}{dt} = 0$ at large t as the expectation stabilizes. For more details, the reader is encouraged to look at references [5, 90].

5.3 Misinformation and Version Age Derivations When Truth Prevails Over Misinformation

Since the accuracy status evolution process and version age evolution process are statistically identical for all user nodes, in the following analysis, the sets A_k and B_m correspond to arbitrary sets of k and m user nodes with $A_k \cap B_m = \varnothing$. Our first test function is $\psi(\boldsymbol{T}, \boldsymbol{X}) = T_{A_k,B_m}$, which is modified upon transition (i,j,h) to $\psi(\phi_{i,j,h}(\boldsymbol{T}, \boldsymbol{X}, t)) = T'_{A_k,B_m}$ and can be characterized using (5.6), (5.7) and (5.8) as follows,

$$T'_{A_k,B_m} = \begin{cases} 1, & i=0, j \in A_k, h=1 \\ \mathbb{1}_{\{T_{A_k,\varnothing}=1, X_{A_k}=0\}}, & i=0, j \in B_m, h=1 \\ T_{A_k,B_{m+1}} & i \in \mathcal{N}\backslash(A_k \cup B_m), j \in A_k \cup B_m, h=0 \\ T_{A_k,B_{m+1}} & i \in \mathcal{N}\backslash(A_k \cup B_m), j \in B_m, h=1 \\ T_{A_{k+1},B_m} & i \in \mathcal{N}\backslash(A_k \cup B_m), j \in A_k, h=1 \\ T_{A_{k+1},B_{m-1}} & i \in B_m, j \in A_k, h=1 \\ T_{A_k,B_m} & \text{otherwise.} \end{cases} \tag{5.9}$$

In (5.9), the effect of transition $i \in \mathcal{N}\backslash(A_k \cup B_m), j \in B_m, h=1$ is interesting. Though node i sends its green packet to node j (as $h=1$ implies honest communication), T_{A_k,B_m} is only concerned with the orange packet at node j, which always maintains an accuracy status of 0. Hence, for the purposes of T_{A_k,B_m}, it appears as if node i sent its orange packet to potentially replace the orange packet at node j. Since $|\mathcal{N}\backslash(A_k \cup B_m)| = n-k-m$ and $|B_m| = m$, there are $(n-k-m)m$ such unique transitions, each with rate $\frac{\lambda}{n-1}$. Let us define

$$t_{k,m} = \lim_{t\to\infty} \mathbb{E}[T_{A_k,B_m}(t)], \tag{5.10}$$

$$c_k = \lim_{t\to\infty} \mathbb{E}[\mathbb{1}_{\{T_{A_k,\varnothing}(t)=1, X_{A_k}(t)=0\}}]. \tag{5.11}$$

Then, using (5.8) with (5.9), we obtain

$$\begin{aligned} 0 =&(1-t_{k,m})k\frac{\lambda_s}{n} + (c_k - t_{k,m})m\frac{\lambda_s}{n} \\ &+ (t_{k,m+1} - t_{k,m})(n-k-m)(pk+m)\frac{\lambda}{n-1} \\ &+ (t_{k+1,m} - t_{k,m})(1-p)(n-k-m)k\frac{\lambda}{n-1} \\ &+ (t_{k+1,m-1} - t_{k,m})(1-p)km\frac{\lambda}{n-1}, \end{aligned} \tag{5.12}$$

which upon rearrangement gives,

$$\begin{aligned} t_{k,m} =&\frac{1}{(k+m)\frac{\lambda_s}{n} + (n-k-m)(k+m)\frac{\lambda}{n-1} + (1-p)km\frac{\lambda}{n-1}} \\ &\times \Bigg(k\frac{\lambda_s}{n} + c_k m\frac{\lambda_s}{n} + t_{k+1,m}(1-p)(n-k-m)k\frac{\lambda}{n-1} \\ &\quad + t_{k,m+1}(pk+m)(n-k-m)\frac{\lambda}{n-1} \\ &\quad + t_{k+1,m-1}(1-p)km\frac{\lambda}{n-1}\Bigg) \end{aligned} \tag{5.13}$$

Note that since A_k and B_m are disjoint sets in (5.9), only $t_{k,m}$ satisfying $k \geq 0, m \geq 0, k+m \leq n$ are encountered. For example, (5.13) might at first give an impression that $t_{n,0}$ is dependent on $t_{n+1,0}$. However $\mathcal{N}\backslash(A_k \cup B_m) = \varnothing$ for $(k,m) = (n,0)$, hence there are no transitions of type $i \in \mathcal{N}\backslash(A_k \cup B_m), j \in B_m, h = 1$ in (5.9), making the coefficient of $t_{n+1,0}$ zero in (5.13). Likewise the coefficients of $t_{n,1}$ and $t_{n+1,-1}$ are also zero, giving $t_{n,0} = 1$ from (5.13).

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

$$\mathbb{1}'_{\{T_{A_k,\varnothing}=1, X_{A_k}=0\}} = \begin{cases} 0, & i=0, j=0, h=1 \\ 1, & i=0, j \in A_k, h=1 \\ \mathbb{1}_{\{T_{A_{k+1},\varnothing}=1, X_{A_{k+1}}=0\}}, & i \in \mathcal{N}\backslash A_k, j \in A_k, h=1 \\ \mathbb{1}_{\{T_{A_k,\varnothing}=1, X_{A_k}=0\}}, & \text{otherwise.} \end{cases} \tag{5.14}$$

Here, if there is a node $j \in A_k$ storing a green packet with accuracy 1 and version age 0, i.e., the freshest truth, then $\mathbb{1}_{\{T_{A_k,\varnothing}=1, X_{A_k}=0\}}$ will remain 1 irrespective of any type of transition, since all incoming packets to such node j will be discarded in accordance with the state reset protocol of Section 5.2. If $\mathbb{1}_{\{T_{A_k,\varnothing}=1, X_{A_k}=0\}} = 0$, then the only way this test function can change value is if the latest truth is honestly communicated to a node in A_k (i.e., a green packet containing the latest truth is communicated), captured by the second and third cases of the reset map in (5.14).

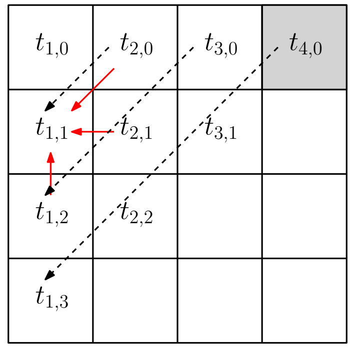


Figure 5.3: Solving $t_{k,m}$ with dynamic programming for $n = 4$.

The corresponding linear equation from (5.14) using (5.8) is,

$$0 =(0 - c_k)\lambda_e + (1 - c_k)k\frac{\lambda_s}{n} + (c_{k+1} - c_k)(1 - p)k(n - k)\frac{\lambda}{n - 1}, \tag{5.15}$$

which upon rearrangement gives,

$$c_k = \frac{k\frac{\lambda_s}{n} + c_{k+1}(1 - p)k(n - k)\frac{\lambda}{n-1}}{\lambda_e + k\frac{\lambda_s}{n} + (1 - p)k(n - k)\frac{\lambda}{n-1}}. \tag{5.16}$$

Since the laws of accuracy and age processes at all user nodes are the same, from (5.3) we get that the long-term expected fraction of nodes with truth is $F = \lim_{t\to\infty} \mathbb{E}[T_1(t)] = t_{1,0}$.

Therefore, for computation of F, we first solve for c_k using (5.16), starting from $k = n$, with $c_n = \frac{\lambda_s}{\lambda_e+\lambda_s}$ and successively substituting for $k = n - 1, \dots, 1$ in an iterative fashion. Once we have computed all c_k, we compute $t_{k,m}$ using (5.13) in a dynamic programming fashion, shown in Fig. 5.3. Since in (5.13) $t_{k,m}$ depends on $t_{k+1,m}$, $t_{k,m+1}$ and $t_{k+1,m-1}$, starting with $t_{n,0} = 1$, we solve $t_{r-s,s}$ following the order $r = n, \dots, 1$ and $s = 0, 1, \dots, r - 1$, to reach $t_{1,0} = F$ at the end, as further guided by the arrows along the diagonals in Fig. (5.3).

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

$$X'_{A_k} = \begin{cases} X_{A_k}+1, & i=0, j=0, h=1 \\ 0, & i=0, j\in A_k, h=1 \\ X_{A_{k+1}}, & i\in\mathcal{N}\backslash A_k, j\in A_k, h=1 \\ X_{A_{k+1}}, & i\in\mathcal{N}\backslash A_k, j\in A_k, h=0 \\ X_{A_k}, & \text{otherwise.} \end{cases} \tag{5.17}$$

Note how information mutation does not impact the version age of a packet, since the transition maps in (5.7) and (5.17) do not depend on h. The linear equation for (5.17) using (5.8) is,

$$\begin{aligned} 0 =&(v_k+1-v_k)\lambda_e + (0-v_k)k\frac{\lambda_s}{n} + (v_{k+1}-v_k)(1-p)k(n-k)\frac{\lambda}{n-1} \\ &+ (v_{k+1}-v_k)pk(n-k)\frac{\lambda}{n-1}, \end{aligned} \tag{5.18}$$

which upon rearrangement gives,

$$v_k = \frac{\lambda_e + v_{k+1}k(n-k)\frac{\lambda}{n-1}}{k\frac{\lambda_s}{n} + k(n-k)\frac{\lambda}{n-1}}. \tag{5.19}$$

Defining $x_i = \lim_{t\to\infty}\mathbb{E}[X_i(t)]$, with $x_1 = \ldots = x_n$ due to network symmetry, similar to c_k, we can backward iterate on (5.19) to compute the $x_1 = v_1$.

Lemma 5.1 *v_1 scales as $O(\log n)$.*

The proof of Lemma 5.1 follows similarly to [102, Lem. 11].

5.4 Misinformation and Version Age Derivations When Misinformation Prevails Over Truth

We now modify the file exchange protocol at nodes and consider the second scenario where misinformation prevails over truth. The information mutation process stays the same, i.e, when node i sends a packet to node j on link (i,j) at time t, node i either honestly forwards the information in its possession with probability $1-p$ or alters the contents of the packet to send misleading information with probability p, spreading misinformation in the latter case. However, when node j receives the packet from node i at time t, $T_j(t)$ and $X_j(t)$ are now reset according to following state reset protocol:

- If $X_i(t^-) > X_j(t^-)$, node j rejects the incoming packet and $T_j(t), X_j(t)$ pair remains unchanged, since node j already has the fresher packet.
- If $X_i(t^-) < X_j(t^-)$, the incoming packet corresponds to a fresher version, hence node j replaces its packet with the incoming packet. Note that node j does not know if node i was honest or not, or the received packet is the truth

or not. Hence $X_j(t) = X_i(t^-)$, and

$$T_j(t) = \begin{cases} T_i(t^-), & \text{node } i \text{ is honest,} \\ 0, & \text{node } i \text{ is not honest.} \end{cases}$$

- If $X_i(t^-) = X_j(t^-)$, both packets are equally fresh and $X_j(t)$ remains unchanged. Further, in this model, misinformation prevails over truth when two information packets of the same version are encountered and either of them carries the misinformation. The accuracy of the incoming packet is $T_i(t^-)$ if node i is honest in its communication, and 0 if node i is dishonest, while the version age of incoming packet is $X_i(t^-)$ in both cases. Hence,

$$T_j(t) = \begin{cases} \min\{T_i(t^-), T_j(t^-)\}, & \text{node } i \text{ is honest,} \\ 0, & \text{node } i \text{ is not honest.} \end{cases}$$

Next, given a set of nodes $A \subseteq \mathcal{N}$ and a continuous state $(\boldsymbol{T}, \boldsymbol{X})$, we define $R_{A,B}$, analogous to $T_{A,B}$ earlier, as follows:

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

Essentially, if we were to collect all the green packets from nodes in set A and all the orange packets from nodes in set B, and pick the freshest packets from this collection of green and orange packets, such that one of these freshest packets happen to have accuracy of 0, then $R_{A,B} = 0$. Based on transition (i, j, h) at time

t, the reset map to $\phi_{i,j,h}(\boldsymbol{T},\boldsymbol{X},t) = [T_1',\ldots,T_n',X_1',\ldots,X_n'] \in \mathbb{R}^{2n}$ can be described as,

$$T_\ell' = \begin{cases} 1 - \mathbb{1}_{\{R_{A_1,\varnothing}=0, X_{A_1}=0\}}, & i=0, j\in\mathcal{N}, h=1, \ell=j \\ R_{\{i,\ell\},\varnothing}, & i,j\in\mathcal{N}, h=1, \ell=j \\ R_{\{\ell\},\{i\}}, & i,j\in\mathcal{N}, h=0, \ell=j \\ T_\ell, & \text{otherwise,} \end{cases} \tag{5.20}$$

and

$$X_\ell' = \begin{cases} X_\ell + 1, & i=0, j=0, h=1, \ell=j \\ 0, & i=0, j\in\mathcal{N}, h=1, \ell=j \\ \min\{X_i, X_\ell\}, & i,j\in\mathcal{N}, h=1, \ell=j \\ \min\{X_i, X_\ell\}, & i,j\in\mathcal{N}, h=0, \ell=j \\ X_\ell, & \text{otherwise.} \end{cases} \tag{5.21}$$

We choose our first test function as $\psi(\boldsymbol{T},\boldsymbol{X}) = R_{A_k,B_m}$, which is modified upon transition (i,j,h) to $\psi(\phi_{i,j,h}(\boldsymbol{T},\boldsymbol{X},t)) = R'_{A_k,B_m}$ and can be characterized using (5.20), (5.21) and (5.8) as follows,

$$R'_{A_k,B_m} = \begin{cases} 1-\mathbb{1}_{\{R_{A_k,B_m}=0, X_{A_k\cup B_m}=0\}}, & i=0, j\in A_k, h=1 \\ 0, & i=0, j\in B_m, h=1 \\ R_{A_k,B_{m+1}} & i\in\mathcal{N}\backslash(A_k\cup B_m), j\in A_k\cup B_m, h=0 \\ R_{A_k,B_{m+1}} & i\in\mathcal{N}\backslash(A_k\cup B_m), j\in B_m, h=1 \\ R_{A_{k+1},B_m} & i\in\mathcal{N}\backslash(A_k\cup B_m), j\in A_k, h=1 \\ R_{A_{k-1},B_{m+1}} & i\in A_k, j\in A_k\cup B_m, h=0 \\ R_{A_{k-1},B_{m+1}} & i\in A_k, j\in B_m, h=1 \\ R_{A_k,B_m} & \text{otherwise.} \end{cases} \tag{5.22}$$

Let us define $r_{k,m} = \lim_{t\to\infty}\mathbb{E}[R_{A_k,B_m}(t)]$ and $d_{k,m} = \lim_{t\to\infty}\mathbb{E}[\mathbb{1}_{\{R_{A_k,B_m}(t)=0, X_{A_k\cup B_m}(t)=0\}}]$. Note that $F = \lim_{t\to\infty}\mathbb{E}[T_1(t)] = r_{1,0}$. Using (5.8) with (5.22), we obtain,

$$\begin{aligned} 0 =& (1-d_{k,m}-r_{k,m})k\frac{\lambda_s}{n} + (0-r_{k,m})m\frac{\lambda_s}{n} \\ &+ (r_{k,m+1}-r_{k,m})(n-k-m)(k+m)p\frac{\lambda}{n-1} \\ &+ (r_{k,m+1}-r_{k,m})(n-k-m)m(1-p)\frac{\lambda}{n-1} \\ &+ (r_{k+1,m}-r_{k,m})(n-k-m)k(1-p)\frac{\lambda}{n-1} \\ &+ (r_{k-1,m+1}-r_{k,m})k(k+m-1)p\frac{\lambda}{n-1} \\ &+ (r_{k-1,m+1}-r_{k,m})km(1-p)\frac{\lambda}{n-1}, \end{aligned} \tag{5.23}$$

or

$$
\begin{aligned}
0 =&(1 - d_{k,m} - r_{k,m})k\frac{\lambda_s}{n} + (0 - r_{k,m})m\frac{\lambda_s}{n} \\
&+ (r_{k,m+1} - r_{k,m})(n-k-m)(pk+m)\frac{\lambda}{n-1} \\
&+ (r_{k+1,m} - r_{k,m})(n-k-m)k(1-p)\frac{\lambda}{n-1} \\
&+ (r_{k-1,m+1} - r_{k,m})k((k-1)p+m)\frac{\lambda}{n-1},
\end{aligned} \tag{5.24}
$$

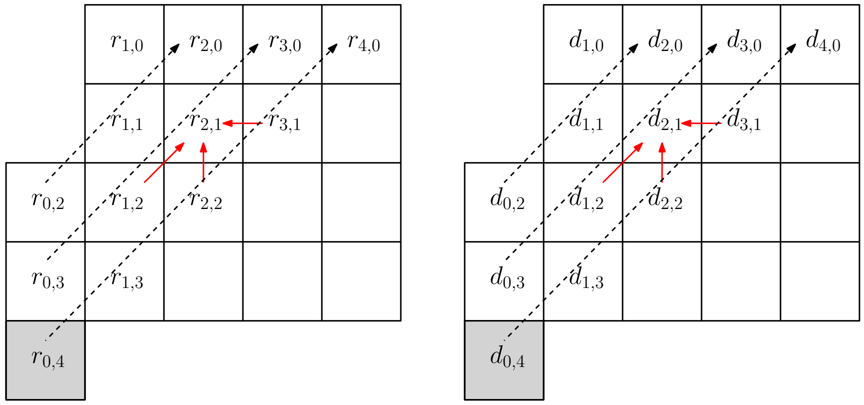
which upon rearrangement gives,

$$
\begin{aligned}
r_{k,m} =&\frac{1}{(k+m)\frac{\lambda_s}{n} + (n-k-m)(k+m)\frac{\lambda}{n-1} + k((k-1)p+m)\frac{\lambda}{n-1}} \\
&\times \Bigg((1-d_{k,m})k\frac{\lambda_s}{n} + r_{k,m+1}(n-k-m)(pk+m)\frac{\lambda}{n-1} \\
&\quad + r_{k+1,m}(n-k-m)k(1-p)\frac{\lambda}{n-1} \\
&\quad + r_{k-1,m+1}k((k-1)p+m)\frac{\lambda}{n-1}\Bigg),
\end{aligned} \tag{5.25}
$$

with $r_{0,n} = 0$.

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

$$
\mathbb{1}'_{\{R_{A_k,B_m}=0, X_{A_k \cup B_m}=0\}} =
$$



(a) (b)

Figure 5.4: Approach to computing (a) $r_{k,g}$ (b) $d_{k,g}$.

$$\begin{cases} 0, & i=0, j=0, h=1 \\ 1, & i=0, j \in B_m, h=1 \\ \mathbb{1}_{\{R_{A_k,B_{m+1}}=0, X_{A_k \cup B_{m+1}}=0\}}, & i \in \mathcal{N} \backslash (A_k \cup B_m), j \in A_k \cup B_m, h=0 \\ \mathbb{1}_{\{R_{A_k,B_{m+1}}=0, X_{A_k \cup B_{m+1}}=0\}}, & i \in \mathcal{N} \backslash (A_k \cup B_m), j \in B_m, h=1 \\ \mathbb{1}_{\{R_{A_{k+1},B_m}=0, X_{A_{k+1} \cup B_m}=0\}}, & i \in \mathcal{N} \backslash (A_k \cup B_m), j \in A_k, h=1 \\ \mathbb{1}_{\{R_{A_{k-1},B_{m+1}}=0, X_{A_{k+1} \cup B_{m+1}}=0\}} & i \in A_k, j \in A_k \cup B_m, h=0 \\ \mathbb{1}_{\{R_{A_{k-1},B_{m+1}}=0, X_{A_{k-1} \cup B_{m+1}=0}\}} & i \in A_k, j \in B_m, h=1 \\ \mathbb{1}_{\{R_{A_k,B_m}=0, X_{A_k \cup B_m}=0\}}, & \text{otherwise.} \end{cases} \tag{5.26}$$

The corresponding linear equation from (5.26) using (5.8) is,

$$\begin{aligned} 0 =& (0 - d_{k,m})\lambda_e + (1 - d_{k,m}) m \frac{\lambda_s}{n} \\ &+ (d_{k,m+1} - d_{k,m})(n-k-m)(k+m)p\frac{\lambda}{n-1} \end{aligned}$$

$$
\begin{aligned}
&+ (d_{k,m+1} - d_{k,m})(n-k-m)m(1-p)\frac{\lambda}{n-1} \\
&+ (d_{k+1,m} - d_{k,m})(n-k-m)k(1-p)\frac{\lambda}{n-1} \\
&+ (d_{k-1,m+1} - d_{k,m})k(k+m-1)p\frac{\lambda}{n-1} \\
&+ (d_{k-1,m+1} - d_{k,m})km(1-p)\frac{\lambda}{n-1},
\end{aligned} \tag{5.27}
$$

or

$$
\begin{aligned}
0 =&(0 - d_{k,m})\lambda_e + (1 - d_{k,m})m\frac{\lambda_s}{n} \\
&+ (d_{k,m+1} - d_{k,m})(n-k-m)(pk+m)\frac{\lambda}{n-1} \\
&+ (d_{k+1,m} - d_{k,m})(n-k-m)k(1-p)\frac{\lambda}{n-1} \\
&+ (d_{k-1,m+1} - d_{k,m})k((k-1)p+m)\frac{\lambda}{n-1},
\end{aligned} \tag{5.28}
$$

which upon rearrangement gives,

$$
\begin{aligned}
d_{k,m} =&\frac{1}{\lambda_e + m\frac{\lambda_s}{n} + (n-k-m)(k+m)\frac{\lambda}{n-1} + k((k-1)p+m)\frac{\lambda}{n-1}} \\
&\times \left(m\frac{\lambda_s}{n} + d_{k,m+1}(n-k-m)(pk+m)\frac{\lambda}{n-1} \right. \\
&\quad + d_{k+1,m}(n-k-m)k(1-p)\frac{\lambda}{n-1} \\
&\quad \left. + d_{k-1,m+1}k((k-1)p+m)\frac{\lambda}{n-1} \right).
\end{aligned} \tag{5.29}
$$

5.5 Numerical Analysis and Simulation Results

Though it is difficult to derive closed-form expressions for F (expected fraction of nodes with truth) and x_1 (expected version age at a node) from the complicated expressions found in (5.13), (5.16), (5.19), (5.25) and (5.29), we next provide some interesting observations and insights, and further support our theoretical analysis with numerical results.

5.5.1 Analysis When Truth Prevails Over Misinformation

We first simulate the fully-connected network of Fig. 5.1, under the first system model of truth prevalence, for up to a total time of 5×10^5 which we use as proxy for $t \to \infty$. We choose parameters $n = 10$, $p = 0.9$, $\lambda_e = 1$, $\lambda_s = 1$ and $\lambda = 1$, and vary one of the parameters at a time to observe their effects on F and x_1, plotting simulation points (blue dots) of $F = t_{1,0}$ on curves (red lines) obtained from equations (5.13), (5.16) and (5.19) in Figs. 5.5-5.9.

Fig. 5.5 shows the plots of F and x_1 with respect to the network size n. The real-time simulation points coincide with the values derived from the iterative calculations, supporting our theoretical results in Section 5.3. Fig. 5.5(a) shows that $t_{1,0} = 1$ when $n = 1$, since there is no scope of information mutating in a network containing just one user node due to the absence of inter-node links. Fig. 5.5 suggests that a larger network size leads to more misinformation and more staleness at nodes, with x_1 scaling logarithmically, as suggested in Lemma 5.1.

Next, Fig. 5.6 shows plots of F and x_1 with respect to λ_e. When λ_e is small,

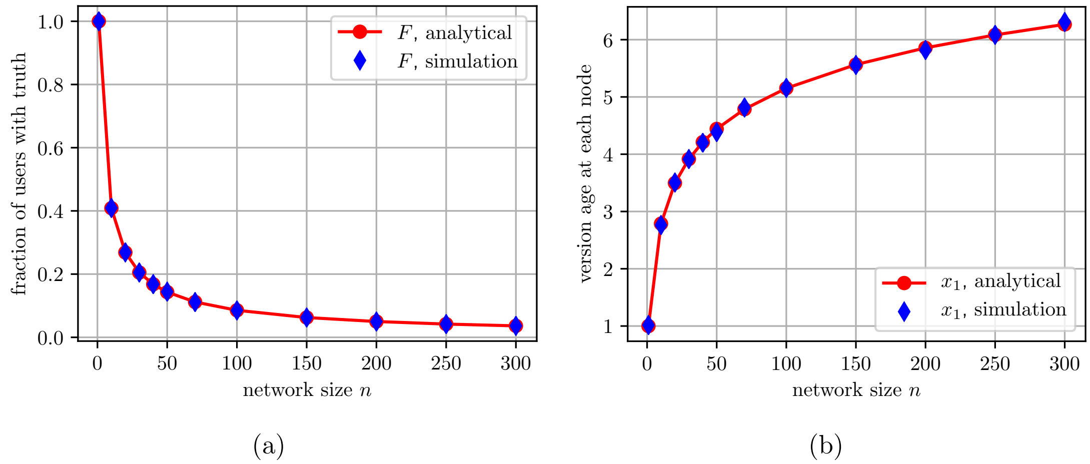


Figure 5.5: Analytical (red) and simulation (blue) results compared for F and x_1 as a function of n, with parameters $\lambda_e = 1, \lambda_s = 1, \lambda = 1, p = 0.9$.

then substituting $\lambda_e \approx 0$ in (5.16) yields $c_k \approx 1$ recursively for $k = n, \ldots, 1$. This is because the source hardly gets updated, which causes version age to mostly remain zero at nodes, and due to the prevalence of truth over misinformation, eventually all nodes get the truth corresponding to the version present at the source, i.e., the latest truth. Consequently, by iterating over (5.13), as suggested in Fig. 5.3, we have $t_{k,m} \approx 1$ for all k, m with $k + m \leq n$.

On the other hand, when λ_e is very high, substituting $\lambda_e = \infty$ in (5.16) gives $c_k \approx 0$ for all k. This is because the current version number at the source is incrementing very fast, causing the versions present at the nodes to become very quickly outdated, driving x_1 very large (Fig. 5.6(b)) and c_k zero. $t_{k,m}$ depends on λ_e only through c_k, and as large λ_e drives c_k to zero, $t_{k,m}$ does become smaller for all k, m (can be seen by iterating in the manner suggested in Fig. 5.3), which suggests that faster version updates at the source causes network nodes to be more misinformed. However, unlike c_k, $t_{k,m}$ does not become zero, due to other positive

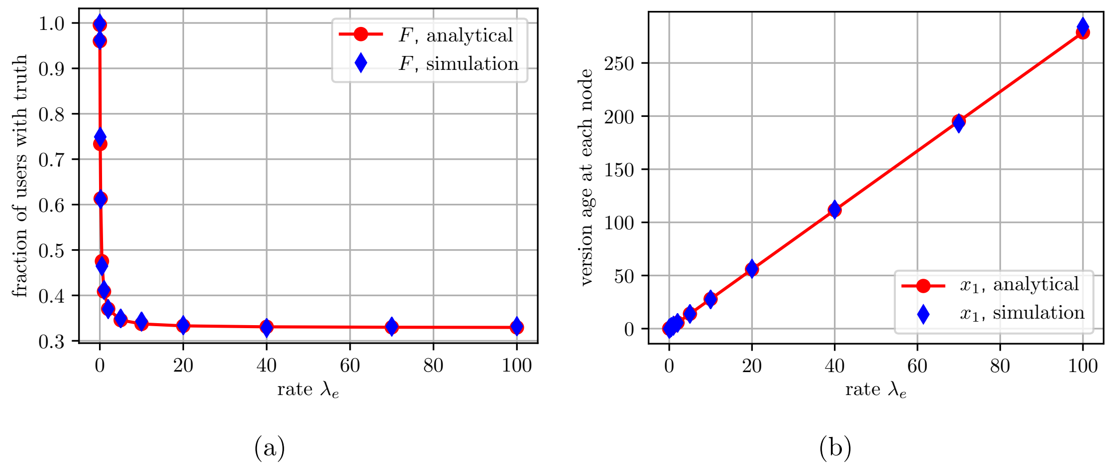


Figure 5.6: F and x_1 as a function of λ_e, with $n = 10, \lambda_s = 1, \lambda = 1, p = 0.9$.

terms in (5.13), owing to the fact that truth prevails over misinformation, thus many nodes will still have the truth corresponding to older versions.

Next, if p is very low, then information hardly mutates and only versions of truth circulate in the network. This is supported by substituting $p = 0$ in (5.13) where $t_{k,0}$ only depends on $t_{k+1,0}$ giving $t_{k,0} = 1$ all for $k = n, \ldots, 1$ (see Fig. 5.7(a)). However when p is high, this does not guarantee that all nodes are misinformed, since the source constantly sends out the latest truth to the network which always gets accepted at the receiving nodes. Substituting $p = 1$ in (5.16) gives $c_k = \frac{k\frac{\lambda}{n}}{\lambda_s + k\frac{\lambda}{n}} > 0$, causing $t_{1,k}$ to have non-zero value for all k from (5.13). From (5.19), version age x_1 remains independent of p, which is also supported by Fig. 5.7(b).

The dependency of F on λ_s and λ is more interesting. When λ_s is very large, the source updates the network very fast, causing all nodes to have the latest truth at all times. Substituting $\lambda_s = \infty$ in (5.13), (5.16) and (5.19) gives $c_k \approx 1$, $t_{k,m} \approx 1$ and $x_1 \approx 0$ (see Fig. 5.8). On the other hand, v_k for all k is a decreasing

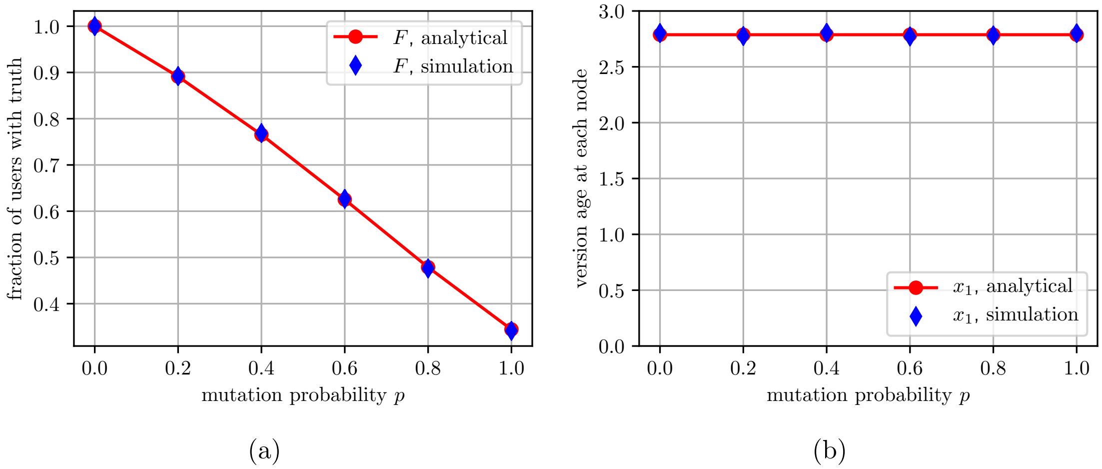
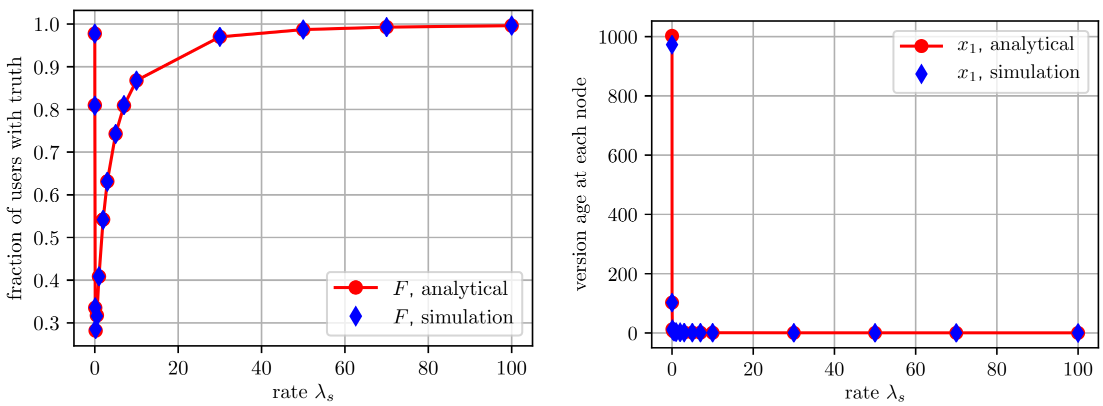


Figure 5.7: F and x_1 as a function of p, with $n = 10, \lambda_e = 1, \lambda_s = 1, \lambda = 1$.

function of λ_s (can be inductively proved from (5.19) starting with $k = n$), thus when λ_s is very small, by substituting $\lambda_s = 0$ in (5.19) we get a large version age of $x_1 = \sum_{k=1}^{n-1} \frac{\lambda_e}{k(n-k\frac{\lambda}{n-1})} + \frac{\lambda_e}{\lambda_s} \approx \frac{\lambda_e}{\lambda_s}$. In Fig. 5.8(b), the first simulation corresponds to $\lambda_s = 0.001$, for which $x_1 = \frac{\lambda_e}{\lambda_s} = \frac{1}{0.001} = 1000$. However, interestingly, since the source rarely sends any packet to the network when λ_s is close to zero, the whole network continues to gossip about the last version sent by the source to any of the network nodes, and eventually the truth corresponding to that version reaches all nodes. Substituting $\lambda_s = 0$ in (5.13) gives $t_{k,m} = 1$, which is also supported by the right extreme of Fig. 5.8(a). However, for intermediate values of λ_s, for $n \geq 2$, we get $c_k < 1$ from (5.16), which makes $t_{1,1} < 1$, which in turn makes $t_{1,0}$ strictly less than 1, as also supported by Fig. 5.8(a).

Finally, if gossiping rate λ is very large, implying that each node has high update sending capacity, then as soon as the source sends an update to a user node, the high gossiping rate allows for instant dissemination of the latest truth to all

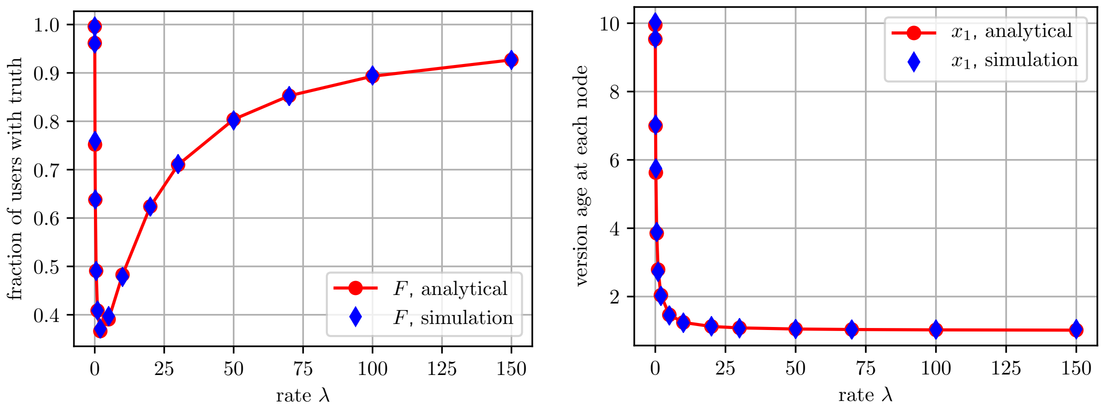


(a) (b)

Figure 5.8: F and x_1 as a function of λ_s, with $n = 10, \lambda_e = 1, \lambda = 1, p = 0.9$.

nodes of the network. This can be verified by substituting $\lambda = \infty$ in (5.13) which gives $t_{k,m} = 1$, and can be observed from Fig. 5.9(a). One would consequently expect the version age to also reduce at all nodes, however, the version age does not become zero in this case, since when the source gets a new version update, the version age at all nodes increments by one, and it continues to have a non-zero value at all nodes until the source sends a packet to some user node in the network. Substituting $\lambda = \infty$ in (5.19) gives $x_1 = v_1 = \ldots = v_n = \frac{\lambda_e}{\lambda_s}$, and in Fig. 5.9(b), version age converges to $\frac{\lambda_e}{\lambda_s} = 1$ as λ becomes large.

On the other hand, when λ is very small, implying there are negligible inter-node transmissions, nodes do not receive mutated packets from other nodes and depend primarily on updates received from the source, which always transmits the truth. Substituting $\lambda = 0$ in (5.13) gives $t_{1,0} = 1$, as observed in Fig. 5.9(a). Further, substituting $\lambda = 0$ in (5.19) gives $x_1 = \frac{n\lambda_e}{\lambda_s}$, which evaluates to 10 for the parameters in Fig. 5.9(b).



(a) (b)

Figure 5.9: F and x_1 as a function of λ, with $n = 10, \lambda_e = 1, \lambda_s = 1, p = 0.9$.

Therefore, $t_{1,0}$ if high (close to 1) with extremely high and low gossiping rates. However, for non-zero finite values of λ, $t_{1,0} < 1$. To show this, we use a proof by contradiction, by assuming $t_{1,0} = 1$, which would recursively require all $c_k = 1$ and $t_{k,m} = 1$ from (5.13) when λ is a finite positive value. However, from (5.16), $c_k < 1$ for $\lambda_e > 0$ and finite λ, leading to a contradiction.

5.5.2 Analysis When Misinformation Prevails Over Truth

We next re-simulate the network model under the second system model wherein misinformation prevails over truth. Fig. 5.10 shows the plots of F with respect to various network parameters. Note that the change in the system model, i.e., the modification in file exchange protocol to cause misinformation prevalence, does not affect age x_1, and hence the corresponding age simulations need not be redone.

Fig. 5.10(a) shows that F increases with network size n, similar to the first system model where truth prevails over misinformation. This suggests that a large

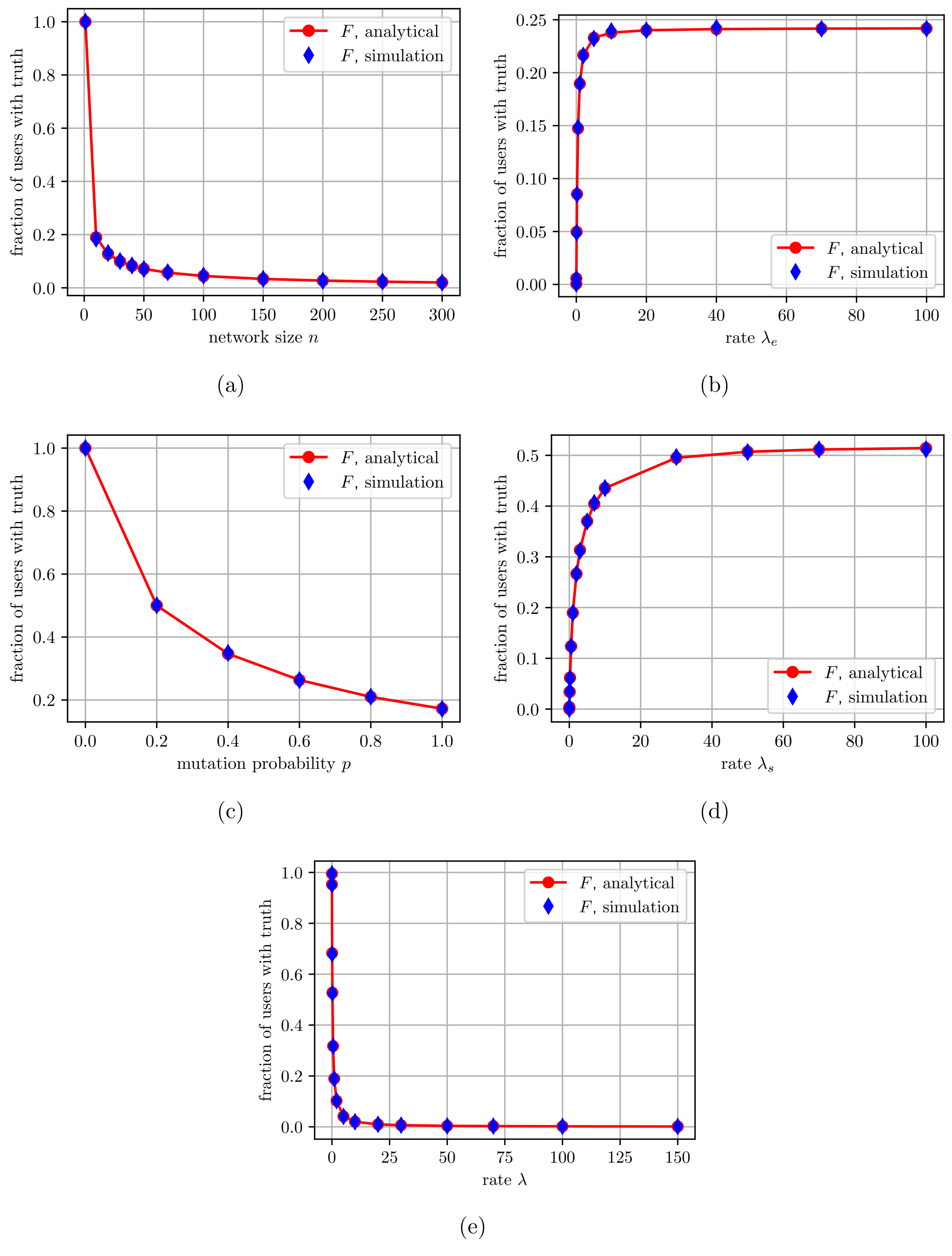


Figure 5.10: Analytical (red) and simulation (blue) results compared for F and x_1 as a function of (a) n, (b) λ_e, (c) p, (d) λ_s, (e) λ, with the fixed parameters chosen from $\lambda_e = 1$, $\lambda_s = 1$, $\lambda = 1$, $n = 10$, and $p = 0.9$.

network size leads to widespread misinformation, irrespective of the prevalence of truth or misinformation.

However, the behavior under λ_e variation is different from the first system model. When λ_e is small, substituting $\lambda_e \approx 0$ in (5.29) and iterating over different k, m in the order suggested in Fig. 5.4(b), we get $d_{k,m} \approx 1$ for all k, m with $k+m \leq n$. This is because the source hardly gets updated, which causes the instantaneous version age to mostly remain zero at all nodes after they catch up to the current version present at the source, and due to the prevalence of misinformation over truth, eventually all nodes get access to the mutated packets corresponding to the current version of the source. Similar to $d_{k,m}$, we iterate (5.25) in the order suggested in Fig. 5.4(a) to get $r_{k,m} \approx 0$, implying $F = r_{1,0} = 0$, as observed in Fig. 5.10(b).

On the other hand, when λ_e is very high, substituting $\lambda_e = \infty$ in (5.29) gives $d_{k,m} \approx 0$ for all feasible k, m, since the current version number at the source is incrementing very fast, causing the versions present at the nodes to become very quickly outdated, driving $d_{k,m}$ to zero. However, (5.25) does not show a direct dependence on variable λ_e. Further, the possibility of $r_{1,0} = 1$ would recursively require from (5.25) $r_{k,m} = 1$ for all k, m, which is infeasible due to the presence of $m\frac{\lambda_s}{n}$ term in the denominator of (5.25). Hence, we observe in Fig. 5.10(b), that though F increases from its initial value of zero, it ultimately converges to a value less than one.

Next, if p is very low, then information hardly mutates and only versions of truth circulate in the network. This is supported by substituting $p = 0$ in (5.29) and (5.25) where $d_{k,0}$ and $r_{k,0}$ only depends on $d_{k+1,0}$ and $r_{k+1,0}$, respectively, giving

$d_{k,0} = 1$ and, consequently, $r_{k,0} = 1$ all for $k = n, \ldots, 1$ (see Fig. 5.10(c)). However, when p is high, every packet gets mutated during transmission by network nodes. Since the source sends accurate packets corresponding to newer versions after the source updates, the low version age preference of network nodes prevents F from reducing all the way to zero, see Fig. 5.10(c). Substituting $p = 1$ in (5.29) gives $d_{k,m} < 1$ due to the λ_e in denominator, thus making $r_{k,m} > 0$ in (5.25).

Next, Fig. 5.10(d) shows the plot of F with respect to λ_s. When λ_s is small, then substituting $\lambda_s = 0$ in (5.25) gives $r_{k,m} = 0$, for all k, m, since the source does not relay fresh updates about the newer versions at the source to the network anymore. In the absence of updates from the source, the whole network continues to gossip about the last version sent by the source to any of the network nodes, and eventually mutated packets of that version prevail over all nodes. Hence, $F = r_{1,0} = 0$ initially in Fig. 5.10(d).

As λ_s becomes large, the source transmits the latest truth to all network nodes very fast. However, since in the second system model, misinformation prevails over truth, an accurate packet received from the source need not be accepted at the receiving node, if the latter has access to a mutated copy of the same version. Substituting $\lambda_s = \infty$ in (5.29) results in $d_{k,m} = 1$ when $m > 0$. Consequently, substituting $\lambda_s = \infty$ in (5.25) gives $r_{1,0} = 1 - d_{1,0} < 1$, where $d_{1,0} > \frac{d_{1,1} p\lambda}{\lambda_e + \lambda} = \frac{p\lambda}{\lambda_e + \lambda} > 0$. As expected, F converges to a value lesser than one in Fig. 5.10(d).

Finally, from Fig. 5.10(e), we can see that F is a decreasing function of gossiping rate λ. Since packets get mutated only during gossip transmissions, substituting $\lambda = 0$ in (5.25) gives $F = r_{1,0} = 1 - d_{1,0} = 1$, i.e., no mutated packets in the network in the absence of gossiping, where $d_{1,0} = 0$ from (5.29). On the other hand, when $\lambda = \infty$, $d_{k,m} = 1$ from (5.25) and $r_{k,m} = 0$ from (5.25). Thus, under high gossiping, all packets create their mutated packets in approximately p fraction of their transmissions, such that these mutated packets spread to all nodes, based on the file exchange protocol.

5.6 Conclusion

In this chapter, we analyzed a network of user nodes receiving updates from a source and employing an age-based gossip protocol to expedite version update dissemination. We studied how information mutations during transmissions can lead to spread of misinformation within the network. We considered two system models: one where truth prevails over misinformation and one where misinformation prevails over truth. Using stochastic hybrid systems (SHS) modeling, we investigated the expected fraction of nodes with accurate information and the expected version age at the nodes, and how these quantities depend on various network parameters. Our analysis revealed distinct behaviors under the two models. When truth prevails, extreme gossiping rates (either very high or very low) effectively mitigate misinformation, while moderate gossiping rates exacerbate its spread. On the other hand, when misinformation prevails, higher gossiping rates lead to more widespread misinformation. These results emphasize the importance of understanding the interplay between gossiping rates and information integrity in networks.

CHAPTER 6

Sacrificing Freshness for Reliable Information in Age-Based Gossiping

6.1 Introduction

In this chapter, we consider a system where a set of n nodes, $\mathcal{N} = \{1, \dots, n\}$, wish to track a continually updating process or event E, which gets updated according to a Poisson process with rate λ_E, see Fig. 6.1. However, to transmit the information about the event to the nodes, two sources are available, such that one of these sources is more reliable than the other and is expected to transmit more accurate information. We call the former as *reliable*, R, source and the latter as *unreliable*, U, source, and they send out packets to the network with total update rate of λ_R and λ_U, respectively. Here, the less reliable source could, for example, be a proxy for a cheap sensor monitoring the physical environment, transmitting quantized or noisy measurements to an IoT network. Another example could be when multiple websites or news sources are delivering scores of a sports event to a group of interested viewers, such that some of these sources are more reliable than others.

Note that in the above settings, the event of interest is dynamic in nature. In

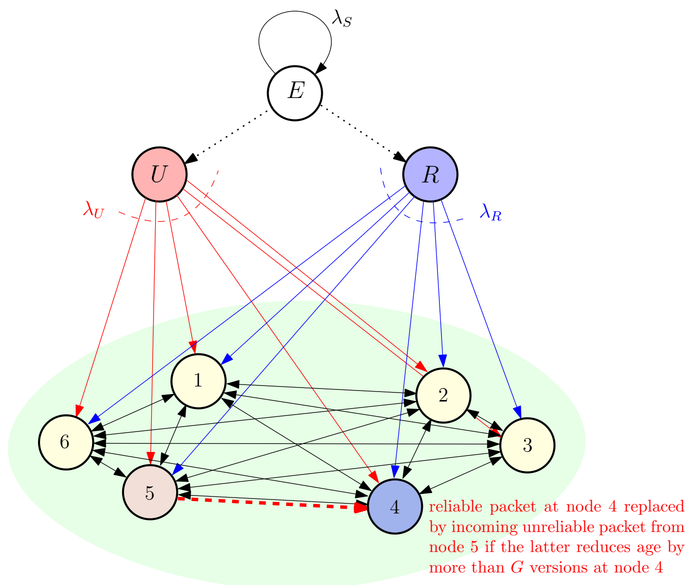


Figure 6.1: System with reliable (R) and unreliable (U) sources observing an event (E) and gossip network of 6 nodes, depicting an inter-node transition from node 5 (possessing unreliable packet) to node 4 (possessing reliable packet). Node 4 accepts the incoming packet only if this action results in a decrease in version age at node 4 by at least G versions.

this work, we employ version age of information metric as we associate a version number to every information about the event. In line with the notation followed in this chapter, if $V_E(t)$ denotes the version number corresponding to the current state of event and $V_i(t)$ denotes the version number of the information about the event present at node i, then the instantaneous version age of information at node i is defined as $X_i(t) = V_E(t) - V_i(t)$, where $V_E(t)$ increments by one every time the event gets updated.

In addition to the sources disseminating updates to the network, the nodes in the network aim to further expedite the dissemination of updates in order to improve their version age of information and information reliability through gossiping [4, 5, 9, 71–76, 78, 93, 94, 101, 103–106]. All these works have a single source of information responsible for transmitting the latest information packets to the network, and the nodes exchange packets with the single goal of improving their freshness of information.

In this chapter, we consider two kinds of sources, a reliable source and an unreliable source, who are assumed to always have the latest information about the event, resulting in their respective version age $X_R(t)$ and $X_U(t)$ to be zero at all times. The user nodes prefer a reliable packet (originated at the reliable source) over an unreliable packet (originated at the unreliable source) even if the former is a little bit outdated compared with the latter, and are willing to sacrifice their freshness of information by a small amount if that allows the node to switch to a reliable packet from an unreliable packet.

Let $S_i(t)$ indicate the reliability status of the information packet present at node i at time t, with $S_i(t) = 0$ and $S_i(t) = 1$ corresponding to reliable and unreliable packet, respectively. At time t, if node i sends an update to node j, node j decides whether to accept or reject the packet in accordance with the following set of rules:

- If $S_i(t) = 1$ and $S_j(t) = 1$, i.e., both nodes possess unreliable information, then node j chooses the packet with lower version age of information.

- If $S_i(t) = 0$ and $S_j(t) = 0$, i.e., both nodes possess reliable information, then node j again chooses the packet with lower version age of information.

- If $S_i(t) = 0$ and $S_j(t) = 1$, i.e., incoming packet has reliable information but node j has unreliable information, then node j will choose the reliable incoming packet as long as $X_i \leq X_j + G$, in other words, the incoming packet is no more than G versions older than the packet already present at node j.

- If $S_i(t) = 1$ and $S_j(t) = 0$, i.e., node j already has reliable information packet and the incoming packet is unreliable, then node j would continue to keep its reliable packet as long as $X_j \leq X_i + G$, in other words, the reliable packet present at node j is no more than G versions older than the incoming unreliable packet.

With this protocol, let $F(t)$ denote the fraction of user nodes that have unreliable information packet at time t, such that,

$$F(t) = \frac{S_1(t) + S_2(t) + \ldots + S_n(t)}{n}. \tag{6.1}$$

We are interested in characterizing the long-term expectations $F = \lim_{t\to\infty} \mathbb{E}[F(t)]$ and $x_i = \lim_{t\to\infty} \mathbb{E}[X_i(t)]$, $i \in \mathcal{N}$.

In the next section, we begin with modelling the problem as an SHS system to derive certain linear equations to characterize F and x_i. We then prove several results that allow us to show that F is a decreasing function of G and x_i are increasing functions of G. Therefore, G induces a trade-off between reliability and freshness of information. Further, we study the dependency of F and x_i on various network parameters. We finally present simulations results to verify our analytical results and give further insights.

6.2 System Model and SHS Characterization

The system model includes a reliable source (R) and an unreliable source (U) that send updates to a set of n user nodes $\mathcal{N} = 1, \ldots, n$ about an event (E), which gets updated as a rate λ_E Poisson process, as shown in Fig. 6.1. Both reliable and unreliable sources always have the latest information and send updates to user node $i \in \mathcal{N}$ as Poisson processes with rates of $\frac{\lambda_R}{n}$ and $\frac{\lambda_U}{n}$, respectively. Additionally, for every $i, j \in \mathcal{N}$, node i sends updates to node j as a Poisson process with rate $\lambda_{ij} = \frac{\lambda}{n-1}$ as part of the underlying gossip network. Given the network symmetry, the reliability status and version age processes at all user nodes will be statistically identical, and hence, $x_1 = \ldots = x_n$ and $F = s_1$, where $s_i = \lim_{t\to\infty} \mathbb{E}[S_i(t)]$. Next, we use SHS modelling [90] to obtain a set of linear equations to derive s_1 and x_1.

We select the continuous state for our SHS model as $(\boldsymbol{S}(t), \boldsymbol{X}(t)) \in \mathbb{R}^{2n}$, where $\boldsymbol{S}(t) = [S_1(t), \ldots, S_n(t)]$ and $\boldsymbol{X}(t) = [X_1(t), \ldots, X_n(t)]$ denote the instantaneous reliability status and instantaneous version age, respectively, at the n user nodes at time t. We say that transition (i, j) takes place when node i sends an update packet to node j, with (E, E) representing event update, and (U, i) and (R, i) representing updates to user node i from unreliable and reliable source, respectively. $S_i(t)$ and $X_i(t)$ do not change between transitions and SHS model operates in a single discrete state, where the continuous state obeys the differential equation $(\dot{\boldsymbol{S}}(t), \dot{\boldsymbol{X}}(t)) = \mathbf{0}_{2n}$. The set of transitions is,

$$\mathcal{L} = \{(E, E)\} \bigcup \{(U, i) : i \in \mathcal{N}\} \bigcup \{(R, i) : i \in \mathcal{N}\} \bigcup \{(i, j) : i, j \in \mathcal{N}\}, \quad (6.2)$$

such that the transition (i,j) resets the state $(\boldsymbol{S},\boldsymbol{X})$ at time t to $\phi_{i,j}(\boldsymbol{S},\boldsymbol{X},t)\in\mathbb{R}^{2n}$ post transition. The rates λ_{ij} associated with each transition (i,j) are given as,

$$\lambda_{ij}=\begin{cases}\frac{\lambda_U}{n}, & i=U, j\in\mathcal{N}\\ \frac{\lambda_R}{n}, & i=R, j\in\mathcal{N}\\ \frac{\lambda}{n-1}, & i,j\in\mathcal{N}\\ \lambda_E, & i=E, j=E.\end{cases} \tag{6.3}$$

Next, we define some variables that will come in handy later. Consider a set A of nodes. Some nodes in this set might have reliable information and others might have unreliable information. Let $R(A)\subseteq A$ denote the subset of nodes that posses reliable information and let $U(A)\subseteq A$ denote the subset of nodes that possess unreliable information. For a continuous state $(\boldsymbol{S},\boldsymbol{X})$ and set of nodes A, we define version age of set A, X_A as follows:

- If $A=\varnothing$, then $X_A=\infty$.
- If $A=R(A)$ or $A=U(A)$, then $X_A=\min_{j\in A}X_j$.
- If $\min_{j\in R(A)}X_j\leq\min_{j\in U(A)}X_j+G$, then $X_A=\min_{j\in R(A)}X_j$.
- If $\min_{j\in U(A)}X_j\leq\min_{j\in R(A)}X_j-G-1$, then $X_A=\min_{j\in U(A)}X_j$.

Next, we define reliability status of set A, S_A, as follows:

- If $X_{R(A)}\leq X_{U(A)}+G$, then $S_A=0$.
- If $X_{U(A)}\leq X_{R(A)}-G-1$, then $S_A=1$.

In essence, determining S_A and X_A requires us to identify the best node in set A in some sense, such that reliability status and version age of that node are also the reliability status and version age of the set. From the definitions of S_A and X_A, we can see that as long as the most recent reliable packet is at most G versions older than the latest unreliable packet in the set of nodes, the node with the latest reliable packet establishes the values of X_A and S_A. Otherwise, the node with the latest unreliable packet determines X_A and S_A.

With this definition, considering the transition (i,j) at time t, the reset map to $\phi_{i,j}(\boldsymbol{S},\boldsymbol{X},t) = [S'_1,\ldots,S'_n,X'_1,\ldots,X'_n] \in \mathbb{R}^{2n}$ can be described as,

$$S'_\ell = \begin{cases} S_{\{U,\ell\}}, & i = U, j \in \mathcal{N}, \ell = j \\ 0, & i = R, j \in \mathcal{N}, \ell = j \\ S_{\{i,\ell\}}, & i,j \in \mathcal{N}, \ell = j \\ S_\ell, & \text{otherwise}, \end{cases} \tag{6.4}$$

$$X'_\ell = \begin{cases} X_\ell + 1, & i = E, j = E, \ell = j \\ \mathbb{1}_{\{X_{R(\{\ell\})}=1\}}, & i = U, j \in \mathcal{N}, \ell = j \\ 0, & i = R, j \in \mathcal{N}, \ell = j \\ X_{\{i,\ell\}}, & i,j \in \mathcal{N}, \ell = j \\ X_\ell, & \text{otherwise}, \end{cases} \tag{6.5}$$

where $\mathbb{1}_{\{.\}}$ is the indicator function.

Next, we will pick a series of test functions $\psi : \mathbb{R}^{2n} \times [0, \infty) \to \mathbb{R}$ that are time-invariant, i.e., their partial derivative with respect to t is $\frac{\partial \psi(\boldsymbol{X}, \boldsymbol{U}, t)}{\partial t} = 0$, such that their long-term expected value $\mathbb{E}[\psi] = \lim_{t\to\infty} \mathbb{E}[\psi(\boldsymbol{S}(t), \boldsymbol{X}(t), t)]$ will be useful for analysis later. Since the test function only depends on the continuous state values $(\boldsymbol{S}, \boldsymbol{X})$ and is time-invariant, for simplicity, we will drop the third input t and write $\psi(\boldsymbol{S}, \boldsymbol{X}, t)$ as $\psi(\boldsymbol{S}, \boldsymbol{X})$, which is assumed to satisfy $\dot{\psi}(\boldsymbol{S}(t), \boldsymbol{X}(t)) = 0$. Defining $\mathbb{E}[\psi(\phi_{i,j})] = \lim_{t\to\infty} \mathbb{E}[\psi(\phi_{i,j}(\boldsymbol{S}(t), \boldsymbol{X}(t), t))]$, [90, Thm. 1] yields

$$0 = \sum_{(i,j)\in\mathcal{L}} \left(\mathbb{E}[\psi(\phi_{i,j})] - \mathbb{E}[\psi]\right) \lambda_{ij}. \tag{6.6}$$

We will be using this equation repeatedly by introducing a set of time-invariant test functions suitable for our analysis. For more details, readers are referred to [5, 90, 107].

6.3 Reliability and Version Age Analysis

Since version age and reliability status processes are statistically identical for all user nodes, let A_k denote an arbitrary subset of k user nodes. Our first test function is $\psi(\boldsymbol{S}, \boldsymbol{X}) = S_{A_k}$, which upon (i, j) transition becomes $\psi(\phi_{i,j}(\boldsymbol{S}, \boldsymbol{X}, t)) = S'_{A_k}$ and can be characterized using (6.4), (6.5) as follows,

$$S'_{A_k} = \begin{cases} S_{A_k\cup\{U\}}, & i=U, j\in A_k \\ 0, & i=R, j\in A_k \\ S_{A_{k+1}}, & i=\mathcal{N}\backslash A_k, j\in A_k \\ S_{A_k}, & \text{otherwise.} \end{cases} \tag{6.7}$$

Defining $a_k = \lim_{t\to\infty}\mathbb{E}[S_{A_k}(t)]$ and $b_k = \lim_{t\to\infty}\mathbb{E}[S_{A_k\cup\{U\}}(t)]$, and using (6.6) gives,

$$0 = (b_k - a_k)\frac{k\lambda_U}{n} + (0 - a_k)\frac{k\lambda_R}{n} + (a_{k+1} - a_k)\frac{k(n-k)\lambda}{n-1}. \tag{6.8}$$

Our second test function is $\psi(\boldsymbol{S},\boldsymbol{X}) = S_{A_k\cup\{U\}}$, which has (i,j) transition map as follows,

$$S'_{A_k\cup\{U\}} = \begin{cases} 1 - \mathbb{1}_{\{X_{R(A_k)}\leq G-1\}}, & i=E, j=E \\ 0, & i=R, j\in A_k \\ S_{A_{k+1}\cup\{U\}}, & i=\mathcal{N}\backslash A_k, j\in A_k \\ S_{A_k\cup\{U\}}, & \text{otherwise.} \end{cases} \tag{6.9}$$

Note that the version age of the most recent unreliable packet in the set $A_k\bigcup\{U\}$ will always remain zero, since the unreliable source has zero version age at all times. Hence, post (E,E) transition, $S'_{A_k\cup\{U\}}$ will be zero, i.e., the node with the latest reliable packet will be the best node, only if it had at most $G-1$ version age before transition, i.e., $\mathbb{1}_{\{X_{R(A_k)} \leq G-1\}}$, since version age increments by one after the (E,E) transition.

Defining $c_{k,g} = \lim_{t\to\infty} \mathbb{E}[\mathbb{1}_{\{X_{R(A_k)}(t) \leq g\}}]$, (6.6) gives,

$$0 = (1 - c_{k,G-1} - b_k)\lambda_E + (0 - b_k)\frac{k\lambda_R}{n} + (b_{k+1} - b_k)\frac{k(n-k)\lambda}{n-1}. \tag{6.10}$$

Our third test function is $\psi(\boldsymbol{S}, \boldsymbol{X}) = \mathbb{1}_{\{X_{R(A_k)} \leq g\}}$, $g \in \{0, \dots, G-1\}$, which has the (i,j) transition map as follows,

$$\mathbb{1}'_{\{X_{R(A_k)} \leq g\}} = \begin{cases} \mathbb{1}_{\{X_{R(A_k)} \leq g-1\}}, & i = E, j = E \\ 1, & i = R, j \in A_k \\ \mathbb{1}_{\{X_{R(A_{k+1})} \leq g\}}, & i = \mathcal{N} \backslash A_k, j \in A_k \\ \mathbb{1}_{\{X_{R(A_k)} \leq g\}}, & \text{otherwise}, \end{cases} \tag{6.11}$$

which upon employing (6.6), gives,

$$0 = (c_{k,g-1} - c_{k,g})\lambda_E + (1 - c_{k,g})\frac{k\lambda_R}{n} + (c_{k+1,g} - c_{k,g})\frac{k(n-k)\lambda}{n-1}. \tag{6.12}$$

Here, note that version age cannot be a negative number, therefore, $\mathbb{1}_{\{X_{R(A_k)} \leq -1\}} = 0$ and $c_{k,-1} = 0$.

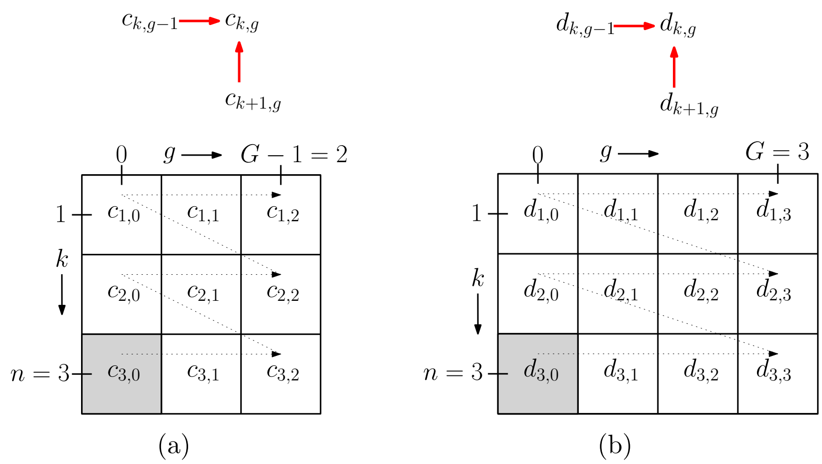


Figure 6.2: Approach to computing (a) $c_{k,g}$ and (b) $d_{k,g}$.

Then, (6.8), (6.10), (6.12) can be rewritten as follows,

$$a_k = \frac{b_k \frac{k\lambda_U}{n} + a_{k+1}\frac{k(n-k)\lambda}{n-1}}{\frac{k\lambda_U}{n} + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}, \tag{6.13}$$

$$b_k = \frac{(1 - c_{k,G-1})\lambda_E + b_{k+1}\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}, \tag{6.14}$$

$$c_{k,g} = \frac{c_{k,g-1}\lambda_E + \frac{k\lambda_R}{n} + c_{k+1,g}\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}. \tag{6.15}$$

Note that $a_1 = s_1 = F$ and from (6.13), we see that computation of a_1 requires solving all a_k and b_k. In turn, b_k from (6.14) requires computation of $c_{k,G-1}$ for all k, which in turn requires computation of $c_{k,g}$ for all k and g from (6.15). Therefore, we first compute $c_{k,g}$ as in Fig. 6.2(a), since $c_{k,g}$ depends on $c_{k,g-1}$ and $c_{k+1,g}$. Starting with $c_{n,0} = \frac{\lambda_R}{\lambda_E+\lambda_R}$, for each k in the order $\{n, \ldots, 1\}$, we compute $c_{k,g}$ for g in order $\{0, \ldots, G-1\}$. Once we have all the $c_{k,G-1}$, which is the last column of Fig. 6.2(a), we substitute them in (6.14) to solve for b_k in the order $k = n, \ldots, 1$. Finally, we use the b_k to compute a_k from (6.13) in the order $k = n, \ldots, 1$.

Next, to characterize the version age at the nodes, we pick the test functions X_{A_k} and $X_{A_k}\mathbb{1}_{\{X_{R(A_k)}\leq g\}}$, which have the following (i,j) transition maps,

$$X'_{A_k}=\begin{cases} X_{A_k}+1, & i=E, j=E \\ X_{A_k}\mathbb{1}_{\{X_{R(A_k)}\leq G\}}, & i=U, j\in A_k \\ 0, & i=R, j\in A_k \\ X_{A_{k+1}}, & i=\mathcal{N}\backslash A_k, j\in A_k \\ X_{A_k}, & \text{otherwise,} \end{cases} \tag{6.16}$$

and

$$X_{A_k}\mathbb{1}'_{\{X_{R(A_k)}\leq g\}}=\begin{cases} (X_{A_k}+1)\mathbb{1}_{\{X_{R(A_k)}\leq g-1\}}, & i=E, j=E \\ 0, & i=R, j\in A_k \\ X_{A_{k+1}}\mathbb{1}_{\{X_{R(A_{k+1})}\leq g\}}, & i=\mathcal{N}\backslash A_k, j\in A_k \\ X_{A_k}\mathbb{1}_{\{X_{R(A_k)}\leq g\}}, & \text{otherwise,} \end{cases} \tag{6.17}$$

where (6.17) is obtained from the product of (6.16) and (6.11), however, each case of (6.17) has the same set size k in all product terms, which prevents the need to evaluate any further test cases. Defining $e_k=\lim_{t\to\infty}\mathbb{E}[X_{A_k}(t)]$ and $d_{k,g}=\lim_{t\to\infty}\mathbb{E}[X_{A_k}\mathbb{1}_{\{X_{R(A_k)}\leq g\}}]$ and using (6.6), these transition maps give the following linear equations,

$$0=(e_k+1-e_k)\lambda_E+(d_{k,G}-e_k)\frac{k\lambda_U}{n}+(0-e_k)\frac{k\lambda_R}{n}+(q_{k+1}-e_k)\frac{k(n-k)\lambda}{n-1}, \tag{6.18}$$

and

$$0 = (d_{k,g-1} + c_{k,g-1} - d_{k,g})\lambda_E + (0 - d_{k,g})\frac{k\lambda_R}{n} + (d_{k+1,g} - d_{k,g})\frac{k(n-k)\lambda}{n-1}, \quad (6.19)$$

which upon rearrangement, along with (6.15), give the following set of equations,

$$e_k = \frac{\lambda_E + d_{k,G}\frac{k\lambda_U}{n} + e_{k+1}\frac{k(n-k)\lambda}{n-1}}{\frac{k\lambda_U}{n} + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}, \quad (6.20)$$

$$d_{k,g} = \frac{(d_{k,g-1} + c_{k,g-1})\lambda_E + d_{k+1,g}\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}, \quad (6.21)$$

$$c_{k,g} = \frac{c_{k,g-1}\lambda_E + \frac{k\lambda_R}{n} + c_{k+1,g}\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}. \quad (6.22)$$

Note that $\mathbb{1}_{\{X_{R(A_k)} \leq -1\}} = 0$, and consequently, $d_{k,-1} = 0$, since version age is non-negative. To solve for $x_1 = e_1$, we first compute $c_{k,g}$ as in Fig. 6.2(a), as discussed previously for solution of F. Then, using $c_{k,g}$, we compute $d_{k,g}$ from (6.21) in the row-wise left to right bottom-up manner shown in Fig. 6.2(b), starting with $d_{n,0} = 0$. Finally, using $d_{k,G}$, which is the last column of Fig. 6.2(b), we compute all e_k from (6.20).

Since F depends on a_k, b_k and $c_{k,g}$, we next use (6.13), (6.14), (6.15) to prove certain structural results that enable us to prove that F decreases with G. Likewise, we prove certain results using (6.20), (6.21), (6.22) to show that x_1 increases with G.

Proposition 6.1 *For any $k \in \{1, \dots, n\}$ and $g \in \{1, \dots, G\}$, we have $c_{k,g} > c_{k,g-1}$.*

Proof: We provide a proof using double induction on variables k and g. We first assume the proposition holds for some $k+1$ and prove it for k. That is, for some $k \in \{1, \ldots, n-1\}$, we assume $c_{k+1,g} > c_{k+1,g-1}$ for all $g \in \{1, \ldots, G\}$ and show that it implies $c_{k,g} > c_{k,g-1}$ for all $g \in \{1, \ldots, G\}$. Finally, we prove the proposition for base case of $k = n$.

Assuming for $(k+1)$ *and proving for* k*:* To show $c_{k,g} > c_{k,g-1}$ for all g, we apply a second induction on variable g. We assume the proposition holds for $g-1$, i.e., $c_{k,g-1} > c_{k,g-2}$ and show that it holds for g as well, i.e., $c_{k,g} > c_{k,g-1}$, for some $g \in \{2, \ldots, G\}$. Replacing g by $g-1$ in (6.15),

$$c_{k,g-1} = \frac{c_{k,g-2}\lambda_E + \frac{k\lambda_R}{n} + c_{k+1,g-1}\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}. \tag{6.23}$$

Subtracting (6.23) from (6.15), we get,

$$c_{k,g} - c_{k,g-1} = \frac{(c_{k,g-1} - c_{k,g-2})\lambda_E}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}} + \frac{(c_{k+1,g} - c_{k+1,g-1})\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}. \tag{6.24}$$

In (6.24), the first term is positive due to the assumption under the second induction for $g-1$, and the second term is positive due to the assumption under the first induction for $k+1$.

For the base case of $g = 1$, the analogue of (6.24) is,

$$c_{k,1} - c_{k,0} = \frac{c_{k,0}\lambda_E}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}} + \frac{(c_{k+1,1} - c_{k+1,0})\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}. \tag{6.25}$$

Here, $c_{k,0} \geq \frac{\frac{k\lambda_R}{n}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}} > 0$ from (6.15), which implies that the first term of

(6.25) is positive. Further, $c_{k+1,1} - c_{k+1,0} > 0$ due to the assumption under the first induction for $(k+1)$, which implies the second term of (6.25) is positive as well.

Base case of $k = n$: To prove the proposition for base case of $k = n$, i.e., $c_{n,g} > c_{n,g-1}$ for all $g \in \{1, \ldots, G\}$, we use induction on g. We assume that it holds for $g-1$, i.e., $c_{n,g-1} > c_{n,g-2}$, and holds for g as well, i.e., $c_{n,g} > c_{n,g-1}$, some $g \in \{2, \ldots, G\}$. From (6.15), we get

$$c_{n,g} = \frac{c_{n,g-1}\lambda_E + \lambda_R}{\lambda_E + \lambda_R}, \tag{6.26}$$

$$c_{n,g-1} = \frac{c_{n,g-2}\lambda_E + \lambda_R}{\lambda_E + \lambda_R}. \tag{6.27}$$

Taking the difference of (6.26) and (6.27), and using the assumption under induction for $g-1$, we get

$$c_{n,g} - c_{n,g-1} = \frac{(c_{n,g-1} - c_{n,g-2})\lambda_E}{\lambda_E + \lambda_R} > 0. \tag{6.28}$$

For the base case of $g = 1$, similarly (6.15) gives

$$c_{n,1} = \frac{c_{n,0}\lambda_E + \lambda_R}{\lambda_E + \lambda_R}, \tag{6.29}$$

$$c_{n,0} = \frac{\lambda_R}{\lambda_E + \lambda_R}. \tag{6.30}$$

From (6.29) and (6.30), we get

$$c_{n,1} - c_{n,0} = \frac{c_{n,0}\lambda_E}{\lambda_E + \lambda_R} = \frac{\lambda_R\lambda_E + \lambda_R}{(\lambda_E + \lambda_R)^2} > 0. \tag{6.31}$$

Hence, $c_{k,g} > c_{k,g-1}$ holds for all $k \in \{1, \ldots, n\}$ and $g \in \{1, \ldots, G\}$. ■

Proposition 6.2 *For any $k \in \{1, \ldots, n\}$, $g \in \{1, \ldots, G\}$, we have $d_{k,g} > d_{k,g-1}$.*

Proof: We prove the proposition using double induction on variables k and g, similar to Proposition 6.1. In our first induction, we assume that the proposition holds for $(k+1)$, i.e., $d_{k+1,g} > d_{k+1,g-1}$ and show that it holds for k as well, i.e., $d_{k,g} > d_{k,g-1}$, for all g. To prove for all g, we apply a second induction, where we assume that the proposition holds for $g-1$, i.e, $d_{k,g-1} > d_{k,g-2}$, and show that it holds for g as well, i.e., $d_{k,g} > d_{k,g-1}$.

Substituting $g-1$ for g in (6.21), we get

$$d_{k,g-1} = \frac{(d_{k,g-2} + c_{k,g-2})\lambda_E + d_{k+1,g-1}\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}. \tag{6.32}$$

Subtracting (6.32) from (6.21), we get

$$\begin{aligned} d_{k,g} - d_{k,g-1} =& \frac{(d_{k,g-1} - d_{k,g-2})\lambda_E}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}} + \frac{(c_{k,g-1} - c_{k,g-2})\lambda_E}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}} \\ &+ \frac{(d_{k+1,g} - d_{k+1,g-1})\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}. \end{aligned} \tag{6.33}$$

In (6.33), the first term is positive due to the assumption under the second induction for $g-1$, the second term is positive due to Proposition 6.1, and the third term is positive due to the assumption under the first induction for $(k+1)$. The base cases of $g = 0$ and $k = n$ can be likewise proved. ■

Proposition 6.3 *b_k is a decreasing function of G, for all $k \in \{1, \ldots, n\}$.*

Proof: For $k = n$, we have from (6.14), that

$$b_n = \frac{(1 - c_{n,G-1})\lambda_E}{\lambda_E + \lambda_R}. \tag{6.34}$$

We know from Proposition 6.1 that $c_{n,G-1}$ increases with G, and therefore, b_n decreases with G. Next, we assume that b_{k+1} is a decreasing function of G, and inductively argue that b_k is also a decreasing function of G. The latter can be seen to be true from (6.14), where $(1 - c_{k,G-1})$ is a decreasing function of G from Proposition 6.1 and b_{k+1} is a decreasing function of G due to the assumption under induction for $(k + 1)$. ■

Lemma 6.1 *For given $n \in \mathbb{N}$, $\lambda, \lambda_E, \lambda_R, \lambda_U \in \mathbb{R}^{++}$, a_1 is a decreasing function of G. Further,* $\lim_{G\to\infty} a_1 = 0$.

Proof: For $k = n$, we have from (6.13), that

$$a_n = \frac{b_n\lambda_U}{\lambda_U + \lambda_R}. \tag{6.35}$$

We know from Proposition 6.3 that b_n decreases with G, and therefore, a_n decreases with G. Next, we assume that a_{k+1} is a decreasing function of G, and inductively argue that a_k is also a decreasing function of G. From (6.13), we see that the denominator does not depend on G and the two terms of the numerator are decreasing functions of G due to Proposition 6.3 and the assumption under the induction for $(k + 1)$.

Therefore, a_1 is a monotonically decreasing function of G. Further, by definition, $a_1 \in [0,1]$, i.e., a_1 is bounded. Thus, by monotone convergence theorem, $\lim_{G\to\infty} a_1$ exists. $G \to \infty$ implies that in the file exchange protocol, nodes do not accept unreliable packets and hold on to their last received reliable packets. Therefore, all network nodes have reliable information at all times and heuristically $\lim_{G\to\infty} a_1$ is zero.

Mathematically, the same can be shown as follows. First, using (6.12), we can inductively show that $\lim_{G\to\infty} c_{k,G-1} = 1$ for all k, by iterating over k in the order $\{n,\ldots,1\}$ and using $\lim_{G\to\infty} c_{k,G-1} = \lim_{G\to\infty} c_{k,G-2}$ (the limit $\lim_{g\to\infty} c_{k,g}$ exists by boundedness and monotonicity of $c_{k,g}$ from Proposition 6.1). Next, it can be inductively shown using (6.14) that $\lim_{G\to\infty} b_k = 0$ for all k, by iterating over k in the order $\{n,\ldots,1\}$. Finally, it can be inductively shown using (6.13) that $\lim_{G\to\infty} a_k = 0$ for all k, by iterating over k in the order $\{n,\ldots,1\}$, which completes the proof. ■

Lemma 6.2 *For given $n \in \mathbb{N}$, $\lambda, \lambda_E, \lambda_R, \lambda_U \in \mathbb{R}^{++}$, e_1 is an increasing function of G.*

Proof: Similar to Lemma 6.1, for $k = n$, (6.20) gives

$$e_n = \frac{\lambda_E + d_{n,G}\lambda_U}{\lambda_U + \lambda_R}. \tag{6.36}$$

We know from Proposition 6.2 that $d_{n,G}$ increases with G, and therefore, e_n increases with G. Next, we assume e_{k+1} is an increasing function of G to inductively argue that e_k is also an increasing function of G. From (6.20), we see that the denominator does not depend on G and the last two terms of the numerator are increasing functions of G due to Proposition 6.2 and the assumption under the induction for $(k+1)$. Therefore, e_1 increases with G. ■

$G=0$ implies that reliability status becomes important only when the version ages of a reliable packet and an unreliable packet are the same, i.e., it has no impact on the version age of user nodes. Hence, for purposes of calculating e_1, we can just assume a total source to network update rate of $\lambda_R+\lambda_U$, which modifies (6.20) into

$$e_k = \frac{\lambda_E + e_{k+1}\frac{k(n-k)\lambda}{n-1}}{\frac{k(\lambda_U+\lambda_R)}{n} + \frac{k(n-k)\lambda}{n-1}}. \tag{6.37}$$

On the other hand, when $G \to \infty$, packets from the unreliable source are rejected by the network, such that (6.20) turns into

$$e_k = \frac{\lambda_E + e_{k+1}\frac{k(n-k)\lambda}{n-1}}{\frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}. \tag{6.38}$$

Note that network nodes desire fresh reliable packets, or in mathematical terms, low $x_1 = e_1$ and low $F = s_1 = a_1$. We note from Lemma 6.1 and Lemma 6.2 that increasing G leads to decrease in a_1, which is desirable, but an increase in e_1, which is undesirable. Therefore, there is a trade-off between x_1 and F induced by G. We next study the dependency of $F = a_1$ and $x_1 = e_1$ on network parameters λ_E, λ_U, λ_R, λ and n.

Proposition 6.4 *For any $k \in \{1, \ldots, n-1\}$, $g \in \{1, \ldots, G-1\}$, we have $c_{k,g} \leq c_{k+1,g}$.*

Proof: $R(A_k) \subseteq R(A_{k+1})$ in (6.11), as the extra node in A_{k+1} need not necessarily have a reliable packet. Consequently, $X_{R(A_{k+1})} \leq X_{R(A_k)}$, for an extended feasible region provides a lower minimum. Hence, $\mathbb{1}_{\{X_{R(A_k)} \leq g\}} \leq \mathbb{1}_{\{X_{R(A_{k+1})} \leq g\}}$, and upon taking expectations on both sides, $c_{k,g} \leq c_{k+1,g}$. ■

Proposition 6.5 *For any $k \in \{1, \dots, n-1\}$, $g \in \{1, \dots, G-1\}$, we have $b_k \geq b_{k+1}$.*

Proof: $S_{A_k \cup \{U\}}$ will be 1, if the set A_k does not have a reliable packet with version age upto G, in which case the packet at the unreliable source U becomes the best packet for the set $A_k \cup \{U\}$. Since $A_k \cup \{U\} \subseteq A_{k+1} \cup \{U\}$, the set $A_{k+1} \cup \{U\}$ is more likely to have a reliable packet of version age upto G due to an extended feasible region. Consequently, $S_{A_{k+1} \cup \{U\}}$ is more likely to be zero, i.e., $S_{A_{k+1} \cup \{U\}} \leq S_{A_k \cup \{U\}}$, consequently $b_{k+1} \leq b_k$. ■

Proposition 6.6 *For any $k \in \{1, \dots, n-1\}$, $g \in \{1, \dots, G-1\}$, we have $a_k \geq a_{k+1}$.*

Proof: Rearranging (6.8) gives

$$(a_k - a_{k+1}) \frac{k(n-k)\lambda}{n-1} = (b_k - a_k) \frac{k\lambda_U}{n} + (0 - a_k) \frac{k\lambda_R}{n} \tag{6.39}$$

$$= (b_k \lambda_U - a_k(\lambda_U + \lambda_R)) \frac{k}{n}. \tag{6.40}$$

To show $a_k - a_{k+1} \geq 0$ on the left side of (6.40), we claim $a_k \leq \frac{\lambda_U}{\lambda_U + \lambda_R} b_k$, which we prove through induction. For the base case $k = n$, we see from (6.13) that $a_n = \frac{b_n \lambda_U}{\lambda_U + \lambda_R}$. Next, we assume that the claim holds for $k+1$, i.e., $a_{k+1} \leq \frac{\lambda_U}{\lambda_U + \lambda_R} b_{k+1}$, and prove that it holds for k as well. Using (6.13), Proposition 6.5 and the assumption under induction, we get

$$a_k = \frac{b_k \frac{k\lambda_U}{n} + a_{k+1}\frac{k(n-k)\lambda}{n-1}}{\frac{k\lambda_U}{n} + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}} \tag{6.41}$$

$$\leq \frac{b_k \frac{k\lambda_U}{n} + \left(\frac{\lambda_U}{\lambda_U+\lambda_R} b_{k+1}\right)\frac{k(n-k)\lambda}{n-1}}{\frac{k\lambda_U}{n} + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}} \tag{6.42}$$

$$\leq \frac{b_k \frac{k\lambda_U}{n} + \left(\frac{\lambda_U}{\lambda_U+\lambda_R} b_{k}\right)\frac{k(n-k)\lambda}{n-1}}{\frac{k\lambda_U}{n} + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}} \tag{6.43}$$

$$= \frac{\lambda_U}{\lambda_U + \lambda_R} b_k, \tag{6.44}$$

completing the proof. ■

F is lowest with network size $n = 1$, i.e., information is most reliable at user nodes when they do not gossip with other user nodes, and thereby, end up individually being a part of single node gossip networks. To prove this, we attach superscripts in variables of (6.13), (6.14), (6.15) to indicate the network size n, resulting in $a_k^{(n)}$, $n_k^{(n)}$ and $c_{k,g}^{(n)}$. From (6.13), (6.14), (6.15), $a_n^{(n)}$ can be computed using the following equations and can be seen to be independent of n,

$$a_n^{(n)} = b_n^{(n)} \frac{\lambda_U}{\lambda_U + \lambda_R}, \tag{6.45}$$

$$b_n^{(n)} = (1 - c_{n,G-1}^{(n)}) \frac{\lambda_E}{\lambda_E + \lambda_R}, \tag{6.46}$$

$$c_{n,g}^{(n)} = c_{n,g-1}^{(n)} \frac{\lambda_E}{\lambda_E + \lambda_R} + \frac{\lambda_R}{\lambda_E + \lambda_R}, \tag{6.47}$$

with $c_{n,0}^{(n)} = \frac{\lambda_R}{\lambda_E+\lambda_R}$. (6.47) has to be iterated over $g = 0, \dots, G-1$, and therefore, the value of $c_{n,G-1}^{(n)}$, and consequently of $b_n^{(n)}$ and $a_n^{(n)}$ from (6.46) and (6.45), is independent of n. The latter implies that $a_1^{(1)} = a_n^{(n)} \leq a_1^{(n)}$, where the inequality results from Lemma 6.6. In Section 6.4, we observe from simulations that $F = a_1^{(n)}$ is an increasing function of n, i.e., unreliable information in the network increases as the network size grows. However, as we show in Lemma 6.3, reliable packets never get weeded out completely from the network even with very large network sizes, as $a_1^{(n)}$ remains strictly less than 1. Since direct fresh updates from the reliable source are always accepted at receiving nodes, $\lambda_R > 0$ ensures the network continues to have access to fresh reliable packets.

Lemma 6.3 *For given $G \in \mathbb{N} \cup \{0\}$, $\lambda, \lambda_E, \lambda_R, \lambda_U \in \mathbb{R}^+$, $\lim_{n\to\infty} a_1^{(n)} < 1$.*

Proof: The proof directly follows from (6.44), resulting in

$$a_1^{(n)} \leq \frac{\lambda_U}{\lambda_U + \lambda_R} b_1^{(n)} < 1, \tag{6.48}$$

where $b_1^{(n)} \leq 1$ by definition. ■

Next, we prove some structural results that allow us to investigate the dependence of F on gossiping rate λ.

Proposition 6.7 *$c_{k,g}$ is an increasing function of λ.*

Proof: Intuitively, since $c_{k,g} = \lim_{t\to\infty} \mathbb{E}[\mathbb{1}_{\{X_{R(A_k)}(t) \leq g\}}]$, with higher transmission rates in the network, nodes in the set A_k are more likely to receive the desirable reliable packets with version age less than g, and therefore, $\mathbb{1}_{\{X_{R(A_k)}(t) \leq g\}}$ is more likely to be 1 and $c_{k,g}$ is expected to increase. We next prove this intuition mathematically.

For gossiping rate λ, $c_{k,g}$ is defined by (6.15), and let $\tilde{c}_{k,g}$ be the analogous variable corresponding to gossiping rate $\lambda + \tilde{\lambda}$, where $\tilde{\lambda} > 0$, such that we claim $\tilde{c}_{k,g} \geq c_{k,g}$. From (6.15),

$$\tilde{c}_{k,g} = \frac{\tilde{c}_{k,g-1}\lambda_E + \frac{k\lambda_R}{n} + \tilde{c}_{k+1,g}\frac{k(n-k)(\lambda+\tilde{\lambda})}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)(\lambda+\tilde{\lambda})}{n-1}}. \tag{6.49}$$

After cross multiplying and subtracting $\tilde{c}_{k,g}\frac{k(n-k)\tilde{\lambda}}{n-1}$ on both sides, we get

$$\tilde{c}_{k,g} = \frac{\tilde{c}_{k,g-1}\lambda_E + \frac{k\lambda_R}{n} + \tilde{c}_{k+1,g}\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}} + \frac{\left(\tilde{c}_{k+1,g} - \tilde{c}_{k,g}\right)\frac{k(n-k)\tilde{\lambda}}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}. \tag{6.50}$$

We prove $\tilde{c}_{k,g} \geq c_{k,g}$ by using induction on variable g. We first show that when $g = 0$, $\tilde{c}_{k,g} \geq c_{k,g}$ holds for all k. Then, we assume that it holds for general $g-1 \in \{0, \ldots, G-2\}$ and prove for g.

Base case of $g = 0$: To prove the proposition for the base case of $g = 0$, i.e., $\tilde{c}_{k,0} \geq c_{k,0}$ for all k, we get from (6.50),

$$c_{k,0} = \frac{\frac{k\lambda_R}{n} + c_{k+1,0}\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}, \tag{6.51}$$

and

$$\tilde{c}_{k,0} = \frac{\frac{k\lambda_R}{n} + \tilde{c}_{k+1,0}\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}} + \frac{\left(\tilde{c}_{k+1,0} - \tilde{c}_{k,0}\right)\frac{k(n-k)\tilde{\lambda}}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}. \tag{6.52}$$

We now use a second induction on variable k. For the base case of $k = n$, we have $\tilde{c}_{n,0} = \frac{\lambda_R}{\lambda_E+\lambda_R} = c_{n,0}$. Next, we assume that $\tilde{c}_{k+1,0} \geq c_{k+1,0}$ holds for some

$k+1 \in \{n, \ldots, 2\}$ and inductively show that it holds for k as well. The latter can be seen by comparing (6.51) and (6.52), since in (6.52), $\tilde{c}_{k+1,0} \geq c_{k+1,0}$ due to the assumption under the induction on k, and $(\tilde{c}_{k+1,0} - \tilde{c}_{k,0}) \geq 0$ from Proposition 6.4.

Hence, for $g = 0$, $\tilde{c}_{k,g} \geq c_{k,g}$ holds for every k. Next, we assume that the claim holds for a general $g-1 \in \{0, \ldots, G-2\}$. To prove that it holds for the case of g as well, we apply a second induction on the variable k. As before, it can be trivially shown for $k = n$, that $\tilde{c}_{n,g} \geq c_{n,g}$ by comparing (6.15) and (6.49) and using $\tilde{c}_{n,g-1} \geq c_{n,g-1}$ due to the assumption under induction on $g-1$. Next, we employ a second induction on variable k and assume $\tilde{c}_{k+1,g} \geq c_{k+1,g}$, i.e., the claim holds for a general $k+1$. We can then inductively show that the claim holds for k as well, i.e., $\tilde{c}_{k,g} \geq c_{k,g}$, by comparing (6.15) and (6.50), since $\tilde{c}_{k,g-1} \geq c_{k,g-1}$ due to the assumption under the induction on g, $\tilde{c}_{k+1,g} \geq c_{k+1,g}$ due to the assumption under the second induction on k, and $(\tilde{c}_{k+1,g} - \tilde{c}_{k,g}) > 0$ due to Proposition (6.4). ■

Proposition 6.8 *b_k is a decreasing function of λ.*

Proof: For gossiping rate λ, b_k is defined by (6.14), and let $\tilde{b}_k$ denote the analogous variable corresponding to gossiping rate $\lambda + \tilde{\lambda}$. We claim $\tilde{b}_k \leq b_k$ for all k, which we prove through induction on variable k. Therefore, we assume that the claim holds for a general $k+1$ and show that it holds for k as well. Similar to (6.49) and (6.50), simple cross multiplication, followed by subtraction of $\tilde{b}_k \frac{k(n-k)\tilde{\lambda}}{n-1}$ from both sides, gives

$$\tilde{b}_k = \frac{(1 - \tilde{c}_{k,G-1})\lambda_E + \tilde{b}_{k+1}\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}} + \frac{(\tilde{b}_{k+1} - \tilde{b}_k)\frac{k(n-k)\tilde{\lambda}}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}. \tag{6.53}$$

Comparing (6.53) with (6.14), we can verify $\tilde{b}_k \leq b_k$, since in their corresponding right-hand side expressions, $\tilde{b}_{k+1} \leq b_{k+1}$ due to the assumption under induction for $k+1$ case, $(\tilde{b}_{k+1} - \tilde{b}_k) \leq 0$ due to Proposition 6.5, and $\tilde{c}_{k,G-1} \geq c_{k,G-1}$ due to Proposition 6.7. Finally, for the base case of $k = n$, we have $\tilde{b}_n = \frac{(1-\tilde{c}_{n,G-1})\lambda_E}{\lambda_E+\lambda_R} \leq \frac{(1-c_{n,G-1})\lambda_E}{\lambda_E+\lambda_R} = b_n$, since $\tilde{c}_{n,G-1} \geq c_{n,G-1}$ from the proof of Proposition 6.7. ■

Lemma 6.4 *For given $n \in \mathbb{N}$, $G \in \mathbb{N} \cup \{0\}$, $\lambda, \lambda_E, \lambda_U \in \mathbb{R}$, a_1 is a decreasing function of λ. Further, the limiting values of a_1 are* $\lim_{\lambda\to\infty} a_1 = \left(\frac{\lambda_E}{\lambda_E+\lambda_R}\right)^{G+1} \frac{\lambda_U}{\lambda_U+\lambda_R}$ *and* $\lim_{\lambda\to 0} a_1 = \left(\frac{\lambda_E}{\lambda_E+\frac{\lambda_R}{n}}\right)^{G+1} \frac{\lambda_U}{\lambda_U+\lambda_R}$.

Proof: We prove that a_k is a decreasing function of λ for all k, similar to the proof of Proposition 6.8. For gossiping rate λ, a_k is defined by (6.13), and let $\tilde{a}_k$ denote the analogous variable corresponding to gossiping rate $\lambda + \tilde{\lambda}$. We claim $\tilde{a}_k \leq a_k$ for all k through induction on variable k. Therefore, we assume that the claim holds for a general $k+1$ and show that it holds for k as well. Similar to (6.49) and (6.50), simple cross multiplication, followed by subtraction of $\tilde{a}_k \frac{k(n-k)\tilde{\lambda}}{n-1}$ from both sides, gives

$$\tilde{a}_k = \frac{\tilde{b}_k \frac{k\lambda_U}{n} + \tilde{a}_{k+1}\frac{k(n-k)\lambda}{n-1}}{\frac{k\lambda_U}{n} + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}} + \frac{(\tilde{a}_{k+1} - \tilde{a}_k)\frac{k(n-k)\lambda}{n-1}}{\frac{k\lambda_U}{n} + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}. \tag{6.54}$$

Comparing (6.54) with (6.13), we can verify $\tilde{a}_k \leq a_k$, since in their corresponding right-hand side expressions, $\tilde{b}_k \leq b_k$ from Proposition 6.8, $\tilde{a}_{k+1} \leq a_{k+1}$ due to the assumption under the induction for $k+1$ case, and $(\tilde{a}_{k+1} - \tilde{a}_k) \leq 0$ due to Proposition 6.6. For the base case of $k = n$, we have $\tilde{a}_n = \frac{\tilde{b}_n\lambda_U}{\lambda_U+\lambda_R} \leq \frac{b_n\lambda_U}{\lambda_U+\lambda_R} = a_n$, since $\tilde{b}_n \leq b_n$ from proof of Proposition 6.8. Using $k = 1$, we finally get that a_1 is a decreasing function of λ.

Substituting $\lambda \to \infty$ in (6.13), (6.14) and (6.15) gives

$$\begin{aligned} a_1 = \ldots = a_n = b_n &\frac{\lambda_U}{\lambda_U + \lambda_R} && (6.55)\\ &= (1 - c_{n,G-1})\frac{\lambda_E}{\lambda_E + \lambda_R}\frac{\lambda_U}{\lambda_U + \lambda_R} && (6.56)\\ &= \left(\frac{\lambda_E}{\lambda_E + \lambda_R}\right)^{G+1}\frac{\lambda_U}{\lambda_U + \lambda_R} > 0, && (6.57)\end{aligned}$$

where $(1 - c_{n,G-1})$ is computed by iterating over g in (6.58),

$$(1 - c_{n,g}) = (1 - c_{n,g-1})\frac{\lambda_E}{\lambda_E + \lambda_R}. \tag{6.58}$$

One might expect that a high gossiping rate would result in the fast dissemination of fresh, reliable information in the network and push a_1 to zero. However, as seen in (6.57), a_1 in this case would still depend on other factors, such as how quickly the reliable source can start sending packets soon after the event is updated and whether the unreliable source is quicker in this regard. Next, substituting $\lambda = 0$ in (6.13), (6.14) and (6.15) gives

$$\begin{aligned} a_1 = b_1 &\frac{\lambda_U}{\lambda_U + \lambda_R} && (6.59)\\ &= (1 - c_{1,G-1})\frac{\lambda_E}{\lambda_E + \frac{\lambda_R}{n}}\frac{\lambda_U}{\lambda_U + \lambda_R} && (6.60)\\ &= \left(\frac{\lambda_E}{\lambda_E + \frac{\lambda_R}{n}}\right)^{G+1}\frac{\lambda_U}{\lambda_U + \lambda_R} < 1, && (6.61)\end{aligned}$$

where $(1 - c_{1,G-1})$ is computed by iterating over g in (6.62),

$$1 - c_{1,g} = (1 - c_{1,g-1}) \frac{\lambda_E}{\lambda_E + \frac{\lambda_R}{n}}. \tag{6.62}$$

Notice how $\lim_{\lambda \to \infty} a_1$ and $\lim_{\lambda \to 0} a_1$ only differ due to the presence of variable n in the denominator, highlighting the importance of gossiping for propagating reliable information in a large network. When $n = 1$, both the values are the same, since there is no gossiping in a single-node network. ■

Next, we investigate the dependence of F on the reliable source update rate λ_R.

Proposition 6.9 *$c_{k,g}$ is an increasing function of λ_R.*

Proof: The proof is similar to Proposition 6.7. With reliable source update rate λ_R, $c_{k,g}$ is defined by (6.15), and let $\tilde{c}_{k,g}$ be the analogous variable corresponding to reliable source update rate $\lambda_R + \tilde{\lambda}_R$, $\tilde{\lambda}_R > 0$. From (6.15),

$$\tilde{c}_{k,g} = \frac{\tilde{c}_{k,g-1}\lambda_E + \frac{k(\lambda_R + \tilde{\lambda}_R)}{n} + \tilde{c}_{k+1,g}\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \frac{k(\lambda_R + \tilde{\lambda}_R)}{n} + \frac{k(n-k)\lambda}{n-1}}, \tag{6.63}$$

which, upon simple cross multiplication, followed by subtraction of $\tilde{c}_{k,g}\frac{k\tilde{\lambda}_R}{n}$ from both sides, gives

$$\tilde{c}_{k,g} = \frac{\tilde{c}_{k,g-1}\lambda_E + \frac{k\lambda_R}{n} + \tilde{c}_{k+1,g}\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}} + (1 - \tilde{c}_{k,g})\frac{k\tilde{\lambda}_R}{n}. \tag{6.64}$$

We claim $\tilde{c}_{k,g} \geq c_{k,g}$, which we prove through double induction on variables g

and k, as in Proposition 6.7. We assume that the claim holds for some $g-1$, i.e., $\tilde{c}_{k,g-1} \geq c_{k,g-1}$ for all k, and prove that it holds for g as well, i.e., the first induction is on variable g.

To prove the claim for g, i.e., $\tilde{c}_{k,g} \geq c_{k,g}$ for all k, we apply a second induction on variable k. Therefore, we assume that the claim holds for $k+1$ and show that it holds for k as well. The latter is obvious from comparing (6.64) with (6.15), where $\tilde{c}_{k,g-1} \geq c_{k,g-1}$ due to the assumption under the first induction for the case of $g-1$, $\tilde{c}_{k+1,g} \geq c_{k+1,g}$ due to the assumption under the second induction for case of $k+1$, and $1 \geq \tilde{c}_{k,g}$ from the definition of $\tilde{c}_{k,g}$. The base case of $k=n$ can be similarly proved by comparing terms of (6.64) with (6.15), which proves the claim for the case of g as well, i.e., $\tilde{c}_{k,g} \geq c_{k,g}$ for all k. Likewise, the base case of $g=0$ under the first induction can be proved by writing $\tilde{c}_{k,0}$ analogous to (6.51), rearranging its terms similar to (6.64) and comparing them. ■

Lemma 6.5 *For given $n \in \mathbb{N}$, $G \in \mathbb{N} \cup \{0\}$, $\lambda, \lambda_E, \lambda_U \in \mathbb{R}$, a_1 is a decreasing function of λ_R, with $\lim_{\lambda_R \to \infty} a_1 = 0$ and $\lim_{\lambda_R \to 0} a_1 = 1$.*

Proof: The proof involves showing that b_k and consequently a_k, for all k, are decreasing functions of λ_R, using proof by induction. Assuming b_{k+1} is a decreasing function of λ_R, we can see from (6.14) that the denominator $\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}$ increases with λ_R, $(1-c_{k,G-1})\lambda_E$ in the numerator increases with λ_R by Proposition 6.9 and $b_{k+1}\frac{k(n-k)\lambda}{n-1}$ decreases with λ_R due to the assumption under the induction. Consequently, b_k is also a decreasing function of λ_R. The base case of b_n can be similarly shown to decrease with λ_R.

Likewise, we can show through induction that a_k decreases with λ_R. From (6.13), the denominator $\frac{k\lambda_U}{n} + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}$ decreases with λ_R, $b_k \frac{k\lambda_U}{n}$ decreases with λ as proved above, and $a_{k+1}\frac{k(n-k)\lambda}{n-1}$ decreases with λ_R due to the assumption under the induction on a_{k+1}. The base case of a_n can be similarly shown to decrease with λ_R.

Substituting $\lambda_R \to \infty$ in (6.13) gives $a_1 = 0$. On the other hand, $\lambda_R = 0$ implies that no reliable packets are sent to the network and the network only has unreliable packets available. Mathematically, substituting $\lambda_R = 0$ in (6.13), (6.14) and (6.15), one can inductively verify that $c_{k,g} = 0$, $b_k = 1$ and $a_k = 1$, implying $\lim_{\lambda_R \to 0} a_1 = 1$. ■

We next investigate the dependence of F on the unreliable source update rate λ_U. We observed in Lemma 6.5 how a large λ_R guarantees that all nodes only possess reliable packets at all times, i.e., $F = 0$. However, we next observe in Lemma 6.6 how a large λ_U does not guarantee similar prevalence of unreliable packets in the network.

Lemma 6.6 *For given $n \in \mathbb{N}$, $G \in \mathbb{N} \cup \{0\}$, $\lambda, \lambda_E, \lambda_U \in \mathbb{R}$, a_1 is an increasing function of λ_U, with $\lim_{\lambda_U \to \infty} a_1 < 1$.*

Proof: Note from (6.14) and (6.15) that b_k and $c_{k,g}$ are independent of λ_U. For unreliable source update rate λ_U, a_k is defined by (6.13), and let $\tilde{a}_k$ be the analogous variable corresponding to the unreliable source update rate $\lambda_U + \tilde{\lambda}_U$. From (6.13),

$$\tilde{a}_k = \frac{b_k \frac{k(\lambda_U + \tilde{\lambda}_U)}{n} + \tilde{a}_{k+1}\frac{k(n-k)\lambda}{n-1}}{\frac{k(\lambda_U + \tilde{\lambda}_U)}{n} + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}, \tag{6.65}$$

which, upon simple cross multiplication, followed by subtraction of $\tilde{a}_k \frac{k\tilde{\lambda}_U}{n}$ from both sides, gives

$$\tilde{a}_k = \frac{b_k \frac{k\lambda_U}{n} + \tilde{a}_{k+1}\frac{k(n-k)\lambda}{n-1}}{\frac{k\lambda_U}{n} + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}} + \frac{(b_k - \tilde{a}_k)\frac{k\tilde{\lambda}_U}{n}}{\frac{k\lambda_U}{n} + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}. \tag{6.66}$$

Note that $b_k \geq \tilde{a}_k$ by definition, since $S_{A_k \cup \{U\}} \geq S_{A_k}$. The latter is true because $S_{A_k \cup \{U\}}$ will be zero only if the set A_k possesses a reliable packet of version age upto G, in which case S_{A_k} will also be zero. Hence, $S_{A_k \cup \{U\}} = 0$ implies $S_{A_k} = 0$, however $S_{A_k \cup \{U\}} = 1$ need not imply $S_{A_k} = 1$.

Employing proof by induction, we assume $\tilde{a}_{k+1} \geq a_{k+1}$, which allows us to readily see $\tilde{a}_k \geq a_k$ by comparing (6.66) and (6.13). Here, $b_k - \tilde{a}_k \geq 0$ as proved above, b_k is independent of λ_U and $\tilde{a}_{k+1} \geq a_{k+1}$ by the assumption under the induction. The base case of $a_n = \frac{b_n \lambda_U}{\lambda_U + \lambda_R}$ can be shown to increase with λ_U by taking the first derivative with respect to λ_U.

From (6.44), $a_1 \leq b_1 < 1$, where b_1 is independent of λ_U and is strictly less than 1, since $b_1 = 1$ is not feasible in (6.14) as long as $\lambda_R > 0$. ■

We next investigate the dependence of F on the event update rate λ_E.

Proposition 6.10 *$c_{k,g}$ is a decreasing function of λ_E.*

Proof: For event update rate λ_E, $c_{k,g}$ is defined by (6.15), and let $\tilde{c}_{k,g}$ be the analogous variable corresponding to event update rate $\lambda_E + \tilde{\lambda}_E$. By (6.15),

$$\tilde{c}_{k,g} = \frac{\tilde{c}_{k,g-1}(\lambda_E + \tilde{\lambda}_E) + \frac{k\lambda_R}{n} + \tilde{c}_{k+1,g}\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \tilde{\lambda}_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}, \tag{6.67}$$

which, upon simple cross multiplication, followed by subtraction of $\tilde{c}_{k,g}\tilde{\lambda}_E$ from both sides, gives

$$\tilde{c}_{k,g} = \frac{\tilde{c}_{k,g-1}\lambda_E + \frac{k\lambda_R}{n} + \tilde{c}_{k+1,g}\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}} + \frac{(\tilde{c}_{k,g-1} - \tilde{c}_{k,g})\tilde{\lambda}_E + \frac{k\lambda_R}{n}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}. \tag{6.68}$$

The proof can be shown by employing a double induction on variables g and k, similar to Proposition 6.9. Comparing (6.68) with (6.15), $\tilde{c}_{k,g-1} \leq c_{k,g-1}$ due to the assumption under first induction for $g-1$ case, $\tilde{c}_{k+1,g} \leq c_{k+1,g}$ due to the assumption under the second induction for $k+1$ case, and $\tilde{c}_{k,g-1} - \tilde{c}_{k,g} \leq 0$ by Proposition 6.1. ■

Proposition 6.11 *b_k is an increasing function of λ_E.*

Proof: We provide an inductive proof similar to that of Proposition 6.8. Analogous to b_k, we define $\tilde{b}_k$ for event update rate $\lambda_E + \tilde{\lambda}_E$ from (6.14) as follows,

$$\tilde{b}_k = \frac{(1 - \tilde{c}_{k,G-1})(\lambda_E + \tilde{\lambda}_E) + \tilde{b}_{k+1}\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \tilde{\lambda}_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}, \tag{6.69}$$

which, upon simple cross multiplication, followed by subtraction of $\tilde{b}_k\tilde{\lambda}_E$ from both sides, gives

$$\tilde{b}_k = \frac{(1 - \tilde{c}_{k,G-1})\lambda_E + \tilde{b}_{k+1}\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}} + \frac{(1 - \tilde{c}_{k,G-1} - \tilde{b}_k)\tilde{\lambda}_E}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}. \tag{6.70}$$

Note that $S_{A_k \cup \{U\}} = 1 - \mathbb{1}_{\{X_{R(A_k)} \leq G\}}$, since only $\mathbb{1}_{\{X_{R(A_k)} \leq G\}} = 1$ will imply that the best node in $A_k \cup \{U\}$ will lie in the set $R(A_k)$. Therefore, $b_k = 1 - c_{k,G}$. Since $c_{k,g}$ from Proposition 6.1 increases with g, we have, $b_k = 1 - c_{k,G} \leq 1 - c_{k,G-1}$.

Now, comparing (6.70) with (6.14), we can see that $\tilde{b}_k \geq b_k$, since $1-\tilde{c}_{k,G-1}-\tilde{b}_k \geq 0$ as proved above, $\tilde{b}_{k+1} \geq b_{k+1}$ by the assumption under the induction for $k+1$ case, and $(1-\tilde{c}_{k,G-1}) \geq (1-c_{k,G-1})$ by Proposition 6.10. ■

Lemma 6.7 *For given $n \in \mathbb{N}$, $G \in \mathbb{N} \cup \{0\}$, $\lambda, \lambda_R, \lambda_U \in \mathbb{R}$, a_1 is an increasing function of λ_E, with* $\lim_{\lambda_E \to \infty} a_1 < 1$.

Proof: We claim that a_k, for all k, is an increasing function of λ_E. We prove the claim by induction, i.e., we assume a_{k+1} increases with λ_E and simply observe in (6.13) that both a_{k+1} and b_k, by Proposition 6.11, are increasing functions of λ_E, implying that the claim holds for a_k as well. Substituting $k=1$ gives the desired result. Further, from (6.44), $a_1 \leq \frac{\lambda_U}{\lambda_U+\lambda_R} b_1 < 1$, for all values of λ_E, where $b_1 \leq 1$ by definition. ■

Next, we study the dependency of the expected version age $x_1 = e_1$ on network parameters λ_E, λ_U, λ_R, λ and n.

Proposition 6.12 $C_{k,g} = \frac{c_{k,g}\frac{k\lambda_U}{n}}{\frac{k\lambda_U}{n}+\frac{k\lambda_R}{n}+\frac{k(n-k)\lambda}{n-1}}$ *and* $D_{k,g} = \frac{d_{k,g}\frac{k\lambda_U}{n}}{\frac{k\lambda_U}{n}+\frac{k\lambda_R}{n}+\frac{k(n-k)\lambda}{n-1}}$ *are decreasing functions of λ.*

Proof: This can be proved through double induction on variables k and g, as employed in proof of Proposition 6.10. Multiplying both sides of (6.22) by $\frac{\frac{k\lambda_U}{n}}{\frac{k\lambda_U}{n}+\frac{k\lambda_R}{n}+\frac{k(n-k)\lambda}{n-1}}$, we get

$$C_{k,g} = \frac{\frac{k\lambda_U}{n}\frac{k\lambda_R}{n}}{(\frac{k\lambda_U}{n}+\frac{k\lambda_R}{n}+\frac{k(n-k)\lambda}{n-1})(\lambda_E+\frac{k\lambda_R}{n}+\frac{k(n-k)\lambda}{n-1})} + \frac{C_{k,g-1}\lambda_E + C_{k+1,g}\frac{k(n-k)\lambda}{n-1}}{\lambda_E+\frac{k\lambda_R}{n}+\frac{k(n-k)\lambda}{n-1}}. \quad (6.71)$$

Here $C_{k,g-1}$ and $C_{k+1,g}$ are decreasing functions of λ due the assumptions under double induction for the case of $g-1$ and $k+1$, similar to proof of Proposition 6.10, and all the denominator terms are an increasing function of λ, implying that $C_{k,g}$ is also a decreasing function of λ. Likewise, we can multiply both sides of (6.21) by $\frac{\frac{k\lambda_U}{n}}{\frac{k\lambda_U}{n}+\frac{k\lambda_R}{n}+\frac{k(n-k)\lambda}{n-1}}$ to get

$$D_{k,g} = \frac{(D_{k,g-1} + C_{k,g-1})\lambda_E + D_{k+1,g}\frac{k(n-k)\lambda}{n-1}}{\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}. \tag{6.72}$$

Again, with a double induction on variables k and g, it can be shown that $D_{k,g}$ decreases with λ, since both $D_{k,g-1}$ and $D_{k+1,g}$ decrease with λ due to the assumptions under double induction for the case of $g-1$ and $k+1$, and $C_{k,g}$ has already been proved to decrease with λ. The denominator term $\lambda_E + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}$ is further an increasing function of λ. ■

Lemma 6.8 *For given $n \in \mathbb{N}$, $G \in \mathbb{N} \cup \{0\}$, $\lambda_E, \lambda_R, \lambda_U \in \mathbb{R}$, x_1 is a decreasing function of λ. Further,* $\lim_{\lambda\to\infty} x_1 > 0$.

Proof: Note that in (6.20), $\frac{d_{k,G}\frac{k\lambda_U}{n}}{\frac{k\lambda_U}{n}+\frac{k\lambda_R}{n}+\frac{k(n-k)\lambda}{n-1}} = D_{k,G}$ is a decreasing function of λ from Proposition 6.12. We can show that e_k, for all k, decreases with λ through a proof by induction. Assuming e_{k+1} decreases with λ, using (6.20), coupled with Proposition 6.12, we can argue that e_k also decreases with λ. To prove the second part, substituting $\lambda \to \infty$ in (6.20) gives,

$$e_1 = \ldots = e_n \geq \frac{\lambda_E}{\lambda_U + \lambda_R}. \tag{6.73}$$

Hence, a high inter-node gossiping rate does not imply all network nodes will be fresh, i.e., $x_1 = e_1$ does not go to zero. This is because, as the event keeps getting updated, the version age at network nodes keeps incrementing until either of the reliable or unreliable source, both of which have finite update rates, sends the newer versions to the network nodes. ■

In Section 6.4, we surprisingly observe from simulations that x_1 is not a decreasing function of λ_R. This can be ascribed to the willingness of the network nodes to switch to staler packets to have more reliable information at low values of λ_R. Substituting $\lambda_R = 0$ in (6.20), (6.21) and (6.22) yields $c_{k,g} = d_{k,g} = 0$, and consequently, e_k is iteratively solved through the following simple recursive equation,

$$e_k = \frac{\lambda_E + e_{k+1}\frac{k(n-k)\lambda}{n-1}}{\frac{k\lambda_U}{n} + \frac{k(n-k)\lambda}{n-1}}. \tag{6.74}$$

On the other hand, when λ_R is large, (6.22) results in $\lim_{\lambda_R\to\infty} c_{k,g} = 1$ and (6.21) results in $\lim_{\lambda_R\to\infty} d_{k,g} = 0$, and consequently, (6.20) gives

$$\lim_{\lambda_R\to\infty} e_k = 0. \tag{6.75}$$

Hence, a large reliable source update rate ensures that the reliable source can directly keep all network nodes updated with the freshest reliable packet at all times, causing them to have zero version age.

We next see, however, that a large λ_U does not analogously guarantee that age will decay to zero, due to the unique nature of the file exchange protocol discussed in Section 6.1, wherein the version age is readily traded for more reliable packets at network nodes.

Lemma 6.9 *For given $n \in \mathbb{N}$, $G \in \mathbb{N} \cup \{0\}$, $\lambda, \lambda_E, \lambda_R \in \mathbb{R}$, x_1 is a decreasing function of λ_U, with* $\lim_{\lambda_U \to \infty} x_1 > 0$.

Proof: Note that $c_{k,g}$ and $d_{k,g}$ are independent of λ_U from (6.21) and (6.22). Next, for unreliable source update rate λ_U, e_k is defined by (6.20), and let $\tilde{e}_k$ be the analogous variable corresponding to event update rate $\lambda_U + \tilde{\lambda}_U$. From (6.20),

$$\tilde{e}_k = \frac{\lambda_E + d_{k,G}\frac{k(\tilde{\lambda}_U+\lambda_U)}{n} + \tilde{e}_{k+1}\frac{k(n-k)\lambda}{n-1}}{\frac{k(\tilde{\lambda}_U+\lambda_U)}{n} + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}, \tag{6.76}$$

which, upon simple cross multiplication, followed by subtraction of $\tilde{e}_k\frac{k\tilde{\lambda}_U}{n}$ from both sides, gives

$$\tilde{e}_k = \frac{\lambda_E + d_{k,G}\frac{k\lambda_U}{n} + \tilde{e}_{k+1}\frac{k(n-k)\lambda}{n-1}}{\frac{k\lambda_U}{n} + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}} + \frac{(d_{k,G} - \tilde{e}_k)\frac{k\tilde{\lambda}_U}{n}}{\frac{k\lambda_U}{n} + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}. \tag{6.77}$$

We provide a proof by induction and assume that e_{k+1} is a decreasing function of λ_U. Though $d_{k,G}$ is independent of λ_U, note that $d_{k,G} \leq e_k$, irrespective of the unreliable source update rate, since $X_{A_k}\mathbb{1}_{\{X_{R(A_k)} \leq g\}} \leq X_{A_k}$. This implies that when unreliable source update rate is $\lambda_U + \tilde{\lambda}_U$, $d_{k,G} \leq \tilde{e}_k$ still holds. Comparing (6.20) and (6.77), we can see that $d_{k,G} \leq \tilde{e}_k$ and $\tilde{e}_{k+1} \leq e_{k+1}$ due to the assumption under induction, implying $\tilde{e}_k \leq e_k$. Hence e_k, for all k, is a decreasing function of λ_U.

When $\lambda_U = 0$, e_1 is simply computed by (6.38). On the other hand, with

$\lambda_U \to \infty$ in (6.38), $e_1 = d_{1,G}$, which is computed from (6.21) and (6.22), and is positive and independent of λ_U. ■

Lemma 6.10 *For given $n \in \mathbb{N}$, $G \in \mathbb{N} \cup \{0\}$, $\lambda, \lambda_R, \lambda_U \in \mathbb{R}$, x_1 scales as $O(\lambda_E)$.*

Proof: For large λ_E, we have from (6.20),

$$e_k = \frac{\lambda_E}{\frac{k\lambda_U}{n} + \frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}} = O(\lambda_E), \tag{6.78}$$

as $\lambda_E + d_{k,G}\frac{k\lambda_U}{n} + e_{k+1}\frac{k(n-k)\lambda}{n-1} \approx \lambda_E$. ■

Lemma 6.11 *For given $G \in \mathbb{N} \cup \{0\}$, $\lambda, \lambda_E, \lambda_R, \lambda_U \in \mathbb{R}$, x_1 scales as $\Theta(\log n)$.*

Proof: From Lemma 6.2, we know the expected age $x_1 = e_1$ is lowest with $G = 0$ and highest with $G = \infty$. These extreme cases can be used to derive a lower bound and an upper bound, respectively, to x_1, both of which scale as $\log n$.

When $G = \infty$, unreliable packets are never accepted at any network node, in which case x_1 is given by (6.38), i.e.,

$$e_k = \frac{\lambda_E + e_{k+1}\frac{k(n-k)\lambda}{n-1}}{\frac{k\lambda_R}{n} + \frac{k(n-k)\lambda}{n-1}}. \tag{6.79}$$

To derive an upper bound, we reduce the update rates λ and λ_R to $L = \min\{\lambda_R, \lambda\}$ in (6.79) and call the expected age of the resulting system model as $\tilde{e}_k$, i.e.,

$$\tilde{e}_k = \frac{\lambda_E + \tilde{e}_{k+1}\frac{k(n-k)L}{n-1}}{\frac{kL}{n} + \frac{k(n-k)L}{n-1}}. \tag{6.80}$$

We claim that $e_k \leq \tilde{e}_k$. If $\lambda_R > \lambda = L$, then replacing λ_R with L reduces the denominator magnitude of (6.79), and consequently, $\tilde{e}_k$ of (6.80) can be recursively shown to be greater than e_k for all k, beginning with $k = n$. On the other hand, if $L = \lambda_R \leq \lambda$, then $e_k \leq \tilde{e}_k$ follows from Lemma 6.8, since replacing λ with a smaller value L leads to increased age. Therefore, (6.80) can be used to derive an upper bound for e_1, and we know from [5, Thm. 2] that $\tilde{e}_k$ in (6.80) scales as $\log n$.

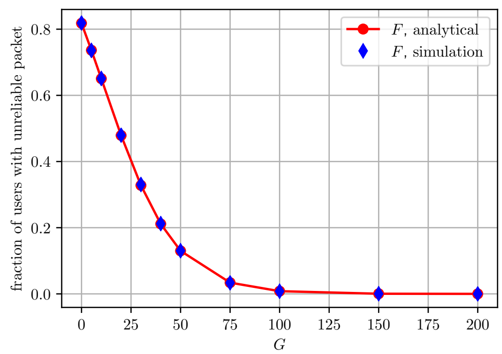
Likewise, for the case of $G = 0$, where e_1 is governed by (6.37), we increase the rate terms $\lambda_R + \lambda_U$ and λ to $M = \max\{\lambda_R + \lambda_U, \lambda\}$ and call the expected age of the resulting system model as $\bar{e}_k$, i.e.,

$$\bar{e}_k = \frac{\lambda_E + \bar{e}_{k+1}\frac{k(n-k)M}{n-1}}{\frac{kM}{n} + \frac{k(n-k)M}{n-1}}, \tag{6.81}$$

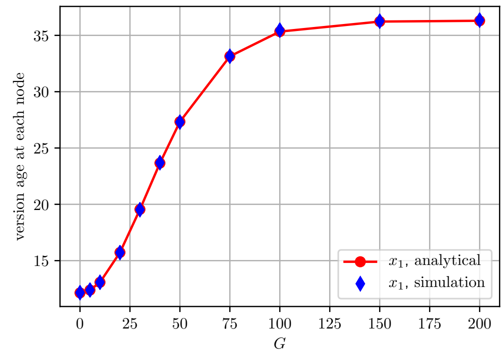
which can be used to derive a lower bound. Note that increasing rates in (6.37) is simply the reverse of reducing rates in (6.38), and $e_k \geq \bar{e}_k$ can be justified similarly as above. Next, $\bar{e}_k$, from [5, Thm. 2], also scales as $\log n$. ■

6.4 Numerical Results

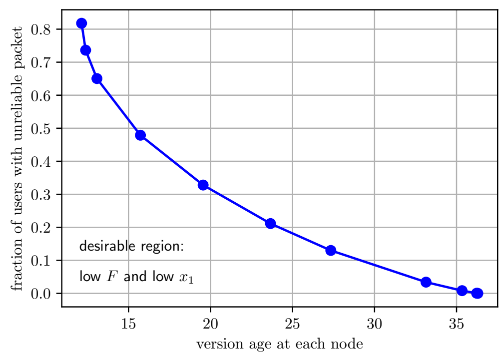
We first simulate a fully-connected network of $n = 50$ nodes with parameters $\lambda_E = 2$, $\lambda_U = 5$, $\lambda_R = 1$ and $\lambda = 0.1$, for up to a total time of 10^5, which we use as proxy for $t \to \infty$. We vary G and plot simulation points (blue dots) of F and x_1 on curves (red lines) obtained from equations (6.13), (6.14), (6.15), (6.20), (6.21) in Fig. 6.3(a) and Fig. 6.3(b), respectively. The real-time simulation points coincide with the iterative calculation of the derived equation curves, lending support to the



(a)



(b)

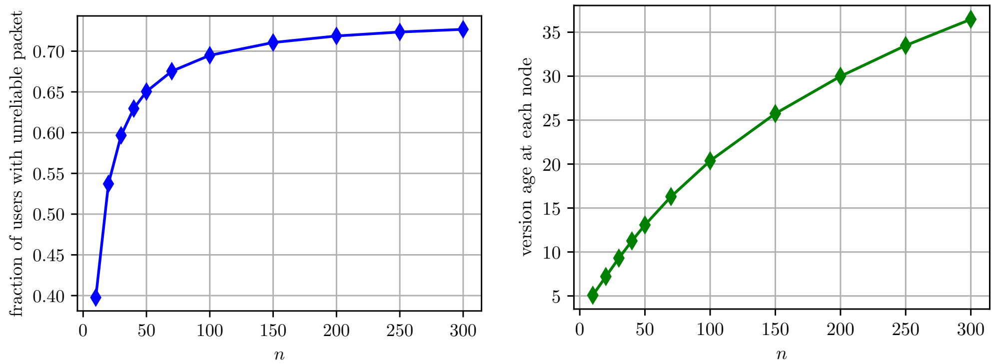


(c)

Figure 6.3: Analytical and simulation results as a function of G for (a) expected fraction of users with unreliable information F, (b) expected version age x_1, (c) trade-off between F and x_1 induced by G.

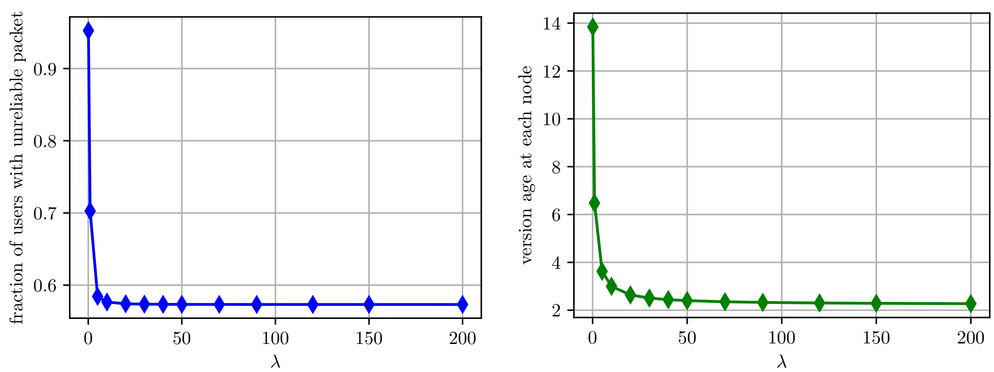
theoretical analysis. Fig. 6.3(a) shows that F decreases with G and converges to zero, as suggested by Lemma 6.1, and Fig. 6.3(b) shows that the version age at nodes increases with G, as suggested by Lemma 6.2. Fig. 6.3(c) shows the trade-off between F and x_1 induced by G, with low values of both variables being desirable.

We next plot F and x_1 as a function of the network parameters n, λ, λ_R, λ_U, and λ_E. Fig. 6.4(a) and Fig. 6.4(b) show the variation in F and x_1 as the network size n grows. Both F and x_1 grow with n, implying that larger network sizes have undesirable effects on the packet quality at nodes. As suggested in Lemma 6.11,



(a) (b)

Figure 6.4: (a) F and (b) x_1 as a function of network size n, with $\lambda_E = 2$, $\lambda_U = 5$, $\lambda_R = 1$, $\lambda = 0.1$ and $G = 10$.

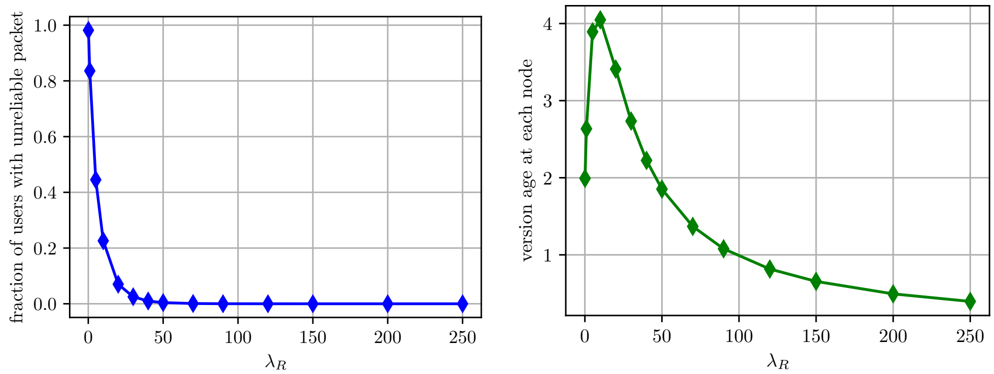


(a) (b)

Figure 6.5: F and (b) x_1 as a function of gossiping rate λ, with $\lambda_E = 2$, $\lambda_U = 5$, $\lambda_R = 0.1$, $n = 50$ and $G = 10$.

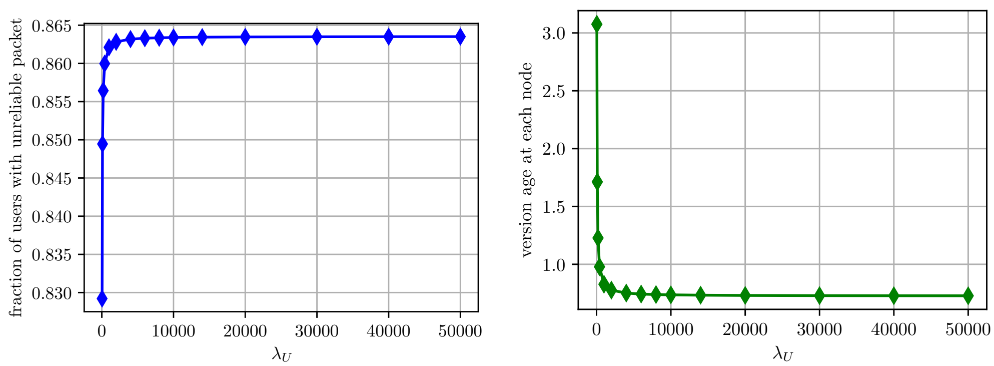
expected age grows logarithmically in Fig. 6.4(b).

Fig. 6.5(a) and Fig. 6.5(b) show the variation in F and x_1 with gossiping rate λ. Fig. 6.5(a) shows that F is a decreasing function of λ, converging to a non-zero value of $\left(\frac{\lambda_E}{\lambda_E+\lambda_R}\right)^{G+1} \frac{\lambda_U}{\lambda_U+\lambda_R} = 0.573$, as predicted by Lemma 6.4. Fig. 6.5(b) shows that x_1 is also a decreasing function of λ, in accordance with Lemma 6.8, converging to a value greater than the lower bound of $\frac{\lambda_E}{\lambda_U+\lambda_R} = 0.39$ given by (6.73).



(a) (b)

Figure 6.6: F and (b) x_1 as a function of reliable source update rate λ_R, with $\lambda_E = 2$, $\lambda_U = 50$, $\lambda = 0.1$, $n = 50$ and $G = 10$.

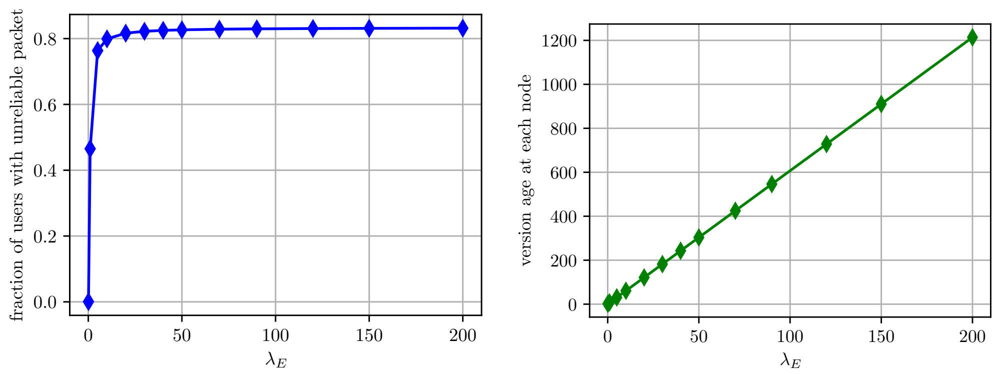


(a) (b)

Figure 6.7: F and (b) x_1 as a function of unreliable source update rate λ_U, with $\lambda_E = 2$, $\lambda_R = 1$, $\lambda = 0.1$, $n = 50$ and $G = 10$.

Next, we simulate the variation of F and x_1 with reliable source update rate λ_R in Fig. 6.6(a) and Fig. 6.6(b). As per Lemma 6.5, F decreases with λ_R in Fig. 6.6(a), ultimately reducing to zero, causing a full prevalence of reliable packets in the network, concurrently also causing x_1 to reduce to zero in Fig. 6.6(b), as suggested by (6.75).

Fig. 6.7(a) and Fig. 6.7(b) show the dependence of F and x_1 on unreliable



(a) (b)

Figure 6.8: F and (b) x_1 as a function of event update rate λ_E, with $\lambda_U = 5$, $\lambda_R = 1$, $\lambda = 0.1$, $n = 50$ and $G = 10$.

source update rate λ_U. As seen in Fig. 6.7(a) and suggested by Lemma 6.6, F is an increasing function of λ_U, however, it does not converge to 1 at large values of λ_U. Likewise, in Fig. 6.7(b), x_1, though a decreasing function of λ_U, does not converge to zero, as suggested by Lemma 6.9.

Finally, we simulate the variation in F and x_1 with event update rate λ_E in Fig. 6.8(a) and Fig. 6.8(b). As per Lemma 6.7, F increases with λ_E in Fig. 6.8(a), however, converges to a value less than the upperbound of $\frac{\lambda_U}{\lambda_U+\lambda_R} = 0.83$. Further, as per Lemma 6.10, x_1 scales linearly with λ_E in Fig. 6.8(b).

6.5 Conclusion

In this chapter, we analyzed a system model where packets containing information about a dynamic event are disseminated to a network of nodes by both reliable and unreliable sources. The file exchange protocol implemented at the network nodes allows for a sacrifice in the freshness of packets, quantified by the version age of information metric, by up to G versions, to enhance packet reliability. Our analysis demonstrated that this sacrificing capacity G creates a trade-off between reliability and information freshness at the network nodes. Additionally, we explored the dependence of network reliability and freshness on various network parameters, including update rates at different nodes and the overall network size.

CHAPTER 7

Timeliness in Cache-Aided Networks with Non-Poisson Updating

7.1 Introduction

In this chapter, we investigate two timeliness metrics- age of information and version age of information, in cache-aided networks when the inter-update times on the links are not necessarily exponentially distributed. We first study age of information in a multi-hop multi-cast network with cache-aided nodes as shown in Fig. 7.1. We assume that the updates on each link are sent according to an ordinary renewal process which is not necessarily a Poisson process, and all nodes forward updates independently of other nodes. As a result, the evolution of the age in the network is defined by a superposition of multiple independent ordinary renewal processes. As proved in [108], the superposition of two ordinary renewal processes is an ordinary renewal process only if all processes are Poisson processes.

Since the exponential distribution (or geometric distribution) is the only continuous (or discrete) probability distribution with memoryless property, most prior works studying timely information dissemination in networks heavily rely on these

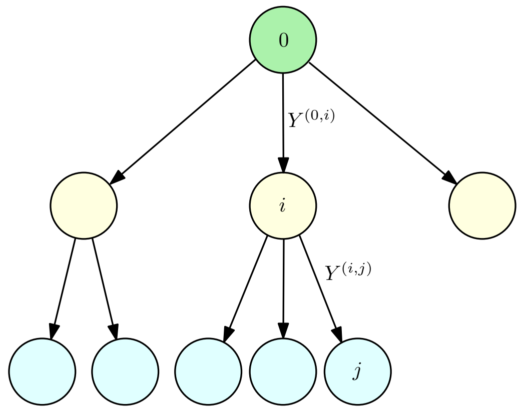


Figure 7.1: Multi-hop multi-cast tree network.

distributions. Here, it is worth noting how networks with nodes as queues [109–111] differ from networks with nodes as caches. When a node has a queue buffer, all three random variables corresponding to, the arrival of a packet, waiting time of a packet in the queue and the service time of a packet, operate in a sequential order. In contrast, in cache-updating networks, each node requests packets from other nodes according to a renewal process, independently of how packets arrive at other nodes. Hence, simultaneous operation of renewal processes is distinctive to cache-updating networks. Nevertheless, when the random variables involved are memoryless, the age processes in cache-updating networks and preemptive-queuing networks are similar. This can be seen in the case of linear or tree networks by comparing the results of [109] and [112] for memoryless exponential random variables, and [113] and [114] for memoryless geometric random variables.

However, when the inter-update times follow general distributions which are not exponential, it becomes difficult to calibrate the different renewal processes with respect to each other. To understand this challenge, let us first consider the single-hop system model of Fig. 7.2(a) which offers the convenience of a single renewal

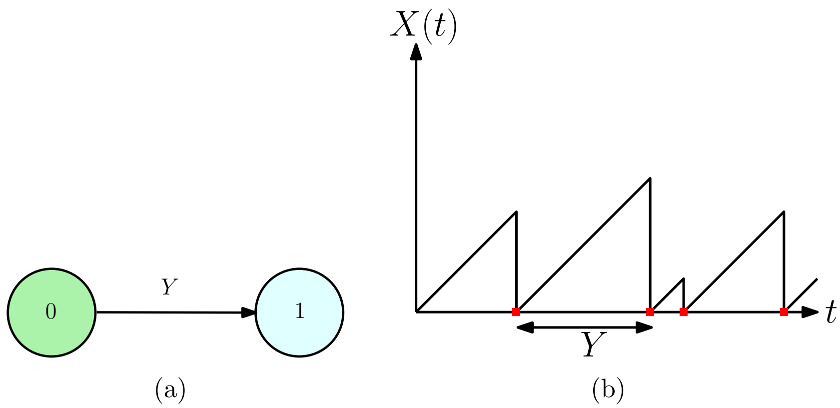


Figure 7.2: (a) One-hop model, consisting of source (node 0) and user (node 1). (b) Red dots show time instants of packet arrival at user from source.

process, and hence, circumvents the need to deal with superposition of arrival processes. In this model, the user downloads packets from the source according to a renewal process where inter-update times Y_i are positive i.i.d. random variables with non-arithmetic distribution F. In this respect, a distribution F is called arithmetic (or periodic) if it is piecewise constant and its points of increase are contained in a set $\{0, d, 2d, \ldots\}$ with the largest such $d > 0$ being the span of such distribution. When F is not arithmetic, it is called non-arithmetic, e.g., a distribution with a continuous part [115].

In the above model, the age of information $X(t)$ at the user, shown in Fig. 7.2(b), is determined by the time elapsed since the last renewal. Since $X(t)$ at time t depends only on location of t within the inter-renewal interval containing t, $X(t)$ qualifies as a renewal reward process [116], and its ensemble-average in limit of large t is the same as its time average which, with probablity 1, provided $\mathbb{E}[Y^2] < \infty$, is

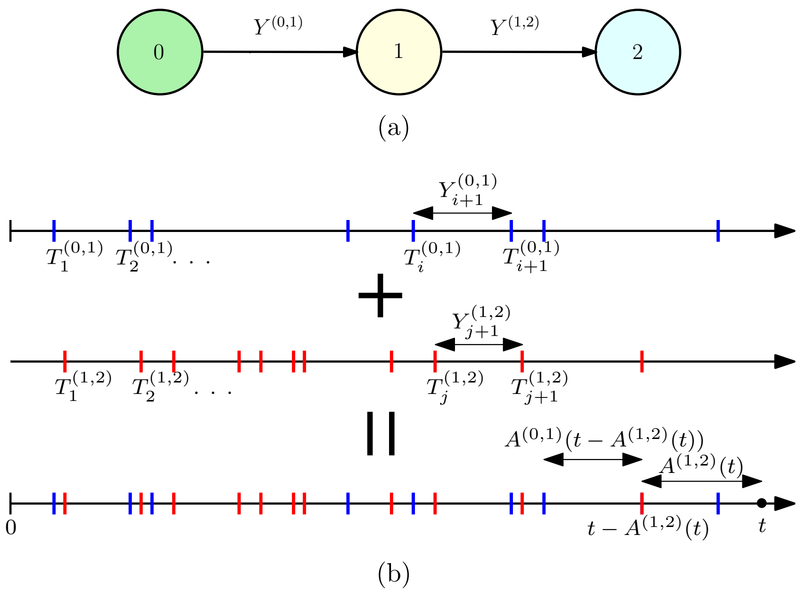


Figure 7.3: (a) Two-hop model. (b) Superposition of two renewal processes, $N^{(0,1)}(t)$ and $N^{(0,1)}(t)$.

given by

$$\lim_{t\to\infty} \mathbb{E}[X(t)] = \lim_{t\to\infty} \frac{\int_{\tau=0}^{t} X(\tau)d\tau}{t} = \frac{\mathbb{E}[Y^2]}{2\mathbb{E}[Y]} \tag{7.1}$$

where Y is a typical random variable with distribution F.

We next add a second link to this model which results in the two-hop network of Fig. 7.3(a), where updates arrive at node k from node $k-1$ at times $T_i^{(k-1,k)}$ according to renewal process k, as shown in Fig. 7.3(b). If we consider a typical inter-renewal interval $[T_i^{(0,1)}, T_{i+1}^{(0,1)}]$, then in the absence of memoryless inter-update times, the distribution of packet arrival instant at node 2 in this interval is dependent on when the last packet arrived at node 2 in previous inter-renewal intervals, which prevents us from characterizing the age process in this interval independently of the past. Similarly, if we were to consider an inter-renewal interval $[T_j^{(1,2)}, T_{j+1}^{(1,2)}]$,

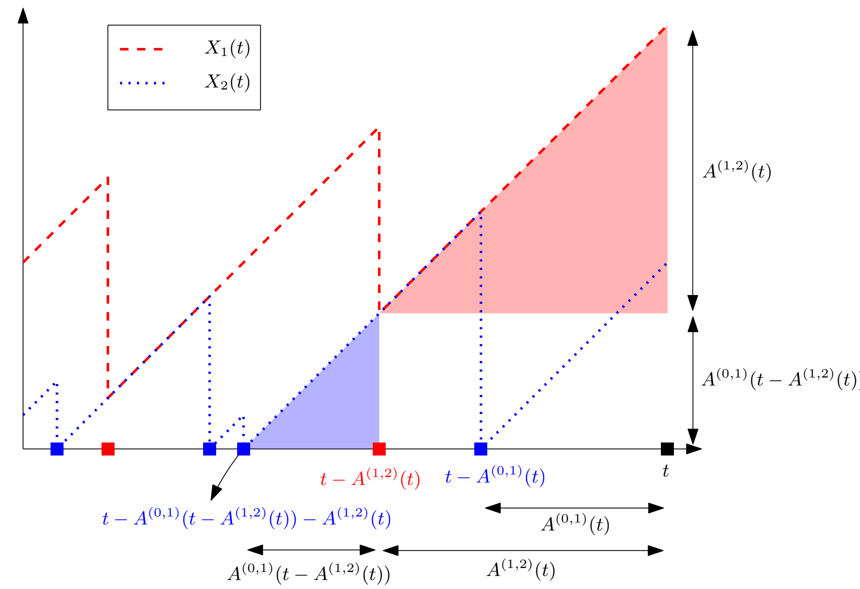


Figure 7.4: Evolution of instantaneous age of information $X_1(t)$ and $X_2(t)$, at node 1 and 2, respectively, till time t in a two-hop network. The red dots represent the time instants when updates arrive at node 2 from node 1, while the blue dots represent the time instants when updates arrive at node 1 from node 0. The instantaneous age at node 2 at time t is composed of two terms: the top part is $A^{(1,2)}(t)$, which arises from the $N^{(1,2)}(t)$ renewal process on the incident link $(1, 2)$, and the bottom part is $A^{(0,1)}(t - A^{(1,2)}(t))$, which depends on both the renewal processes $N^{(0,1)}(t)$ and $N^{(1,2)}(t)$. Hence, age evolution at node 2 after the red dot at time instant $t - A^{(1,2)}(t)$ cannot be characterized independently of the time before the red dot, since the height of the blue shaded triangle depends on the past. Likewise, age at the last blue dot time instant $t - A^{(0,1)}(t)$ is determined by time before the blue dot, highlighting correlations between different inter-renewal intervals. Notice how $X_1(t) \leq X_2(t)$, since packets arrive at node 2 from the single preceding node 1.

then the user age at the beginning or end of this interval depends on arrivals under renewal process 1 in prior intervals. Therefore, the age evolutions in different time intervals are correlated throughout the timeline, see Fig. 7.4, and an interesting question to ask here is, whether it is possible to somehow characterize the ensemble average of age at user 2 in the regime of large t.

Compared to traditional age of information, there are significant differences in the analysis of version age of information, by virtue of the additional renewal point process at the source superimposed on other renewal processes. This is because traditional age of information increases at unit rate, does not account for update process at the source, and consequently can be fully described at a node in terms of the time since the last update from the immediate server node and the age at the server node. On the other hand, version age of information is a discrete metric that gets incremented in steps of one whenever the source gets updated to a newer version, and involves counting of renewals at the source between update arrivals from the immediate server node. The case of version age of information, consequently, gets further convoluted and is presented in later sections.

The works that are most relevant to this work are as follows: In [43], age of information is studied in a single-hop network with N files having varying degrees of popularity. In [51], freshness of a citation index is studied for n researchers with Poisson citation arrivals. [112] and [7] study multi-hop networks with Poisson arrivals for version age of information metric and binary freshness metric, respectively. [8] focuses on two-hop parallel relay networks with Poisson arrivals. [4] and [5] have initiated the study of age of information and version age of information, respectively, in arbitrary networks with Poisson arrivals, using the stochastic hybrid system (SHS) framework [4–6, 9, 105, 106, 112, 117, 118]. The particular convenience of Poisson updates in SHS stems from the fact that exponential distribution is the only distribution with constant hazard rate, which simplifies expressions involving expectations of products of test functions and transition intensities or hazard rates. For multi-hop networks operating under ordinary renewal processes, [67] derived the distribution of traditional age of information for the case of strictly stationary age processes; see [67, Section V]. Note that our work does not require the age processes to be strictly stationary for analysis of age of information metric in any of the network topologies considered. Other notable works on information freshness in cache updating systems include [44, 66, 119].

The key contributions of this work are as follows:

- We first attempt to characterize the age of information in cache-aided update systems for non-Poisson renewal processes. Though getting expressions for the long-term expected age proves difficult for general networks, we provide a closed-form expression for the expected age in multi-cast networks which exhibit a tree topology as shown in Fig. 7.1.

- We observe from (7.1) that for a fixed mean arrival time $\mathbb{E}[Y]$, the age of information in a single-hop network increases with an increase in the variance $var[Y]$, since $\mathbb{E}[Y^2] = var[Y] + \mathbb{E}[Y]^2$. We show that the age at the end-user in a multi-hop network also exhibits a similar relationship with the variance of the inter-arrival distribution of the user node, which gives all nodes an opportunity to lower their age, independent of the dynamics of the network.

- We provide analogous results for version age of information by employing a pre-limit refinement of Blackwell's theorem [120] using a semi-martingale representation of a renewal process $N(t)$, because the classical Blackwell's result, $\lim_{t\to\infty} \mathbb{E}[N(t+a) - N(t)] = \frac{a}{\mathbb{E}[Y]}$, only provides expected number of renewals for process $N(t)$ in a constant time interval a, which proves insufficient for version age analysis in this work. We show that the expected version age at the end-users is inversely proportional to the mean update interval at the source, and also exhibits an additive structure.

- We finally study the expected age of information in networks with the property that the update processes on all links become sparse for large network sizes, such as a fully connected network, and remark that the expected age scales as $O(\log n)$ in symmetric fully connected networks.

- We present simulation results to support our findings.

7.2 Model and Notations for Age Analysis

Packets are assumed to arrive from node i at node j on link (i,j) according to $N^{(i,j)}(t)$ renewal process, and the finite random times $0 \leq T_1^{(i,j)} \leq T_2^{(i,j)} \leq \ldots$ denote the corresponding renewal times, such that, inter-arrival times $Y_n^{(i,j)} = T_n^{(i,j)} - T_{n-1}^{(i,j)}$ are i.i.d. with common distribution $F^{i,j}$. Given $N^{(i,j)}(t) = \max\{n : T_n^{(i,j)} \leq t\}$, the regenerative process $A^{(i,j)}(t) = t - T_{N^{(i,j)}(t)}^{(i,j)}$ denotes the corresponding backward recurrence time (or current life or age of renewal process) at t, which is the time since the last renewal prior to t. Note that $0 \leq A^{(i,j)}(t) \leq t$, which will be repeatedly used later. For more details, please see references [115, 116].

Consider a typical node j in an arbitrary network of nodes and let S_j denote the set of nodes from which packets arrive at node j. The most recent packet from node $i \in S_j$ arrives at node j before time t at time instant $t - A^{(i,j)}(t) = T_{N^{(i,j)}(t)}^{(i,j)}$, at which point, node j compares generation times of the arriving packet with the packet present at its cache, discarding the staler packet in favor of the fresher packet.

Let $X_j(t)$ denote the instantaneous age of information at node j at time t. Then, $X_j(t)$ can be written as

$$X_j(t) = \sum_{i\in S_j}\Big[\prod_{k\in S_j\setminus\{i\}} \chi_{\{A^{(i,j)}(t)<A^{(k,j)}(t)\}}\Big] \times\Big[\min\big\{X_i(t-A^{(i,j)}(t)), X_j(t-A^{(i,j)}(t)\big\} + A^{(i,j)}(t)\Big] \tag{7.2}$$

where $\chi_{\mathcal{A}}$ represents the indicator random variable for the measurable set $\mathcal{A}$ and $X_j(0)=0$. Since the source always has the latest packet, $X_0(t)=0$ at all times.[1]

In (7.2), $\prod_{k\in S_j\setminus\{i\}} \chi_{\{A^{(i,j)}(t)<A^{(k,j)}(t)\}}$ corresponds to the scenario when the last packet that arrived at node j before time t came from node i, which would be the case when the backward recurrence times of all other relevant renewal processes at time t are larger than $A^{(i,j)}(t)$. The last term $A^{(i,j)}(t)$ in (7.2) comes from the fact that age grows at unit rate post the last packet arrival.

In the next step, $\min\{X_i(t-A^{(i,j)}(t)), X_j(t-A^{(i,j)}(t)\}$, which is a minimum over two age processes, can be further characterized in a manner similar to (7.2), and the corresponding expression will have terms of the form that involve taking a minimum over three age processes, for example $\min\{X_i(t'), X_j(t'), X_k(t')\}$ with $t' = t - A^{(i,j)}(t) - A^{(k,j)}(t-A^{(i,j)}(t))$. By recursively repeating this process we finally encounter the expression $\min\{X_1(t''), X_2(t''), \ldots, X_n(t'')\}$, $t'' = t-\Delta(t)$, where $\Delta(t)$ represents a stochastic process whose exact expression depends on the

[1](7.2) holds true for the case where the distributions of inter-update times do not have atom points. When the distributions have atom points, packets from different nodes might arrive at node j at the same time with non-zero probability. This situation can be remedied by choosing a priority order for different incoming links, which would change some of the "$<$" to "$\leq$" in the indicator variable in (7.2).

network topology. Since source node is the only node external to the set of n nodes, the last min expression can be completely defined in terms of backward recurrence times of the form $A^{(0,\ell)}(t''')$ such that $0 \in S_\ell$. This recursive approach will become more clear for multi-cast networks with tree topology in Section 7.3.

On first glance, this might give an impression that since all renewal processes and their associated recurrence times are independent processes, by reducing $X_j(t)$ to a function composed purely of backward recurrence times, one could conveniently compute the expectation of $X_j(t)$. However, note that, in the first and second steps of the recursion above, we encountered the term $A^{(k,j)}(t)$ in the product of indicator variables in (7.2), and the term $A^{(k,j)}(t - A^{(i,j)}(t))$ in the definition of t'. Though both terms correspond to the same renewal process $N^{(k,j)}(t)$, these backward recurrence times could be correlated through time which complicates analysis.

However, this complication does not arise if we assume that each node in the network has only one incoming link, as shown in the tree network of Fig. 7.1. This is because, $|S_j| = 1$, and the product term in (7.2) vanishes. Additionally, since packets now arrive at node j from a single preceding node i, $X_i(t) \leq X_j(t)$ for all t, see Fig. 7.4, which simplifies the min term as,

$$\min\left\{X_i(t - A^{(i,j)}(t)), X_j(t - A^{(i,j)}(t)\right\} = X_i(t - A^{(i,j)}(t)) \tag{7.3}$$

In the next section, we derive a closed form expression for the long-term expected age $\lim_{t\to\infty} \mathbb{E}[X_j(t)]$ at each node j in networks that have a tree topology.

7.3 Age in Networks with Tree Structure

7.3.1 Two-Hop Network

Consider the two-hop network in Fig. 7.3(a), where we wish to determine the long-term expected age at node 2. For this network, as depicted graphically in Fig. 7.4, the instantaneous age $X_2(t)$ can be written as

$$X_2(t) = X_1(t - A^{(1,2)}(t)) + A^{(1,2)}(t) \tag{7.4}$$

where $X_1(t - A^{(1,2)}(t))$ in turn can be expressed as

$$X_1(t - A^{(1,2)}(t)) = X_0(t - A^{(1,2)}(t) - A^{(0,1)}(t - A^{(1,2)}(t))) + A^{(0,1)}(t - A^{(1,2)}(t)) \tag{7.5}$$

Let us define

$$\Delta_1(t) = A^{(1,2)}(t) \tag{7.6}$$

$$\Delta_2(t) = A^{(0,1)}(t - A^{(1,2)}(t)) \tag{7.7}$$

Substituting (7.5), (7.6) and (7.7) into (7.4), and using $X_0(t - \Delta_1(t) - \Delta_2(t)) = 0$, we obtain

$$X_2(t) = \Delta_1(t) + \Delta_2(t) \tag{7.8}$$

Lemma 7.1 *Given independent stochastic processes* $S_1(t)$ *and* $S_2(t)$ *with* $\sup_{t\geq 0} |\mathbb{E}[S_1(t)]| < \infty$ *and* $0 \leq S_2(t) \leq t$*, such that* $\lim_{t\to\infty} \mathbb{E}[S_1(t)]$ *and* $\lim_{t\to\infty} \mathbb{E}[S_2(t)]$ *exist, we have*

$$\lim_{t\to\infty} \mathbb{E}[S_1(t - S_2(t))] = \lim_{t\to\infty} \mathbb{E}[S_1(t)] \tag{7.9}$$

Proof: Let us define expectation functions

$$\mu_1(t) = \mathbb{E}[S_1(t)] \tag{7.10}$$

$$\mu_2(t) = \mathbb{E}[S_2(t)] \tag{7.11}$$

such that $\mu_1(t) \to a_1$ and $\mu_2(t) \to a_2$, where a_1 and a_2 are some constants. Thus, for every $\epsilon_2 > 0$ there exists time T_2 such that for all $t > T_2$, we have $|\mu_2(t) - a_2| < \epsilon_2$. Choosing $\epsilon_2 = a_2$, and using Markov inequality, we obtain

$$\mathbb{P}(S_2(t) \geq \sqrt{t}) \leq \frac{\mu_2(t)}{\sqrt{t}} < \frac{2a_2}{\sqrt{t}} \tag{7.12}$$

Likewise, for every $\epsilon_1 > 0$, there exists T_1 such that for all $t > T_1$, we have $|\mu_1(t) - a_1| < \epsilon_1$. Since $S_1(t)$ and $S_2(t)$ are independent stochastic processes, for all $t \geq 0$, we have

$$\mathbb{E}[S_1(t - S_2(t))|S_2(t)] = \mu_1(t - S_2(t)) \tag{7.13}$$

Therefore,

$$
\begin{aligned}
\Big|\mathbb{E}\big[S_1(t - S_2(t))\big] - a_1\Big| =& \Big|\mathbb{E}\big[\mu_1(t - S_2(t)) - a_1\big]\Big| && (7.14)\\
=& \Big|\mathbb{E}\Big[(\mu_1(t - S_2(t)) - a_1)\left(\chi_{\{S_2(t)\geq\sqrt{t}\}} + \chi_{\{S_2(t)<\sqrt{t}\}}\right)\Big]\Big| && (7.15)\\
\leq& \Big|\mathbb{E}\Big[\mu_1(t - S_2(t))\chi_{\{S_2(t)\geq\sqrt{t}\}}\Big]\Big| + \Big|\mathbb{E}\left[a_1\chi_{\{S_2(t)\geq\sqrt{t}\}}\right]\Big|\\
&+ \Big|\mathbb{E}\left[(\mu_1(t - S_2(t)) - a_1)\chi_{\{S_2(t)<\sqrt{t}\}}\right]\Big| && (7.16)\\
\leq& \mathbb{E}\left[\Big|\mu_1(t - S_2(t))\Big|\chi_{\{S_2(t)\geq\sqrt{t}\}}\right] + \mathbb{E}\left[a_1\chi_{\{S_2(t)\geq\sqrt{t}\}}\right]\\
&+ \mathbb{E}\left[\Big|\mu_1(t - S_2(t)) - a_1\Big|\chi_{\{S_2(t)<\sqrt{t}\}}\right] && (7.17)
\end{aligned}
$$

where (7.16) follows from the triangle inequality.

If $t > T' = \frac{1+2T_1+\sqrt{1+4T_1}}{2}$, then when $S_2(t) < \sqrt{t}$, we have $t - S_2(t) > t - \sqrt{t} > T_1$ and hence $|\mu_1(t - S_2(t)) - a_1| < \epsilon_1$. For all other cases, $|\mu_1(t - S_2(t))| \leq \sup_{t\geq 0}|\mathbb{E}[S_1(t)]|$. For $t > \max\{T', T_2\}$, using (7.12), (7.17) gives

$$
\Big|\mathbb{E}[S_1(t - S_2(t))] - a_1\Big| < \sup_{t\geq 0}|\mathbb{E}[S_1(t)]|\,\frac{2a_2}{\sqrt{t}} + \frac{2a_1a_2}{\sqrt{t}} + \epsilon_1 \tag{7.18}
$$

Since ϵ_1 can be chosen to be arbitrarily small,

$$
\lim_{t\to\infty}\Big|\mathbb{E}[S_1(t - S_2(t))] - a_1\Big| = 0 \tag{7.19}
$$

Thus, $\mathbb{E}[S_1(t - S_2(t))] \to a_1$, which completes the proof. ■

From (7.8), $\mathbb{E}[X_2(t)]$ requires computing the terms $\mathbb{E}[\Delta_1(t)]$ and $\mathbb{E}[\Delta_2(t)]$.

Since $A^{(1,2)}(t)$ evolves as in Fig. 7.2(b), (7.1) gives

$$\lim_{t\to\infty} \mathbb{E}[\Delta_1(t)] = \lim_{t\to\infty} \mathbb{E}[A^{(1,2)}(t)] = \frac{\mathbb{E}\left[\left(Y^{(1,2)}\right)^2\right]}{2\mathbb{E}\left[Y^{(1,2)}\right]} \tag{7.20}$$

Likewise, we have

$$\lim_{t\to\infty} \mathbb{E}[A^{(0,1)}(t)] = \frac{\mathbb{E}\left[\left(Y^{(0,1)}\right)^2\right]}{2\mathbb{E}\left[Y^{(0,1)}\right]} \tag{7.21}$$

Since the limit $\lim_{t\to\infty} \mathbb{E}[A^{(0,1)}(t)]$ exists, there exists T such that for all $t > T$, $\mathbb{E}[A^{(0,1)}(t)] < \lim_{t\to\infty} \mathbb{E}[A^{(0,1)}(t)] + \epsilon$ for some $\epsilon > 0$. Further, since $0 \leq A^{(0,1)}(t) \leq t$ by definition, we have $\mathbb{E}[A^{(0,1)}(t)] < T$ for $t \leq T$. Hence,

$$\sup_{t\geq 0} \left|\mathbb{E}[A^{(0,1)}(t)]\right| \leq \max\left\{ \lim_{t\to\infty} \mathbb{E}[A^{(0,1)}(t)] + \epsilon, T\right\} < \infty \tag{7.22}$$

Hence by Lemma 7.1,

$$\lim_{t\to\infty} \mathbb{E}[\Delta_2(t)] = \lim_{t\to\infty} \mathbb{E}[A^{(0,1)}(t)] = \frac{\mathbb{E}\left[\left(Y^{(0,1)}\right)^2\right]}{2\mathbb{E}\left[Y^{(0,1)}\right]} \tag{7.23}$$

Thus, the long-term expected age at node 2 is

$$\lim_{t\to\infty} \mathbb{E}[X_2(t)] = \frac{\mathbb{E}\left[\left(Y^{(1,2)}\right)^2\right]}{2\mathbb{E}\left[Y^{(1,2)}\right]} + \frac{\mathbb{E}\left[\left(Y^{(0,1)}\right)^2\right]}{2\mathbb{E}\left[Y^{(0,1)}\right]} \tag{7.24}$$

Remark 7.1 *Interestingly, the age at node* 2 *is determined by independent contributions of the links in the path from node* 0 *to node* 2*.*

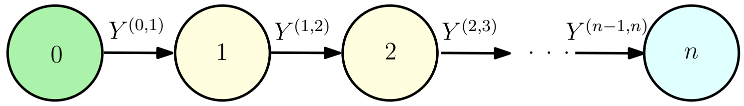


Figure 7.5: n-hop model.

In general, for tree networks, only the links involved in the path between the source and an end-user are critical to the age dynamics of the end-user. In the next subsection, we therefore study n-hop linear networks.

7.3.2 Multi-Hop Network

Consider the n-hop network of Fig. 7.5 where we wish to determine the long-term expected age at node n of the network. We define time segments $\Delta_i(t)$, $i \geq 1$ through the following recurrence equation

$$\Delta_i(t) = A^{(n-i,n-i+1)}(t - \sum_{j=0}^{i-1} \Delta_j(t)) \tag{7.25}$$

with $\Delta_0(t) = 0$, see Fig. 7.6. Note that $\Delta_i(t)$ is smaller than $t - \sum_{j=0}^{i-1} \Delta_j(t)$ by definition of $A^{(n-i,n-i+1)}(t)$. Similar to (7.4), the instantaneous age $X_n(t)$ at node n can be written as

$$X_n(t) = X_{n-1}(t - A^{(n-1,n)}(t)) + A^{(n-1,n)}(t) \tag{7.26}$$

This can be alternately represented as

$$X_n(t - \Delta_0(t)) = X_{n-1}(t - \Delta_0(t) - \Delta_1(t)) + \Delta_1(t) \tag{7.27}$$

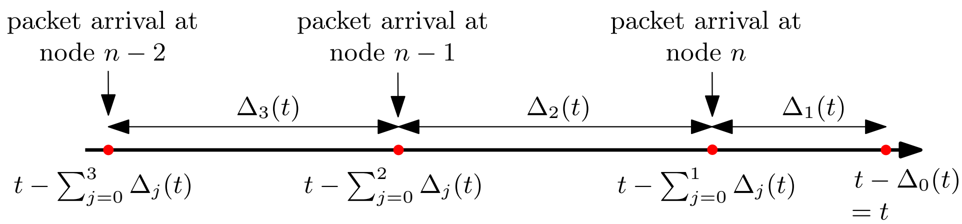


Figure 7.6: Time segments $\Delta_i(t)$ for $i \in \{1, \ldots, n\}$.

In the next step, $X_{n-1}(t-A^{(n-1,n)}(t))$ of (7.26) will be again characterized in a similar manner and the full set of equations encountered in this recursive approach is of the form

$$X_{n-i}(t-\sum_{j=0}^{i}\Delta_j(t)) = X_{n-i-1}(t-\sum_{j=0}^{i+1}\Delta_j(t)) + \Delta_{i+1}(t) \tag{7.28}$$

for $0 \le i \le n-1$ with $X_0(t-\sum_{j=1}^{n}\Delta_j(t)) = 0$ as node 0 represents the source node. Then, it follows from (7.28) that

$$X_n(t) = \sum_{j=1}^{n}\Delta_j(t) \tag{7.29}$$

Similar to (7.20), we have

$$\lim_{t\to\infty}\mathbb{E}[\Delta_1(t)] = \lim_{t\to\infty}\mathbb{E}[A^{(n-1,n)}(t)] = \frac{\mathbb{E}\left[\left(Y^{(n-1,n)}\right)^2\right]}{2\mathbb{E}\left[Y^{(n-1,n)}\right]} \tag{7.30}$$

Further, using the approach of (7.22), we get $\sup_{t\ge 0}\left|\mathbb{E}[A^{(n-i,n-i+1)}(t)]\right| < \infty$. Since $\sum_{j=0}^{i-1}\Delta_j(t) \le t$, we can prove $\lim_{t\to\infty}\mathbb{E}[\Delta_i(t)] = \lim_{t\to\infty}\mathbb{E}[A^{(n-i,n-i+1)}(t)]$ recursively for $i = 2, 3, \ldots, n$ from (7.25) using Lemma 7.1.

Hence, from (7.29), we obtain

$$\lim_{t\to\infty} \mathbb{E}[X_n(t)] = \sum_{j=1}^{n} \lim_{t\to\infty} \mathbb{E}[\Delta_j(t)] = \sum_{j=1}^{n} \frac{\mathbb{E}\left[\left(Y^{(n-j,n-j+1)}\right)^2\right]}{2\mathbb{E}\left[Y^{(n-j,n-j+1)}\right]} \tag{7.31}$$

Remark 7.2 *In the special case when the renewal processes are Poisson, (7.31) reduces to the sum of inverse of rates of the Poisson processes, which was observed in [109] for preemptive queues, and in [112] for cache-updating systems, using the SHS method.*

Remark 7.3 *Interestingly, the age at node* n *depends on independent contributions of the intermediate links* $(i, i+1)$, $0 \leq i \leq n-1$ *and is invariant to ordering of these links. Hence, each node can minimize its age by optimizing its individual packet request renewal process, irrespective of the statistical properties of other nodes and links in the network. Since the constant random variable has zero variance, for a fixed mean* $\mathbb{E}[Y^{(n-j,n-j+1)}]$, *(7.31) hints that all nodes should request packets at constant time intervals to reduce variance. Here, we would like to point out that the constant distribution is an example of arithmetic distribution where the limit* $\lim_{t\to\infty} \mathbb{E}[X(t)]$ *does not exist. This is because the ages at all nodes evolve as deterministic functions of time and the expectations do not stabilize at large* t. *Time average instead is the more reasonable choice of performance metric for arithmetic distributions and a counter part of Lemma 7.1 needs to be proved for their case.*

These findings imply that in a cache updating system, such as within a communication network consisting of a cloud, macro-base station, small-cell base station, and users, the end-users should download packets at constant intervals from their nearest base stations. As we will demonstrate in Section 7.8 later, the expected age at user can significantly differ based on whether the renewal processes are correctly identified or merely approximated as Poisson processes.

Remark 7.4 *Using approach similar to Lemma 7.1 and (7.31), one could proceed to compute higher moments of age of information as well. For instance, the second moment,* $\lim_{t\to\infty}\mathbb{E}[X_n(t)^2]$*, useful for computing variance of age at node* n*, can be computed, from (7.29), to be*

$$\lim_{t\to\infty}\mathbb{E}[X_n(t)^2] = \lim_{t\to\infty}\mathbb{E}\left[\left(\sum_{j=1}^{n}\Delta_j(t)\right)^2\right] \tag{7.32}$$

$$= \lim_{t\to\infty}\mathbb{E}\left[\left(\sum_{j=1}^{n}A^{(n-j,n-j+1)}(t)\right)^2\right] \tag{7.33}$$

$$= \lim_{t\to\infty}\mathbb{E}\left[\sum_{i\neq j}\left(2A^{(n-i,n-i+1)}(t)A^{(n-j,n-j+1)}(t)\right)\right] + \lim_{t\to\infty}\mathbb{E}\left[\sum_{j=1}^{n}\left(A^{(n-j,n-j+1)}(t)\right)^2\right] \tag{7.34}$$

$$= \lim_{t\to\infty}\sum_{i\neq j}2\frac{\mathbb{E}\left[\left(Y^{(n-j,n-j+1)}\right)^2\right]}{2\mathbb{E}\left[Y^{(n-i,n-i+1)}\right]}\frac{\mathbb{E}\left[\left(Y^{(n-i,n-i+1)}\right)^2\right]}{2\mathbb{E}\left[Y^{(n-j,n-j+1)}\right]} + \sum_{j=1}^{n}\frac{\mathbb{E}\left[\left(Y^{(n-j,n-j+1)}\right)^3\right]}{3\mathbb{E}\left[Y^{(n-j,n-j+1)}\right]} \tag{7.35}$$

where the equivalence of (7.32) and (7.33) can be established through a convergence proof similar to Lemma 7.1 and Cauchy-Schwarz inequality $|\mathbb{E}[X_1X_2]|^2 \leq \mathbb{E}[X_1^2]\mathbb{E}[X_2^2]$*. The equivalence of (7.34) and (7.35) follows from moment derivations of backward recurrence times in [121] and independence of renewal processes on*

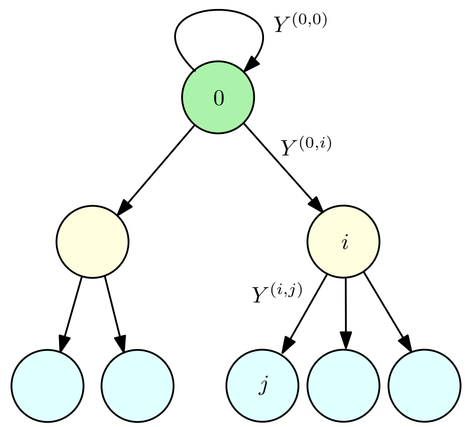


Figure 7.7: Multi-hop multi-cast tree network with version updates at source.

various links.

7.4 Version Age with Multi-Hop Renewal Updates

Consider a cache-enabled network consisting of a source node, server nodes and user nodes in a tree topology, with source as the root node and users as leaf nodes, as shown in Fig. 7.7. The source gets updated according to an ordinary renewal process and uses a logical clock to mark the updates with an incrementing numeric value, which we refer to as the version number of the update packet. The user nodes attempt to retrieve the latest possible version update from the source through a sequence of cache-aided server nodes, such that updates on all links are forwarded according to ordinary renewal processes that are not necessarily Poisson processes. In this setting, we use version age of information to quantify the freshness of information at the user nodes. At time t, the instantaneous version age at node i is defined as $X_i(t) = W_0(t) - W_i(t)$, where at $W_i(t)$ is the latest version of a file available at node i and $W_0(t)$ is the current version prevailing at the source.

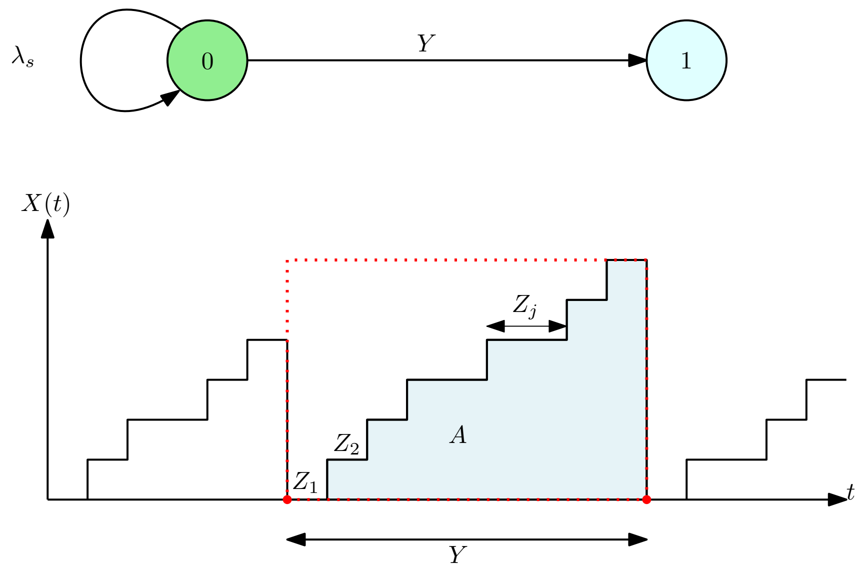


Figure 7.8: One-hop model, where the sources gets updates according to a Poisson process with rate λ_s and updates arrive at node 1 according to a renewal process with typical inter-renewal period Y.

Consider the single-hop network of Fig. 7.8, where a user downloads packets from the source according to an ordinary renewal point process with inter-update times as positive i.i.d. random variables, denoted by typical random variable Y with non-arithmetic distribution F. Let us first consider the simpler case when the source (node 0) gets updated according to a Poisson process with rate λ_s. In this case, the instantaneous version age $X(t)$ at time t at node 1 is determined by the number of renewals at the source since node 1 last downloaded a packet from the source. In Fig. 7.8, Y represents a typical inter-renewal interval between two consecutive downloads at node 1, and Z_j corresponds to the inter-renewal interval j between two updates at the source in interval Y, with Z_1 denoting the time between a download at node 1 and the first update at the source since the download. Due to the memoryless property of Poisson process, Z_1 is exponentially distributed, like other Z_js. Hence, $X(t)$ at any time t within the particular inter-renewal interval Y will only depend

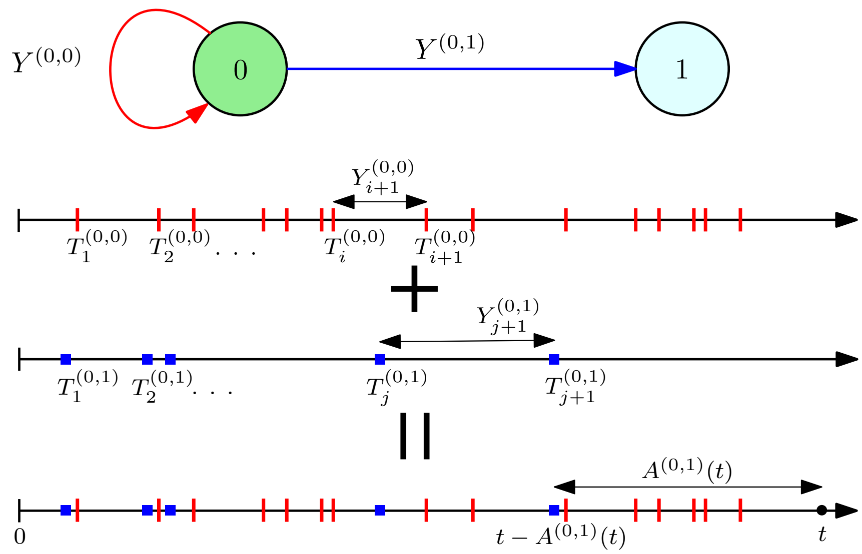


Figure 7.9: Superposition of renewal processes, $N^{(0,0)}(t)$ and $N^{(0,1)}(t)$, in the one-hop model.

on the location of t in Y and random variables Z_js local to Y, consequently, $X(t)$ qualifies as a renewal reward process. We define A to be the accumulated reward in the inter-renewal interval Y, which corresponds to the area of the shaded region in Fig. 7.8. We assume in this work that inter-renewal distributions of all renewal processes have finite first and second moments. Therefore, $\mathbb{E}[Y] < \infty$ and $\mathbb{E}[Y^2] < \infty$, and with probability 1, we have from [116]

$$\lim_{t\to\infty} \mathbb{E}[X(t)] = \frac{\mathbb{E}[A]}{\mathbb{E}[Y]} = \frac{\lambda_s \mathbb{E}[Y^2]}{2\mathbb{E}[Y]} \tag{7.36}$$

where the shaded area A in Fig. 7.8 can be computed as in [51]. However, when the source gets updated according to a general renewal process, as in Fig. 7.9, in the absence of memoryless property of inter-update times, the distribution of Z_1 depends on the last source update instant in the previous inter-renewal interval, which prevents us from characterizing version age in the interval Y independently

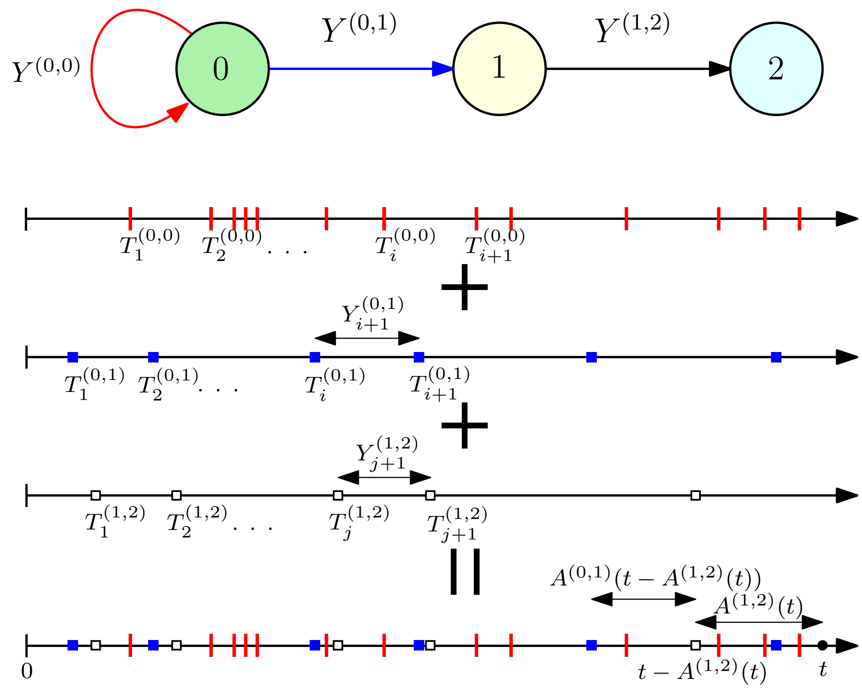


Figure 7.10: Superposition of source renewal process $N^{(0,0)}(t)$ on the two renewal processes, $N^{(0,1)}(t)$ and $N^{(0,1)}(t)$ in the two-hop model.

of the past.

Further, if we added a second node to this model, this results in the two-hop model of Fig. 7.10, where updates arrive at node k from node $k-1$ at times $T_i^{(k-1,k)}$ according to renewal process k, and similar to the case of traditional age of information metric, it becomes difficult to calibrate the different renewal processes with respect to each other for version age evaluation. That is, if we consider a typical inter-renewal interval $[T_i^{(0,1)}, T_{i+1}^{(0,1)}]$, then in the absence of memoryless inter-update times, the distribution of packet arrival instant at node 2 and the first source renewal update instant in this interval is dependent on when the last packet arrived at node 2 in previous inter-renewal intervals, which prevents us from characterizing the version age process at node 2 in this interval independently of the past. Similarly, if we were to consider an inter-renewal interval $[T_j^{(1,2)}, T_{j+1}^{(1,2)}]$, then the user version age at the beginning of this interval depends on the last packet arrival instant at node 1 in prior intervals and how much source renewal updates occurred since then. Therefore, the version age evolutions in different time intervals are correlated throughout the timeline, and in the following sections we aim to characterize the ensemble average of the version age at the end user in the regime of large t, by employing a pre-limit refinement of Blackwell's theorem [120] using a semi-martingale representation of a renewal processes.

7.5 Model and Notations for Version Age Analysis

The source receives version updates according to a renewal process $N^{(0,0)}(t)$ and packets from node i arrive at node j on link (i,j) according to a renewal process $N^{(i,j)}(t)$. The corresponding finite random times $0 \leq T_1^{(i,j)} \leq T_2^{(i,j)} \leq \ldots$ denote the renewal times, such that the inter-arrival times $Y_n^{(i,j)} = T_n^{(i,j)} - T_{n-1}^{(i,j)}$ are positive i.i.d. random variables with common distribution $F^{i,j}$, which is assumed to be non-arithmetic with finite first and second moments. Given $N^{(i,j)}(t) = \max\{n : T_n^{(i,j)} \leq t\}$, the regenerative process $A^{(i,j)}(t) = t - T_{N^{(i,j)}(t)}^{(i,j)}$ denotes the corresponding backward recurrence time (or age of renewal process) at time t, which is the time since the last renewal prior to t. Likewise, the regenerative process $B^{(i,j)}(t) = T_{N^{(i,j)}(t)+1}^{(i,j)} - t$ denotes the corresponding forward recurrence time (or residual renewal time of renewal process) at time t, which is the time to the next renewal after t.

Consider a typical node j in an arbitrary network of nodes and let S_j denote the set of nodes from which packets arrive at node j. The most recent packet from node $i \in S_j$ arrives at node j before time t at time instant $t - A^{(i,j)}(t) = T^{(i,j)}_{N^{(i,j)}(t)}$, at which point, node j compares the version number of the arriving packet with the packet present at its cache, and discards the staler version in favor of the fresher version.

Let $X_j(t)$ denote the instantaneous version age of information at node j at time t. Then, $X_j(t)$ can be written as

$$\begin{aligned} X_j(t) = \sum_{i \in S_j} \Big[\prod_{k \in S_j \setminus \{i\}} \chi_{\{A^{(i,j)}(t) < A^{(k,j)}(t)\}} \Big] \\ \times \Big[\min\big\{X_i(t - A^{(i,j)}(t)), X_j(t - A^{(i,j)}(t)\big\} \\ + N^{(0,0)}(t) - N^{(0,0)}(t - A^{(i,j)}(t)) \Big] \end{aligned} \tag{7.37}$$

where $\chi_{\mathcal{A}}$ represents the indicator random variable for the measurable set $\mathcal{A}$ and $X_j(0) = 0$. Since the source always has the latest packet, $X_0(t) = 0$ at all times.

In (7.37), $\prod_{k \in S_j \setminus \{i\}} \chi_{\{A^{(i,j)}(t) < A^{(k,j)}(t)\}}$ corresponds to the scenario when the last packet that arrived at node j before time t came from node i, which would be the case when the backward recurrence times of all other relevant renewal processes at time t are larger than $A^{(i,j)}(t)$. The last term $N^{(0,0)}(t) - N^{(0,0)}(t - A^{(i,j)}(t))$ in (7.37) comes from the fact that version age at node j increments by one every time the source gets updated post the last packet arrival at $t - A^{(i,j)}$.

In the next step, $\min\{X_i(t - A^{(i,j)}(t)), X_j(t - A^{(i,j)}(t)\}$, which is a minimum over two age processes, can be further characterized in a manner similar to (7.37) by accounting for arrivals at nodes $\{i, j\}$ from the set $S_i \cup S_j$, and the corresponding expression will have terms of the form that involve taking a minimum over three version age processes, in addition to counting number of source renewals between relevant updates. By recursively repeating this process we finally encounter the expression $\min\{X_1(t''), X_2(t''), \ldots, X_n(t'')\}$, $t'' = t - \Delta(t)$, where $\Delta(t)$ represents a stochastic process whose exact expression depends on the network topology. Since the source node is the only node external to the set of n nodes and the version age at source is always zero, the last min expression can be completely defined in terms of the number of source renewals in time intervals that are backward recurrence times of the form $A^{(0,\ell)}(t''')$ for all ℓ with $0 \in S_\ell$. This recursive approach will become more clear for multi-cast networks in Section 7.6. Similar to Section 7.2, we encounter terms, corresponding to the same renewal process, correlated through time which complicates analysis.

However, when each node in the network has only one incoming link, as shown in the tree network of Fig. 7.7, then $|S_j| = 1$, and the product term in (7.37) vanishes. Additionally, as packets now arrive at node j from a single preceding node i, $X_i(t) \leq X_j(t)$ for all t, this simplifies the min term as

$$\min\left\{X_i(t - A^{(i,j)}(t)), X_j(t - A^{(i,j)}(t))\right\} = X_i(t - A^{(i,j)}(t)) \tag{7.38}$$

In the next section, we derive a closed form expression for the long-term expected version age $\lim_{t\to\infty} \mathbb{E}[X_j(t)]$ at each node j in networks that have a tree topology.

7.6 Version Age in Networks with Tree Structure

To facilitate the exposition of the derivation of version age in general tree networks, we again initially begin by examining the special cases of one-hop and two-hop networks, which serve to motivate the necessity of lemmas proved in the work.

7.6.1 One-Hop Network

We consider a single-hop network, where the source gets updated according to a renewal process that is not necessarily Poisson; Fig. 7.9. At time t, the last packet arrival at node 1 from node 0 happens at time $t - A^{(0,1)}(t)$, thus, the instantaneous version age at node 1 depends on the number of version updates at the source in the interval $(t - A^{(0,1)}(t), t]$, giving

$$X_1(t) = N^{(0,0)}(t) - N^{(0,0)}(t - A^{(0,1)}(t)) + X_0(t - A^{(0,1)}(t)) \tag{7.39}$$

Lemma 7.2 *Let $N(t)$ be a renewal process with i.i.d. inter-renewal times, denoted by typical random variable Y, with non-arithmetic distribution and finite first and second moments. Let $S(t)$ be a stochastic process that is independent of $N(t)$, such that, $0 \leq S(t) \leq t$ and $\lim_{t\to\infty} \mathbb{E}[S(t)]$ exists. Then,*

$$\lim_{t\to\infty} \mathbb{E}[N(t) - N(t - S(t))] = \frac{\lim_{t\to\infty} \mathbb{E}[S(t)]}{\mathbb{E}[Y]} \tag{7.40}$$

Proof: Taking $\mu = \mathbb{E}[Y]$, [120] provides the following semi-martingale representation for a renewal process $N(t)$,

$$N(t) = \frac{t + B(t)}{\mu} + M(t) \tag{7.41}$$

where $B(t)$ is the forward recurrence time associated with the renewal process $N(t)$ and $M(t) = N(t) + 1 - \frac{T_{N(t)+1}}{\mu}$ is a martingale. Let $m(t) = \mathbb{E}[N(t)]$ and $b(t) = \mathbb{E}[B(t)]$, then since $\mathbb{E}[Y] < \infty$ and $\mathbb{E}[Y^2] < \infty$, renewal reward theorem [116] gives $\lim_{t\to\infty} b(t) = \frac{\mathbb{E}[Y^2]}{2\mathbb{E}[Y]} = b_1$. This implies that there exists T, such that, for all $t \geq T$, $b(t) < b_1 + \epsilon$, for some $\epsilon > 0$. Further, for any $t < T$, $B(t) \leq (T - t) + B(T) \leq T + b_1 + \epsilon$, in the worst case, no renewal occurs in the time interval (t, T) which leads to $B(t) = (T - t) + B(T)$. Hence,

$$\sup_{t\geq 0} |b(t)| \leq T + b_1 + \epsilon < \infty \tag{7.42}$$

Using the fact that $\mathbb{E}[M(t)] = \mathbb{E}[M(0)] = 0$ for all t (see Wald identity, [116]) since $M(t)$ is a martingale, we get

$$\mathbb{E}[N(t) - N(t - S(t))] = \mathbb{E}\left[\frac{t + B(t)}{\mu} - \frac{t - S(t) + B(t - S(t))}{\mu}\right] \tag{7.43}$$

$$= \mathbb{E}\left[\frac{S(t) + B(t) - B(t - S(t))}{\mu}\right] \tag{7.44}$$

Taking limit $t \to \infty$ on both sides and using Lemma 7.1 with $S_1(t) = B(t)$ and $S_2(t) = S(t)$ gives the desired result. ■

Coming back to computing $\lim_{\to\infty} \mathbb{E}[X_1(t)]$ in Fig. 7.9, using (7.39) and Lemma 7.2

along with $X_0(t - A^{(0,1)}(t)) = 0$, we have

$$\lim_{t\to\infty} \mathbb{E}[X_1(t)] = \frac{\lim_{t\to\infty} \mathbb{E}[A^{(0,1)}(t)]}{\mathbb{E}[Y^{(0,0)}]} = \frac{\mathbb{E}[\left(Y^{(0,1)}\right)^2]}{2\mathbb{E}[Y^{(0,0)}]\mathbb{E}[Y^{(0,1)}]} \tag{7.45}$$

Remark 7.5 *Note that if both of the processes in Fig. 7.9 are Poisson, then the expected version age at the user node is known to be* $\frac{\lambda_s}{\lambda}$ *[5, 51]. Therefore, it is interesting to note that* λ_s *here is the proxy for* $\frac{1}{\mathbb{E}[Y^{(0,0)}]}$*, while* λ *is the proxy for* $\frac{2\mathbb{E}[Y^{(0,1)}]}{\mathbb{E}[\left(Y^{(0,1)}\right)^2]}$*.*

7.6.2 Two-Hop Network

Consider the two-hop network in Fig. 7.10 where we wish to determine the long-term expected version age at node 2. Then, the instantaneous version age $X_2(t)$ can be written as

$$X_2(t) = N^{(0,0)}(t) - N^{(0,0)}(t - A^{(1,2)}(t)) + X_1(t - A^{(1,2)}(t)) \tag{7.46}$$

where $X_1(t - A^{(1,2)}(t))$ in turn can be expressed as

$$\begin{aligned} X_1(t - A^{(1,2)}(t)) =& N^{(0,0)}(t - A^{(1,2)}(t)) - N^{(0,0)}(t - A^{(1,2)}(t) - A^{(0,1)}(t - A^{(1,2)}(t))) \\ &+ X_0(t - A^{(1,2)}(t) - A^{(0,1)}(t - A^{(1,2)}(t))) \end{aligned} \tag{7.47}$$

Let us define

$$\Delta_1(t) = A^{(1,2)}(t) \tag{7.48}$$

$$\Delta_2(t) = A^{(0,1)}(t - A^{(1,2)}(t)) \tag{7.49}$$

Substituting (7.47), (7.48) and (7.49) in (7.46) and using $X_0(t-\Delta_1(t)-\Delta_2(t)) = 0$, we get

$$\begin{aligned} X_2(t) =& N^{(0,0)}(t) - N^{(0,0)}(t-\Delta_1(t)) + N^{(0,0)}(t-\Delta_1(t)) \\ & - N^{(0,0)}(t-\Delta_1(t)-\Delta_2(t)) && (7.50) \\ =& N^{(0,0)}(t) - N^{(0,0)}(t-\Delta_1(t)-\Delta_2(t)) && (7.51) \end{aligned}$$

To compute the expectation in (7.51), $\lim_{t\to\infty}\mathbb{E}[X_2(t)]$, we use Lemma 7.2, which requires computing the terms $\mathbb{E}[\Delta_1(t)]$ and $\mathbb{E}[\Delta_2(t)]$ at $t \to \infty$. The backward recurrence time $A^{(1,2)}(t)$ has the following limiting expectation [116]

$$\lim_{t\to\infty}\mathbb{E}[\Delta_1(t)] = \lim_{t\to\infty}\mathbb{E}[A^{(1,2)}(t)] = \frac{\mathbb{E}\left[\left(Y^{(1,2)}\right)^2\right]}{2\mathbb{E}\left[Y^{(1,2)}\right]} \tag{7.52}$$

Likewise, we have

$$\lim_{t\to\infty}\mathbb{E}[A^{(0,1)}(t)] = \frac{\mathbb{E}\left[\left(Y^{(0,1)}\right)^2\right]}{2\mathbb{E}\left[Y^{(0,1)}\right]} \tag{7.53}$$

Since the limit $\lim_{t\to\infty}\mathbb{E}[A^{(0,1)}(t)]$ exists, there exists T such that for all $t > T$, $\mathbb{E}[A^{(0,1)}(t)] < \lim_{t\to\infty}\mathbb{E}[A^{(0,1)}(t)] + \epsilon$ for some $\epsilon > 0$. Further, since $0 \le A^{(0,1)}(t) \le t$

by definition, we have $\mathbb{E}[A^{(0,1)}(t)] < T$ for $t \leq T$. Hence,

$$\sup_{t\geq 0} \left|\mathbb{E}[A^{(0,1)}(t)]\right| \leq \max\left\{\lim_{t\to\infty} \mathbb{E}[A^{(0,1)}(t)] + \epsilon, T\right\} < \infty \tag{7.54}$$

Hence, by Lemma 7.1,

$$\lim_{t\to\infty} \mathbb{E}[\Delta_2(t)] = \lim_{t\to\infty} \mathbb{E}[A^{(0,1)}(t)] = \frac{\mathbb{E}\left[\left(Y^{(0,1)}\right)^2\right]}{2\mathbb{E}\left[Y^{(0,1)}\right]} \tag{7.55}$$

Since $0 \leq \Delta_1(t) + \Delta_2(t) \leq t$ and $\lim_{t\to\infty} \mathbb{E}[\Delta_1(t) + \Delta_2(t)]$ exists, by Lemma 7.2 the long-term expected version age at node 2 is

$$\lim_{t\to\infty} \mathbb{E}[X_2(t)] = \frac{\lim_{t\to\infty} \mathbb{E}[\Delta_1(t) + \Delta_2(t)]}{\mathbb{E}[Y^{(0,0)}]} \tag{7.56}$$

$$= \frac{1}{\mathbb{E}[Y^{(0,0)}]} \left(\frac{\mathbb{E}\left[\left(Y^{(1,2)}\right)^2\right]}{2\mathbb{E}\left[Y^{(1,2)}\right]} + \frac{\mathbb{E}\left[\left(Y^{(0,1)}\right)^2\right]}{2\mathbb{E}\left[Y^{(0,1)}\right]} \right) \tag{7.57}$$

Hence, the version age at node 2 is determined by the sum of independent contributions of links in the path from node 0 to node 2, divided by $\mathbb{E}[Y^{(0,0)}]$. In general, for tree networks, only the links involved in the path between the source and an end-user are critical to the version age dynamics of the end-user apart from the update process at the source, thus, in the next subsection, we study n-hop linear networks.

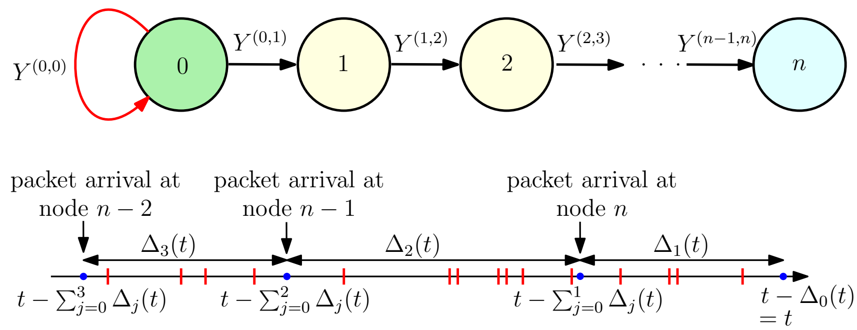


Figure 7.11: Time segments $\Delta_i(t)$ for $i \in \{1, \ldots, n\}$ in n-hop model. The points marked by red represent version updates at the source.

7.6.3 Multi-Hop Network

Consider the n-hop network of Fig. 7.11 where we wish to determine the long-term expected version age at node n of the network and time segments $\Delta_i(t)$ are defined as in (7.25). Similar to (7.46), the instantaneous age $X_n(t)$ at node n can be written as

$$X_n(t) = N^{(0,0)}(t) - N^{(0,0)}(t - A^{(n-1,n)}(t)) + X_{n-1}(t - A^{(n-1,n)}(t)) \tag{7.58}$$

This can be alternately represented as

$$X_n(t-\Delta_0(t)) = N^{(0,0)}(t) - N^{(0,0)}(t-\Delta_0(t)-\Delta_1(t)) + X_{n-1}(t-\Delta_0(t)-\Delta_1(t)) \tag{7.59}$$

In the next step, $X_{n-1}(t-A^{(n-1,n)}(t))$ of (7.58) will be again characterized in a similar manner and the full set of equations encountered in this recursive approach is of the form

$$X_{n-i}(t-\sum_{j=0}^{i}\Delta_j(t)) =N^{(0,0)}(t-\sum_{j=0}^{i}\Delta_j(t))-N^{(0,0)}(t-\sum_{j=0}^{i+1}\Delta_j(t))$$
$$+X_{n-i-1}(t-\sum_{j=0}^{i+1}\Delta_j(t)) \tag{7.60}$$

for $0\leq i\leq n-1$ with $X_0(t-\sum_{j=1}^{n}\Delta_j(t))=0$ as node 0 represents the source node. Then, it follows from (7.60) that

$$X_n(t)=N^{(0,0)}(t)-N^{(0,0)}(t-\sum_{j=1}^{n}\Delta_j(t)) \tag{7.61}$$

Similar to (7.52), we have

$$\lim_{t\to\infty}\mathbb{E}[\Delta_1(t)]=\lim_{t\to\infty}\mathbb{E}[A^{(n-1,n)}(t)]=\frac{\mathbb{E}\left[\left(Y^{(n-1,n)}\right)^2\right]}{2\mathbb{E}\left[Y^{(n-1,n)}\right]} \tag{7.62}$$

Further, using the approach of (7.54), we get $\sup_{t\geq 0}\left|\mathbb{E}[A^{(n-i,n-i+1)}(t)]\right|<\infty$. Since $\sum_{j=0}^{i-1}\Delta_j(t)\leq t$, we can prove $\lim_{t\to\infty}\mathbb{E}[\Delta_i(t)]=\lim_{t\to\infty}\mathbb{E}[A^{(n-i,n-i+1)}(t)]$ recursively for $i=2,3,\ldots,n$ from (7.25) using Lemma 7.1.

Hence, from (7.61), we obtain

$$\lim_{t\to\infty}\mathbb{E}[X_n(t)] =\frac{\sum_{j=1}^{n}\lim_{t\to\infty}\mathbb{E}[\Delta_j(t)]}{\mathbb{E}[Y^{(0,0)}]}=\frac{1}{\mathbb{E}[Y^{(0,0)}]}\sum_{j=1}^{n}\frac{\mathbb{E}\left[\left(Y^{(n-j,n-j+1)}\right)^2\right]}{2\mathbb{E}\left[Y^{(n-j,n-j+1)}\right]} \tag{7.63}$$

Remark 7.6 *We observe that similar to the traditional age metric, the version age* *at node n depends on independent contributions of the intermediate links $(i, i+1)$, $0 \leq i \leq n-1$, and is invariant to the ordering of these links, giving each node the ability to minimize its version age by optimizing its individual packet request renewal process. Since the constant random variable has zero variance, for a given average inter-update time $\mathbb{E}[Y^{(n-j,n-j+1)}]$, nodes should request packets at near constant time intervals to reduce variance and consequently version age, from (7.63). Further, the version age at node n is inversely proportional to $\mathbb{E}[Y^{(0,0)}]$, implying the version age at nodes would be larger for a fast updating source on average. Further, if all renewal processes in Fig. 7.11 were Poisson, with λ_s as the source update rate and λ_j as the link $(n-j, n-j+1)$ update rate, respectively, then (7.63) simplifies to*

$$\lim_{t\to\infty} \mathbb{E}[X_n(t)] = \lambda_s \sum_{j=1}^{n} \frac{1}{\lambda_j} \tag{7.64}$$

which also results from [5, Thm. 1] or [112, Eqn. (11)] with interesting parallelism to [109, Thm. 2].

7.7 Age of information in Networks with Rare Inter-node Updates

The Poisson limit theorem for partitions of point process (restated from [115] for convenience of the reader) studies the Poisson convergence of a renewal point process $N(t)$ that is partitioned by the rule: Each point of N is assigned to a subprocess $\ell \in I$ (a countable set) with partitioning probability $p(\ell)$, independent of everything else, where $\sum_{\ell\in I} p(\ell) = 1$, such that $N = \sum_{\ell\in I} N_\ell$, where N_ℓ denotes the subprocess

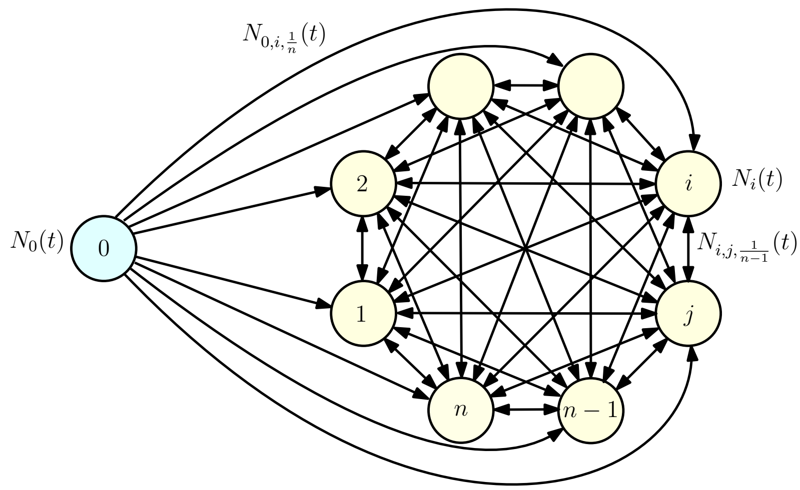


Figure 7.12: Fully connected network of n nodes, where node i transmits update packets according to $N_i(t)$ renewal process.

ℓ. To ensure the subprocesses are sparse, the partitioning probabilities are assumed to be function of n, i.e., $p_n(\ell)$, satisfying $\lim_{n\to\infty} p_n(\ell) \to 0$, and the corresponding subprocesses are denoted by $N_{n\ell}(t)$. Under the following conditions, the subprocesses $N_{n\ell}$ can be approximated by independent Poisson processes: *Condition 1.* There exists a positive constant α such that $\frac{N(t)}{t} \xrightarrow{d} \alpha$ in distribution. *Condition 2.* There exist positive constants $r(\ell)$ and $a_n \to \infty$ such that $\lim_{n\to\infty} a_n p_n(\ell) = r(\ell)$, $\ell \in I$.

Under these conditions, the process $N_{n\ell}(a_n t)$, obtained from $N_{n\ell}(t)$ under a rescaling of time by factor of a_n, converges in distribution to a Poisson process with rate $r(\ell)\alpha$. This convergence theorem justifies that we can reasonably approximate $N_{n\ell}(t)$ by a Poisson process with rate $p_n(\ell)\alpha$ when $p_n(\ell)$ is small or n is large [115]. In the next subsection, we use Poisson limits of partitions to examine age scaling of fully connected networks.

7.7.1 Investigating Fully Connected Networks Via Sparse Updates in Large Networks

Consider a large fully connected network consisting of a source node and n nodes, denoted by set $\mathcal{N} = \{1, \ldots, n\}$, see Fig.7.12. Node i transmits update packets according to a renewal point process $N_i(t)$ with typical inter-renewal period Y_i, a random variable with distribution F_i. Source node (alternatively referred to as node 0), independently at each point of $N_0(t)$, uniformly at random chooses a node in $\mathcal{N}$ as the destination node for the packet to be transmitted. Since each node $i \in \mathcal{N}$ gets chosen as the destination node with probability $\frac{1}{n}$, the packets arrive from node 0 at node i according to a $\frac{1}{n}$-thinned sub-renewal process of $N_0(t)$, which we denote by $N_{0,i,\frac{1}{n}}(t)$. Likewise, node $i \in \mathcal{N}$, independently at each point of $N_i(t)$, uniformly at random chooses a node $j \in \mathcal{N}\backslash\{i\}$ as the destination node for the update to be transmitted, and consequently, packets arrive from node i at node j according to a $\frac{1}{n-1}$-thinned sub-renewal process of $N_i(t)$, denoted by $N_{i,j,\frac{1}{n-1}}(t)$. Due to the discussions of Section 7.2, it is not trivial to derive the expected age at a network node with multiple incoming links when the update processes on those links are not Poisson. However, in large fully connected networks, all sub-renewal processes become sparse in the regime of large n, and therefore, converge to Poisson processes.

For process $N_0(t)$, *Condition 1* is satisfied since by the Blackwell renewal theorem, we have $\frac{N_0(t)}{t} \to \frac{1}{\mathbb{E}[Y_0]}$, and *Condition 2* is satisfied by taking $p_n(\ell) = \frac{1}{n}$ and $a_n = n$. Hence, $N_{0,i,\frac{1}{n}}(t)$ can be approximated by an independent Poisson process with rate $\frac{1}{n\mathbb{E}[Y_0]}$ for all $i \in \mathcal{N}$. Similarly, $N_{i,j,\frac{1}{n-1}}(t)$ can be approximated by an independent Poisson process with rate $\frac{1}{(n-1)\mathbb{E}[Y_i]}$ for all $j \in \mathcal{N}$. When all the updates on all the links are sent according to Poisson processes, the expected age at any node can be derived by solving the recursive equation of [4, Thm. 1]. For the special case of symmetric fully connected networks, where all renewal processes N_i have the same distribution with mean of inter-arrival distribution denoted as $\mathbb{E}[Y]$, [4, Thm. 2] provides that the expected age at each node scales as $\mathbb{E}[Y] \times \left(\sum_{i=1}^{n} \frac{1}{k}\right) = O(\log n)$ for large n. Consequently, the expected age scales as $\log n$ even when all nodes transmit packets with non-Poisson renewal processes.

Remark 7.7 *Interestingly, unlike previous sections, the variance of the renewal processes do not play a role in the age of user nodes in large fully connected networks.*

7.8 Numerical Results

We first simulate the model in Fig. 7.5 for $n = 3$, i.e., a 3-hop model with links $(0, 1), (1, 2)$ and $(2, 3)$. We consider three different continuous probability distributions: Rayleigh with scale $\sigma = 1$, Chi-Square with degree of freedom $k = 1$, and Beta with shape parameters $\alpha = 2$, $\beta = 3$, i.e., Rayleigh(1), $\chi^2(1)$ and Beta(2,3), for inter-update times, such that they have different means. For convenience, the mean and variance of these distributions are listed in Table 7.1, along with predicted contribution of the corresponding link to age of end-user $\frac{\mathbb{E}[Y^2]}{2\mathbb{E}[Y]}$.

Note that Beta is supported on bounded interval [0,1], while Chi-Square and Rayleigh are supported on semi-infinite interval $[0, \infty)$. Therefore, these are all non-negative probability distributions. There are $3! = 6$ ways to map the three

distribution Y	mean $\mathbb{E}[Y]$	variance $var[Y]$	age $\frac{\mathbb{E}[Y^2]}{2\mathbb{E}[Y]}$
Rayleigh(1)	$\frac{\sigma\sqrt{\pi}}{\sqrt{2}} = 1.25$	$\frac{(4-\pi)\sigma^2}{2} = 0.43$	0.7979
$\chi^2(1)$	$k = 1$	$2k = 2$	1.5
Beta(2,3)	$\frac{\alpha}{\alpha+\beta} = \frac{2}{5}$	$\frac{\alpha\beta(\alpha+\beta)^{-2}}{(\alpha+\beta+1)} = \frac{1}{25}$	0.25

Table 7.1: Distributions, means, variances, and contributions to age.

distributions to the three links (Table 7.2).

link (0,1)	link (1,2)	link (2,3)	$\mathbb{E}[X_n(T)]$
Rayleigh(1)	$\chi^2(1)$	Beta(2,3)	2.5487
Rayleigh(1)	Beta(2,3)	$\chi^2(1)$	2.5495
$\chi^2(1)$	Rayleigh(1)	Beta(2,3)	2.5449
$\chi^2(1)$	Beta(2,3)	Rayleigh(1)	2.5480
Beta(2,3)	$\chi^2(1)$	Rayleigh(1)	2.5453
Beta(2,3)	Rayleigh(1)	$\chi^2(1)$	2.5481

Table 7.2: Mapping of distributions to links.

For each mapping, we simulate the network for a large duration, $T = 10^3$. We take average of $X_n(T)$, which is the instantaneous age at $T = 10^3$, over 2×10^5 iterations for each mapping to approximate $\mathbb{E}[X_n(T)]$ by the law of large numbers, which is used as a proxy for $\lim_{t\to\infty} \mathbb{E}[X_n(t)]$. Table 7.2 shows that $\mathbb{E}[X_n(T)]$ numerically obtained for all mappings is close to the theoretical value of $0.7979 + 1.5 + 0.25 = 2.5479$ suggested by (7.31), which is obtained by summing the link contributions of each distribution from Table 7.1.

Likewise, we also numerically evaluate the second moment of age at the end-user in the above model, such that $\mathbb{E}[X_n(T)]$ turns out to be 9.6331. The expression $\frac{\mathbb{E}[Y^3]}{3\mathbb{E}[Y]}$, with distribution Y as Rayleigh(1), $\chi^2(1)$ and Beta(2,3), turns out to be 1, 5 and $\frac{2}{21}$, respectively. Consequently, (7.35) evaluates to $1 + 5 + \frac{2}{21} + 2 \times 0.7979 \times$

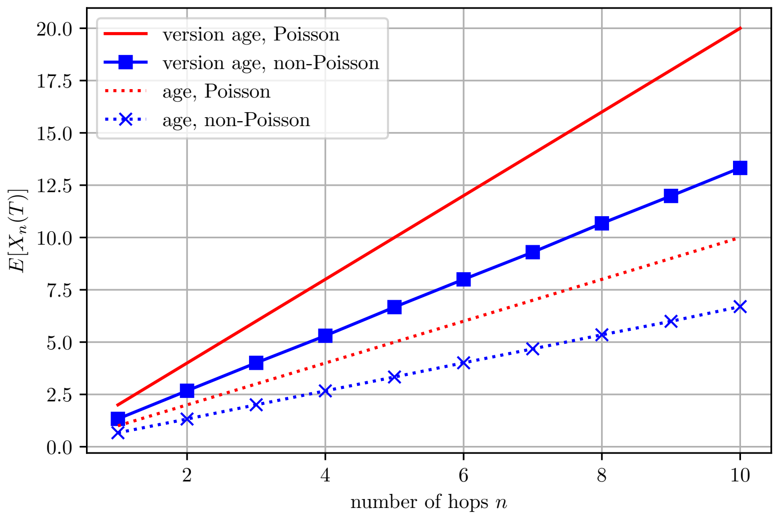


Figure 7.13: The dotted blue curve shows the expected age $\mathbb{E}[X_n(T)]$ in n-hop network with $Y^{(i,i+1)} \sim \mathcal{U}_{[0,2]}$. The solid blue curve shows the expected version age $\mathbb{E}[X_n(T)]$ in n-hop network with $Y^{(i,i+1)} \sim \mathcal{U}_{[0,2]}$ and $Y^{(0,0)} \sim Pareto(3, \frac{1}{3})$. The dotted red and solid red curves depict the expected age and expected version age, respectively, when all renewal processes are approximated as Poisson processes with the same distributional mean of update intervals on all links.

$1.5 + 2 \times 0.7979 \times 0.25 + 2 \times 1.5 \times 0.25 = 9.6378$, which is close to the numerically observed value, lending support to Remark 7.4.

Next, we simulate an n-hop network where update intervals of all links follow uniform distribution on the interval $[0, 2]$, i.e., $Y^{(i,i+1)} \sim Y \sim \mathcal{U}_{[0,2]}$, such that $\frac{\mathbb{E}[Y^2]}{2\mathbb{E}[Y]} = \frac{2}{3}$. We plot $\mathbb{E}[X_n(T)]$ as a function of n in Fig. 7.13 with dotted blue line. The linearity of the graph with the number of hops n in Fig. 7.13 demonstrates the additive structure of the age at the end-user as found in (7.31). Since all links have the same distribution for inter-update times, the graph in Fig. 7.13 follows a linear equation in n as $\lim_{t\to\infty} \mathbb{E}[X_n(t)] = \frac{2}{3}n$, as predicted by (7.31). Further, to highlight the significance of accurately identifying renewal processes and determining the expected age at users using the correct formulae derived in this work, we approximate all renewal processes as Poisson processes while keeping the distributional mean of

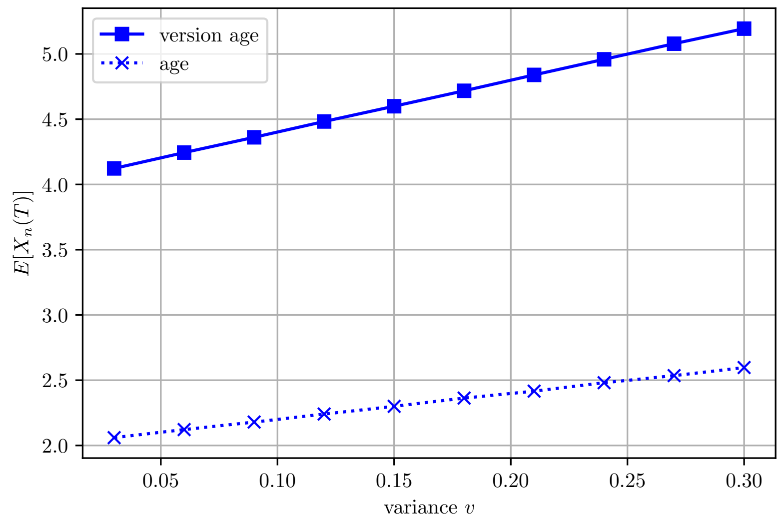


Figure 7.14: The dotted blue curve shows the expected age $\mathbb{E}[X_n(T)]$ in 4-hop network with $Y^{(i,i+1)} \sim \mathcal{U}_{[1-\sqrt{3v},1+\sqrt{3v}]}$. The solid blue curve shows the expected version age $\mathbb{E}[X_n(T)]$ in 4-hop network with $Y^{(i,i+1)} \sim \mathcal{U}_{[1-\sqrt{3v},1+\sqrt{3v}]}$ and $Y^{(0,0)} \sim Pareto(3, \frac{1}{3})$.

update intervals on all links the same. The expected age, $\mathbb{E}[X_n(T)]$, in the latter case is plotted as a dotted red line in Fig. 7.13, which deviates significantly from the correct expected age, shown as a dotted blue line in Fig. 7.13, as the number of hops increases.

Finally, we simulate a 4-hop network, where inter-update times on all links follow the uniform distribution $Y^{(i,i+1)} \sim Y \sim \mathcal{U}_{[1-\sqrt{3v},1+\sqrt{3v}]}$, such that $\mathbb{E}\left[Y\right] = 1$ and $var\left[Y\right] = v$. Note that for fixed mean 1, the maximum value of v is $\frac{1}{3}$ to ensure that the probability distribution has non-negative support. The dotted blue line of Fig. 7.14 shows that for fixed mean, $\mathbb{E}[X_n(T)]$ increases linearly with variance v, and $\lim_{t\to\infty} \mathbb{E}[X_n(t)] = 4 \times \frac{v+1}{2} = 2v + 2$ by (7.31), as discussed in Section 7.1 and Section 7.3.

Next, we present numerical results for version age in multi-hop networks.

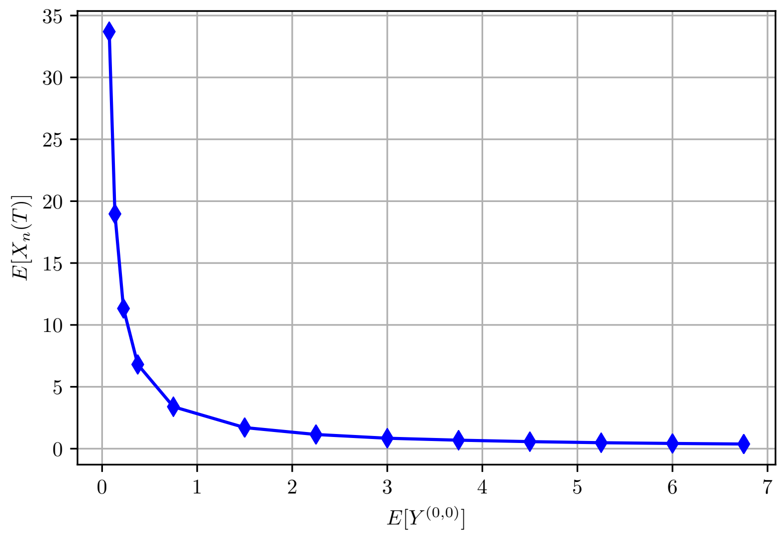


Figure 7.15: Expected version age $\mathbb{E}[X_n(T)]$ in 3-hop network with $Y^{(0,0)} \sim Pareto(3, m)$, $Y^{(0,1)} \sim Rayleigh(1)$, $Y^{(1,2)} \sim \chi^2(1)$ and $Y^{(2,3)} \sim Beta(2, 3)$, as plotted against different values of $\mathbb{E}[Y^{(0,0)}]$ obtained by varying m.

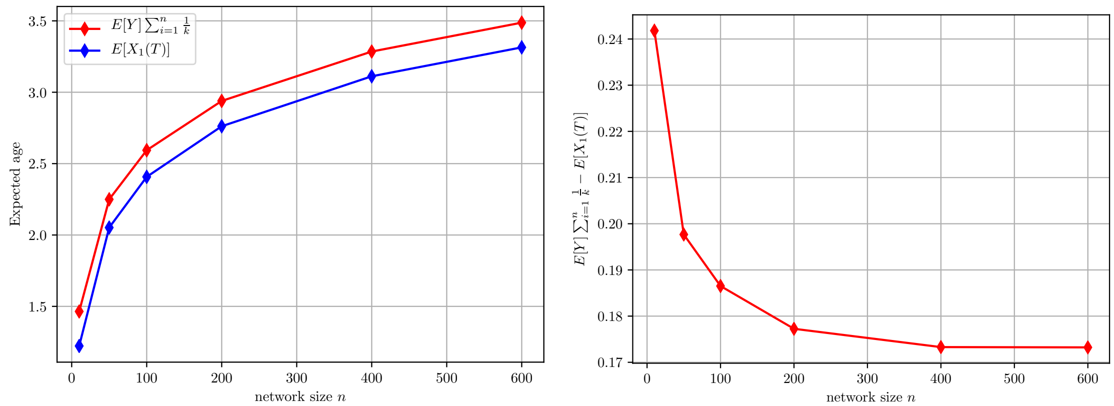
We update the source according to Pareto (Type I) distribution which has mean $\mathbb{E}[Y^{(0,0)}] = \frac{am}{a-1}$ for shape parameter a and scale parameter m. We first simulate the model in Fig. 7.11 for $n = 3$, i.e., a 3-hop model with links $(0, 1), (1, 2)$ and $(2, 3)$ following the inter-renewal distributions: Rayleigh with scale $\sigma = 1$, Chi-Square with degree of freedom $k = 1$, and Beta with shape parameters $\alpha = 2$, $\beta = 3$, respectively. We take average of $X_n(T)$, instantaneous version age at time T, over 2×10^5 iterations to approximate $\mathbb{E}[X_n(T)]$ by the law of large numbers, which is used as a proxy for $\lim_{t\to\infty} \mathbb{E}[X_n(t)]$. Fig. 7.15 shows the plot of $\mathbb{E}[X_n(T)]$ as a function of $\mathbb{E}[Y^{(0,0)}]$, obtained by varying scale parameter m while keeping $a = 3$ in Pareto distribution. The plot supports the theoretical prediction of $\frac{2.5479}{\mathbb{E}[Y^{(0,0)}]}$ from (7.63).

Next, we simulate an n-hop network where update intervals of all links $(i, i+1)$ follow uniform distribution on the interval $[0, 2]$, i.e., $Y^{(i,i+1)} \sim Y \sim \mathcal{U}_{[0,2]}$, such that

$\frac{\mathbb{E}[Y^2]}{2\mathbb{E}[Y]} = \frac{2}{3}$, and the source gets updated according to Pareto (Type I) distribution with $a = 3$ and $m = \frac{1}{3}$, giving $\mathbb{E}[Y^{(0,0)}] = 0.5$. We plot $\mathbb{E}[X_n(T)]$ as a function of n in Fig. 7.13 with solid blue line. The linearity of the graph with the number of hops n in Fig. 7.13 demonstrates the additive structure of the age at the end user as found in (7.63). Since all links have the same distribution for inter-update times, the graph in Fig. 7.13 follows a linear equation in n as $\lim_{t\to\infty} \mathbb{E}[X_n(t)] = \frac{4}{3}n$, as predicted by (7.63), which deviates significantly from the Poisson approximation-based expected age plotted as the solid red curve in Fig. 7.13.

Finally, we simulate a 4-hop network, where inter-update times on all links $(i, i+1)$ follow the uniform distribution $Y^{(i,i+1)} \sim Y \sim \mathcal{U}_{[1-\sqrt{3v},1+\sqrt{3v}]}$, such that $\mathbb{E}[Y] = 1$ and $Var[Y] = v$, with $Y^{(0,0)} \sim Pareto(3, \frac{1}{3})$, $\mathbb{E}[Y^{(0,0)}] = 0.5$. Note that for fixed mean 1, the maximum value of v is $\frac{1}{3}$ to ensure that the probability distribution has non-negative support. The solid blue line of Fig. 7.14 shows that for fixed mean, $\mathbb{E}[X_n(T)]$ increases linearly with variance v, with $\lim_{t\to\infty} \mathbb{E}[X_n(t)] = \frac{1}{0.5} \times (4 \times \frac{v+1}{2}) = 4v + 4$, as predicted by (7.63).

Next, we present numerical results for the expected age of information in a symmetric fully connected network of n nodes, where all nodes transmit update packets according to a renewal process with typical inter-renewal period $Y \sim \mathcal{U}_{[0,1]}$. In Fig. 7.16(a), the red curve corresponds to $\mathbb{E}[Y] \times \left(\sum_{i=1}^{n} \frac{1}{k}\right)$, which is the theoretically predicted expected age at each node for large n from Section 7.7. Since the expected age is the same at all nodes in a symmetric network, we plot expected age at node 1, i.e., $\mathbb{E}[X_1(T)]$, in blue color on the same plot. Fig. 7.16(b) shows the



(a) (b)

Figure 7.16: (a) The blue curve shows the expected age at a node 1 in a symmetric fully connected network where all update processes have inter-update times distributed as $\mathcal{U}_{[0,1]}$. The red curve shows the predicted expected $\mathbb{E}[Y] \times \left(\sum_{i=1}^{n} \frac{1}{k}\right)$ for large n in a fully connected network. (b) The gap $\mathbb{E}[Y] \times \left(\sum_{i=1}^{n} \frac{1}{k}\right) - \mathbb{E}[X_1(T)]$ decreases as n becomes large.

difference between the red and blue curves of Fig. 7.16(a), where we see that the difference $\mathbb{E}[Y] \times \left(\sum_{i=1}^{n} \frac{1}{k}\right) - \mathbb{E}[X_1(T)]$ decreases as n becomes larger.

7.9 Conclusion

In this chapter, we first studied the age of information and the version age of information metrics in cache-aided networks with tree structures where updates on all links are sent according to ordinary renewal processes that are not necessarily Poisson. We showed that the difficulty in age characterization at network nodes arising out of multiple correlated terms vanishes for the case of tree networks since each node has only one incoming link. We then provided recursive equations for instantaneous age and version age in multi-hop networks and used them to derive closed form expressions for expected age and expected version age at an end-user in tree networks. We observed that only the links involved in the path between the source and an end-user are relevant to the age dynamics of the end-user, and saw that both expected age and expected version age have additive structures composed of independent contributions of the intermediate links along the path. We saw that for a given update rate, the age of a node is directly proportional to the variance of inter-update times in these links, and therefore, nodes should request packets at near constant time intervals to increase their freshness. Finally, we used Poisson limits of partitions to study expected age of information in large fully connected networks for arbitrary i.i.d. inter-update times, and remarked that the expected age scales as $O(\log n)$ in symmetric fully connected networks.

CHAPTER 8

Strategic Profit Generation in Age-Based Systems

8.1 Introduction

As we transition into the 5G/6G era, we are experiencing a surge in real-time applications where the value of data is tied to its freshness. Examples of such applications could be navigation systems in vehicular networks, environmental monitoring systems, financial markets, or simply news channels that cater to people's desire to stay informed about the latest events and developments in their local area. Maintaining fresh data, however, can be rather expensive, as it involves frequent sampling of the underlying time-varying processes or events, in addition to processing and transmission-related operational costs. As a result, freshness becomes a valuable commodity in the commercial realm, where IoT service providers or news agencies can monetize these services by pricing the delivery of fresh information [122].

In this respect, we examine the profit maximization strategy at such a service provider (or simply a server), which tracks and publishes updates about a continually updating process or event E. A set of users $\mathcal{N}$ wishes to stay closely informed about the latest updates regarding the event and can explore two avenues: they can either

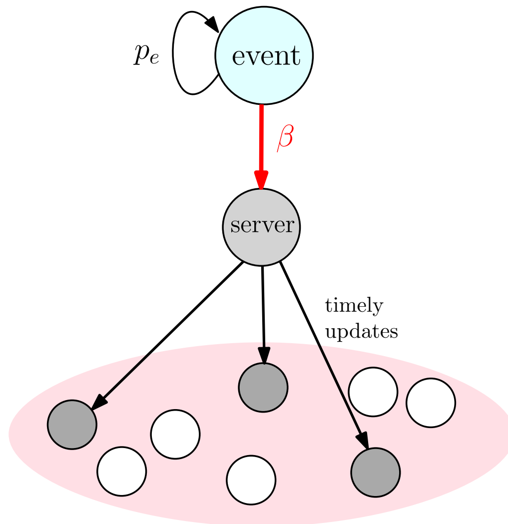


Figure 8.1: System model consisting of a server and user nodes. The event process is updated with probability p_e and the server gets updates with rate β.

directly subscribe to the server for timely updates, or they can rely on intermittent updates, i.e., gossip from their neighboring users, as long as this approach ensures that they remain reasonably informed about the event. In the latter case, the timeliness of a user would depend on the underlying network topology, and the multiplicity and proximity of subscribing nodes.

Prior works on gossip-based publish-subscribe systems [123, 124] have focused on a fixed message dissemination, as opposed to dynamic information dissemination as in this work. For instance, [123] analyzes the average number of interested users to which a fixed message percolates in random networks built using Poisson degree distribution. In [124], the server carefully manipulates a message to persuade users into taking a desired action without disincentivizing users from subscribing, particularly when the gossiping users might default to an undesirable action in the absence of such a message. In these works, with probability p, each user forwards the message to a neighbor, which is modeled using the Erdős-Rényi random graph approach where a link of the gossip network disappears with probability $1-p$. This prior line of work in [123,124] has important differences from real-time network models, where each user forwards its most recently received version to a neighbor with probability p in every timeslot, and thus, a particular version could potentially have multiple forwarding attempts on the same link. Further, under fixed message dissemination, a high gossip probability can ensure that a message reaches all users from only one subscribing user. However, a single subscriber might not be good enough for real-time systems since a non-subscriber situated far away from this subscriber may constantly receive outdated information due to multiple hops in information dissemination path, forcing it to opt for the first option of directly subscribing to the server. Hence, the portion of subscribers will be higher in real-time applications.

The set of questions we wish to investigate in this work is the following: Given a freshness-based gossip network, which users will choose to subscribe to the server, and how often should the server sample the event process to sustain a desired fraction of subscribing users from the population while keeping the cost of sampling the process at minimum in order to maximize its overall profit? The asymmetry in these two objectives leads to a non-cooperative game between the server and the users. Specifically, we model this as a Stackelberg game, where the server acts as the leader by committing to a sampling strategy for the event process. The users, on the other hand, act as followers who decide whether to subscribe or not to the server based on the server's chosen sampling strategy. We formalize the system model and study equilibrium subscription strategies for a range of gossip network topologies. We start with simpler network topologies, so that the results established in these initial topologies would be helpful in the analysis of more complex topologies later.

In the first half of this chapter, we study directed networks such as line, tree, and star networks, where updates take fixed directed paths from the source to the users. Here, each user node accepts all packets sent by its predecessor node, since in directed paths, nodes that are fewer hops away from the source are fresher at all times than users that are larger hops away. In the second half of the chapter, we extend our analysis to the more general case of undirected networks, where information flow is bidirectional between every pair of neighbors. In bidirectional networks, user nodes simultaneously receive packets from multiple nodes, all of which need not have a fresher version than the receiving user, causing the user to discard some of the arriving packets for maintaining freshness. Finally, we provide simulation results to confirm the theoretical findings and provide additional insights.

8.2 System Model and Age Analysis

We consider a communication system consisting of a server node R that samples a real-time data source or event E that is continuously updated and a set of user nodes $\mathcal{N} \subseteq \mathbb{Z}$ arranged into a gossip network; see Fig. 8.1. We represent the gossip network of user nodes through the directed graph $\overrightarrow{G} = (\mathcal{N}, \overrightarrow{E})$ with edge set $\overrightarrow{E}$, such that node i gossips with node j if there is an edge $e_{ij} \in \overrightarrow{E}$. Time is divided into discrete time slots, such that the event gets updated in each timeslot with probability p_e, with the time-interval between two successive event updates distributed as a geometric distribution with parameter p_e. We employ the version age of information metric [5, 51, 66] to characterize timeliness at nodes in this work as we associate a version number to every information about the event. Denoting the version number corresponding to the current state of the event by $V_E(t)$, each event update increments $V_E(t)$ by 1. Similarly, we denote the version number of the information present at node i by $V_i(t)$ and define the instantaneous version age at node i at time t as $X_i(t) = V_E(t) - V_i(t)$, capturing the notion of how much the user lags behind the state-of-the-art version of the event.

The server node gets an update about the event process with probability β during each timeslot t, resetting its version number $V_R(t+1) = V_E(t)$ at the beginning of the following timeslot post update reception. Let $U_{i,j}(t)$ indicate whether an update packet is transmitted from node i to node j at time t: $U_{i,j}(t) = 1$ implies that an update was sent such that the transmission consumed the whole timeslot t and was available to the receiving node j at the beginning of timeslot $t+1$, and $U_{i,j}(t) = 0$ implies that no packet was sent. Analogously, we use $U_{E,R}(t)$ and $U_E(t)$ to indicate whether an update was received at the server and a new event update occurred, respectively. When $U_E(t) = 1$, that is when a new event update occurs during timeslot t, the instantaneous versions of all other nodes increase by 1 in the next timeslot $t+1$ due to unit increment in $V_E(t+1)$ at the event.

8.2.1 Average Age Analysis at the Server

When an update packet arrives at the server at time t, the instantaneous version age at the server $X_R(t) = V_E(t) - V_R(t)$ drops to 0 in the next timeslot $(t+1)$ in the absence of any event updates. However, if a new event update simultaneously occurs at timeslot t, we assume that this update takes the whole timeslot to process and is not ready to be forwarded to the server within the same timeslot, in which case the instantaneous version age at the server increases by 1 in the next timeslot. In the absence of any event updates or packet arrivals at the server, the instantaneous version age at the server remains unchanged. Hence, the four combinations resulting from transitions $U_E(t)$ and $U_{E,R}(t)$ cause $X_R(t)$ to evolve as

$$\begin{aligned} X_R(t+1) =& 0 \times (1-U_E(t))U_{E,R}(t) + 1 \times U_E(t)U_{E,R}(t) \\ &+ X_R(t) \times (1-U_E(t))(1-U_{E,R}(t)) \\ &+ (X_R(t)+1) \times U_E(t)(1-U_{E,R}(t)). \end{aligned} \tag{8.1}$$

It is clear that (8.1) together with some initial distribution defines a discrete time Markov chain with state space $\mathbb{Z}_+$. Showing that the Markov chain consists of single irreducible countably infinite states that is non-periodic and also positive recurrent is straightforward given the Bernoulli strategies on all update links, and we skip these details since they are non-instructive.

Define $x_R = \lim_{t\to\infty}\mathbb{E}[X_R(t)]$. Note that $U_E(t)$ and $U_{E,R}(t)$ are independent random variables, with mean values p_e and β, respectively. Taking the long-term expectation on both sides of (8.1) gives

$$x_R = 0 \times (1-p_e)\beta + 1 \times p_e\beta + x_R \times (1-p_E)(1-\beta) + (x_R+1)p_e(1-\beta), \tag{8.2}$$

which simplifies to $x_R = \frac{p_e}{\beta}$.

When a user subscribes to the server, the server transmits the latest version it has at the beginning of each timeslot t to the subscribing user (referred to as the subscriber), such that the subscriber receives the complete packet by the start of the next timeslot $t+1$. Therefore, the instantaneous version age at a subscriber, denoted by $X_S(t)$ (with $x_S = \lim_{t\to\infty} \mathbb{E}[X_S(t)]$), is influenced by the version age at the server and any event updates in the preceding timeslot. Mathematically, this can be expressed as follows

$$X_S(t+1) = X_R(t) \times (1 - U_E(t)) + (X_R(t)+1) \times U_E(t). \tag{8.3}$$

Taking long-term expectation and using $x_R = \frac{p_e}{\beta}$, we get

$$x_S = x_R + p_e = p_e \left(\frac{1}{\beta} + 1 \right). \tag{8.4}$$

Now, we state the subscribing condition for the users. User i prefers a bounded expected age $x_i = \lim_{t\to\infty} \mathbb{E}[X_i(t)]$, i.e.,

$$x_i < L x_S, \tag{8.5}$$

for some $L \geq 1$, which we will call the *age compatibility (AC) constraint*. If the condition in (8.5) is satisfied without subscription, user i will not subscribe to the server, i.e., its subscription action will be $a_i = 0$; otherwise, user i will subscribe to the server, i.e., its subscription action will be $a_i = 1$. When a user is a non-subscriber, it can still maintain timely information by requesting packets from its neighbors in the network, which implies that the network topology determines the user's action to subscribe or not subscribe.

In a disconnected network, where each user is isolated from others, every user will have to decide to subscribe because non-subscription would result in an infinite expected age at the user, as they have no alternative way of getting updates other than from the server. For a general network structure, the following definition will be useful in identifying equilibrium strategies.

Definition 8.1 *For a given network structure and the server's update rate β, we say that a user is AC-stable if the user*

- *is a non-subscriber and its expected age satisfies (8.5); or*
- *is a subscriber and the alternate decision to unsubscribe while keeping other users' decisions unchanged would cause its expected age to violate (8.5).*

Remark 8.1 *Even though there is no cost associated with the subscription, due to the definition of AC-stable equilibrium, the users prefer not to subscribe if they can satisfy their age requirement through their neighbors.*

Thus, when the users are *AC-stable,* no user wants to deviate from their subscription decisions (which will be the notion of Nash equilibrium defined in the following subsection). Next, we look at certain network topologies to understand server-preferred subscription equilibrium strategies defined as, for a given strategy of the server, among all AC-stable equilibrium strategies, users are assumed to follow the one that will work best for the server.

8.2.2 Game Formulation

We next describe the non-cooperative game between the server and the users. The server wants to maximize the fraction of population that subscribes, denoted by $F_S = \frac{\sum_{i=1}^{|\mathcal{N}|} a_i}{|\mathcal{N}|}$ where $|\mathcal{N}|$ is the cardinality of the user set $\mathcal{N}$; to improve earnings, however, it also wishes to keep β low to minimize its cost of tracking the event process. Thus, the server's utility function for given β and the action vector $\boldsymbol{a} = [a_1, a_2, \cdots, a_{|\mathcal{N}|}]^T$ is given by the expression

$$L_R(\beta, \boldsymbol{a}) = F_S - c(\beta), \tag{8.6}$$

where $c(\beta)$ is the sampling cost with rate β. Here, we consider a general sampling cost $c(\beta)$ that is an increasing function of β. On the other hand, users take their actions based on their ability to satisfy the AC constraint of (8.5) as discussed earlier. In this game, the server R has the information structure $I_R = \{\overrightarrow{G}, p_e, p\}$, and the user $i \in \mathcal{N}$ has $I_i = \{\overrightarrow{G}, p_e, p, \beta, \boldsymbol{a}_{-i}\}$, respectively. Here, we use $\boldsymbol{a}_{-i}$ to denote the action of the users other than user i. Based on the information structure, server R commits to a strategy $\eta_R : I_R \to [0, 1]$. Similarly, let user i's strategy be $\gamma_i : I_i \to \{0, 1\}$. More specifically, in this work, we consider user i's action $a_i = \gamma_i(I_i) = \mathbf{1}\{x_{i,ns} > L x_S\}$ where $x_{i,ns}$ is the average age when user i does not subscribe to the server, for given β and $\gamma_{-i}(\beta)$.

Finally, the utility function of server R is given by

$$J_R(\eta_R; \gamma_1, \gamma_2, \ldots, \gamma_{|\mathcal{N}|}) = L_R(\beta, \boldsymbol{a}), \tag{8.7}$$

where the dependency of strategies to actions are given by $a_i = \gamma_i(I_i)$ and $\beta = \eta_R(I_S)$. In this game, we consider a hierarchical commitment where first the server R announces and commits to a strategy η_R, and then users take actions to satisfy the AC condition in (8.5). This structure leads to a Stackelberg-Nash game [**?**] between the server R and the users. More precisely, the server plays a hierarchical game against the N users, and the users play a Nash game among themselves to determine their best responses to the server's announced policy. We say that the set of strategies $(\eta_R^*, \{\gamma_i^*(\eta_R^*, \gamma_{-i}^*)\}_{i=1}^{|\mathcal{N}|})$ is in equilibrium if

$$J_R(\eta_R^*, \{\gamma_i^*(\eta_R^*, \gamma_{-i}^*(\eta_R^*))\}_{i=1}^{|\mathcal{N}|}) \geq J_R(\eta_R, \{\gamma_i^*(\eta_R, \gamma_{-i}^*(\eta_R))\}_{i=1}^{|\mathcal{N}|}) \tag{8.8}$$

$$\gamma_i^*(\eta_R, \gamma_{-i}^*(\eta_R)) = \mathbf{1}\{x_{i,ns} \geq Lx_S\}, \tag{8.9}$$

where user i's subscription strategy will be based on the term $x_{i,ns}$ affected by the server's strategy and other users' subscription strategies.[1] Thus, at the equilibrium, given the understanding of how the users' subscription strategies depend on β, the server will commit to $\beta^* = \eta_R^*(I_R)$ that will maximize its utility, and for each i the user adopting γ_i^* will not deviate from its action given the server's and

[1]Here, the Nash game is defined over the actions of the users instead of the user's utility function. We say that the Nash equilibrium is achieved when none of the users deviates from its action for a given strategy of the others. Note that the Nash equilibrium is achieved when all the users are AC-stable.

other users' subscription strategies. In this work, we will see that depending on the network structure, there might be multiple Nash equilibrium solutions with different subscription structures. In these scenarios, we assume that the users take the *server-preferred* subscription strategy.[2]

In the next section, we consider directed networks and find the equilibrium subscription strategies.

8.3 Directed Networks

We begin by analyzing equilibrium strategies for directed networks, where information can only flow in one direction between any two nodes i and j, that is, $e_{ij}e_{ji} = 0$. This analysis not only aids in building a strong intuition, but also provides tools useful for analyzing more complex networks.

8.3.1 Directed Line Networks

Consider the directed line network shown in Fig. 8.2. User 0 will have to subscribe because not subscribing would lead to an infinite expected age. Let us assume that node K decides to subscribe, where $K > 1$, such that nodes $\{1, \ldots, K-1\}$ do not subscribe. Given a specific value of β, we aim to determine the value of K for which all the users are AC-stable (or in Nash equilibrium), i.e., any subscribing or non-subscribing user would not wish to change its subscription decision contingent upon the decisions of the other users in the population being fixed. For any node

[2]Another approach could be to consider the worst subscription strategy for the server, and the goal could be to maximize its minimum possible utility.

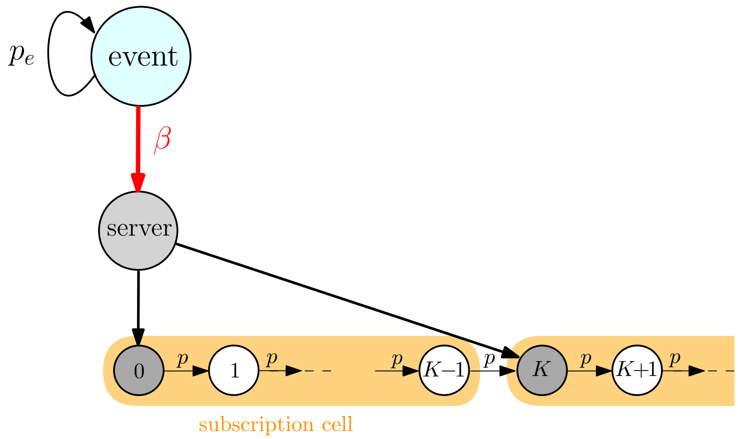


Figure 8.2: Directed line network where the users are subscribing to the server's information in every K users.

$k \in \{1, \dots, K-1\}$, similar to (8.1) and (8.3), we can characterize the instantaneous version age at node k from all combinations of relevant transitions $U_{k-1,k}(t)$ and $U_E(t)$ as follows

$$\begin{aligned} X_k(t+1) =& X_{k-1}(t) \times (1-U_E(t))U_{k-1,k}(t) \\ &+ X_k(t) \times (1-U_E(t))(1-U_{k-1,k}(t)) \\ &+ (X_{k-1}(t)+1) \times U_E(t)U_{k-1,k}(t) \\ &+ (X_k(t)+1) \times U_E(t)(1-U_{k-1,k}(t)), \end{aligned} \tag{8.10}$$

which, similar to (8.2) and (8.4), yields

$$x_k = x_{k-1}(1-p_e)p + x_k(1-p_e)(1-p) + (x_{k-1}+1)p_e p + (x_k+1)p_e(1-p). \tag{8.11}$$

From (8.11), we obtain the following recursive relation

$$x_k = x_{k-1} + \frac{p_e}{p}. \tag{8.12}$$

Since user 0 is a subscriber node, from (8.4), we have

$$x_0 = x_S = p_e \left(\frac{1}{\beta} + 1 \right), \tag{8.13}$$

and therefore, expected age at node k can be computed by using (8.12), (8.13) as

$$x_k = x_0 + k\frac{p_e}{p} = p_e \left(\frac{1}{\beta} + 1 \right) + k\frac{p_e}{p}, \tag{8.14}$$

such that x_k is inversely proportional to gossiping frequency p and server effort β. Note that the expected age increases with distance from the previous subscriber node which is node 0, i.e., $x_1 < \ldots < x_{K-1}$, and thus, variable K (the index of the next subscriber node after node 0) must satisfy

$$x_{K-1} < Lx_S \leq \tilde{x}_K, \tag{8.15}$$

where we use $\tilde{x}_i$ to denote the expected age at node i if it chooses an alternate subscription decision. The subscriber node K would be AC-stable if the alternative decision to unsubscribe would lead to expected age $\tilde{x}_K = x_0 + K\frac{p_e}{p}$, which would cause it to no longer satisfy the AC constraint of (8.5). For a given β, this critical

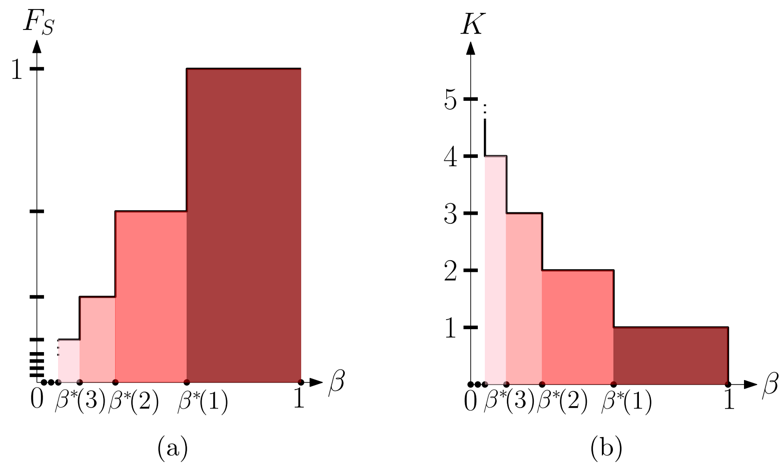


Figure 8.3: (a) F_S and (b) K values at different β.

value of K from (8.15) is

$$x_S + (K-1)\frac{p_e}{p} < Lx_S \leq x_S + K\frac{p_e}{p}, \tag{8.16}$$

which leads to,

$$K - 1 < p(L-1)\left(\frac{1}{\beta} + 1\right) \leq K. \tag{8.17}$$

Then, we solve for the critical value of K as

$$K = \left\lceil p(L-1)\left(\frac{1}{\beta} + 1\right)\right\rceil, \tag{8.18}$$

where $\lceil . \rceil$ denotes the ceil function; see Fig. 8.3. Further, the non-subscriber nodes would also be AC-stable as long as they continue to satisfy the AC constraint, which corresponds to the left-side inequality in (8.15). The fraction of the population that subscribes is thus given by

$$F_{S,line}(\beta) = \frac{1}{K} = \left(\left\lceil p(L-1)\left(\frac{1}{\beta}+1\right) \right\rceil \right)^{-1}, \tag{8.19}$$

which, the server would desire to maximize while maintaining a low β, where we show the dependency of F_S on β by writing $F_S(\beta)$. The minimum $\beta = \beta^*(K)$ required to maintain the subscription frequency K can be given, using (8.4), (8.14), and (8.15) by the following relation

$$Lx_S = \tilde{x}_K = x_S + K\frac{p_e}{p}, \tag{8.20}$$

which yields

$$\beta^*(K) = \left(\frac{K}{p(L-1)} - 1 \right)^{-1}. \tag{8.21}$$

As the server effort β^* increases, subscribers appear on the network more frequently, since K decreases, as shown in Fig. 8.3. Next, we use the server's minimum $\beta^*(K)$ to find β^*_{line} that maximizes the server's utility $L_R(\beta, \boldsymbol{a})$ given in (8.6). Thus, the server's optimization problem becomes,

$$\beta^*_{line} = \arg\max_{\beta^*(K), K=1,2,\cdots} F_{S,line}(\beta^*(K)) - c(\beta^*(K)). \tag{8.22}$$

In other words, the server's optimum β^*_{line} selection can be found easily by evaluating the server's cost at the critical $\beta^*(K)$ values found in (8.21). After the selection of

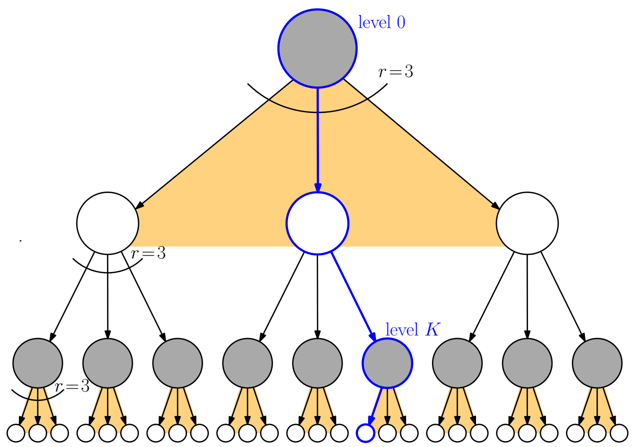


Figure 8.4: Directed tree network with $K = 2$ levels.

β^*_{line} by the server, the users will be in Nash equilibrium with a subscriber followed by $K-1$ non-subscribers, such that $\beta^*(K) = \beta^*_{line}$, as shown in Fig. 8.2.

8.3.2 Directed Tree Networks

Next, we focus on the network topology of Fig. 8.4, where users are connected according to an r-regular tree, such that each user can transmit updates to r new users, $r > 1$. In the tree network, therefore, there are r^k nodes at level k; see Fig. 8.4. The root node here must be a subscriber because not subscribing would result in an infinite expected age. A tree network provides a more realistic representation of how information spreads in a society. Here, each directed path in the tree is simply a directed line network of Subsection 8.3.1; therefore, the nodes in levels $\{1, \dots, K-1\}$ will not subscribe, and all nodes at level K will subscribe, where K is given by (8.18), i.e., the whole level subscribes at intervals of K; see subscription cells of Fig. 8.4. The number of users in each subscription cell of Fig. 8.4 is $r^0+r^1+\dots+r^{K-1} = \frac{r^K-1}{r-1}$, and therefore, the fraction of the subscribers is given by

$$F_{S,tree}(\beta) = \frac{r-1}{r^K - 1} = \frac{r-1}{r^{\lceil p(L-1)(\beta^{-1}+1) \rceil} - 1}. \tag{8.23}$$

From (8.18) and (8.23), we note the inverse relationship between $F_{S,tree}$ and β. Comparing (8.19) and (8.23), we see that with the same server effort β, we get fewer subscribers in a tree network due to the exponential decay of (8.23), compared to when users are connected in a line network. The tree network has one user directly updating r other users, and the server needs to increase β to stay relevant for such well-connected communities. After characterizing the AC-stable user's subscription strategies for a given β, next we find the optimal β^*_{tree} that will maximize the server's utility. Thus, the server's optimization problem in tree networks is given by

$$\beta^*_{tree} = \arg\max_{\beta^*(K), K=1,2,\cdots} F_{S,tree}(\beta^*(K)) - c(\beta^*(K)). \tag{8.24}$$

The server's optimal β^*_{tree} can be found by evaluating the server's cost function at $\beta^*(K)$ values for $K \geq 1$. After determining β^*_{tree} , the users in the tree network will be in Nash equilibrium with a subscriber node at level 0 followed by non-subscribers that are in levels $1, \cdots, K-1$, such that $\beta^*(K) = \beta^*_{tree}$, as shown in Fig. 8.4.

8.3.3 Star Networks

We next evaluate the star network topology of Fig. 8.5, where a central user (node 0) is connected to r peripheral users. This is an undirected network, that is, in-

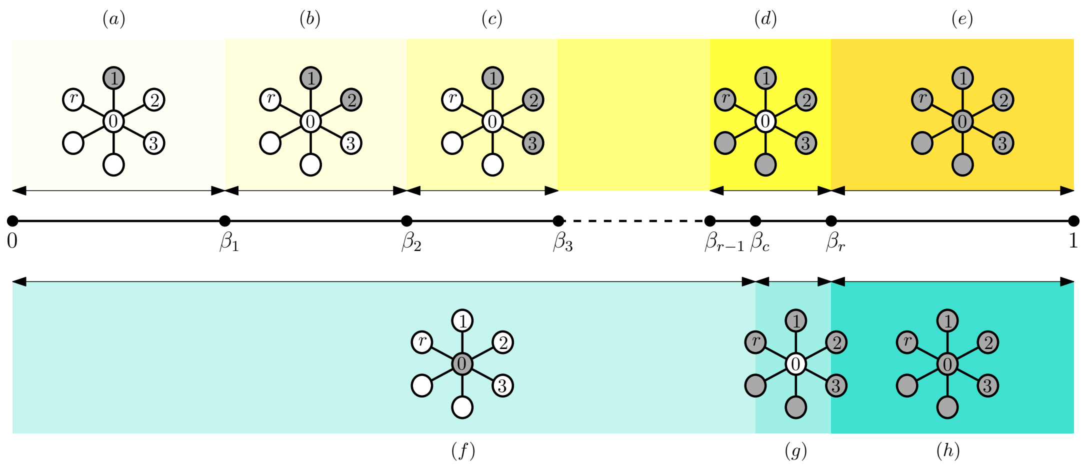


Figure 8.5: Equilibrium subscription strategies on star network as functions of β.

formation flow is bi-directional on all edges of $\overrightarrow{G}$, however, for any given strategies $\gamma_1, \gamma_2, \cdots, \gamma_{|\mathcal{N}|}$, this becomes a directed network. This is because every non-subscribing peripheral user must obtain updates from the central user, and if the central user does not subscribe then it must obtain updates from the peripheral subscribers, resulting in every link becoming a directed link. Let us denote the age of a peripheral non-subscribing user by x_{NP}. Since a peripheral non-subscriber gets updates from the central user, using (8.12) we get

$$x_{NP} = x_0 + \frac{p_e}{p}. \tag{8.25}$$

We first consider strategies in which the central user is a subscriber. When the central user subscribes, $x_0 = x_S$. Suppose that there are k peripheral subscribers along with the central user in the network. The remaining $r - k$ peripheral non-subscribers will be AC-stable, if each of them satisfies the AC constraint. Using

(8.4), (8.5), and (8.25), this gives

$$x_S + \frac{p_e}{p} < Lx_S, \tag{8.26}$$

which in turn yields

$$\beta < \left(\frac{1}{p(L-1)} - 1 \right)^{-1} = \beta_c. \tag{8.27}$$

However, if $\beta < \beta_c$, any peripheral subscriber will not be AC-stable, since a peripheral subscriber can change its decision and unsubscribe, which will still satisfy the AC constraint from (8.26). This results in the user strategies of Fig. 8.5(f), where given $\beta < \beta_c$, only the central user subscribes and all peripheral users do not subscribe. In this case, the central user is AC-stable since not subscribing would result in an infinite age.

If $\beta \geq \beta_c$, and we still consider that the central node is a subscriber, then no peripheral non-subscribers can be AC-stable as the AC constraint of (8.26) will not be satisfied. Hence, all peripheral nodes must subscribe. However, to make sure that the central user is also AC-stable, we need to verify that the alternate age $\tilde{x}_0$, when the central user chooses the alternate decision to not subscribe, violates the AC constraint. Under this alternate decision, the probability that the central node does not receive a packet from any of the r peripheral subscribers in a timeslot is $(1-p)^r$. However, if a packet is received from at least one peripheral subscriber, then $X_0(t+1) = X_S(t)$ in the absence of an event update (which happens with probability $1-p_e$) and $X_0(t+1) = X_S(t)+1$ in the case of an event update (with probability p_e). Approaching similar to (8.1) and (8.3), we get

$$\tilde{x}_0 = \frac{p_e}{1-(1-p)^r} + x_S. \tag{8.28}$$

For $\tilde{x}_0$ to violate the AC constraint ($\tilde{x}_0 \geq Lx_S$), we require

$$\beta \geq \left(\frac{1}{(1-(1-p)^r)(L-1)} - 1\right)^{-1} = \beta_r. \tag{8.29}$$

Since $p = 1-(1-p) \leq 1-(1-p)^r$, we have from (8.27) and (8.29) that $\beta_c \leq \beta_r$. Thus, for $\beta \geq \beta_r$, we get the user subscription strategies of Fig. 8.5(h), where all users subscribe. When $\beta \in [\beta_c, \beta_r)$, we will see in the following discussion that the subscription strategies of Fig. 8.5(g) follow, where all the r peripheral users subscribe, but the central user does not.

We now look at user strategies where the central user is not a subscriber. Suppose that there are k peripheral subscribers with $k \geq 1$ since $k=0$ will lead to infinite age for all users. We find the values of β which guarantee that the non-subscribing and subscribing users will be AC-stable, which yield, respectively, an upper bound and a lower bound on β.

If there is at least one peripheral non-subscriber ($k \leq r-1$), the central and peripheral non-subscribers are AC-stable if

$$x_0 < x_{NP} = x_0 + \frac{p_e}{p} < Lx_S. \tag{8.30}$$

To determine x_0 in (8.30), note that the central node will receive an update from at least one of the peripheral subscribers (with probability $1-(1-p)^k$) in a timeslot t, leading to $X_0(t+1) = X_S(t)$ in the absence of an event update ($U_E(t) = 0$). Receiving packets from peripheral non-subscribers will have no effect on $X_0(t+1)$ as the central user will always be fresher than peripheral non-subscribers in every timeslot. Thus, proceeding similar to (8.1) and (8.3), we get

$$x_0 = \frac{p_e}{1-(1-p)^k} + x_S. \tag{8.31}$$

Therefore, the AC-stable condition of (8.30) yields

$$\frac{p_e}{1-(1-p)^k} + \frac{p_e}{p} < (L-1)x_S, \tag{8.32}$$

which, using (8.4), yields an upper bound on β given by

$$\beta < \left(\left(\frac{1}{1-(1-p)^k} + \frac{1}{p}\right)\frac{1}{L-1} - 1\right)^{-1} = \beta_k, \tag{8.33}$$

such that β_k is an increasing function of k.

Likewise, we also verify whether all the k peripheral subscribers are AC-stable. For a particular peripheral subscriber, the alternate decision to unsubscribe, would result in a network with $k-1$ peripheral subscribers, in which case, the alternate age after unsubscribing can be obtained from (8.25) and by substituting k by $k-1$

in (8.31). Since this alternate age should violate the AC constraint, we have

$$\frac{p_e}{1-(1-p)^{k-1}} + \frac{p_e}{p} \geq (L-1)x_S, \tag{8.34}$$

which gives the lower bound as

$$\beta \geq \left(\left(\frac{1}{1-(1-p)^{k-1}} + \frac{1}{p} \right) \frac{1}{L-1} - 1 \right)^{-1} = \beta_{k-1}, \tag{8.35}$$

with $\beta_0 = 0$ since $k = 1$ peripheral subscriber could get infinite age upon unsubscribing. Therefore, when $\beta \in [\beta_{k-1}, \beta_k)$ for $k \in \{1, \ldots, r-1\}$, k peripheral users will subscribe and all other nodes will not subscribe, as shown in Figs. 8.5(a)-(c).

Lastly, we consider the case of $k = r$ peripheral subscribers. Here the central user is the only non-subscriber, which will be AC-stable for $\beta_{r-1} \leq \beta < \beta_r$ with the upper bound given by β_r in (8.29) (which is derived by using (8.5) and (8.31) with $k = r$). For the r peripheral subscribers to be AC-stable, we need to show that for any peripheral subscriber, the alternate decision to unsubscribe (resulting in a network of $r-1$ peripheral subscribers) would cause it to violate the AC constraint. This yields the lower bound $\beta_{r-1} \leq \beta$ where β_{r-1} is given by substituting $k = r$ in (8.35).

As a result, when $\beta \in [\beta_{k-1}, \beta_k)$ for $k \in \{1, \ldots, r\}$, k peripheral users will subscribe and all other nodes will not subscribe, as shown in Figs. 8.5(a)-(d), with

$$\beta_k = \begin{cases} \left(\left(\frac{1}{1-(1-p)^k} + \frac{1}{p} \right) \frac{1}{L-1} - 1 \right)^{-1}, & 1 \leq k \leq r-1, \\ \left(\frac{1}{(1-(1-p)^r)(L-1)} - 1 \right)^{-1}, & k = r, \end{cases} \tag{8.36}$$

such that β_k is an increasing function of k. Since $\frac{1}{p} < \frac{1}{1-(1-p)^{k-1}} + \frac{1}{p}$, comparing (8.27) and (8.35), we get $\beta_{r-1} < \beta_c$. Therefore, the critical values of β can be ordered as[3]

$$0 < \beta_1 < \ldots < \beta_{r-1} < \beta_c < \beta_r < 1. \tag{8.37}$$

We see that there are multiple AC-stable networks, since the Nash equilibrium is not unique. However, since we assume that the users take the server-preferred subscription strategy, the subscription ratio is equal to

$$F_{S,star}(\beta) = \begin{cases} \frac{k}{r+1}, & \beta \in [\beta_{k-1}, \beta_k), \; k = 1, \ldots, r, \\ 1, & \beta \geq \beta_r. \end{cases} \tag{8.38}$$

The minimum $\beta = \beta^*(k)$ required to maintain the k subscribers can be given by $\beta^*(k) = \beta_{k-1}$. After characterizing the AC-stable user's subscription strategies for a given β, next, we find the optimal β^*_{star} maximizing the server's utility. Thus, the

[3]Given the overlap of Fig. 8.5(f) with strategies of Fig. 8.5(a)-(d) for $\beta < \beta_c$, if a news agency plans to advertise promotional offers to increase the number of its subscribers despite a low β, it should target peripheral users with low connectivity instead of the central user, since subscription by the central user disincentivizes a large number of users from subscribing.

server's optimization problem in star networks is given by

$$\beta_{star}^* = \arg\max_{\beta_{k-1},k=1,2,\cdots,r+1} F_{S,star}(\beta_{k-1}) - c(\beta_{k-1}). \tag{8.39}$$

The optimal β_{star}^* can be found by evaluating the server's cost at β_{k-1} values in (8.36) for different numbers of subscribers k. After the selection of β_{star}^* by the server, the users will be in Nash equilibrium with k subscribers and $r+1-k$ non-subscribers, such that $\beta_{k-1} = \beta_{star}^*$, as shown in Fig. 8.5.

8.4 Undirected Networks

In this section, we delve into subscription policies over a broad range of network connectivity spectrum in undirected networks. We first focus on two extremes: the bidirectional line networks, representing the most constrained extreme of connectivity spectrum, and the fully-connected networks, representing the most deeply connected extreme of connectivity spectrum. Later, we delve into subscription strategies in higher dimensional spaces.

8.4.1 Bidirectional Line Network

We study symmetric AC-stable two-way networks shown in Fig. 8.6, where the number of non-subscribers between any two nearest subscribers is the same. Consider the subscription set $\mathcal{S} = \{mk | k \in \mathbb{Z}\}$, i.e., subscribers appear with frequency of m. Here, $X_i(t+1)$, for any non-subscribing user i, depends on the prior age $X_i(t)$ and transitions $U_E(t)$, $U_{i-1,i}(t)$, and $U_{i+1,i}(t)$. If both $U_{i-1,i}(t) = 1$ and $U_{i+1,i}(t) = 1$, that is,

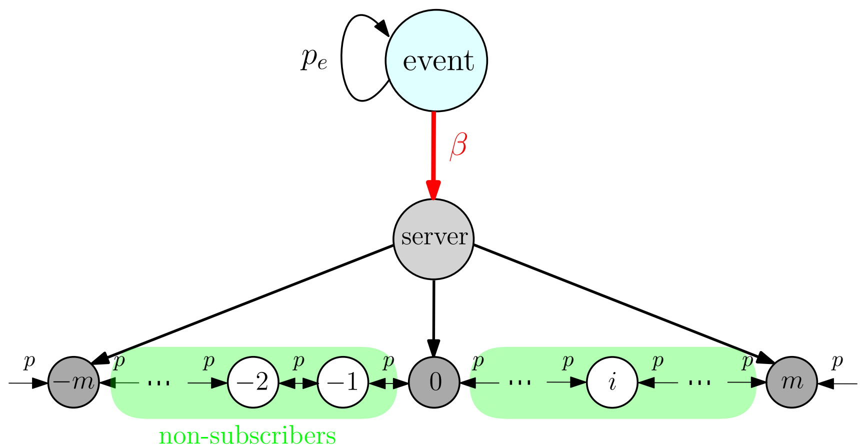


Figure 8.6: An illustration of a typical two-way line network.

user i receives update packets from both users $i-1$ and $i+1$, it keeps the packet with the lowest instantaneous version age, i.e., $X_i(t+1) = \min\{X_{i-1}(t), X_i(t), X_{i+1}(t)\}$, given $U_E(t) = 0$. Likewise, we can evaluate all combinations of the three transitions.

Define $X_{[j,h]}(t) = \min\{X_\ell(t) : \ell \in S_{[j,h]}\}$ to be the instantaneous version age of the set of users $S_{[j,h]} = \{j, \dots, h\}$, $\forall\, 0 \leq j \leq h \leq m$, with $x_{[j,h]} = \lim_{t\to\infty} \mathbb{E}[X_{[j,h]}(t)]$. Note that for $j = 0$ or $h = m$, $x_{[j,h]} = x_S$ since subscribers have strictly lower instantaneous version age than any non-subscribing user. For other cases, $x_{[j,h]} > x_S$ and $X_{[j,h]}(t+1)$ depends on the prior age $X_{[j,h]}(t)$ and transitions $U_E(t)$, $U_{j-1,j}(t)$, $U_{h+1,h}(t)$ as

$$\begin{aligned} x_{[j,h]} =& x_{[j-1,h]}(1-p_e)(1-p)p + x_{[j,h+1]}(1-p_e)p(1-p) \\ &+ x_{[j-1,h+1]}(1-p_e)p^2 + x_{[j,h]}(1-p_e)(1-p)^2 \\ &+ (x_{[j-1,h]}+1)p_e(1-p)p + (x_{[j,h+1]}+1)p_e p(1-p) \\ &+ (x_{[j-1,h+1]}+1)p_e p^2 + (x_{[j,h]}+1)p_e(1-p)^2, \end{aligned} \tag{8.40}$$

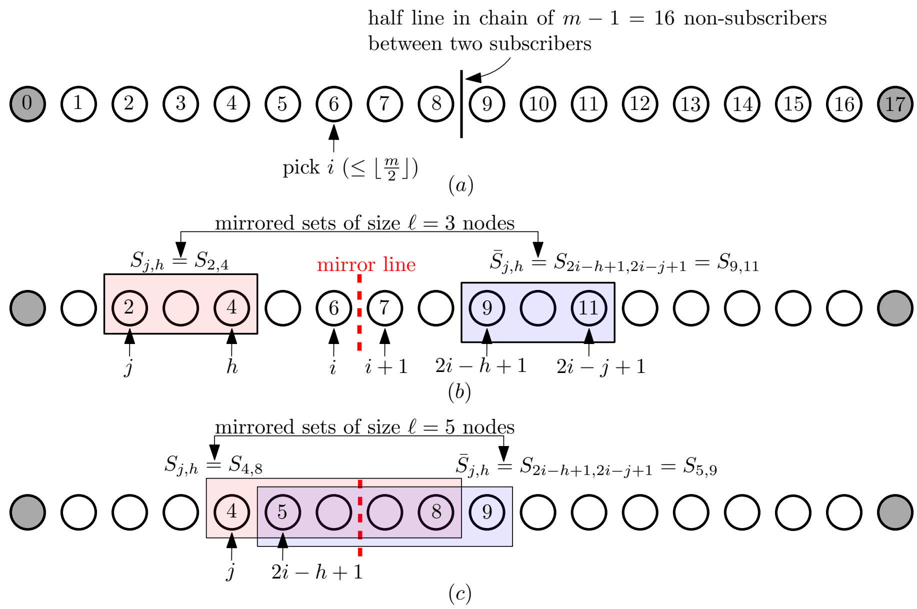


Figure 8.7: $S_{j,h}$ (pink blocks) and $\bar{S}_{j,h} = S_{2i-h-1,2i-j-1}$ (blue blocks) positioned symmetrically about the dotted line between nodes i and $i+1$.

which transforms into (with $x_{[i,i]} = x_i$),

$$x_{[j,h]} = \left(p_e + x_{[j-1,h+1]}p^2 + x_{[j-1,h]}(1-p)p + x_{[j,h+1]}p(1-p)\right) \times \left(1-(1-p)^2\right)^{-1}. \quad (8.41)$$

Lemma 8.1 *The highest age in the non-subscribers $S_{[1,m-1]}$ is at nodes mid-point between subscribers, i.e., $\lfloor \frac{m}{2} \rfloor$ and $\lceil \frac{m}{2} \rceil$.*

Proof: We show that $x_i \leq x_{i+1}$, $\forall i \leq \lfloor \frac{m}{2} \rfloor$ by induction. Fix an $i \leq \lfloor \frac{m}{2} \rfloor$ and draw a dotted line between user i and user $i+1$, as shown in Fig. 8.7. We consider sets $S_{[j,h]}$ with size $\ell = h - j + 1$, where at least half of the users fall on the left side of the dotted line (in $S_{[1,i]}$). Define $\bar{S}_{[j,h]}$ as the mirror image of the set $S_{[j,h]}$ about the dotted line, with $\bar{S}_{[j,h]} = S_{[2i-h+1,2i-j+1]}$, and the expected age of this set as $\bar{x}_{[j,h]} = x_{[2i-h+1,2i-j+1]}$. We claim that $x_{[j,h]} \leq \bar{x}_{[j,h]}$ for all sets of sizes

$\ell \in \{2i+1, \ldots, 1\}$ with $0 \leq j \leq i+1-\frac{\ell}{2}$.

We begin with the biggest sets of $\ell = 2i+1$ where the only admissible set is $S_{0,2i}$ with $x_{[0,2i]} = x_S \leq \bar{x}_{[0,2i]}$, satisfying the claim. Further, for set size $\ell = 2i$, the admissible sets are: (a) $S_{1,2i} = \bar{S}_{1,2i}$, giving $x_{[1,2i]} = \bar{x}_{[1,2i]}$ which satisfies the claim, and (b) $S_{0,2i-1}$ with $x_{[0,2i-1]} = x_S \leq \bar{x}_{[0,2i-1]}$ which satisfies the claim. Next, we show the general inductive step.

We assume that the claim holds for all admissible sets of sizes ℓ and $\ell+1$, and prove the claim for all admissible sets of size $\ell-1$. The version age of size $\ell-1$ set depends on version ages of size ℓ and $\ell+1$ sets by (8.41) as follows

$$x_{[j,j+\ell-2]} = \big(p_e + x_{[j-1,j+\ell-1]}p^2 + x_{[j-1,j+\ell-2]}(1-p)p + x_{[j,j+\ell-1]}p(1-p)\big) \times \big(1-(1-p)^2\big)^{-1} \tag{8.42}$$

$$\bar{x}_{[j,j+\ell-2]} = x_{[2i-j-\ell+3,2i-j+1]} = \big(p_e + \bar{x}_{[j-1,j+\ell-1]}p^2 + \bar{x}_{[j,j+\ell-1]}p(1-p) + \bar{x}_{[j-1,j+\ell-2]}p(1-p)\big) \times \big(1-(1-p)^2\big)^{-1}. \tag{8.43}$$

The right sides of (8.42) and (8.43) deal with size ℓ and $\ell+1$ sets. Hence, by assumptions of the inductive step, we have the terms of (8.42) smaller than terms of (8.43), resulting in $x_{[j,j+\ell-2]} \leq \bar{x}_{[j,j+\ell-2]}$. Setting $\ell = 1$ and $j = i$ gives $x_i \leq x_{i+1}$. ■

Let $x_i^{(m)}$ and $x_{[j,h]}^{(m)}$ denote the expected age of user i and set $S_{[j,h]}$, respectively, given subscription set $\mathcal{S} = \{mk | k \in \mathbb{Z}\}$.

Lemma 8.2 *$x_{[j_1,m-h_1]}^{(m)}$ is independent of m for fixed j_1, h_1.*

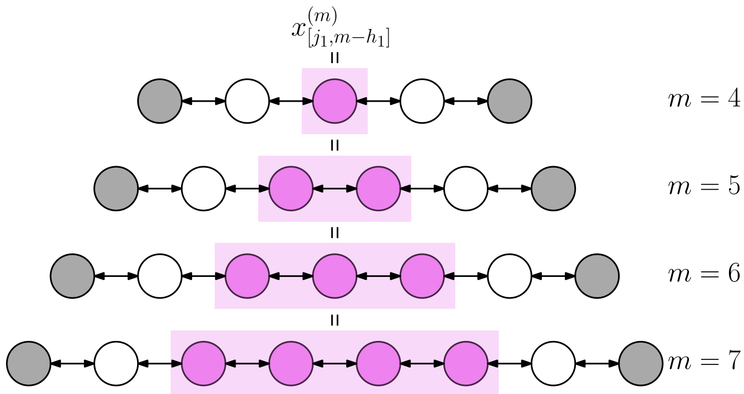


Figure 8.8: Sets $\{j_1, \ldots, m-h_1\}$ with $j_1 = 2$ and $h_1 = 2$, have the same expected age irrespective of the value of m.

Proof: The set $S_{[j_1,m-h_1]}$ has the same distance from the nearest left and right subscriber for every m; see Fig. 8.8. Hence, the age processes of these sets are statistically similar, resulting in identical iterative equations from (8.41). ■

Lemma 8.3 *We have: (a) $x^{(m)}_{\lfloor \frac{m}{2} \rfloor}$ is an increasing function of m; (b) $x^{(m)}_{\lfloor \frac{m}{2} \rfloor}\Big|_{m=1} = x_S$; and (c) $\lim_{m\to\infty} x^{(m)}_{\lfloor \frac{m}{2} \rfloor} = \infty$.*

Proof: The proof involves showing that $x^{(m)}_{\lfloor \frac{m}{2} \rfloor} \leq x^{(m+1)}_{\lfloor \frac{m+1}{2} \rfloor}$. This follows from the fact that the set $S_{[\lfloor \frac{m}{2} \rfloor, \lceil \frac{m}{2} \rceil]}$ with subscription set $\mathcal{S} = \{mk | k \in \mathbb{Z}\}$ and the set $S_{[\lfloor \frac{m}{2} \rfloor, \lceil \frac{m}{2} \rceil + 1]}$ with subscription set $\mathcal{S}' = \{(m+1)k | k \in \mathbb{Z}\}$ have the same distance from the nearest subscribers on both sides, and therefore, by Lemma 8.2, these sets have the same expected age. Further $\lfloor \frac{m+1}{2} \rfloor$ belongs to set $S_{[\lfloor \frac{m}{2} \rfloor, \lceil \frac{m}{2} \rceil + 1]}$, and the instantaneous age of each node in this set cannot be smaller than the overall age of the set. Hence, $x^{(m+1)}_{\lfloor \frac{m+1}{2} \rfloor} \geq x^{(m+1)}_{[\lfloor \frac{m}{2} \rfloor, \lceil \frac{m}{2} \rceil + 1]} = x^{(m)}_{\lfloor \frac{m}{2} \rfloor}$.

Next, $m = 1$ implies that all users are subscribers, and hence expected age at all users is x_S. Further, even with the best case scenario of $p = 1$ when updates immediately get forwarded on all links, the expected age at the middle user from

[125, (13)] would be $x_0 + \lfloor \frac{m}{2} \rfloor p_e \to \infty$ as $m \to \infty$. ■

Lemma 8.4 *$x_i^{(m)}$ is of the form $x_S + p_e g_1(p, i, m)$, where $g_1(\cdot)$ is a function of gossip probability p, node index i, and frequency of subscribers m, and is independent of x_S and p_e.*

Proof: We claim that $x_{[j,h]}^{(m)} = x_S + p_e g_2(p, j, h, m)$ for all $0 \leq j, h \leq m$, where $g_2(\cdot)$ is a function of j, h, p, and m, and is independent of x_S and p_e. The claim can be shown for sets $S_{[j,h]}$ of size $m+1$ and m, in which case $x_{[j,h]}^{(m)} = x_S$ due to the presence of subscribers in the set, resulting in $g_2(p, j, h, m) = 0$. Next, similar to proof of Lemma 8.1, we can inductively prove the claim for admissible sets of smaller sizes using (8.41).

Finally, note that $g_1(p, i, m) = g_2(p, i, i, m)$ can be obtained by computing $g_2(p, j, h, m)$ from (8.41) as follows,

$$\begin{aligned} g_2(p, j, h, m) = \big(1-(1-p)^2\big)^{-1} \times \big(1 + g_2(p, j-1, h+1, m)p^2 + g_2(p, j-1, h, m)(1-p)p \\ + g_2(p, j, h+1, m)p(1-p)\big) \end{aligned} \tag{8.44}$$

starting with size $m+1$ and m sets, progressively evaluating for smaller sets upto the singleton set with $j = h = i$. ■

Next, for a given β, we determine the values of m for which the subscription set $\mathcal{S} = \{mk | k \in \mathbb{Z}\}$ guarantees all the users to be AC-stable in the network. We first verify that the subscribers are AC-stable. Subscribing user 0 is AC-stable, if the alternate decision to not subscribe results in an alternate expected age $\tilde{x}_0$ which violates the AC constraint. Without subscription, user 0 would depend on updates all the way from subscriber $-m$ and subscriber m, leading to

$$\tilde{x}_0 = x_m^{(2m)} \geq Lx_S. \tag{8.45}$$

From Lemma 8.3, $x_m^{(2m)}$ increases with m, $\lim_{m\to\infty} x_m^{(2m)} = \infty$. Hence, there exists a finite lower bound $m^* = \min\{m : x_m^{(2m)} \geq Lx_S, m \in \mathbb{N}\}$. From Lemma 8.4, $x_m^{(2m)} = x_S + p_e g_1(p, m, 2m)$, with $x_S = p_e(\beta^{-1}+1)$. Therefore, $g_1(p, m, 2m)$ increases with m and (8.45) becomes

$$p_e g_1(p, m, 2m) \geq (L-1)x_S = (L-1)p_e \left(\beta^{-1} + 1\right). \tag{8.46}$$

Consequently, the lower bound can be defined as

$$m^* = \min\left\{m : g_1(p, m, 2m) \geq (L-1)\left(\beta^{-1}+1\right), m \in \mathbb{N}\right\} \tag{8.47}$$

such that, for $m \geq m^*$, the subscribing users will be AC-stable. Next, we verify that the non-subscribing users are AC-stable, which would provide an upper-bound m^{**} on m. From Lemma 8.1, it suffices to satisfy AC constraint for user $\lfloor m/2 \rfloor$

$$x_{\lfloor \frac{m}{2} \rfloor}^{(m)} < Lx_S. \tag{8.48}$$

Define $m^{**} = \max\{m : x^{(m)}_{\lfloor \frac{m}{2} \rfloor} < Lx_S\}$. Then, Lemma 8.4 gives

$$m^{**}{=}\max\left\{m{:}\,g_1(p,\lfloor\frac{m}{2}\rfloor,m){<}(L{-}1)\big(\beta^{-1}{+}1\big),m{\in}\mathbb{N}\right\}. \tag{8.49}$$

Observe from (8.47) and (8.49) that both m^* and m^{**} decrease with increase in β, allowing for more frequent subscribers.

Lemma 8.5 *For a given β, the symmetric bidirectional network is AC-stable $\forall\ m \in \{m^*, \ldots, m^{**}\}$, with $m^* = \lfloor m^{**}/2 \rfloor + 1$.*

Proof: The last part follows from the fact that if we choose $m = 2m^*$, then the central non-subscriber $\lfloor \frac{m}{2} \rfloor$ will not be AC-stable due to expected age $x^{(m)}_{\lfloor \frac{m}{2} \rfloor} = x^{(2m^*)}_{m^*} \geq Lx_S$ from initial definition of m^*. Therefore, $m \leq m^{**} < 2m^*$.

Conversely, for $m = 2m^* - 2$, all non-subscribers are AC-stable, implying $m^{**} \geq 2m^* - 2$. This is because the central nonsubscriber $\lfloor \frac{m}{2} \rfloor$ is AC-stable and will not subscribe, as in the latter case, subscribers will appear at a frequency of $m^* - 1$ and will not be AC-stable by the definition of m^*. ■

In the case of multiple AC-stable networks, the Nash equilibrium is not unique and there are multiple Nash equilibria. In this work, we choose the one that maximizes the server's utility by choosing the symmetric AC-stable network with the maximum number of subscribers, i.e., by $S^* = \{m^*k | k \in \mathbb{Z}\}$.[4] Thus, we have $F_{S,\textit{2-line}}(\beta) = \frac{1}{m^*}$ where m^*, defined in (8.47), depends on β. The minimum $\beta = \beta^*(m)$ required to maintain a subscription frequency of m, using equality in

[4]As we mentioned earlier, another approach might be to consider the AC-stable network that would lead to the lowest utility. In that case, we would choose the subscription set as $S^{**} = \{m^{**}k | k \in \mathbb{Z}\}$ and follow the same steps.

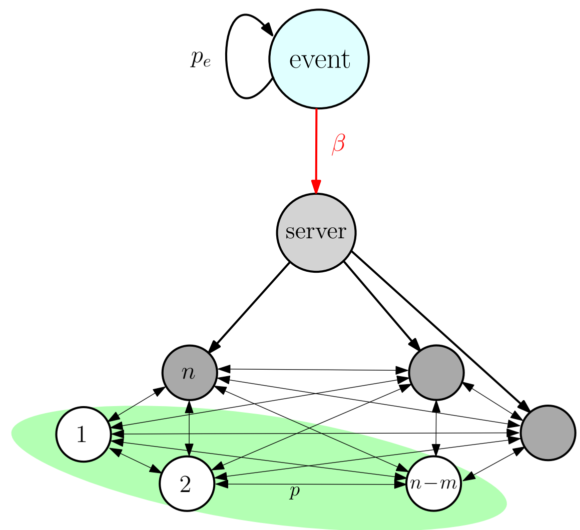


Figure 8.9: A typical fully-connected network with m subscribers.

(8.47), is $\beta^*(m) = \left(\frac{g_1(p,m,2m)}{(L-1)} - 1\right)^{-1}$, where $g_1(p,m,2m) = g_2(p,m,m,2m)$ is obtained by solving (8.44) recursively as explained in Lemma 8.4. Next, we use the server's minimum $\beta^*(m)$ to find $\beta^*_{\textit{2-line}}$ that maximizes the server's utility, that is, $\beta^*_{\textit{2-line}} = \arg\max_{\beta^*(m), m\in\mathbb{N}} F_{S,\textit{2-line}}(\beta^*(m)) - c(\beta^*(m))$. After the selection of $\beta^*_{\textit{2-line}}$ by the server, the users will be in Nash equilibrium with a subscriber followed by $m-1$ non-subscribers, such that $\beta^*(m) = \beta^*_{\textit{2-line}}$, as shown in Fig. 8.6.

8.4.2 Fully-Connected Network

In a fully-connected network, each of the total n users transmits an update to every other user in each timeslot with probability p; see Fig. 8.9. There are m subscribers and $n-m$ non-subscribers, with the subscription set as $\mathcal{S} = S_{[n-m+1,n]}$.

Let $x_{NS}^{(m)}$ denote expected age at a typical non-subscribing user in a fully-connected network with m subscribers. Due to network symmetry, every set of k non-subscribing users will have the same expected age, which we represent by

$x^{(m)}_{[1,k]}$. To analytically characterize $x^{(m)}_{[1,k]}$, we consider how combinations of various transitions affect $X_{[1,k]}(t)$ evolution. The set $S_{[1,k]}$ has k non-subscribers, and it can externally receive update packets from the m subscribers and the remaining $n-m-k$ non-subscribers. For the case of no event update in timeslot t, the probability of not receiving an update from any subscriber is $(1-p)^{km}$, thus with probability $1-(1-p)^{km}$, $X_{[1,k]}(t+1) = X_S(t)$. In the absence of any updates from subscribers, the probability of receiving updates from exactly i out of the remaining $n-m-k$ non-subscribers is $\binom{n-m-k}{i}(1-(1-p)^k)^i(1-p)^{k(n-m-k-i)}$. For the case of updates arriving from the i non-subscribers in $S_{[k+1,k+i]}$, we will have $X_{[1,k]}(t+1) = X_{[1,k+i]}(t)$. Accounting for all possible cases, we get

$$x^{(m)}_{[1,k]} = \Bigg[(1-p)^{km}\sum_{i=0}^{n-m-k}\binom{n-m-k}{i}x^{(m)}_{[1,k+i]}(1-(1-p)^k)^i(1-p)^{k(n-m-k-i)} + (1-(1-p)^{km})x_S + p_e\Bigg], \tag{8.50}$$

which can equivalently be written as

$$x^{(m)}_{[1,k]} = \Bigg[p_e + \sum_{i=1}^{n-m-k}\binom{n-m-k}{i}x^{(m)}_{[1,k+i]}(1-(1-p)^k)^i(1-p)^{k(n-k-i)} + (1-(1-p)^{km})x_S\Bigg] \times (1-(1-p)^{k(n-k)})^{-1}. \tag{8.51}$$

We can use (8.51) to iteratively compute $x^{(m)}_{[1,k]}$ in the order $k = n-m,\ldots,1$, to finally yield $x^{(m)}_{[1,1]} = x^{(m)}_1 = x^{(m)}_{NS}$.

Lemma 8.6 *(a)* $x^{(m)}_{[1,k]}$ *is in the form of* $x_S + p_e g(p,m,k)$.

(b) $x^{(m)}_{[1,k]}$ and $g(p,m,k)$ are decreasing functions of m.

Proof: Similar to Lemma 8.4, the first part can be proven by induction, starting from $k = n-m$ and iteratively proving for $k = n-m-1, \ldots, 1$ using (8.51). Similar to (8.44), $g(p,m,k)$ can be computed iteratively for $k = n-m, \ldots, 1$ using the following equation obtained from (8.51),

$$g(p,m,k) = \left[1 + \sum_{i=1}^{n-m-k} \binom{n-m-k}{i} g(p,m,k+i) \times (1-(1-p)^k)^i (1-p)^{k(n-k-i)}\right] \times \left(1-(1-p)^{k(n-k)}\right)^{-1}. \tag{8.52}$$

To prove the second part, we claim $x^{(m+1)}_{[1,k]} \leq x^{(m)}_{[1,k]}$, $\forall k \leq n-m$, which we prove inductively in the order $k = n-m, \ldots, 1$,

$$x^{(m+1)}_{[1,k]} = \left[\sum_{i=1}^{n-m-k-1} \binom{n-m-k-1}{i} x^{(m+1)}_{[1,k+i]} \times (1-(1-p)^k)^i (1-p)^{k(n-k-i)} + p_e + (1-(1-p)^{k(m+1)}) x_S\right] \times \left(1-(1-p)^{k(n-k)}\right)^{-1}. \tag{8.53}$$

Assuming the claim holds for non-subscribing sets of size greater than k, we prove the claim for set size k by comparing terms of (8.50) and (8.53), where $\binom{n-m-k}{i} \geq \binom{n-m-k-1}{i}$, $(1-(1-p)^{km}) \geq (1-(1-p)^{k(m+1)})$, and $x^{(m)}_{[1,k+i]} \geq x^{(m+1)}_{[1,k+i]}$, and (8.50) has an additional summation term when $i = n-m-k$. The proof for $g(p,m,k)$ then follows from the first part. ■

Lemma 8.7 *For a given β, there exists a unique $m = m^*_{FC}$, for which the fully-connected network is AC-stable with m^*_{FC} subscribers and $n - m^*_{FC}$ non-subscribers,* *such that,*

$$m^*_{FC} = \max\{m : x_S + p_e g(p, m-1, 1) \geq Lx_S\}. \tag{8.54}$$

Proof: For non-subscribers to be AC-stable, we need $x_1^{(m)} < Lx_S$. From Lemma 8.6 part (b), $x_1^{(m)}$ decreases with m, with $x_1^{(m)}|_{m=0} = \infty > Lx_S$ and $x_1^{(m)}|_{m=n} = x_S < Lx_S$. Hence, there exists a unique $m = m^*_{FC} \geq 1$, such that,

$$x_1^{(m^*_{FC})} < Lx_S \leq x_1^{(m^*_{FC}-1)}. \tag{8.55}$$

For $m < m^*_{FC}$, the non-subscribers will not be AC-stable from the right inequality of (8.55) and Lemma 8.6(b). For $m > m^*_{FC}$, any subscriber will not be AC-stable, and will unsubscribe and still satisfy AC constraint with the rest $m-1$ subscribers from the left inequality of (8.55) and Lemma 8.6(b). Thus, $m^*_{FC} = \max\{m : x_1^{(m-1)} = x_S + p_e g(p, m-1, 1) \geq Lx_S\}$, by combining the right inequality of (8.55) and Lemma 8.6(a). ■

Hence, $F_{S,FC}(\beta) = \frac{m^*_{FC}}{n}$, where m^*_{FC} depends on β from (8.54). The minimum $\beta = \beta^*(m)$ required to maintain m subscribers, from equality in (8.54), is $\beta^*(m) = \left(\frac{g(p,m-1,1)}{(L-1)} - 1\right)^{-1}$, which can be used to determine the server's optimum β^*_{FC} via $\beta^*_{FC} = \arg\max_{\beta^*(m), m=1,2,\cdots} F_{S,FC}(\beta^*(m)) - c(\beta^*(m))$. After the selection of β^*_{FC} by the server, the users will be in Nash equilibrium with m subscribers and $n-m$ non-subscribers, such that $\beta^*(m) = \beta^*_{FC}$, as shown in Fig. 8.9.

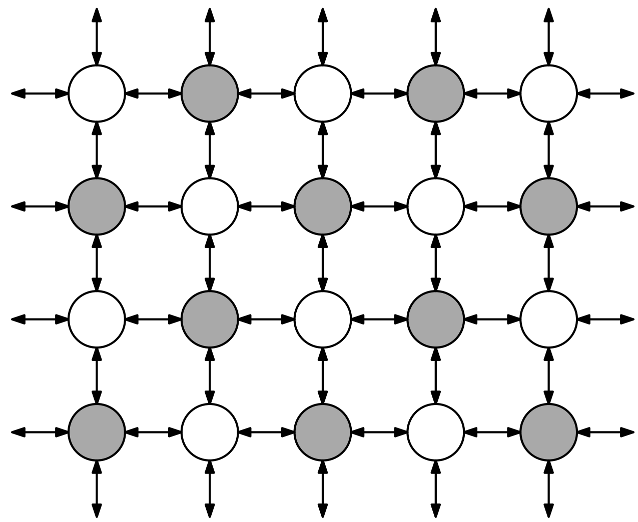


Figure 8.10: $\mathcal{S}_2$ with half of the population as subscriber.

8.4.3 Undirected Networks in Higher Dimensions

Consider the network $\mathbb{Z}^m$ and consider the set of subscribers

$$\mathcal{S}_m = \{(i_1, i_2, \ldots, i_m) : (i_1 + i_2 + \ldots + i_m) \texttt{ mod } 2 = 0, i_1, \ldots, i_m \in \mathbb{Z}\}, \tag{8.56}$$

such that half the population is a subscriber (see Fig. 8.10 for $m = 2$ case). We next analyze the range for β that can keep all nodes of such networks AC-stable.

For this purpose, we first have to check if the non-subscribers are AC-stable. As depicted in Fig. 8.10, a non-subscribing node can potentially get updates from $d = 2^m$ subscribers in timeslot t, all of whom have the same instantaneous version age at all times. We need to consider whether no update arrives at the non-subscribing node in timeslot t (which happens with probability $(1-p)^d$) and whether there is an event update (with probability p_e). Therefore, approaching similar to (8.40) and

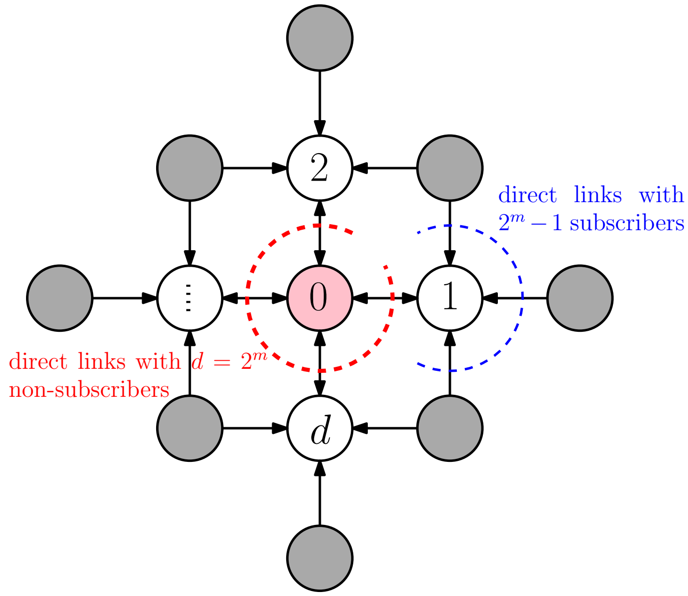


Figure 8.11: Each node in $\mathcal{S}_m$ is surrounded by 2^m nodes.

(8.41), the expected age at the non-subscribing node is

$$x_S + \frac{p_e}{1-(1-p)^d}, \tag{8.57}$$

which needs to satisfy the AC constraint of (8.5), given by

$$\frac{p_e}{1-(1-p)^d} < (L-1)x_S = (L-1)p_e\left(\frac{1}{\beta}+1\right), \tag{8.58}$$

resulting in

$$\beta < \frac{1}{\frac{1}{(1-(1-p)^{2^m})(L-1)} - 1} = \beta_U(m). \tag{8.59}$$

Next, we need to ensure that a typical subscriber (represented by node 0 in Fig. 8.11) will be AC-stable, by showing that its expected age $\tilde{x}_0$ from the alternate action of not to subscribe violates the AC constraint.

When subscribing node 0 chooses to unsubscribe in Fig. 8.11, it can still get updates from the surrounding non-subscribing nodes $\{1, \ldots, d = 2^m\}$, each of which is in contact with $d-1$ subscribers. Since the version age processes at all nodes in the set $\{1, \ldots, d\}$ are statistically similar, we will restrict our attention to the expected version age of arbitrary sets of $k+1$ nodes, which we will denote by $\{0, 1, \ldots, k\}$, $k \leq d$. This set contains the middle non-subscriber node 0 and k out of the d surrounding non-subscribers, and has $k(d-1)$ direct links from subscriber nodes, hence with probability $(1-(1-p)^{k(d-1)})$, $X_{[0,k]}(t+1) = X_S(t)$. In the event a packet does not come to this set from any of the subscribers, we need to consider packet arrivals from i out of $d-k$ remaining surrounding non-subscriber nodes, which happens with probability $p^i(1-p)^{d-k-i}$, and there are $\binom{d-k}{i}$ unique ways of choosing i out of $d-k$ nodes. Since the age evolution of all the resulting sets of size $k+i$ is statistically identical, we will denote the expected age of these resultant sets by $x_{[0,k+i]}$. Considering the event updates with probability p_e, and approaching similar to (8.40) and (8.41), we can get

$$x_{[0,k]} = (1-p)^{k(d-1)} \sum_{i=0}^{d-k} \binom{d-k}{i} p^i (1-p)^{d-k-i} x_{[0,k+i]} + (1-(1-p)^{k(d-1)}) x_S + p_e. \quad (8.60)$$

Starting from $k = d$, which gives $x_{[0,d]} = x_S + \frac{p_e}{1-(1-p)^{d(d-1)}}$, we can compute $x_{[0,k]}$ iteratively in the order of $k = d, d-1, \ldots, 0$, to finally obtain $\tilde{x}_0 = x_{[0,0]}$, which needs to fail the AC constraint, $\tilde{x}_0 \geq L x_S$.

Lemma 8.8 *$x_{[0,k]}$ is of the form $x_S + p_e g_3(p, m, k)$.*

Proof: The proof is similar to that of Lemma 8.4, wherein the lemma statement is verified inductively for all $k = d, d-1, \ldots, 0$. ■

Hence $\tilde{x}_0 = x_S + p_e g_3(p, m, 0) \geq L x_S$ or $g_3(p, m, 0) \geq (L-1)\left(\frac{1}{\beta} + 1\right)$, which will yield $\beta \geq \beta_L(m)$.

Discussion: As $d = 2^m$ increases, the factor $(1 - (1-p)^{k(d-1)})$ becomes the dominant factor in (8.60) for $k \leq 1$. Intuitively, this is due to each non-subscriber node being in direct contact with a higher number of subscribers in large dimensional space, improving timeliness in the network. Thus, $\lim_{m \to \infty} x_{[0,k]} = x_S + p_e$, $k \leq 1$, and consequently, $\tilde{x}_0 = \lim_{m \to \infty} x_{[0,0]} = x_S + 2p_e$. Note that $x_S + kp_e$ is the lowest possible attainable value of expected age at a non-subscriber node that is k steps away from its closest subscriber node (see (8.14) with gossip frequency $p = 1$). Consequently, both $\beta_U(m)$ from (8.59) and $\beta_L(m)$ also increase with m. This suggests that higher degree of connectivity between nodes decreases the incentive for nodes to subscribe, requiring increased server effort β to maintain relevance in highly connected networks.

Existence of equilibrium solutions under high gossiping: We now show how in high dimensional network with high gossiping rates ($p = 1$), for a given β, there exist equilibrium subscription solutions. We consider symmetric networks $\mathbb{Z}^m$ with the set of subscribers of the form

$$\mathcal{S}_m = \{(hi_1, hi_2, \ldots, hi_m) : (i_1 + i_2 + \ldots + i_m) \texttt{ mod } 2 = 0, i_1, \ldots, i_m \in \mathbb{Z}, \}, \quad (8.61)$$

where $h \in \mathbb{N}$ and fraction of subscribers $F_{S,m-\text{dim}} = \frac{1}{2h^m}$.

Under the special case of $p = 1$, each node shares its latest version of information with all its neighbors in every timeslot. In such a case, each node posseses packets arriving from its nearest subscriber node. From (8.14), if a node is k hops away from its nearest subscriber, its expected version age will be $x_S + kp_e$. To check if the network is AC-stable, we first check if all the non-subscribers are AC-stable. Clearly, the non-subscriber with the highest age will be h hops away from its nearest subscribers (the middle most node between two adjacent subscribers, see Fig. 8.11), and therefore the highest age of a non-subscriber, independent of the dimensionality of the space, is $x_S + hp_e$ which satisfies the AC-constraint, i.e., $x_S + hp_e < Lx_S$. Further, we check if the subscribers are AC-stable, i.e., if a subscriber chooses to unsubscribe, then its expected age $x_0 + 2hp_e$, determined by its distance from the closest subscriber which is $2h$ hops away, violates AC-constraint, i.e., $x_S + 2hp_e \geq Lx_S$. Substituting the expression for x_S, for a given value of β, the values of h for which the network is in equilibrium are

$$h < (L-1)\left(\frac{1}{\beta}+1\right) \leq 2h. \tag{8.62}$$

Note that $(L-1)\left(\frac{1}{\beta}+1\right)$ is a positive value, and as long as its value is greater than 1, there will exist an h that will make the network AC-stable. However, if $(L-1)\left(\frac{1}{\beta}+1\right) \leq 1$, then all nodes of the network will subscribe, which we will refer to as $h = 0$ case for convenience of notation. In such a case, the resulting network will be AC-stable, since any subscriber, upon unsubscribing, will become equivalent to a non-subscriber of $h = 1$ case, and therefore, will not satisfy the

AC-constraint, since $h = 1$ is not a feasible solution to (8.62).

Further note that, certain values of L can rule out feasibility of certain values of F for an AC-stable m-dimensional network irrespective of the server effort β employed. As an example, if $L > 1.5$, implying $1 = (1.5 - 1)\left(\frac{1}{1} + 1\right) < (L - 1)\left(\frac{1}{\beta} + 1\right)$ implies a network with all users as subscribers can never be AC-stable, since in the right inequality of (8.62)

$$2(L-1) \leq (L-1)\left(\frac{1}{\beta} + 1\right) \leq 2h, \tag{8.63}$$

which gives $1 \leq h$. Hence the maximum achievable F is $\frac{1}{2}$.

Since we consider AC-stable networks with maximum subscribers in this work, we consider the lowest possible h, which is $h^* = \left\lfloor \frac{(L-1)}{2}\left(\frac{1}{\beta} + 1\right) \right\rfloor$. Further, for given h frequency, the minimum $\beta = \beta^*(h) = \frac{1}{\frac{2h}{(L-1)} - 1}$. Next, we use the server's minimum $\beta^*(h)$ to find $\beta^*_{m\text{-}dim}$ that maximizes the server's utility, that is, $\beta^*_{m\text{-}dim} = \arg\max_{\beta^*(h), h \in \mathbb{N} \cup \{0\}} F_{S,m\text{-}dim}(\beta^*(h)) - c(\beta^*(h))$. After the selection of $\beta^*_{m\text{-}dim}$ by the server, the users will be in Nash equilibrium with the set of subscribers $\mathcal{S}_m$ defined by (8.61), such that h in (8.61) satisfies $\beta^*(h) = \beta^*_{m\text{-}dim}$.

8.5 Numerical and Simulation Results

8.5.1 Directed Networks

We first simulate the directed line network covered in Subsection 8.3.1 with parameters $L = 10$, $p = 0.2$, $\beta = 0.6$, and $p_e = 0.3$ for a total time of $t = 10^4$, which we

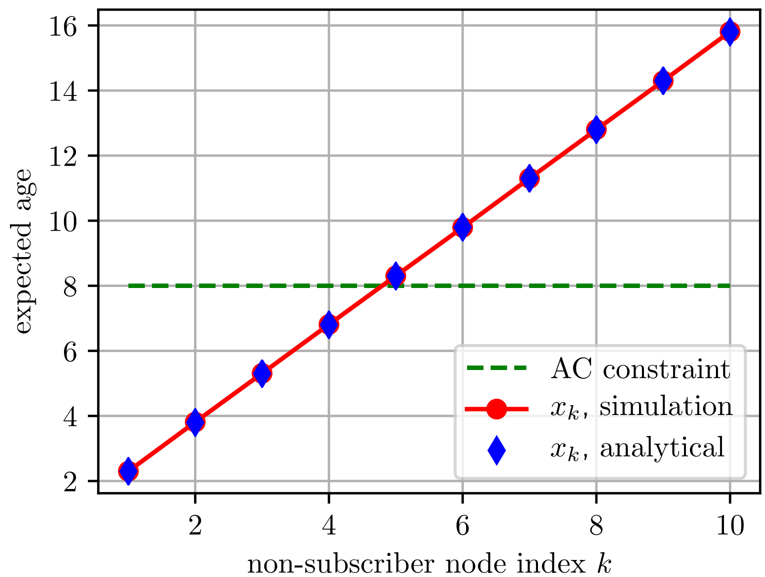


Figure 8.12: Age evolution in one-way line network when $L = 10$, $p = 0.2$, $\beta = 0.6$, and $p_e = 0.3$. For all users to be AC-stable, users in $5\mathbb{N}$ will have to subscribe.

use as a proxy for $t \to \infty$, and take the average over 2×10^5 iterations to approximate $\lim_{t\to\infty} \mathbb{E}[X_i(t)]$ by the law of large numbers. In Fig. 8.12, we observe that the real-time simulation points (red) coincide with the analytically derived values (blue) from (8.14), verifying the theoretical analysis. The graph shows how expected age increases in a directed path linearly with the number of hops from the nearest subscriber; thus, we have $K = 5$ for all users to be AC-stable, as in (8.18).

Next, we analyze the equilibrium strategies for the server and the users in a line network, 2-regular tree network (in Subsection 8.3.2), and a star network with 100 peripheral users (in Subsection 8.3.3), with parameters $L = 10$, and $p = 0.2$, for the server utility function $F_S(\beta) - 80 \times \beta^2$. This utility function penalizes high values for β, but is indifferent to smaller values of β due to the $80 \times \beta^2$ term. Fig. 8.13 shows β^* and the corresponding F_S and the server utility function $L_R(\beta, \boldsymbol{a})$ achieved in all three networks. We note that a higher β^* is required in the tree network to

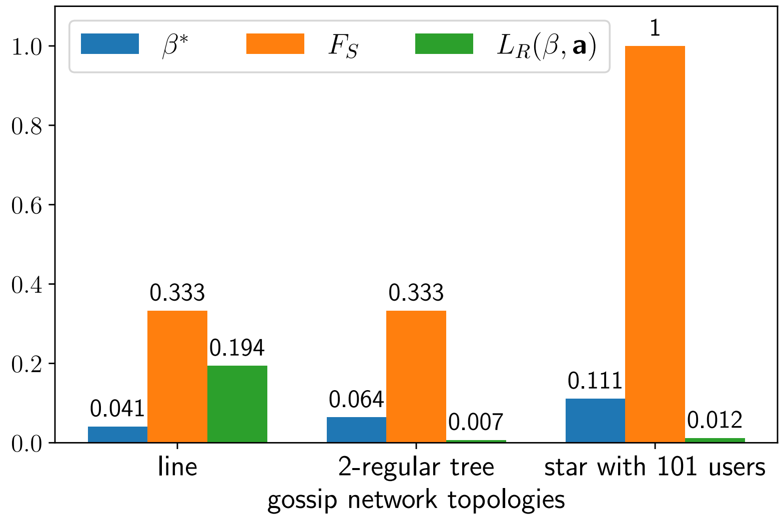


Figure 8.13: The comparison of the server's optimal β^*, subscription rate F_S, and the server's optimal cost function for the line, tree, and star networks when $L = 10$, $p = 0.2$, $\beta = 0.6$, and $p_e = 0.3$.

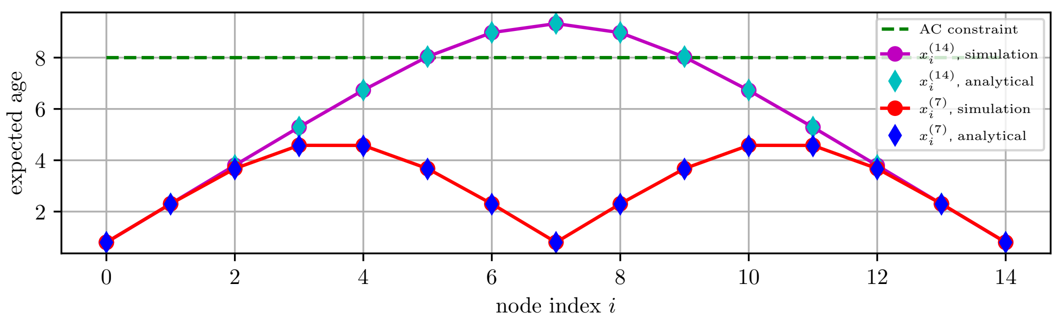


Figure 8.14: The expected age of nodes in a two-way (bidirectional) line network with $m = \{7, 14\}$.

achieve the same F_S, and the corresponding server utility is lower since the nodes are more well-connected in a tree. On the other hand, in the star network, we choose the highest $\beta^* = 0.111$ that causes all users to subscribe; however, the highest β^* negatively affects the server utility, which is the lowest for the star network.

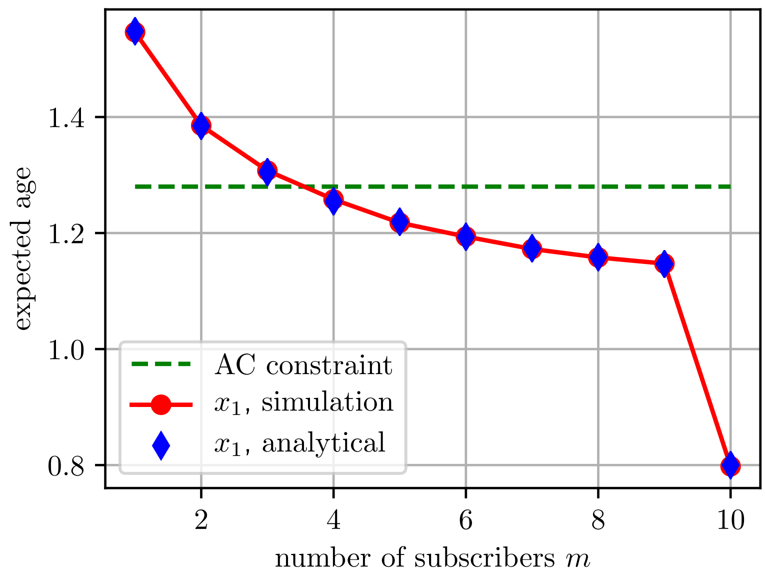


Figure 8.15: The expected age of a node in a fully-connected network of $n = 10$.

8.5.2 Undirected Networks

We next simulate the bidirectional line network provided in Subsection 8.4.1 with parameters $L = 10$, $p = 0.2$, $\beta = 0.6$, $p_e = 0.3$. In Fig. 8.14, the real-time simulation points (red) coincide with the analytically derived values (blue) from (8.40), verifying the theoretical analysis. Observe how all non-subscribers are AC-stable with $m^* = 7$, i.e., $x_i^{(7)}$ is below the AC constraint threshold (green) for all i, and the subscribers are AC-stable, since the alternate decision to unsubscribe increases their age to $x_7^{(14)}$, which violates the AC constraint.

Next, we focus on the expected age of user 1 in a fully-connected network provided in Subsection 8.4.2 with $n = 10$ users, $L = 1.6$, $p = 0.2$, $\beta = 0.6$, and $p_e = 0.3$. The decrease of $x_1^{(10)}$ with increase in m is in accordance with Lemma 8.6(b), with $m^*_{FC} = 4$ making all users AC-stable, by (8.54). The sudden jump in Fig. 8.15

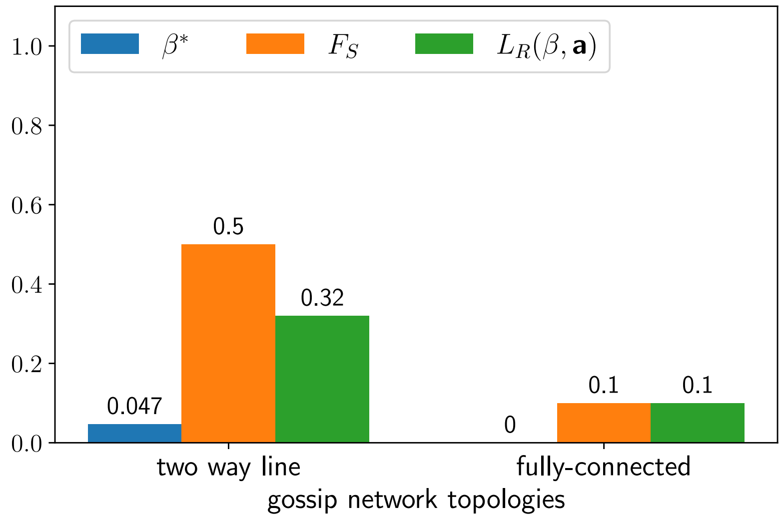


Figure 8.16: The comparison of the server's optimal β^*, subscription rate F_S, and the server's optimal cost function for two way-line network and fully connected network.

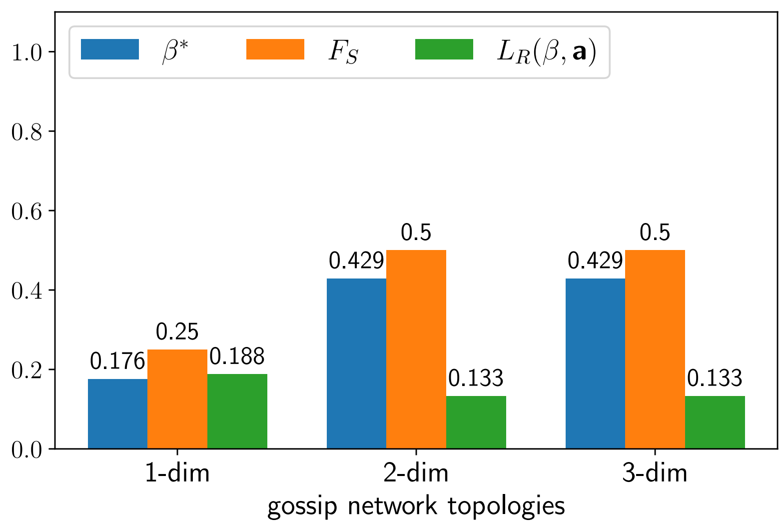


Figure 8.17: The comparison of the server's optimal β^*, subscription rate F_S, and the server's optimal cost function in m-dimensional network.

at $m = n = 10$ comes from user 1 itself turning to a subscriber.

Fig. 8.16 shows β^*, the corresponding F_S, and server utility function $L_R(\beta, \boldsymbol{a})$ in both networks with $L = 1.6$, $p = 0.2$, and server utility function $F_S(\beta) - 80 \times \beta^2$. The deep connectivity of the fully-connected network disincentivizes users from subscribing, resulting in just one subscriber and $\beta^*_{FC} \to 0 = \beta^*(1)$ that suffices to sustain one subscriber. Contrarily, the bidirectional line has every alternate user as a subscriber, with higher $\beta^*_{2\text{-}line}$, F_S and utility.

Fig. 8.17 shows β^*, the corresponding F_S, and server utility function $L_R(\beta, \boldsymbol{a})$ in m-dimensional networks given in Subsection 8.4.3 with $L\!=\!1.6$, $p\!=\!1$, and server utility function $F_S(\beta)\!-2\!\times\!\beta^2$. In higher dimensions, $F_{S,m-\text{dim}}$ reduces exponentially with m. Hence, we observe from Fig. 8.17 that networks employ a high β to maintain a high $F_{S,m-\text{dim}}$ to optimize the cost function. Since $L > 1.5$, as per discussion at the end of Subsection 8.4.3, $h = 0$ case is not possible, and servers simply attempt to achieve the lowest possible $h = 1$ frequency in higher dimensions.

8.6 Conclusion and Future Directions

In this chapter, we have investigated a Stackelberg-Nash game between a server and users, where the server wishes to maximize its profit by increasing the number of subscribers and reducing costs associated with the frequent sampling of the event, while the users make their subscription decisions based on their timeliness requirements. We first analyzed directed networks and showed how subscribers repeat periodically along directed paths in line and tree networks. The analyses in tree and star networks showed how higher connectivity and the presence of central agents can discourage other nodes from subscribing, and well-connected communities tend to have fewer subscribers.

Next, we extended our analysis to undirected networks with bidirectional (two-way) flow of information on inter-node links. In the bidirectional line network, multiple Stackelberg equilibrium solutions may exist, with higher plausible periodicity of subscribers under frequent server sampling. The fully-connected network gives a single equilibrium solution and fewer subscribers owing to the deep network connectivity disincentivizing users from subscribing. We also proved existence of equilibrium solutions in higher dimensions. Our theoretical results revealed an inverse relationship between the expected age and both gossiping and server sampling rates.

In networks with multiple equilibrium solutions, our focus was on server-preferred subscription strategies that maximized the fraction of users as subscribers, thereby maximizing the server's utility. It would be interesting to explore the opposite scenario, considering the worst-case subscription strategy for the server, which results in the fewest subscribers. The goal here would be to maximize the server's minimum possible utility. Additionally, while this chapter examined a wide range of gossip networks, practical networks often exhibit arbitrary inter-node connections, leading to diverse gossip network topologies. In such cases, bandit or machine learning-based approaches could be employed to adaptively determine subscription decisions for nodes. Finally, whereas this chapter has considered nodes aiming to bound their timeliness by a constant multiple of the timeliness of a direct subscriber, another approach could involve setting a hard constraint on a node's timeliness, independent of the timeliness of a subscriber.

CHAPTER 9

Timely Gossiping with File Slicing and Network Coding

9.1 Introduction

It was shown in [5] that in a complete symmetric network of n nodes with a single file, the average age at each node is $O(\log n)$. An interesting question to ask is whether we can do better than $O(\log n)$ version age in a fully connected network. Similarly, if we disseminate n files simultaneously to all n nodes in real time in a network by dividing all rate resources among n files in [5], we obtain a version age of order $O(n \log n)$. Another interesting question is whether we can do better than $O(n \log n)$ for n files.

In this chapter, we achieve $O(1)$ version age for single-file dissemination and $O(n)$ version age for n-file dissemination (improving both results by $\log n$), in a network of n nodes, that communicate using a random phone call model. We show that complete dissemination of a file to the entire network is not necessary to achieve these age bounds. Further, periodic pausing of dissemination of newer file versions from the source does not hinder the achievement of these bounds. We show that specific file splitting and network coding based gossip protocols achieve the mentioned bounds. Finally, we present simulation results to validate the aforementioned bounds.

9.2 Gossiping with a Single File

We consider slotted time, where one time slot is the duration of time needed to receive an entire file. The network consists of n nodes, and a node, called the *source*, receives updates for the file according to an arbitrary distribution. All source updates start at the beginning of a time slot. Hence, the source may get a new version of the file in every time slot. The n users wish to have the latest possible version of the file at any time. We follow the random phone call model (i.e., rumor mongering model), where the algorithm works in rounds, and in every round, each user, uniformly at random, selects another user, called the *target* node, to communicate with.

Theorem 9.1 *In a fully connected network of n users, which uses a gossip protocol, if there exists $p \in [0, 1)$ and $n_0 \in \mathbb{N}$, such that for all $n > n_0$, the probability of a random node receiving the file in $O(1)$ time is greater than $1 - p$, then the protocol achieves an $O(1)$ age at each node.*

Proof: Let $c = O(1)$ time slots, $c \in \mathbb{N}$, which is also taken to be the duration of a cycle. The timeline is thus divided into consecutive cycles, with each cycle consisting of c time slots; see Fig. 9.1. At the beginning of each cycle, the source begins to transmit the latest version it has received by the end of the previous time slot to the network. The network is allowed to gossip for c time slots according to the given

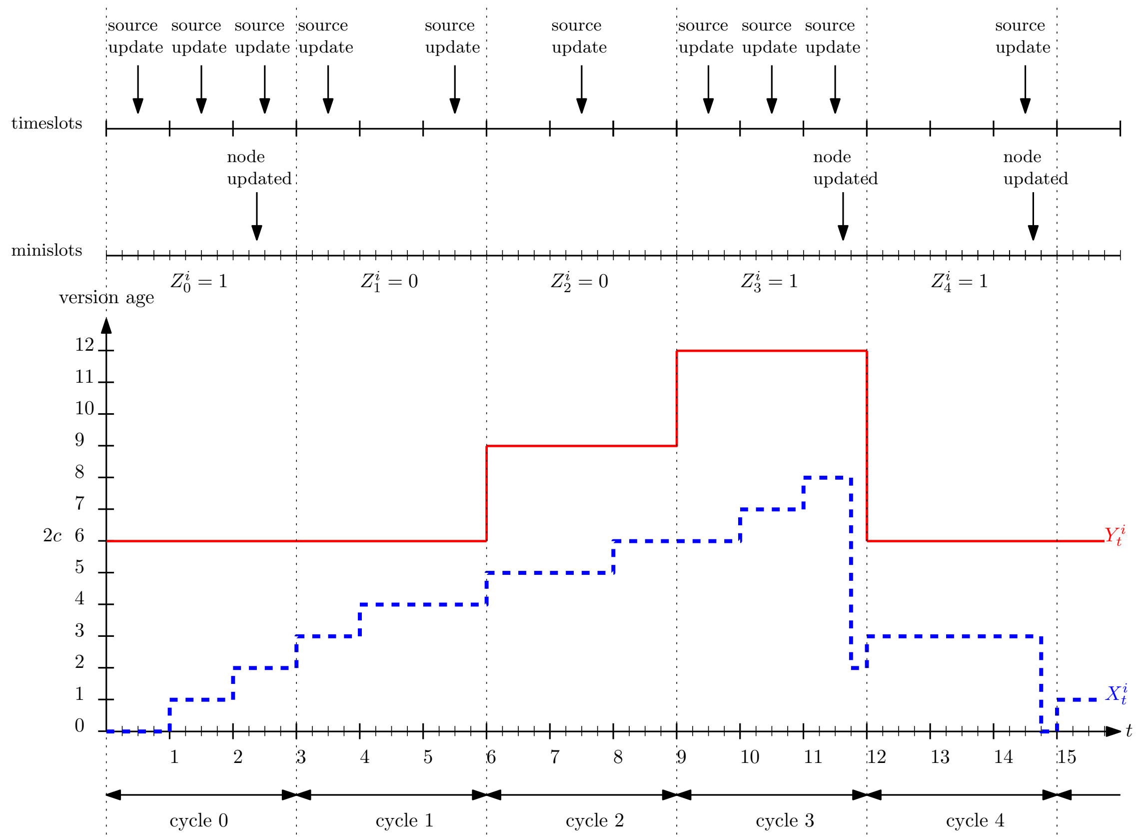


Figure 9.1: Sample version age evolution at typical node i, with parameters $c = 3$, $\ell = 4$. The red curve corresponds to Y_t^i which remains above $2c = 6$, and increases by $c = 3$ whenever the node is unsuccessful in receiving the update in the preceding cycle. The blue curve corresponds to X_t^i which drops to lower values in the minislot after it gathers all pieces of a particular file version.

protocol. To avoid mixing of different versions of a file, when the cycle ends, the nodes lay off transmission of the file versions already in their possession, and the circulation of a newer version is set in motion with the beginning of the next cycle.

In time slot t, let N_t^i be the latest version of the file available at node i, and let N_t^s be the latest version present at the source. Thus, the instantaneous version age of information at node i at time t is $X_t^i = N_t^s - N_t^i$. Therefore, every time the source receives an update, N_t^s, and hence, the instantaneous age at each node in the network, increment by one.

We assume that all nodes start with zero age at $t = 0$. At $t = kc$, $k \in \mathbb{N}$, the cycle k begins, and the source node starts the transmission of version N^s_{kc} to the network. During cycle k, i.e., between time slot kc and $(k+1)c$, the source may receive a maximum of c updates in the c time slots, but it continues to circulate the version N^s_{kc} and does not transmit the newer versions. If node i receives the version N^s_{kc} at any time t, $kc \leq t < (k+1)c$, then the instantaneous version age at node i drops to $X^i_t = N^s_t - N^s_{kc}$.

Let Z^i_k indicate if node i successfully receives the file in cycle k. Hence,

$$Z^i_k = \begin{cases} 1, & \text{if node } i \text{ receives version } N^s_{kc} \text{ in cycle } k \\ 0, & \text{otherwise} \end{cases} \tag{9.1}$$

Note that Z^i_k subscript k denotes the cycle index whereas N^s_t, N^i_t and X^i_t subscripts t denote the time slot index (see Fig. 9.1).

At the end of cycle k,

$$X^i_{(k+1)c} = \begin{cases} X^i_{kc} + N^s_{(k+1)c} - N^s_{kc}, & \text{if } Z^i_k = 0 \\ N^s_{(k+1)c} - N^s_{kc}, & \text{if } Z^i_k = 1 \end{cases} \tag{9.2}$$

where we note that the source would have moved from version N^s_{kc} to version $N^s_{(k+1)c}$ through cycle k, and $X^i_{kc} = N^s_{kc} - N^i_{kc}$. Here, observe that if $Z^i_k = 0$, then $N^i_{(k+1)c} = N^i_{kc}$, and if $Z^i_k = 1$, then $N^i_{(k+1)c} = N^s_{kc}$.

Next, we define Y^i_t for node i (see Fig. 9.1), where the subscript t is the time slot index. For $kc \leq t < (k+1)c$,

$$Y_t^i \triangleq 2cZ_{k-1}^i + (1 - Z_{k-1}^i)(Y_{kc-1}^i + c) \tag{9.3}$$

with $Y_t^i = 2c$ for cycle 0. Note that Y_t^i remains constant over the duration of any cycle.

Next, we show that Y_t^i forms an upper bound on X_t^i for all t. First, note that, since the source can get a maximum of c new updates in a cycle of c slots, we have $N_{kc}^s - N_{(k-1)c}^s \leq c$. Then, if $Z_{k-1}^i = 1$, i.e., if node i is successfully updated in cycle $k-1$, from (9.2), we have $X_{kc}^i \leq c$, i.e., at the beginning of the kth cycle, the ith node's version age is upper bounded by c. By repeating this argument, $N_{(k+1)c}^s - N_{kc}^s \leq c$, and re-employing (9.2) gives $X_t^i \leq 2c$ throughout the kth cycle, irrespective of the value of Z_k^i, i.e., whether the ith node is updated or not in the kth cycle. Note from (9.3) that when $Z_{k-1}^i = 1$ we have $Y_t^i = 2c$ for the entire kth cycle. Thus, when $Z_{k-1}^i = 1$, we have $X_t^i \leq Y_t^i = 2c$ throughout the kth cycle. On the other hand, if $Z_{k-1}^i = 0$, since the source can get at most c updates over c slots, $X_t^i \leq X_{kc-1}^i + c$ for $t \in [kc, (k+1)c)$. In this case, from (9.3), $Y_t^i = Y_{kc-1}^i + c$. All of these imply that $X_t^i \leq Y_t^i$ for all t, as graphically shown in Fig. 9.1.

Next, we bound $\mathbb{E}[Y_t^i]$ which in turns bounds $\mathbb{E}[X_t^i]$ due to $X_t^i \leq Y_t^i$. Consider $kc \leq t < (k+1)c$. Recall that if node i successfully receives a file in cycle $(k-1)$, i.e., $Z_{k-1}^i = 1$, then Y_t^i drops to $2c$, otherwise it increases by c. Hence, for $m \in \{2, \dots, k+1\}$, $Y_t^i = mc$ is possible only if $Z_{k-m+1}^i = 1$ and $Z_j^i = 0$ for all $j \in \{k-m+2, \dots, k-1\}$, $j \in \mathbb{N}$. That is, the node was updated $m-1$ slots ago, and could not get updated after that, accumulating an increment of c in Y_t^i value in each of the later slots. Further, if $Y_t^i = (k+2)c$, then that means that node i failed to receive any updates since $t = 0$. Since Z_k^i are i.i.d. in k for node i, define $p_n = \mathbb{P}(Z_k^i = 0)$. Since we consider a symmetric fully connected network, all nodes have the same per cycle failure probability p_n in an n node network. Hence, we have the following bound,

$$\mathbb{E}[Y_t^i] = \sum_{m=2}^{k+1} mc(1-p_n)p_n^{m-2} + (k+2)cp_n^k \tag{9.4}$$

$$< \frac{c}{p_n}\sum_{m=2}^{k+2} mp_n^{m-1} \tag{9.5}$$

$$< \frac{c(2-p_n)}{(1-p_n)^2} \tag{9.6}$$

$$= \frac{c}{(1-p_n)^2} + \frac{c}{(1-p_n)} \tag{9.7}$$

Note that the expression in (9.7) is an increasing function of p_n. As the statement of the theorem states $\mathbb{P}(Z_k^i = 1) > 1 - p$ for all $n > n_0$, thus $p_n < p$, and consequently

$$\mathbb{E}[Y_t^i] < \frac{c}{(1-p)^2} + \frac{c}{(1-p)} \tag{9.8}$$

Hence, $\mathbb{E}[X_t^i] < \mathbb{E}[Y_t^i] < \frac{c}{(1-p)^2} + \frac{c}{(1-p)}$ for network size $n > n_0$, and the protocol achieves $O(1)$ age at each node. ■

We have made several relaxations and sub-optimalities in the analysis above. Before we look back at all these relaxations, we first consider an example of such a gossip protocol, the INTERLEAVE protocol proposed in [75]. Motivated by file splitting employed in protocols such BitTorrent, INTERLEAVE is a file splitting based dissemination scheme that is able to satisfy the conditions of Theorem 9.1 as we show next.

9.2.1 An Achievable Gossiping Protocol: INTERLEAVE

The INTERLEAVE protocol of [75] employs an alternating combination of push and pull actions at each user in the network to disseminate a file. The file is split into ℓ pieces by the source, and the pieces are numbered before dissemination to the network. Each piece consumes $\frac{1}{\ell}$ fraction of a time slot for transmission, which we define to be one *minislot*. Thus, in the timeline, each time slot is further divided into ℓ minislots.

In an odd minislot, each user chooses a target node uniformly at random from the network, and pushes to its target node the highest numbered piece it has received in the previous odd minislots. The source pushes a fresh piece to a target node in the network in every odd minislot, where the pieces are chosen in the order they are numbered. In an even minislot, each user again chooses a target node uniformly at random from the network, and makes a pull request for the lowest numbered piece that is missing in its collection from its target node. Node i is said to have received a particular version of the file when it has accumulated all pieces of that version.

[75, Thm. 6] provides that INTERLEAVE disseminates all ℓ pieces of a file to all n nodes of the network within $9(\ell + \log n)$ minislots, with probability $1 - 5n^{-s}$, for any $s < 0.5$. If we choose $\ell \geq \alpha \log n$ for some constant $\alpha > 0$, then considering a minislot to be $\frac{1}{\ell}$ duration of a full time slot, INTERLEAVE takes $9(1+\frac{\log n}{\ell}) = O(1)$ time for spreading a particular file to the entire network with high probability. Since the probability $1 - 5n^{-s}$ is an increasing function of n, the INTERLEAVE protocol satisfies the requirements of Theorem 9.1 and hence can help achieve $O(1)$ expected age at any node in the network.

The inner-workings of INTERLEAVE may look complicated at first, and it may look as though it requires more information than the gossip algorithm in [5]. To this, we point out that, in INTERLEAVE all nodes are oblivious of the collection of file pieces existing at other nodes of the network. Each node makes a pull request for the lowest numbered piece missing in its private collection, and pushes out the highest numbered piece in its private collection. Therefore, the decision making in INTERLEAVE is independent of the information at other nodes, and thus, the INTERLEAVE protocol is a gossip protocol, as the one in [5].

Next, we note the relaxations made in the calculations in Theorem 9.1: 1) The probability of successfully receiving a file in a cycle, $\mathbb{P}(Z_k^i = 1)$, is replaced by its lower bound $1-p$. 2) At the beginning of each new cycle, all nodes are made to halt transmissions of all files in their possession up till the previous cycle. This could result in some nodes sitting idle in the beginning of a cycle waiting for a piece of the newer version, while they could have continued disseminating pieces of previous version missing in other nodes. 3) Y_t^i is a crude upper bound to X_t^i; Y_t^i only changes value at the end of a cycle, though it is possible for node to gather all pieces of a file before the end of the cycle. 4) The evolution of Y_t^i assumes the worst case scenario where the source gets the maximum possible c version updates in every cycle, and hence file version received by the end of the cycle is always c slots old. Further, Y_t^i is set constant over cycle k, upper bounding the maximum value of X_t^i within that cycle. 5) The protocol may take less than c time slots for larger values of n for disseminating the file to all nodes with same probability. Hence, use of c time slots in a cycle might be sub-optimal.

Finally, as a general remark on Theorem 9.1, interestingly, a protocol can achieve $O(1)$ age even if the success probability $1-p$ has a very small value, causing very fewer nodes to really receive a file in a particular cycle. One reason for this can be that every cycle presents a fresh opportunity for a node to receive a new version and reduce its age. Hence it is okay to miss out some updates in a couple of cycles.

9.3 Gossiping with Multiple Files

We consider a system consisting of a network of n nodes and a library of n files. Each node is responsible for creating version updates for a unique file. We label the node acting as source for file f as node f. Then, file f is termed as the source file for node f. We consider slotted time, and define one time slot to be the time required to receive an entire file. All file transmissions start at the beginning of a time slot and consume an entire time slot. We follow the random phone call model, where in every time slot, each user uniformly at random picks another user, referred to as the target, and communicates with either a pull or a push action based on a gossip protocol.

Node f receives version updates for its source file f according to an arbitrary distribution. Thus, in each time slot, node f may or may not get a newer version update. The updates at all nodes for their respective source files are considered uncoordinated, implying that some files may get updated more frequently at their respective source nodes than others. All users wish to obtain the latest possible version of all files at any point of time.

Theorem 9.2 *In a fully connected network of n users and n files, where users communicate according to a gossip protocol, if there exists $p \in [0, 1)$ and $n_0 \in \mathbb{N}$, such that for all $n > n_0$, the probability of a random node receiving all n files in $O(n)$ time is greater than $1 - p$, then the protocol achieves $O(n)$ age for each file at each node.*

Proof: Let $cn = O(n)$ time slots, $c \in \mathbb{N}$, which is also taken to be the duration of one cycle. The timeline is divided into consecutive cycles where each cycle consists of cn time slots. At the beginning of each cycle, all nodes begin dissemination of the latest version of their respective source files received by the end of the previous time slot to the network. The network is allowed to gossip over one cycle or cn time slots according to the given protocol. To avoid mixing of multiple versions of different files, at the end of the cycle, all nodes lay off transmission of file versions already in their possession, and circulation of newer versions of all files starts with the beginning of the next cycle.

In time slot t, let N_t^{if} denote the latest version of file f available at node i and let N_t^f be the latest version of file f at its source node f. The instantaneous version age for file f at node i is $X_t^{if} = N_t^f - N_t^{if}$. Therefore, every time source node f is updated with a newer version for file f, the instantaneous version age at every other node for file f increments by one.

The evolution of X_t^{if} is similar to Section 9.2. At the beginning of cycle k, each node f begins transmission of version N_{knc}^f of its source file f to the network. During the cycle, every source node can receive a maximum of cn updates for its source file. The source nodes hold back the newer versions and so the system continues circulation of version N_{knc}^f for each file f. We define Z_k^{if} for cycle k as follows

$$Z_k^{if} = \begin{cases} 1, & \text{node } i \text{ receives version } N_{knc}^f \text{ in cycle } k \\ \\ 0, & \text{node } i \text{ does not receive } N_{knc}^f \text{ in cycle } k \end{cases} \tag{9.9}$$

For upper bounding the expected version age at time t for file f at node i, $\mathbb{E}[X_t^{if}]$, we define Y_t^{if}, for $kcn \leq t < (k+1)cn$, as follows

$$Y_t^{if} = 2ncZ_{k-1}^{if} + (1 - Z_{k-1}^{if})(Y_{knc-1}^{if} + nc) \tag{9.10}$$

with $Y_t^{if} = 2nc$ for cycle 0. Note that Y_t^{if} remains constant over the duration of any cycle.

Since any source node can get a maximum of nc new updates in a cycle of nc slots for its source file, $N_{(k+1)nc}^f - N_{knc}^f \leq nc$. Using arguments similar in the proof of Theorem 9.1 allows us to conclude $Y_t^{if} \geq X_t^{if}$ for all t. In turn, $\mathbb{E}[Y_t^{if}]$ can be upper bounded as in Theorem 9.1, as given below

$$\mathbb{E}[Y_t^{if}] < \frac{cn}{(1-p)^2} + \frac{cn}{(1-p)} \tag{9.11}$$

Hence $\mathbb{E}[X_t^{if}] < \frac{cn}{(1-p)^2} + \frac{cn}{(1-p)} = O(n)$. ■

9.3.1 An Achievable Gossiping Protocol: RLC

We now show that the random linear coding (RLC) gossip protocols proposed in [73] fall under the category of such protocols, which work on the principle of algebraic mixing of files for faster simultaneous dissemination of all files. In RLC with PUSH protocol, each node creates a linear combination of all the coded packets in its possession at the beginning of each time slot, and transmits it to a randomly chosen target node. The coefficients are randomly chosen from the finite field $\mathbb{F}_q$, with $q \geq n$. In RLC with PULL protocol, nodes pull such random coded messages from their target nodes in every time slot. In both protocols, once a node has collected n independent linear combinations, it can decode all messages.

[73, Thm. 3.1 and Thm. 3.2] provide that both RLC protocols can disseminate n files to n nodes in $O(n)$ time with high probability $1-O(\frac{1}{n})$. Since the probability $1-O(\frac{1}{n})$ is an increasing function of n, both RLC protocols satisfy conditions of Theorem 9.2 in this chapter.

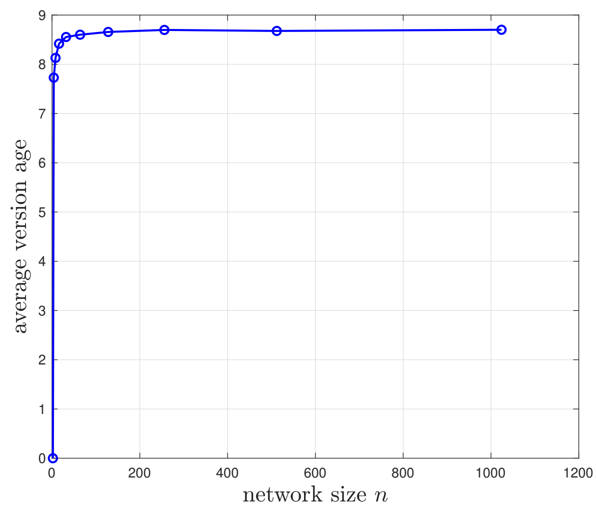


Figure 9.2: Version age versus network size n with INTERLEAVE protocol.

9.4 Numerical Results

We now further validate our results with numerical evaluations. First, we simulate single-file dissemination using the INTERLEAVE protocol in real-time over 200 cycles. We choose number of pieces $\ell = \lfloor \log_2 n \rfloor$ and $c = 18$ in Section 9.2. Since ℓ changes value only when n crosses a power of two, we choose $n = 2^m$, $m \in \mathbb{N}$ to have ℓ increase consistently with n. We assume the source node receives version updates according to Bernoulli(0.7) distribution in every time slot. Fig. 9.2 shows the expected version age per node as a function of network size n. We observe that as the network size grows larger, the version age does not increase, and remains below 9 for the choice of parameters above. For $c = 18$, the upper bound derived in Theorem 9.1 is $\frac{c}{(1-p)^2} + \frac{c}{(1-p)} \geq 36$, which shows that the derived upper bound is an overestimate.

Next, we use RLC with PUSH mechanism for real-time dissemination of n

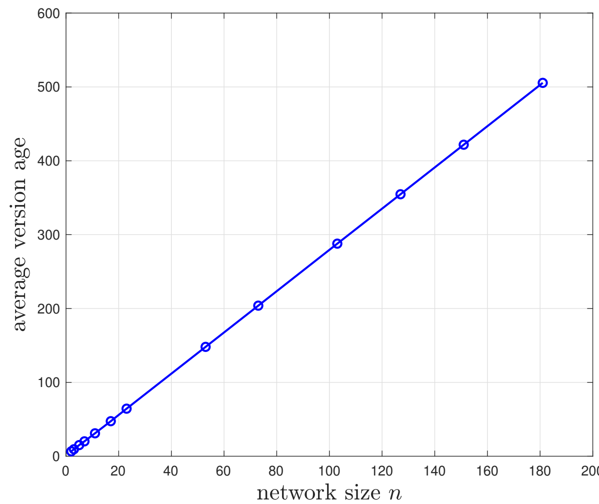


Figure 9.3: Version age versus network size n with RLC with PUSH mechanism.

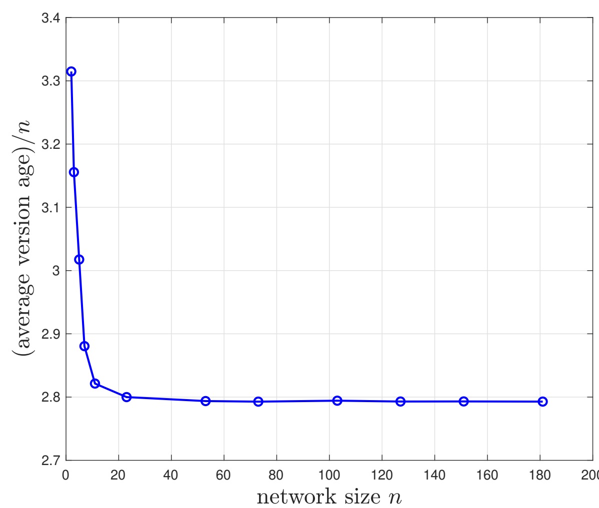


Figure 9.4: (Version age)/n versus n with RLC with PUSH mechanism.

files in an n node network. The authors in [73] point out that their upper bound in [73, Thm. 3.2] is not tight and their simulations suggest that the dissemination time is close to $1.5n + \log_2 n$. We choose $c = 6$ and run the scheme of Section 9.3 over 100 cycles. We assume all nodes receive version updates for their respective source files according to Bernoulli(0.7) distribution in every time slot, and choose n to be prime numbers to keep field size $q = n$ as in [73]. Fig. 9.3 shows that per node per file expected version age grows almost linearly with network size n. Fig. 9.4 plots version age divided by network size n, and we see the graph to be lower than 3 for above choice of parameters, which suggests that the upper bound in Theorem 9.2 (which is ≥ 12) is an overestimate. We note that, in n-file simulations, we consider all files to be successfully decoded by a node at the same time, when n linearly independent coded packets arrive at the node. For some $r < n$, the node might receive r linearly independent combinations of r files and decode them before receiving n linearly independent combinations. Hence, Fig. 9.3 is an upper bound to the actual average version age.

9.5 Conclusion

In this chapter, we studied classes of gossip protocols which achieve $O(1)$ age at each node in a single-file n-node system, and $O(n)$ age for each file at each node in an n-file n-node system. We saw that the bounds hold even though files might not be successfully received by the nodes in every cycle, as each cycle presents a new opportunity to the nodes to get updated. We demonstrated that gossip protocols based on file splitting and network coding fall under the above category of protocols.

CHAPTER 10

Age-Based Cache Updating Under Timestomping

10.1 Introduction

In this chapter, we consider the discrete-time cache updating system shown in Fig. 10.1, where we examine the evolution of the age of information at the user node over the time horizon T. Due to resource constraints, it is not feasible for the source to update the user in every time slot. The source is able to update the user node directly over T_1 time slots and the cache node over T_2 time slots, such that $T_1 \ll T_2$ and $T_1 + T_2 < T$. We assume that if the source sends an update to the cache in a time slot, it is possible for the cache to relay the packet to the user in the same time slot, i.e., packet can reach from source to user both directly or through cache within one time slot. For some of the results in this work to go through, we assume that the source can transmit only once to the user directly between two consecutive source to cache updates. A more general version of the problem can be an interesting future work.

In the absence of any adversary in this system, two observations can be made about the updating policy that should be adopted in this system. First, whenever

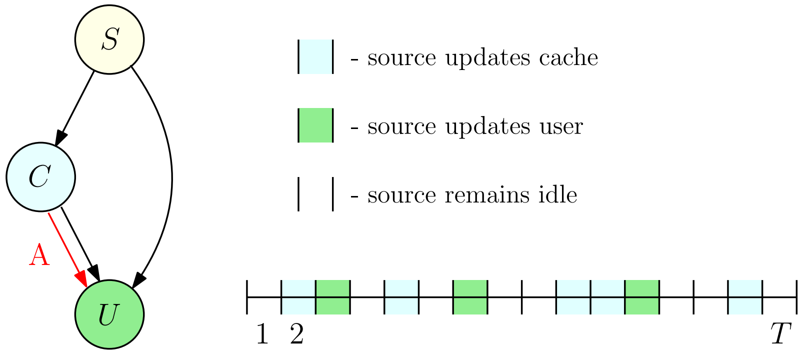


Figure 10.1: System model for caching system.

the cache gets updated, then instead of waiting, the cache should immediately relay the update to the user in the same time slot, which will cause the age of the user to drop to 1 in the very next time slot. Second, both direct update and cache-aided updates have the same impact on age of information at the user, with age dropping to 1 in the following time slot. Hence, from [43], the user should receive an update, from either the cache or the source, at constant time intervals, with inter-update intervals of size $\frac{T}{T_1+T_2+1}$ to minimize its average age.

We next consider the presence of a timestomping adversary at the outgoing link of the cache which aims to deteriorate the average age at the user, through timestamp manipulation. If the user possesses a more recent packet than the packet present at the cache, the adversary would be inclined to send a copy of the cached packet to the user after changing its timestamp to current time t. The manipulated timestamp would trick the user into thinking that the received packet is fresher, making it discard its own packet in favor of the cached packet. On the other hand, if the user's packet is staler than the cached packet, than the adversary would be inclined to not send the cached packet, in other words, stay idle.

In the recent literature, multiple studies have been conducted on the age of information in the presence of an adversary, see [6,126–135]. All these works consider either an adversary that completely eliminates the update packet [129–134] or an adversary that decreases the signal to noise ratio of a communication link [126,127] or an adversary that blocks the communication channel for a duration of time which results in higher age for the communication network [128] or an adversarial gossip network [6,135]. Different than all this work, we consider an adversary that changes the time stamp of an update packet.

In Chapter 4, we previously considered a timestomping adversary. Traditionally, timestomping is used by malware adversaries to make malicious files appear to be out of an attack timeframe and consequently bypass detection. Chapter 4 introduces how age-based cache updating systems are uniquely vulnerable to timestomping based attacks. This is because, timestomping alters the timestamps of packets, and in age-based systems, the decision to accept or reject an incoming packet is based on the comparison of timestamps of various packets.

In Chapter 4, the timestomping adversary attacks one node out of a gossip network of n nodes, and probabilistically alters the timestamps of all the incoming and outgoing packets of the attacked node, with the goal of introducing staleness at all nodes of the network. Further, in Chapter 4, all nodes push updates according to a Poisson process with fixed rates in the continuous time setting and the adversary increases or decreases the timestamps probabilistically, without knowing the age or timestamps at other nodes. Different from Chapter 4, in this work, we consider a discrete time system where the adversary can transmit the cached update packet to the user at any time slot over the time horizon T and the adversary has the full knowledge of the packet transmission policy employed by the source and the timestamp at the user, which allows the adversary to choose its actions to transmit or not in every time slot wisely.

In this chapter, we formulate the timestomping adversarial cache update problem as an online learning problem and study the competitive ratio for this problem. We assume the adversary is oblivious in nature, i.e., the adversary only has knowledge about the source transmission policy, however, it is required to generate the adversarial action for each timeslot over the entire time horizon T before the beginning of the first time slot. In this work, we first provide a deterministic policy and show that this policy is $\left(\frac{1+T+T_1}{1+T_2+T_1+T}\right)\left(1+\frac{T_2}{T_1+1}\right)$ competitive. Then, we use Yao's minimax theorem [136] to find a fundamental lower bound on the competitive ratio for the considered system model.

10.2 System Model and Problem Formulation

We consider a wireless communication network where a source aims to minimize the age of a user with timely delivery of update packets to it. Due to power constraints, the source can only transmit update packets directly to the user for T_1 time slots over the time horizon of T. There is a cache in the system, and the source can transmit update packets to the cache for T_2 time slots over the same time horizon of T. As the transmission cost from the source to the cache is typically less than the transmission cost from the source to the user, we have $T_1 \ll T_2$. In addition, due to the power constraint of the source $T_1 + T_2 < T$. Without loss of generality, let $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{N}$. At a given time slot, the source can only transmit to either the user or to the cache. Each update packet contains its generation time as a time stamp. The cache and the user only keep their respective freshest update packets and discard any staler update packets.

There is also an adversary in the system. The adversary completely controls the communication link between the cache and the user, however, it cannot control the link between the source and the user. Whenever the user receives an update packet from the cache, it compares its time stamp with the time stamp of the already existing packet. If the packet from the cache has relatively fresher time stamp, the user accepts it and discards the existing packet. The adversary can transmit the cached update to the user at any time slot, i.e., there is no power constraint for the adversary. At any given time slot t, the age of the user is defined as $v(t) = t - u(t)$, where $u(t)$ is the generation time of the freshest packet that the user had before time slot t. At time t, if the user receives an update packet directly from the source, and also a cached update packet, then the user always accepts the packet from the source and discards the packet from the cache.

The adversary can manipulate the age of the user by changing the time stamp of the cached packet. For example, at time t, let the time stamp of the packet at the user be $t-2$, and the time stamp of the packet at the cache be $t-4$. The adversary may change the time stamp of the cached packet to t while transmitting it to the user. Upon receiving this packet, the user discards the fresher existing packet and accepts the staler packet which results in an increment of age of the system. Thus, the adversary can choose one of two actions, namely, it can either change the time stamp of the cached packet as the freshest packet, i.e., it changes the time stamp of the cached packet to current time t, or it can change the time stamp of the cached packet as the stalest packet, i.e., it changes the time stamp of the cached packet to the oldest time 0. We denote the former action as 1 and the latter action as 0. For time horizon T, we define a sequence σ as the adversarial sequence where $\sigma \in \{0,1\}^T$. A pictorial representation of the system model is given in Fig. 10.1.

The goal of the source is to design a transmission policy π to reduce the average age of the system, while the goal of the adversary is to design an adversarial sequence σ to increase the age of the system. We consider an oblivious adversary, i.e., the adversary has to generate the whole sequence σ at time 0. We define the average age of the system corresponding to a transmission algorithm π and an adversarial sequence σ as,

$$A^{\pi,\sigma} = \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}_{\pi}[v^{\pi,\sigma}(t)] \tag{10.1}$$

In this work, to analyze the performance of an online algorithm we use the competitive ratio metric [137]. The competitive ratio for any online algorithm π is

$$c^{\pi} = \sup_{\sigma} \frac{A^{\pi,\sigma}}{A^{o,\sigma}} \tag{10.2}$$

where superscript o stands for the optimal algorithm for the adversarial sequence σ.

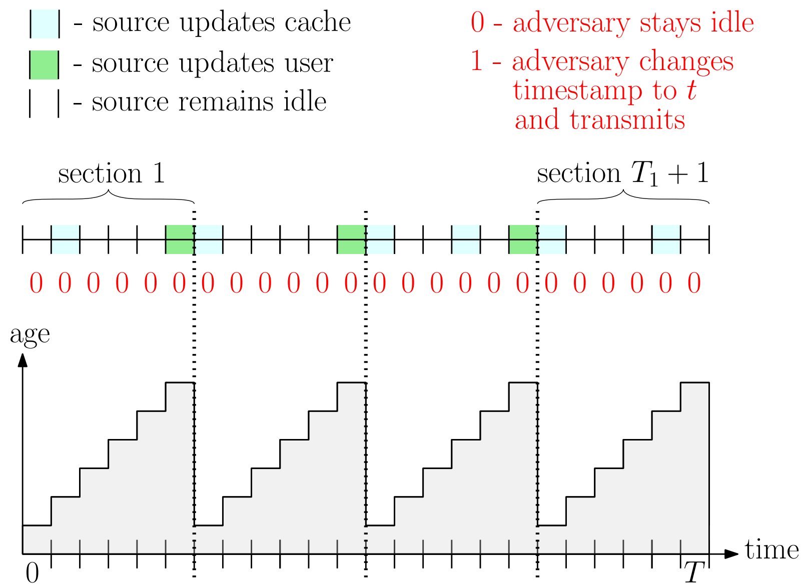


Figure 10.2: Suggested deterministic algorithm, dividing timelines into $T_1 + 1$ sections of equal size. $T_1 = 3$, $T_2 = 6$, $T = 24$.

10.3 Algorithms and Analysis

We consider a greedy deterministic policy $\hat{\pi}$ for the source. The source divides the whole time horizon T into equal $T_1 + 1$ blocks. Thus, each block in the partition has a length of $\frac{T}{T_1+1}$. Under the policy $\hat{\pi}$, the source directly transmits an update packet to the cache at the beginning of each block, starting from the second block and the source directly transmits an update packet to the user at the end of each block, except the last block. Finally, the source chooses any arbitrary $T_2 - T_1$ time slots from the remaining time slots and transmits to the cache. This scheme is pictorially shown in Fig. 10.2. Theorem 10.1 gives an upper bound for the competitive ratio of $\hat{\pi}$.

Theorem 10.1 *The policy $\hat{\pi}$ is $\left(\frac{1+T_1+T}{1+T_2+T_1+T}\right)\left(1 + \frac{T_2}{T_1+1}\right)$ competitive.*

Proof: First, note that the source can transmit to the user (directly or via the cache), only for $T_1 + T_2$ time slots. Now, we can show that if this transmission occurs with uniform spacing over time horizon T, i.e., if we divide the whole time horizon into equal length sections with the length of each section being $\frac{T}{T_1+T_2+1}$, and the age of each section evolves as $1, 2, \cdots, \frac{T}{T_1+T_2+1}$, this provides a universal lower bound on the age for the optimal policy corresponding to any adversarial action σ. Thus, for any adversarial action σ,

$$A^{o,\sigma} \geq \frac{1}{2}\left(1 + \frac{T}{T_2 + T_1 + 1}\right) \tag{10.3}$$

The optimal action for the adversary corresponding to the policy $\hat{\pi}$, is to never transmit any cached packet to the user, i.e., the optimal adversarial action is all zeros. For this adversarial action, the age of the user with policy $\hat{\pi}$, evolves as $1, 2, \cdots, \frac{T}{T_1+1}$ in each section. Thus,

$$A^{\hat{\pi},\sigma} = \frac{1}{2}\left(1 + \frac{T}{T_1 + 1}\right) \tag{10.4}$$

Thus,

$$c^{\hat{\pi}} = \sup_{\sigma} \frac{A^{\hat{\pi},\sigma}}{A^{o,\sigma}} \tag{10.5}$$

$$\leq \frac{\sup_{\sigma} A^{\hat{\pi},\sigma}}{\inf_{\sigma} A^{o,\sigma}} \tag{10.6}$$

$$\leq \left(\frac{1 + T_1 + T}{1 + T_2 + T_1 + T}\right)\left(1 + \frac{T_2}{T_1 + 1}\right) \tag{10.7}$$

which completes the proof. ■

Next, we find a universal lower bound on the competitive ratio for this problem. For this, we leverage Yao's minimax principle [137]. For online learning, Yao's minimax principle states that the competitive ratio for the best randomized policy,

$$\inf_{\pi \in R} c^{\pi} = \sup_{P} \inf_{\pi \in D} c^{\pi,P} \tag{10.8}$$

where P is the arbitrary distribution from which the adversary samples an adversarial sequence, R is the set of randomized policies. Thus, the left hand side of (10.8) provides the best competitive ratio possible. We denote the set of deterministic policies as D. Then, $c^{\pi,P}$ is the smallest competitive ratio for π under the distribution P, i.e., $c^{\pi,P}$ is the infimum of all c which satisfies the following,

$$\mathbb{E}_{\sigma \sim P}[A^{\pi,\sigma}] \leq c\mathbb{E}_{\sigma \sim P}[A^{o,\sigma}] + a \tag{10.9}$$

where a is a constant.

Now, we define a particular probability distribution P_1 to generate the adversarial sequence σ. In P_1, the adversary chooses action 0 with probability $\frac{1}{2}$ and action 1 with probability $\frac{1}{2}$. Thus, we can obtain a fundamental lower bound on the competitive ratio for this problem from Yao's minimax principle as follows. From (10.8),

$$\inf_{\pi \in R} c^{\pi} \geq \inf_{\pi \in D} c^{\pi,P_1} \tag{10.10}$$

$$\geq \frac{\mathbb{E}_{\sigma\sim P}[A^{\pi^*_D,\sigma}]}{\mathbb{E}_{\sigma\sim P}[A^{o,\sigma}]} \tag{10.11}$$

where π^*_D is the best deterministic policy under the probability distribution P_1. Now, we find the numerator and an upper bound for the denominator of (10.11). In the next few lemmas, we first find the best deterministic policy for distribution P_1.

In the next lemma, we study the optimal locations for the cache updates for any arbitrary user update locations. We consider a policy $\bar{\pi}$, for which the source divides the whole time horizon into T_2+1 sections, at the end of each section the source transmits an update packet to the cache except the very last section, the length of a section in which the source transmits an update packet directly to the user is $\frac{T-T_1}{T_2+1}+1$, and the length of a section in which the source does not transmit an update packet directly to the user is $\frac{T-T_1}{T_2+1}$.

Lemma 10.1 *Under the constraint of our system, i.e., the source cannot transmit directly to the user in two consecutive sections, the policy $\bar{\pi}$ assigns the optimal cache update location for any arbitrary user update location.*

Proof: We consider an arbitrary ith section for the policy $\bar{\pi}$. We call the source to cache update for the $(i-1)$th section as the left cache update for the ith section and the source to cache update for the ith section as the right cache update for the ith section. First, we show that if there is a direct source to user update in the ith section, then moving the left cache update for the ith section to the left or to the right increases the expected age of the system. Then, we show that if there is no direct source to user update in the ith section, moving the left cache update for the

ith section to the right or to the left by one time slot increases the expected age of the system.

We consider that there is a direct source to user update in the ith section. Now, we consider a source policy $\tilde{\pi}$, such that it is similar to the policy $\bar{\pi}$, except it shifts the left cache update for the ith section by one time slot left compared to the policy $\bar{\pi}$. Now, we show that the expected age for the policy $\tilde{\pi}$ is higher than the expected age for the policy $\bar{\pi}$. Let us assume that for the policy $\bar{\pi}$ the $(i-1)$th section ends at the time slot t_1. That means that the left cache update for the ith section occurs at the time slot t_1. We define $\frac{T-T_1}{T_2+1} = x_2$, then the ith section has $x_2 + 1$ time slots and the right cache update for the ith time slot occurs at the time slot $t_1 + x_2 + 1$. Let us assume that the direct source to user update occurs at the time slot $t_1 + x_3$. Note that, for any adversarial sequence the difference of the evolution of the age between the policies $\bar{\pi}$ and $\tilde{\pi}$ starts at the time slot t_1 and ends at an arbitrary time slot t_2, where t_2 either depends on at which time slot the source again directly transmits to the user or it depends on at which time slot the adversarial action is 1 after the ith section.

Now, we find several adversarial sequences for which the policy $\bar{\pi}$ provides lower expected age compared to the policy $\tilde{\pi}$. Similarly, we find several adversarial sequences for which the policy $\tilde{\pi}$ provides lower expected age compared to the policy $\bar{\pi}$. Finally, we show that combining the effects of all these sequences we obtain a lower expected age for the policy $\bar{\pi}$ compared to the policy $\tilde{\pi}$.

First, let us consider an adversarial sequence, σ_1, such that $\sigma_1(t_1 - 1) = 0$, $\sigma_1(t_1) = 1$, $\sigma_1(t_1 + x_3 + 1) = 1$ and $\sigma_1(t_1 + x_2 + 1) = 1$. Note that, for σ_1 the ages for the policy $\bar{\pi}$ and $\tilde{\pi}$ differ till the time slot t_1+x_2+1. Let us assume that the age of the system for any of the two policies corresponding to σ_1 at time slot t_1-1 is Δ. The evolution of the age from the time slot t_1 to the time slot t_1+x_2+1 corresponding to the policy $\bar{\pi}$ is $\Delta+1 \to 1 \to 2 \cdots \to x_3 \to 1 \to x_3+2 \to x_3+2 \to \cdots \to x_2+1$. Similarly, the evolution of the age from the from the time slot t_1 to the time slot t_1+x_2 corresponding to the policy $\tilde{\pi}$ is $\Delta+1 \to 1 \to 2 \to \cdots \to x_3+1 \to 1 \to x_3+3 \to x_3+4 \to \cdots \to x_2+2$. Thus, the difference of the ages between the policies $\tilde{\pi}$ and $\bar{\pi}$ corresponding to σ_1 is x_2. Note that, the adversary chooses the sequence σ_1 with probability $\frac{1}{2^4}$. Now, consider that at time slot $t_1+x_2+1+x_4$, the source again directly transmits an update packet to the user. Now, we assume another adversarial sequence σ_2, such that $\sigma_2(t_1-1)=0$, $\sigma_2(t_1)=1$, $\sigma_2(t_1+x_3+1)=1$, $\sigma_2(t_1+x_2+1)=0=\sigma_2(t_1+x_2+2)=\cdots=\sigma_2(t_1+x_2+k)$ and $\sigma_2(t_1+x_2+k+1)=1$, $k<x_4$. With a similar argument made for the sequence σ_1, we observe that for σ_2 the difference of the ages between $\tilde{\pi}$ and $\bar{\pi}$ is x_2+k. Note that the adversary chooses the sequence σ_2 with probability $\frac{1}{2^{4+k}}$.

From the above study it is evident that, given an adversarial sequence, σ such that $\sigma(t_1-1)=0$ and $\sigma(t_1)=1$, the difference of the expected ages between the policies $\tilde{\pi}$ and $\bar{\pi}$ increases with x_4. Let us denote the set of all adversarial sequences, σ such that $\sigma(t_1-1)=0$ and $\sigma(t_1)=1$, as $\mathcal{S}$. Now, we find the minimum difference of the ages between the policies $\bar{\pi}$ and $\tilde{\pi}$ with respect to x_4, whenever the adversary chooses a sequence from $\mathcal{S}$, i.e., when $x_4 = 1$. Thus, we have

$$\mathbb{E}[A^{\tilde{\pi},\sigma} - A^{\bar{\pi},\sigma} | \sigma \in \mathcal{S}] = \sum_{i=1}^{x_2-x_3} \frac{x_2-i+1}{2^{1+i}} + \frac{2x_3}{2^{x_2-x_3+1}} + \sum_{j=0}^{x_2-x_3-2} \frac{x_2-i+1}{2^{1+i}} \tag{10.12}$$

$$= \frac{3x_2}{2} + \frac{1}{2^{x_2-x_3+1}} - \frac{x_3}{2^{x_2-x_3+1}} - \frac{1}{2} \tag{10.13}$$

The first term in (10.13) corresponds to the adversarial sequences in $\mathcal{S}$, for which $\sigma(t_1 + x_2 + 1) = 1$ and in any of the time slot t', from $t_1 + x_3 + 1$ to $t_1 + x_2$, $\sigma(t') = 1$. The second term in (10.13) corresponds to the adversarial sequences in $\mathcal{S}$, for which the adversarial action is 0, from the time slot $t_1 + x_3 + 1$ to the time slot $t_1 + x_2$. The last term in (10.13) corresponds to the adversarial sequences in $\mathcal{S}$, for which $\sigma(t_1 + x_2 + 1) = 0$ and in any of the time slot t', from $t_1 + x_3 + 1$ to $t_1 + x_2$, such that $\sigma(t') = 1$.

Now, we consider all the adversarial sequences, σ such that $\sigma(t_1 - 1) = 1$ and $\sigma(t_1) = 0$. We denote the set of all such adversarial sequences as $\mathcal{S}_1$. Following a similar analysis as we have done to obtain (10.13), we have the following

$$\begin{aligned}\mathbb{E}[A^{\bar{\pi},\sigma} - A^{\tilde{\pi},\sigma} | \sigma \in \mathcal{S}_1] = &\sum_{i=1}^{x_3-1} \sum_{j=1}^{x_2-x_3} \frac{ix_2 + j - 2}{2^{1+i+j}} + \sum_{i=1}^{x_3-1} \frac{(i+1)x_2 - x_3 - 1}{2^{i+x_2-x_3}} \\ &+ \sum_{j=1}^{x_2-x_3} \frac{x_3x_2 + j - 2}{2^{x_3+j}} + \sum_{i=1}^{x_3-1} \sum_{j=1}^{x_2-x_3} \frac{ix_2 + j - 3}{2^{1+i+j}} \\ &+ \sum_{j=1}^{x_2-x_3} \frac{x_3x_2 + j - 3}{2^{x_3+j}} + \frac{x_3x_2 + x_2 - x_3 - 1}{2^{x_2-1}}\end{aligned} \tag{10.14}$$

$$= 2x_2 - \frac{2x_2}{2^{x_3}} - \frac{1}{2} - \frac{1}{2^{x_2-x_3+1}} \tag{10.15}$$

Next, we consider all the adversarial sequences σ such that $\sigma(t_1 - 1) = 1$ and $\sigma(t_1) = 1$. We denote the set of all such adversarial sequences as $\mathcal{S}_2$. Following a similar analysis as we have done to obtain (10.13), we have the following

$$\mathbb{E}[A^{\bar{\pi},\sigma} - A^{\tilde{\pi},\sigma}|\sigma \in \mathcal{S}_2] = \sum_{i=-1}^{x_2-x_3-2} \frac{i}{2^{3+i}} + \frac{(x_2 - x_3 - 1)}{2^{x_2-x_3}} \sum_{i=-2}^{x_2-x_3-3} \frac{i}{2^{4+i}} \tag{10.16}$$

$$= -\frac{1}{2^{x_2-x_3+1}} - \frac{1}{2} \tag{10.17}$$

Finally, we consider all the adversarial sequences σ such that $\sigma(t_1 - 1) = 0$ and $\sigma(t_1) = 0$. We denote the set of all such adversarial sequences as $\mathcal{S}_3$. Following a similar analysis as we have done to obtIN (10.13), we have the following

$$\begin{aligned}\mathbb{E}[A^{\tilde{\pi},\sigma} - A^{\bar{\pi},\sigma}|\sigma \in \mathcal{S}_3] = &\sum_{i=1}^{x_3-1}\sum_{j=1}^{x_2-x_3} \frac{x_2 - i - j + 1}{2^{i+j}} + \sum_{j=1}^{x_2-x_3} \frac{x_2 - x_3 - j + 1}{2^{x_3+j-1}} \\ &+\sum_{i=1}^{x_3-1} \frac{x_3 - i}{2^{x_2-x_3+i}} + \sum_{i=1}^{x_3-1}\sum_{j=1}^{x_2-x_3} \frac{1}{2^{i+j+1}} + \sum_{j=1}^{x_2-x_3} \frac{1}{2^{x_3+j}} \end{aligned}\tag{10.18}$$

$$= x_2 + \frac{2}{2^{x_3}} + \frac{1}{2^{x_2-x_3+1}} - 2.5 \tag{10.19}$$

Now, combining (10.13), (10.15), (10.17) and (10.19), we obtain

$$\mathbb{E}[A^{\bar{\pi},\sigma} - A^{\tilde{\pi},\sigma}] = 2 - \frac{x_2}{2} - \frac{2}{2^{x_2-x_3}} - \frac{2x_2}{2^{x_3}} + \frac{x_3}{2^{x_2-x_3+1}} - \frac{2}{2^{x_3}} \tag{10.20}$$

We see that (10.20) is an increasing function with respect to x_3. Thus, we take $x_3 = x_2$, for which we see that $\mathbb{E}[A^{\bar{\pi},\sigma} - A^{\tilde{\pi},\sigma}] < 0$. In a similar fashion, we can show that, if we shift the left cache update for the ith section to the right, then the average age of the system increases. Also, in a similar fashion, we can show that, if there is no direct source to user update in the ith section, then if we move the left cache update for the ith section to the right or to the left by one time slot, then the expected age of the system increases. ■

In the next lemma, we find the optimal placement for the source to user update, with respect to the source to cache update given by Lemma 10.1.

Lemma 10.2 *Between two cache updates, i.e., the $(i-1)$th and the ith cache updates, the source achieves the minimum age by updating the user directly just before the ith cache update.*

Proof: Consider the source policy given in Lemma 10.1, i.e., policy $\bar{\pi}$. Consider that at the ith section, the source transmits an update packet directly to the user. Let us assume that the ith section starts at t_1, the source to user update occurs at $t_1 + x_3$ and the age of the user at time slot $t_1 + x_3$ is Δ. Now, consider another policy $\tilde{\pi}$ which is similar to $\bar{\pi}$, except it transmits the direct source to user update at the ith section at time slot $t_1 + x_3 + 1$, instead of time slot $t + x_3$. Now, we show that under the distribution P_1, the policy $\tilde{\pi}$ results in lower expected age for the user compared to policy $\bar{\pi}$.

Following a similar analysis as of the proof of Lemma 10.1,

$$\mathbb{E}[A^{\tilde{\pi},\sigma} - A^{\bar{\pi},\sigma}] \leq 0 \tag{10.21}$$

which concludes the proof of this lemma. ■

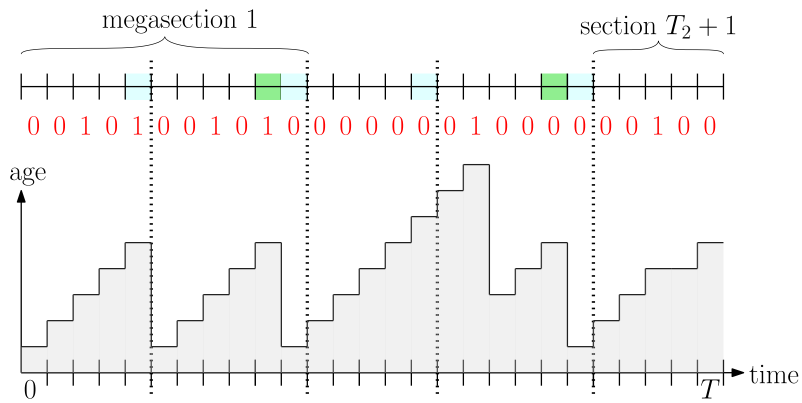


Figure 10.3: Optimal deterministic algorithm, dividing timelines into T_2+1 sections of equal size. $T_1 = 2$, $T_2 = 4$, $T = 27$.

From Lemma 10.1 and Lemma 10.2, we see that the optimal action for the source is to divide the whole time horizon into $T_2 + 1$ sections, and transmit an update packet to the cache at the end of each section except for the last section. The length of a section in which there is no user update is $\frac{T-T_1}{T_2+1}$ and the length of a section in which there is a user update is $\frac{T-T_1}{T_2+1} + 1$. The optimal source to user update locations are just before the source to cache updates. From the symmetry of the cache updates over the time horizon, we claim that the optimal source to user update locations are equidistant over the time horizon T, that is there are $\frac{T_2}{T_1}$ source to cache updates between two consecutive source to user updates. Let us denote this source policy as $\breve{\pi}$. According to (10.11), to get a lower bound on the competitive ratio, we first find the average age for the policy $\breve{\pi}$; see Fig. 10.3 for a pictorial representation of the policy $\breve{\pi}$.

10.3.1 Average Age for Policy $\breve{\pi}$

For the notational convenience, in this section, we denote the source to cache update as the cache update and the source to user update as the user update. Recall that for the policy $\breve{\pi}$ a section i ends with the ith cache update and starts right after the $(i-1)$th cache update. We define a megasection which consists of $\frac{T_2}{T_1}$ slots. The ith mega section ends with a source update followed by a cache update, and starts right after the $(i-1)$th megasection. Recall that, for policy $\breve{\pi}$, each section which does not consist of a user update, has x_2 time slots, where $x_2 = \frac{T-T_1}{T_2+1}$ and the sections which consist of a user update have x_2+1 time slots. Recall that, for a policy π and an adversarial sequence σ, the age of the user at the tth time slot is defined as $v^{\pi,\sigma}(t)$. For notational convenience, in any arbitrary section, we redefine the age of the user at the jth time slot of that section as v_j, where either $j \in 1, 2, \cdots, x_2$ or $j \in 1, 2, \cdots, x_2+1$, depending on the section. We define v_0 to be the age of the user at the last time slot of the preceding section (indicated by time slot 0). If a user does not receive a direct update from the source in time slot $j-1$, then we have

$$\mathbb{E}[v_j|v_{j-1}] = \frac{(v_{j-1}+1)}{2} + \frac{j}{2} \tag{10.22}$$

$$= \frac{v_{j-1}}{2} + \frac{(j+1)}{2} \tag{10.23}$$

Note that, $\mathbb{E}[v_j|v_{j-1}, v_0] = \mathbb{E}[v_j|v_{j-1}]$ as v_j is completely determined by v_{j-1} and the action taken by the adversary in time slot $j-1$, which gives

$$\mathbb{E}[v_j|v_0] = \frac{\mathbb{E}[v_{j-1}|v_0]}{2} + \frac{(j+1)}{2} \tag{10.24}$$

Iteratively repeating this for $\mathbb{E}[v_{j-1}|v_0]$, we obtain

$$\mathbb{E}[v_j|v_0] = \frac{v_0}{2^j} + \sum_{k=2}^{j+1} \frac{k}{2^{j+2-k}} \tag{10.25}$$

Note that the user updates occur only in time slot x_2 of the last section of a megasection, hence (10.25) is always applicable to time slots $\{1, \ldots, x_2\}$ of every section.

Conditioned on v_0, the sum of age of time slots $\{0, \ldots, x_2 - 1\}$ of this section, is

$$\begin{aligned}
\mathbb{E}\left[\sum_{j=0}^{x_2-1} v_j \middle| v_0\right] &= \sum_{j=0}^{x_2-1} \left(\frac{v_0}{2^j} + \sum_{k=2}^{j+1} \frac{k}{2^{j+2-k}}\right) && (10.26)\\
&= \sum_{j=0}^{x_2-1} \frac{v_0}{2^j} + \sum_{j=1}^{x_2-1} \sum_{k=2}^{j+1} \frac{k}{2^{j+2-k}} && (10.27)\\
&= \sum_{j=0}^{x_2-1} \frac{v_0}{2^j} + \sum_{j=1}^{x_2-1} \frac{1}{2^{j+2}} \sum_{k=2}^{j+1} 2^k k && (10.28)\\
&= v_0\alpha + \beta && (10.29)
\end{aligned}$$

where α is computed as

$$\alpha = \sum_{j=0}^{x_2-1} \frac{1}{2^j} = \frac{1-\frac{1}{2^{x_2}}}{1-0.5} = 2\left(1-\frac{1}{2^{x_2}}\right) \tag{10.30}$$

and β is computed as

$$\beta = \sum_{j=1}^{x_2-1} \frac{1}{2^{j+2}} \sum_{k=2}^{j+1} 2^k k \tag{10.31}$$

$$= \sum_{j=1}^{x_2-1} \frac{1}{2^{j+2}} (2\frac{1-(j+2)2^{j+1}+(j+1)2^{j+2}}{(1-2)^2} - 2) \tag{10.32}$$

$$= \frac{1}{(1-2)^2}\left(\sum_{j=1}^{x_2-1} \frac{1}{2^{j+1}} + \sum_{j=1}^{x_2-1} j\right) - \sum_{j=1}^{x_2-1} \frac{1}{2^{j+1}} \tag{10.33}$$

$$= \sum_{j=1}^{x_2-1} j = \frac{(x_2-1)x_2}{2} \tag{10.34}$$

Using nested expectations, we obtain

$$\mathbb{E}\left[\sum_{j=0}^{x_2-1} v_j\right] = \mathbb{E}[v_0]\alpha + \beta \tag{10.35}$$

Let us assume that $\frac{T_2}{T_1} = x_3$. Next, instead of considering the age of arbitrary sections, let the sections of a megasection be indexed by i, where $i \in \{1, \ldots, x_3\}$ and let $v_{j,i}$ denote the age of time slot i of section j of arbitrary megasection. Building on (10.29), let $v_{0,i}$ denote the age of last time slot of the section preceding section i, i.e., $v_{x_2,i-1} = v_{0,i}$ for $i > 1$. Note that $v_{0,1} = v_{x_2+1,x_3} = 1$, since a user update occurs in the second last time slot of every megasection, causing the age in the last time slot of a megasection to drop to 1. Consequently, the sum of expected age of all time slots in a megasection is

$$\mathbb{E}\Bigg[\sum_{i=1}^{x_3-1}\sum_{j=1}^{x_2} v_{j,i} + \sum_{j=1}^{x_2+1} v_{j,x_3}\Bigg] = \mathbb{E}\Bigg[\sum_{i=1}^{x_3}\sum_{j=0}^{x_2-1} v_{j,i} + v_{x_2,x_3}\Bigg] \tag{10.36}$$

$$= \mathbb{E}\Bigg[\sum_{i=1}^{x_3} v_{0,i}\Bigg]\alpha + \beta_1 \tag{10.37}$$

with $\beta_1 = \mathbb{E}\big[v_{x_2,x_3}\big] + x_3\beta$, where $\mathbb{E}\big[v_{x_2,x_3}\big]$ can be obtained from (10.25).

Given $v_{0,1} = 1$, we first compute the expected value of $v_{0,i} = v_{x_2,i-1}$ in terms of $v_{0,i-1}$ using (10.25) by substituting $j = x_2$,

$$\mathbb{E}[v_{0,i}|v_{0,i-1}] = \frac{v_{0,i-1}}{2^{x_2}} + \sum_{k=2}^{x_2+1} \frac{k}{2^{x_2+2-k}} \tag{10.38}$$

$$= r v_{0,i-1} + s \tag{10.39}$$

where r is computed as

$$r = \frac{1}{2^{x_2}} \tag{10.40}$$

and s is computed as

$$s = \sum_{k=2}^{x_2+1} \frac{k}{2^{x_2+2-k}} = \frac{1}{2^{x_2+2}} \sum_{k=2}^{x_2+1} 2^k k = x_2 \tag{10.41}$$

Iteratively repeating this for all $v_{0,\ell}$, we obtain

$$\mathbb{E}[v_{0,i}] = r^{i-1} v_{0,1} + \sum_{\ell=0}^{i-2} r^{\ell} s \tag{10.42}$$

where $v_{0,1} = 1$. Combining (10.37) and (10.42), we have

$$\mathbb{E}\Bigg[\sum_{i=1}^{x_3-1}\sum_{j=1}^{x_2} v_{j,i} + \sum_{j=1}^{x_2+1} v_{j,x_3}\Bigg] = \Bigg[\sum_{i=1}^{x_3} r^{i-1} + s\sum_{i=2}^{x_3}\sum_{\ell=0}^{i-2} r^{\ell}\Bigg]\alpha + \beta_1 \tag{10.43}$$

$$= \frac{x_3x_2(x_2-1)}{2} + 2x_2x_3 + 2 - \frac{1}{2^{x_2x_3}} + \frac{x_2 2^{x_2}}{2^{x_2}-1}\left(\frac{1}{2^{x_2x_3}} - 1\right) \tag{10.44}$$

Let us define the total age of the ith megasection as A_i, $i \in \{1, 2, \cdots, x_3\}$. Note that, the random variables A_i are independent and identically distributed, thus the average expected age of the user for the policy $\check{\pi}$ and the adversarial sequence following a uniform probability distribution is,

$$\mathbb{E}_{\sigma\sim P_1}[A^{\check{\pi},\sigma}] = \frac{1}{T}\sum_{i=1}^{T_1} A_i = \frac{T_1}{T}\frac{1}{T_1}\sum_{i=1}^{T_1} A_i \tag{10.45}$$

From (10.45), for large T_1, as $\frac{T}{T_1}$ is constant, i.e., for large T,

$$\mathbb{E}_{\sigma\sim P_1}[A^{\check{\pi},\sigma}] = \frac{T_1}{T}\mathbb{E}\Bigg[\sum_{i=1}^{x_3-1}\sum_{j=1}^{x_2} v_{j,i} + \sum_{j=1}^{x_2+1} v_{j,x_3}\Bigg] \tag{10.46}$$

10.3.2 Upper Bound on the Offline Optimal Policy

Next, we find an upper bound on the expected age of the offline optimal policy, where the adversarial sequences follow the probability distribution P_1. Let us define a policy $\check{\pi}_1$ which is similar to the policy $\check{\pi}$, except for policy $\check{\pi}_1$ the source to cache update does not occur at the end of each section but a cache update can occur at any

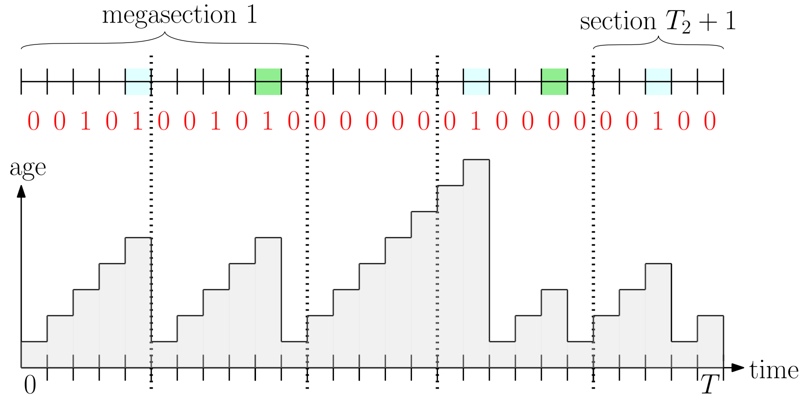


Figure 10.4: A policy $\breve{\pi}_1$ to find an upper bound on the offline optimal policy, dividing timelines into $T_2 + 1$ sections of equal size. $T_1 = 2$, $T_2 = 4$, $T = 27$.

time slot between the end of a section and the end of the next section depending on whenever the adversarial sequence is 1 for the first time. A pictorial representation of this policy is shown in Fig. 10.4.

As the structure of both the policies $\breve{\pi}$ and $\breve{\pi}_1$ are the same, we again define the megasection and the section. As before, the number of sections in a megasection is x_3, the number of time slots in a section in which there is no direct source to user update is x_2 and the other sections have $x_2 + 1$ time slots. Then, we have

$$\mathbb{E}[v_j|v_0] = \frac{v_0 + j}{2^j} + \sum_{k=0}^{j-1} \frac{j-k}{2^{k+1}} \tag{10.47}$$

This is because, if the adversary does not transmit any packet in the time slots $\{0, \dots, j-1\}$, which has the probability $\frac{1}{2^j}$, the user age increments by j units over v_0 in time slot j. However, if for some $k < j$, the adversary forwards a packet to the user for the first time in time slot k, which has probability $\frac{1}{2^{k+1}}$, the user age at time slot j would be $j - k$.

As before, this formula is applicable in absence of user update before timeslot

j in that section, which occurs only at the second last timeslot of the last section of a megasection, hence this formula is always applicable for $j = \{1, \ldots, x_2\}$.

Conditioned on a_0, the sum of age of timeslots $\{0, \ldots, x_2 - 1\}$ of this section is

$$\mathbb{E}\left[\sum_{j=0}^{x_2-1} v_j \middle| v_0\right] = \sum_{j=0}^{x_2-1}\left(\frac{v_0 + j}{2^j} + \sum_{k=0}^{j-1}\frac{j-k}{2^{k+1}}\right) \tag{10.48}$$

$$= v_0\bar{\alpha} + \bar{\beta} \tag{10.49}$$

where $\bar{\alpha}$ is

$$\bar{\alpha} = 2\left(1 - \frac{1}{2^{x_2}}\right) \tag{10.50}$$

and $\bar{\beta}$ is

$$\bar{\beta} = 4 - \frac{2x_2}{2^{x_2}} - \frac{4}{2^{x_2}} + \frac{(x_2 - 3)x_2}{2} \tag{10.51}$$

Approaching as in (10.37), let $a_{j,i}$ denote the age of the jth timeslot of the ith section of a megasection, such that $a_{0,1} = 1 = a_{x_2+1,x_3}$. Then, the sum of expected age of all timeslots in a megasection is

$$\mathbb{E}\left[\sum_{i=1}^{x_3-1}\sum_{j=1}^{x_2} v_{j,i} + \sum_{j=1}^{x_2+1} v_{j,x_3}\right] = \mathbb{E}\left[\sum_{i=1}^{x_3}\sum_{j=0}^{x_2-1} v_{j,i} + v_{x_2,x_3}\right] \tag{10.52}$$

$$= \mathbb{E}\left[\sum_{i=1}^{x_3} v_{0,i}\right]\bar{\alpha} + \bar{\beta}_1 \tag{10.53}$$

with $\bar{\beta}_1 = \mathbb{E}[v_{x_2,x_3}] + x_3\bar{\beta}$, where v_{x_2,x_3} is obtained from (10.47). Given $v_{0,1} = 1$, we compute the expected value of $v_{0,i} = v_{x_2,i-1}$ in terms of $v_{0,i-1}$ using (10.47) by substituting $j = x_2$

$$\mathbb{E}[v_{0,i}|v_{0,i-1}] = \frac{v_{0,i-1} + x_2}{2^{x_2}} + \sum_{k=0}^{x_2-1} \frac{x_2 - k}{2^{k+1}} \tag{10.54}$$

$$= \bar{r}v_{0,i-1} + \bar{s} \tag{10.55}$$

where $\bar{r}$ is

$$\bar{r} = \frac{1}{2^{x_2}} \tag{10.56}$$

and $\bar{s}$ is

$$\bar{s} = \frac{x_2}{2^{x_2}} + x_2 + \frac{1}{2^{x_2}} - 1 \tag{10.57}$$

Iteratively repeating this for all $v_{0,\ell}$, we obtain

$$\mathbb{E}[v_{0,i}] = \bar{r}^{i-1}v_{0,1} + \sum_{\ell=0}^{i-2} \bar{r}^{\ell}\bar{s} \tag{10.58}$$

where $v_{0,1} = 1$. Summing over all i, (10.52) becomes

$$\mathbb{E}\left[\sum_{i=1}^{x_3-1}\sum_{j=1}^{x_2} v_{j,i} + \sum_{j=1}^{x_2+1} v_{j,x_3}\right] = \left[\sum_{i=1}^{x_3} \bar{r}^{i-1} + \bar{s}\sum_{i=2}^{x_3}\sum_{\ell=0}^{i-2} r^{\ell}\right]\bar{\alpha} + \bar{\beta}_1 \tag{10.59}$$

$$= 3 - \frac{2x_3}{2^{x_2}} + 2x_3 + x_2 - \frac{2}{2^{x_2x_3}} + \frac{x_3x_2(x_2-3)}{2} + 2x_3x_2$$

$$+\frac{x_2(2^{x_2}+1)}{2^{x_2x_3}(2^{x_2}-1)}-\frac{2x_22^{x_2}}{2^{x_2}-1} \tag{10.60}$$

Similar to (10.46), we obtain

$$\mathbb{E}_{\sigma\sim P_1}[A^{o,\sigma}] \leq \frac{T_1}{T}\mathbb{E}\Big[\sum_{i=1}^{x_3-1}\sum_{j=1}^{x_2} v_{j,i} + \sum_{j=1}^{x_2+1} v_{j,x_3}\Big] \tag{10.61}$$

Thus, from (10.11), (10.46) and (10.61), we obtain a universal lower bound for the competitive ratio for this system model.

10.4 Conclusion

In this chapter, we considered a system where a source aims to minimize the age of a user by transmitting fresh update packets to the user over a time horizon T time slots. The source can directly transmit update packets to the user for T_1 time slots. The source can also transmit update packets to a cache in the system for T_2 time slots, where $T_1 \ll T_2$ and $T_1 + T_2 < T$. There is an adversary in the system which completely controls the cache to user communication link. Whenever the user receives a cached update packet, the user compares the time stamp of the packet it has with the time stamp of the received packet, and keeps only the fresher packet while discards the staler packet. The adversary can change the time stamp of the cached update packet, and by doing so, it can deceive the user by making it store a staler update packet with the cost of discarding a fresher update packet. The goal of the adversary is to increase the age of the user, while the goal of the source is to minimize the age of the user. We formulated this problem as an online learning problem and studied the competitive ratio for this problem. First, we proposed a deterministic algorithm and provided an upper bound on the competitive ratio for the proposed algorithm. Then, we proposed a universal lower bound for the studied system model. The extensions of the model with multiple users and/or multiple caches are interesting future directions.

CHAPTER 11

Conclusions

In this dissertation, we focused on improving timely information dissemination in large gossip networks through innovative strategies such as optimal update paths, file slicing, network coding, and intelligent subscription approaches. Additionally, we explored vulnerabilities of dense networks to new threats such as timestomping, jamming, misinformation, and unreliable information propagation, and showed how the very mechanisms that make these networks efficient, such as age-based packet exchange and gossiping, can become vulnerabilities under certain attacks.

In Chapter 2, we investigated timeliness of multiple time-varying files in a parallel cache network using binary freshness, age and version age metrics. We derived closed-form expressions for these metrics and determined that a file achieves maximum freshness when it does not have to split its updates rates across routes. This insight guided the development of an approximate rate allocation policy for all files, derived by solving an auxiliary single-cache problem and adapting its solution to our multi-cache network to offer a sub-optimal solution. Further, we established an upper bound for the gap between the optimal policy and the approximate policy.

In Chapter 3, we studied the impact of jamming attacks on network age. We found for a ring network of n nodes that with $\tilde{n} = cn^\alpha$ jammers, the average version age scales as $\sqrt{n}$ for $\alpha \in \left[0, \frac{1}{2}\right)$ and as n^α for $\alpha \in \left[\frac{1}{2}, 1\right]$. This resilience allows ring networks to withstand up to $\sqrt{n}$ jammers, since version age of a ring network without any jammers scales as $\sqrt{n}$. We also examined fully connected networks, where through a greedy placement of $\tilde{n}$ jammers which consolidates links into a mini-fully connected structure, we demonstrated an average version age scaling of $O(\log n)$ for $\tilde{n} = O(n \log n)$ and $O(n^{\alpha-1})$ for $\tilde{n} = O(n^\alpha)$ with $1 < \alpha \leq 2$. Thus, fully connected networks resist up to $n \log n$ jammers, since version age of a fully connected network without jammers scales as $\log n$.

In Chapter 4, we studied the effects of a new attack, timestomping attack, on the age of gossip, first, in a large fully connected network. We showed that one infected node in such a network can increase the age at all other nodes from $O(\log n)$ to $O(n)$ through timestamp manipulation. Further, we showed that the optimal behavior for the adversary is to reset the timestamps of all outgoing packets to current time thereby disguising them as current packets and of all incoming packets to an outdated time to prevent their acceptance at the infected node. We then analyzed the unidirectional ring network, where we showed that the adversarial effect on age scaling of a node is limited by its distance from the adversary, and the age scaling for a large fraction of the network continues to be $O(\sqrt{n})$, unchanged from the case with no adversary.

In Chapter 5, we examined how packet mutations during transmissions can propagate misinformation within the network. For the case when a receiving node encounters both accurate information and misinformation corresponding to the same version, we considered two models: one where truth prevails over misinformation and one where misinformation prevails over truth. We characterized the expected fraction of nodes with accurate information and the version age at the nodes as a function of various network parameters. We found that when truth prevails, extreme gossiping rates (either very high or very low) effectively counter misinformation, while moderate gossiping rates worsen its spread. Conversely, when misinformation prevails, higher gossiping rates lead to more widespread misinformation.

In Chapter 6, we analyzed a system model where packets containing information about a dynamic event are disseminated to a network of nodes by both reliable and unreliable sources. The file exchange protocol implemented at the network nodes allows them to sacrifice their freshness of packets, quantified by the version age of information metric, by up to G versions to enhance packet reliability. Our analysis demonstrated that this sacrificing capacity G creates a trade-off between reliability and information freshness at the network nodes. Additionally, we explored the dependence of network reliability and freshness on various network parameters, including update rates at different nodes and the overall network size.

In Chapter 7, we derived closed-form expressions for expected age and expected version age in cache-aided networks with tree structures where updates on all links are sent according to ordinary renewal processes that are not necessarily Poisson. We observed that only the links involved in the path between the source and an end-user are relevant to the age dynamics of the end-user, and saw that both expected age and expected version age have additive structures composed of independent contributions of the intermediate links along the path. We saw that the age at the end-user is directly proportional to the variance of inter-update times in these links, and therefore, nodes should request packets at near constant time intervals to increase their freshness. Finally, we used Poisson limits of partitions to study expected age of information in large fully connected networks for arbitrary i.i.d. inter-update times, and remarked that the expected age scales as $O(\log n)$ in symmetric fully connected networks.

In Chapter 8, we investigated a Stackelberg game between a server and users in a communication system, where the server, tracking a time-varying event, aims to maximize its profit by increasing subscriptions from users and reducing event sampling costs, while users choose between subscribing to server or relying on gossip from their neighbors based on their timeliness needs. We analyzed equilibrium strategies in several directed and undirected networks, observing the general theme that well-connected networks have fewer subscribers since well-connected users can discourage multiple neighboring nodes from subscribing. Our results showed an inverse relation between expected age and both gossiping and server sampling rates.

In Chapter 9, we studied classes of gossip protocols which achieve $O(1)$ age at each node in a single-file n-node system, and $O(n)$ age for each file at each node in an n-file n-node system (improving both results by $\log n$ from the conventional gossiping algorithms). We saw that the bounds hold even though files might not be successfully received by the nodes in every cycle, as each cycle presents a new opportunity to the nodes to get updated. We demonstrated that gossip protocols based on file splitting and network coding fall under the above category of protocols.

In Chapter 10, we studied timestomping attacks in a simplified communication model, where a source attempts to minimize the age of a user by transmitting fresh update packets to the user, but due to a power constraint, can only transmit updates directly to the user for T_1 timeslots over a time horizon of T. A cache node, which can afford more frequent transmissions, lies in between the source and the user, however the communication link between the cache and the user is under attack by a timestomping adversary that wishes to increase the age of the user. We formulated this problem as an online learning problem and studied the competitive ratio for this problem. First, we proposed a deterministic algorithm and provided an upper bound on the competitive ratio for the proposed algorithm. Then, we proposed a universal lower bound for the studied system model.

The contents of Chapter 2 are published in [8, 138], Chapter 3 in [6, 135], Chapter 4 in [118,139], Chapter 5 in [140,141], Chapter 6 in [102,142,143], Chapter 7 in [144–146], Chapter 8 in [125, 147, 148], Chapter 9 in [104], Chapter 10 in [149]. Other related works not discussed in this thesis are [119, 150].

Bibliography

[1] A. Kosta, N. Pappas, and V. Angelakis. Age of information: A new concept, metric, and tool. *Foundations and Trends in Networking*, 12(3):162–259, November 2017.

[2] Y. Sun, I. Kadota, R. Talak, and E. Modiano. Age of information: A new metric for information freshness. *Synthesis Lectures on Communication Networks*, 12(2):1–224, December 2019.

[3] R. D. Yates, Y. Sun, R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus. Age of information: An introduction and survey. *IEEE Journal on Selected Areas in Communications*, 39(5):1183–1210, May 2021.

[4] R. D. Yates. Timely gossip. In *IEEE SPAWC*, September 2021.

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

[6] P. Kaswan and S. Ulukus. Age of gossip in ring networks in the presence of jamming attacks. In *Asilomar Conference*, October 2022.

[7] M. Bastopcu and S. Ulukus. Information freshness in cache updating systems. *IEEE Transactions on Wireless Communications*, 20(3):1861–1874, March 2021.

[8] P. Kaswan, M. Bastopcu, and S. Ulukus. Freshness based cache updating in parallel relay networks. In *IEEE ISIT*, July 2021.

[9] M. Bastopcu, B. Buyukates, and S. Ulukus. Gossiping with binary freshness metric. In *IEEE Globecom*, December 2021.

[10] A. Maatouk, S. Kriouile, M. Assaad, and A. Ephremides. The age of incorrect information: A new performance metric for status updates. *IEEE/ACM Transactions on Networking*, 28(5):2215–2228, October 2020.

[11] S. K. Kaul, M. Gruteser, V. Rai, and J. Kenney. Minimizing age of information in vehicular networks. In *IEEE Infocom*, March 2011.

[12] S. Kaul, R. D. Yates, and M. Gruteser. Real-time status: How often should one update? In *IEEE Infocom*, March 2012.

[13] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly. Distributed multi-hop scheduling and medium access with delay and throughput constraints. In *IEEE MobiCom*, July 2001.

[14] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah. Throughput-delay trade-off in wireless networks. In *IEEE Infocom*, March 2004.

[15] N. Bisnik and A. Abouzeid. Delay and throughput in random access wireless mesh networks. In *IEEE ICC*, June 2006.

[16] P. Li, Y. Fang, and J. Li. Throughput, delay, and mobility in wireless ad hoc networks. In *IEEE Infocom*, March 2010.

[17] N. Lu and X. S. Shen. Scaling laws for throughput capacity and delay in wireless networks — A survey. *IEEE Communications Surveys & Tutorials*, 16(2):642–657, August 2013.

[18] E. Najm, R. D. Yates, and E. Soljanin. Status updates through M/G/1/1 queues with HARQ. In *IEEE ISIT*, June 2017.

[19] A. Soysal and S. Ulukus. Age of information in G/G/1/1 systems: Age expressions, bounds, special cases, and optimization. *IEEE Transactions on Information Theory*, 67(11):7477–7489, November 2021.

[20] R. D. Yates and S. K. Kaul. The age of information: Real-time status updating by multiple sources. *IEEE Transactions on Information Theory*, 65(3):1807–1827, March 2019.

[21] P. Zou, O. Ozel, and S. Subramaniam. Optimizing information freshness through computation–transmission tradeoff and queue management in edge computing. *IEEE/ACM Transactions on Networking*, 29(2):949–963, February 2021.

[22] S. Farazi, A. G. Klein, and D. R. Brown III. Average age of information for status update systems with an energy harvesting server. In *IEEE Infocom*, April 2018.

[23] X. Wu, J. Yang, and J. Wu. Optimal status update for age of information minimization with an energy harvesting source. *IEEE Transactions on Green Communications and Networking*, 2(1):193–204, March 2018.

[24] A. Baknina, O. Ozel, J. Yang, S. Ulukus, and A. Yener. Sending information through status updates. In *IEEE ISIT*, June 2018.

[25] S. Leng and A. Yener. Age of information minimization for an energy harvesting cognitive radio. *IEEE Transactions on Cognitive Communications and Networking*, 5(2):427–43, June 2019.

[26] A. Arafa, J. Yang, S. Ulukus, and H. V. Poor. Age-minimal transmission for energy harvesting sensors with finite batteries: Online policies. *IEEE Transactions on Information Theory*, 66(1):534–556, January 2020.

[27] M. A. Abd-Elmagid, H. S. Dhillon, and N. Pappas. A reinforcement learning framework for optimizing age of information in RF-powered communication systems. *IEEE Transactions on Communications*, 68(8):4747–4760, May 2020.

[28] N. Pappas, Z. Chen, and M. Hatami. Average AoI of cached status updates for a process monitored by an energy harvesting sensor. In *CISS*, March 2020.

[29] A. Arafa and S. Ulukus. Timely updates in energy harvesting two-hop networks: Offline and online policies. *IEEE Transactions on Wireless Communications*, 18(8):4017–4030, August 2019.

[30] J. Cho and H. Garcia-Molina. Effective page refresh policies for web crawlers. *ACM Transactions on Database Systems*, 28(4):390–426, December 2003.

[31] A. Kolobov, Y. Peres, E. Lubetzky, and E. Horvitz. Optimal freshness crawl under politeness constraints. In *ACM SIGIR Conference*, July 2019.

[32] E. T. Ceran, D. Gunduz, and A. Gyorgy. A reinforcement learning approach to age of information in multi-user networks. In *IEEE PIMRC*, September 2018.

[33] M. Bastopcu and S. Ulukus. Minimizing age of information with soft updates. *Journal of Communications and Networks*, 21(3):233–243, June 2019.

[34] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano. Scheduling policies for minimizing age of information in broadcast wireless networks. *IEEE/ACM Transactions on Networking*, 26(6):2637–2650, December 2018.

[35] Y. Hsu. Age of information: Whittle index for scheduling stochastic arrivals. In *IEEE ISIT*, June 2018.

[36] A. M. Bedewy, Y. Sun, S. Kompella, and N. B. Shroff. Age-optimal sampling and transmission scheduling in multi-source systems. In *ACM MobiHoc*, July 2019.

[37] E. Ozfatura, B. Buyukates, D. Gunduz, and S. Ulukus. Age-based coded computation for bias reduction in distributed learning. In *IEEE Globecom*, December 2020.

[38] H. H. Yang, A. Arafa, T. Q. S. Quek, and H. V. Poor. Age-based scheduling policy for federated learning in mobile edge networks. In *IEEE ICASSP*, May 2020.

[39] N. Rajaraman, R. Vaze, and R. Goonwanth. Not just age but age and quality of information. *IEEE Journal on Selected Areas in Communications*, 39(5):1325–1338, March 2021.

[40] M. Bastopcu and S. Ulukus. Age of information for updates with distortion: Constant and age-dependent distortion constraints. *IEEE/ACM Transactions on Networking*, 29(6):2425–2438, December 2021.

[41] O. Ayan, M. Vilgelm, M. Klugel, S. Hirche, and W. Kellerer. Age-of-information vs. value-of-information scheduling for cellular networked control systems. In *ACM ICCPS*, April 2019.

[42] S. Banerjee, R. Bhattacharjee, and A. Sinha. Fundamental limits of age-of-information in stationary and non-stationary environments. In *IEEE ISIT*, June 2020.

[43] R. D. Yates, P. Ciblat, A. Yener, and M. Wigger. Age-optimal constrained cache updating. In *IEEE ISIT*, June 2017.

[44] H. Tang, P. Ciblat, J. Wang, M. Wigger, and R. D. Yates. Age of information aware cache updating with file- and age-dependent update durations. In *IEEE WiOpt*, June 2020.

[45] J. Liu, X. Wang, and H. Dai. Age-optimal trajectory planning for UAV-assisted data collection. In *IEEE Infocom*, April 2018.

[46] M. A. Abd-Elmagid and H. S. Dhillon. Average peak age-of-information minimization in UAV-assisted IoT networks. *IEEE Transactions on Vehicular Technology*, 68(2):2003–2008, February 2019.

[47] M. Bastopcu and S. Ulukus. Timely group updating. In *CISS*, March 2021.

[48] B. Buyukates, A. Soysal, and S. Ulukus. Scaling laws for age of information in wireless networks. *IEEE Transactions on Wireless Communications*, 20(4):2413–2427, April 2021.

[49] B. Buyukates, A. Soysal, and S. Ulukus. Age of information in multihop multicast networks. *Journal of Communications and Networks*, 21(3):256–267, July 2019.

[50] M. Wang, W. Chen, and A. Ephremides. Reconstruction of counting process in real-time: The freshness of information through queues. In *IEEE ICC*, July 2019.

[51] M. Bastopcu and S. Ulukus. Who should Google Scholar update more often? In *IEEE Infocom*, July 2020.

[52] Y. Sun, Y. Polyanskiy, and E. Uysal-Biyikoglu. Remote estimation of the Wiener process over a channel with random delay. In *IEEE ISIT*, June 2017.

[53] M. Bastopcu and S. Ulukus. Timely tracking of infection status of individuals in a population. In *IEEE Infocom*, May 2021.

[54] J. Yun, C. Joo, and A. Eryilmaz. Optimal real-time monitoring of an information source under communication costs. In *IEEE CDC*, December 2018.

[55] C. Kam, S. Kompella, and A. Ephremides. Age of incorrect information for remote estimation of a binary Markov source. In *IEEE Infocom*, July 2020.

[56] J. Chakravorty and A. Mahajan. Remote estimation over a packet-drop channel with Markovian state. *IEEE Transactions on Automatic Control*, 65(5):2016–2031, July 2020.

[57] P. Mayekar, P. Parag, and H. Tyagi. Optimal source codes for timely updates. *IEEE Transactions on Information Theory*, 66(6):3714–3731, March 2020.

[58] M. Bastopcu, B. Buyukates, and S. Ulukus. Selective encoding policies for maximizing information freshness. *IEEE Transactions on Communications*, 69(9):5714–5726, September 2021.

[59] D. Ramirez, E. Erkip, and H. V. Poor. Age of information with finite horizon and partial updates. In *IEEE ICASSP*, May 2020.

[60] B. Buyukates and S. Ulukus. Timely distributed computation with stragglers. *IEEE Transactions on Communications*, 68(9):5273–5282, September 2020.

[61] M. Bastopcu and S. Ulukus. Maximizing information freshness in caching systems with limited cache storage capacity. In *Asilomar Conference*, November 2020.

[62] S. Zhang, J. Li, H. Luo, J. Gao, L. Zhao, and X. S. Shen. Towards fresh and low-latency content delivery in vehicular networks: An edge caching aspect. In *IEEE WCSP*, October 2018.

[63] W. Gao, G. Cao, M. Srivatsa, and A. Iyengar. Distributed maintenance of cache freshness in opportunistic mobile networks. In *IEEE ICDCS*, June 2012.

[64] C. Kam, S. Kompella, G. D. Nguyen, J. Wieselthier, and A. Ephremides. Information freshness and popularity in mobile caching. In *IEEE ISIT*, June 2017.

[65] Y. Gu, Q. Wang, H. Chen, Y. Li, and B. Vucetic. Optimizing information freshness in two-hop status update systems under a resource constraint. *IEEE Journal on Selected Areas in Communications*, 39(5):1380–1392, March 2021.

[66] B. Abolhassani, J. Tadrous, A. Eryilmaz, and E. Yeh. Fresh caching for dynamic content. In *IEEE Infocom*, May 2021.

[67] R. D. Yates. The age of information in networks: Moments, distributions, and sampling. *IEEE Transactions on Information Theory*, 66(9):5712–5728, September 2020.

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

[69] A. J. Demers, D. H. Greene, C. H. Hauser, W. Irish, J. Larson, S. Shenker, H. E. Sturgis, D. C. Swinehart, and D. B. Terry. Epidemic algorithms for replicated database maintenance. In *ACM PODC*, August 1987.

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

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

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

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

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

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

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

[77] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels. Dynamo: Amazon's highly available key-value store. *ACM SIGOPS Oper. Syst. Rev.*, 41(6):205–220, 2007.

[78] A. Lakshman and P. Malik. Cassandra: A decentralized structured storage system. *ACM SIGOPS Oper. Syst. Rev.*, 44(2):35–40, 2010.

[79] S. Zhang, N. Zhang, X. Fang, P. Yang, and Xuemin S. Shen. Self-sustaining caching stations: Toward cost-effective 5G-enabled vehicular networks. *IEEE Communications Magazine*, 55(11):202–208, November 2017.

[80] Z. Gu, H. Lu, M. Zhang, H. Sun, and C. W. Chen. Association and caching in relay-assisted mmWave networks: A stochastic geometry perspective. *IEEE Transactions on Wireless Communications*, 20(12):8316–8332, December 2021.

[81] B. T. Bacinoglu, E. T. Ceran, and E. Uysal-Biyikoglu. Age of information under energy replenishment constraints. In *IEEE ITA*, February 2015.

[82] G. D. Nguyen, S. Kompella, C. Kam, J. E. Wieselthier, and A. Ephremides. Impact of hostile interference on information freshness: A game approach. In *IEEE WiOpt*, May 2017.

[83] A. Garnaev, W. Zhang, J. Zhong, and R. D. Yates. Maintaining information freshness under jamming. In *IEEE Infocom*, May 2019.

[84] Y. Xiao and Y. Sun. A dynamic jamming game for real-time status update. In *IEEE Infocom*, April 2018.

[85] S. Banerjee and S. Ulukus. Age of information in the presence of an adversary. In *IEEE Infocom*, May 2022.

[86] S. Banerjee and S. Ulukus. Game theoretic analysis of an adversarial status updating system. In *IEEE ISIT*, June 2022.

[87] J. Augustine, C. Avin, M. Liaee, G. Pandurangan, and R. Rajaraman. Information spreading in dynamic networks under oblivious adversaries. In *DISC*, 2016.

[88] C. Georgiou, S. Gilbert, R. Guerraoui, and D. R. Kowalski. On the complexity of asynchronous gossip. In *PODC*, 2008.

[89] W. Minnaard. Timestomping NFTS. Master's thesis, University of Amsterdam, July 2014.

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

[91] S. Vosoughi, D. Roy, and S. Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

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

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

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

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

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

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

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

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

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

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

[102] P. Kaswan and S. Ulukus. Sacrificing age for reliable information in age-based gossiping. Available at `http://www.ece.umd.edu/~ulukus/journal/`.

[103] G. DeCandia, D. Hastorun, et al. Dynamo: Amazon's highly available key-value store. *ACM SIGOPS Oper. Syst. Rev.*, 41(6):205–220, 2007.

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

[105] P. Mitra and S. Ulukus. ASUMAN: Age sense updating multiple access in networks. In *Allerton Conference*, September 2022.

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

[107] R. D. Yates and S. K. Kaul. The age of information: Real-time status updating by multiple sources. *IEEE Transactions on Information Theory*, 65(3):1807–1827, March 2019.

[108] S. M. Samuels. A characterization of the Poisson process. *Journal of Applied Probability*, 11(1):72–85, 1974.

[109] R. D. Yates. Age of information in a network of preemptive servers. In *IEEE Infocom*, April 2018.

[110] C. Kam, J. P. Molnar, and S. Kompella. Age of information for queues in tandem. In *IEEE MILCOM*, October 2018.

[111] R. D. Yates. Status updates through networks of parallel servers. In *IEEE ISIT*, June 2018.

[112] B. Buyukates, M. Bastopcu, and S. Ulukus. Version age of information in clustered gossip networks. *IEEE Journal on Selected Areas in Information Theory*, 3(1):85–97, March 2022.

[113] R. Talak, S. Karaman, and E. Modiano. Minimizing age-of-information in multi-hop wireless networks. In *Allerton Conference*, October 2017.

[114] J. Selen, Y. Nazarathy, L. L. H. Andrew, and H. L. Vu. The age of information in gossip networks. In *Analytical and Stochastic Modeling Techniques and Applications*, pages 364–379. Springer Berlin, Heidelberg, 2013.

[115] R. Serfozo. *Basics of Applied Stochastic Processes*. Springer Berlin, Heidelberg, 2009.

[116] R. Gallager. *Discrete Stochastic Processes*. OpenCourseWare: Massachusetts Institute of Technology, 2011.

[117] A. Maatouk, M. Assaad, and A. Ephremides. Analysis of an age-dependent stochastic hybrid system. In *ISIT*, June 2022.

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

[119] P. Kaswan, P. Mitra, A. Srivastava, and S. Ulukus. Age of information in gossip networks: A friendly introduction and literature survey. 2023. Available at arXiv:2312.16163.

[120] D. J. Daley and M. Miyazawa. A martingale view of Blackwell's renewal theorem and its extensions to a general counting process. *Journal of Applied Probability*, 56(2):602–623, July 2019.

[121] J. Medhi. *Stochastic Processes*. New Age International, 1994.

[122] M. Zhang, A. Arafa, J. Huang, and H. V. Poor. Pricing fresh data. *IEEE J. Selected Areas in Communications*, 39(5):1211–1225, May 2021.

[123] S. Ferretti. Publish-subscribe systems via gossip: a study based on complex networks. In *ACM SIMPLEX*, June 2012.

[124] G. Egorov and K. Sonin. Persuasion on networks. *Technical report, National Bureau of Economic Research*, July 2020.

[125] P. Kaswan, M. Bastopcu, S. Ulukus, S. R. Etesami, and T. Başar. How to make money from fresh data: Subscription strategies in age-based systems. 2024. Available at arXiv:2404.16009.

[126] A. Garnaev, W. Zhang, J. Zhong, and R. D. Yates. Maintaining information freshness under jamming. In *IEEE Infocom*, May 2019.

[127] G. D. Nguyen, S. Kompella, C. Kam, J. E. Wieselthier, and A. Ephremides. Impact of hostile interference on information freshness: A game approach. In *IEEE WiOpt*, May 2017.

[128] Y. Xiao and Y. Sun. A dynamic jamming game for real-time status updates. In *IEEE Infocom*, April 2018.

[129] S. Banerjee, R. Bhattacharjee, and A. Sinha. Fundamental limits of age-of-information in stationary and non-stationary environments. In *IEEE ISIT*, June 2020.

[130] R. Bhattacharjee and A. Sinha. Competitive algorithms for minimizing the maximum age-of-information. In *ACM Perf. Eval.*, September 2020.

[131] S. Banerjee and S. Ulukus. Age of information in the presence of an adversary. In *IEEE Infocom*, May 2022.

[132] S. Banerjee and S. Ulukus. Game theoretic analysis of an adversarial status updating system. In *IEEE ISIT*, June 2022.

[133] S. Banerjee and S. Ulukus. The freshness game: Timely communications in the presence of an adversary. *Available at arXiv:2302.14024*.

[134] S. Banerjee, S. Ulukus, and A. Ephremides. Age of information of a power constrained scheduler in the presence of a power constrained adversary. In *IEEE Infocom*, May 2023.

[135] P. Kaswan and S. Ulukus. How robust are timely gossip networks to jamming attacks? *IEEE Journal on Selected Areas in Information Theory*, 4:820–832, 2023.

[136] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge university press, 1995.

[137] S. Albers. *Competitive Online Algorithms*. Citeseer, 1996.

[138] P. Kaswan, M. Bastopcu, and S. Ulukus. Timely cache updating in parallel multi-relay networks. *IEEE Transactions on Wireless Communications*, 23(1):2–15, January 2024.

[139] P. Kaswan and S. Ulukus. Timestomping vulnerability of age-sensitive gossip networks. *IEEE Transactions on Communications*, 72(7):4193–4205, July 2024.

[140] P. Kaswan and S. Ulukus. Information mutation and spread of misinformation in timely gossip networks. In *Globecom*, December 2023.

[141] P. Kaswan and S. Ulukus. Misinformation spread in gossip networks: The influence of transmission mutations. Available at `http://www.ece.umd.edu/~ulukus/journal/`.

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

[143] P. Kaswan and S. Ulukus. Choosing outdated information to achieve reliability in age-based gossiping. In *IEEE ICC*, June 2024.

[144] P. Kaswan and S. Ulukus. Age of information with non-Poisson updates in cache-updating networks. In *IEEE ISIT*, June 2023.

[145] P. Kaswan and S. Ulukus. Timely tracking of a remote dynamic source via multi-hop renewal updates. In *IEEE CDC*, December 2024.

[146] P. Kaswan and S. Ulukus. Timeliness in cache-aided networks with non-poisson updating. Available at arXiv:2304.01989.

[147] P. Kaswan, M. Bastopcu, S. Ulukus, S. R. Etesami, and T. Başar. Optimizing profitability in timely gossip networks. In *IEEE SPAWC*, September 2024.

[148] P. Kaswan, M. Bastopcu, S. Ulukus, S. R. Etesami, and T. Başar. Strategic profit generation in age-based systems. Available at `http://www.ece.umd.edu/~ulukus/journal/`.

[149] S. Banerjee, P. Kaswan, and S. Ulukus. Age-based cache updating under timestomping. In *IEEE WiOpt*, August 2023.

[150] P. Kaswan and S. Ulukus. Timeliness: A new design metric and a new attack surface. 2023. Available at arXiv:2312.17220.