

AGE OF INFORMATION BASED FLOW CONTROL IN COMMUNICATION  
NETWORKS

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SAJJAD BAGHAEE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF DOCTOR OF PHILOSOPHY  
IN  
ELECTRICAL AND ELECTRONICS ENGINEERING

JULY 2023



Approval of the thesis:

**AGE OF INFORMATION BASED FLOW CONTROL IN  
COMMUNICATION NETWORKS**

submitted by **SAJJAD BAGHAE** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar  
Dean, Graduate School of **Natural and Applied Sciences** \_\_\_\_\_

Prof. Dr. İlkey Ulusoy  
Head of Department, **Electrical and Electronics Engineering** \_\_\_\_\_

Prof. Dr. Elif Uysal  
Supervisor, **Electrical and Electronics Engineering, METU** \_\_\_\_\_

**Examining Committee Members:**

Prof. Dr. İlkey Ulusoy  
Electrical and Electronics Engineering, METU \_\_\_\_\_

Prof. Dr. Elif Uysal  
Electrical and Electronics, METU \_\_\_\_\_

Prof. Dr. Tolga Girici  
Electrical and Electronics Engineering, TOBB ETU \_\_\_\_\_

Assoc. Prof. Dr. Hüseyin Uğur Yıldız  
Electrical and Electronics Engineering, TEDU \_\_\_\_\_

Assist. Prof. Dr. Elif Tuğçe Ceran Arslan  
Electrical and Electronics Engineering, METU \_\_\_\_\_

Date: 25.07.2023

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Surname: Sajjad Baghaee

Signature :

## **ABSTRACT**

### **AGE OF INFORMATION BASED FLOW CONTROL IN COMMUNICATION NETWORKS**

Baghaee, Sajjad

Ph.D., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Elif Uysal

July 2023, 169 pages

This thesis begins with a study of AoI through measurements on bandwidth-limited links in emulated and real-world applications, where its results reveal similarities between the empirical measurements with the theoretical results. Furthermore, this study analyzes AoI variations over TCP/IP links that use WiFi, LTE, 3G, 2G, and Ethernet. It follows by addressing the issue of synchronization errors in AoI measurements and a solution for it. To achieve a deeper understanding of AoI behavior in practical scenarios, this study extends to measure AoI related to IoT nodes over TCP, UDP, and Web Sockets protocols. Also, we present a deep reinforcement learning-based approach that can minimize the AoI without any prior knowledge about the network attributes. These observations provide the basis for the design of Age-Aware Application Layer Forward Error Correction (A<sup>3</sup>L-FEC), a role-based data flow controller in the application layer of OSI with the aim of improving data freshness by preventing age violation. A<sup>3</sup>L-FEC ensures the reliable transmission of fresh data by leveraging forward error correction and user datagram protocol. The protocol proposes two variations, A<sup>3</sup>L-FEC-FSFB and A<sup>3</sup>L-FEC-VSVB for different types of applications, where each one operates with different sample generation, transmission

intervals, and adaptive blocklength selection. Evaluations of A<sup>3</sup>L-FEC under different network conditions and application requirements have been performed and its age violation control performance has been compared with TCP-BBR in the mininet emulator environment and ACP+ in MATLAB. The outcomes reveal the effectiveness of A<sup>3</sup>L-FEC in minimizing age violations and improving data freshness compared to TCP-BBR and ACP+.

Keywords: Age of information, Age violation, Communication protocol, Age-aware congestion control, Age-aware application layer forward error correction

## ÖZ

### HABERLEŞME AĞLARINDA BILGI YAŞINA DAYALI AKIŞ KONTROLÜ

Baghaee, Sajjad

Doktora, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Elif Uysal

Temmuz 2023 , 169 sayfa

Bu tez sınırlı bant genişliğine sahip bağlantılar üzerinde çalışan benzetilmiş ve gerçek dünya uygulamalarındaki AoI'nin ölçümler yoluyla incelenmesiyle başlar; burada elde edilen sonuçlar, deneysel ölçümler ile teorik sonuçlar arasında benzerlikleri ortaya koymaktadır. Ayrıca, bu çalışma WiFi, LTE, 3G, 2G ve Ethernet gibi farklı ağ teknolojileri tarafından sağlanan TCP/IP bağlantıları üzerinde AoI değişikliklerini analiz etmektedir. Bunun ardından AoI ölçümlerinde senkronizasyon hatalarının ele alınması ve bunun için bir çözüm sunulması gelir. Bu çalışmada pratik senaryolarda AoI davranışını daha derinlemesine anlamak için IoT düğümleri üzerinde TCP, UDP ve Web Soketleri protokollerine göre AoI ölçümleri de yapılmıştır. Ayrıca, ağ öznitelikleri hakkında önceden herhangi bir bilgi sahibi olmadan AoI'yi minimize edebilen derin öğrenmeye dayalı bir yaklaşım sunulmaktadır. Bu gözlemler, OSI'nin uygulama katmanında Age-Aware Application Layer Forward Error Correction (A<sup>3</sup>L-FEC) adlı, yaş ihlalini önleyerek veri tazeliğini artırmayı amaçlayan rol tabanlı bir veri akış denetleyicisinin tasarımının temelini oluşturmaktadır. A<sup>3</sup>L-FEC, ileri hata düzeltme ve kullanıcı datagram protokolünü kullanarak taze verinin güvenilir iletimini sağlar. Protokol, farklı uygulama türleri için A<sup>3</sup>L-FEC-FSFB ve A<sup>3</sup>L-FEC-VSVB olmak üzere

iki varyant önerir; her biri farklı örnek oluşturma, iletim aralıkları ve uyarlanabilir blok uzunluğu seçimi ile çalışır. A<sup>3</sup>L-FEC'nin farklı ağ koşulları ve uygulama gereksinimleri altında değerlendirmeleri gerçekleştirilmiş ve yaş ihlali kontrol performansı mininet emülatör ortamında TCP-BBR ve MATLAB'da ACP+ ile karşılaştırılmıştır. Sonuçlar, A<sup>3</sup>L-FEC'nin yaş ihlallerini en aza indirmede ve veri tazeliğini artırmada TCP-BBR ve ACP+'ya göre etkili olduğunu göstermektedir.

**Anahtar Kelimeler:** Bilgi yaşı, Yaş ihlali, İletişim protokolü, Yaşa duyarlı tıkanıklık kontrolü, Yaşa duyarlı uygulama katmanı ileri hata düzeltme



To my family

## ACKNOWLEDGMENTS

I would like to extend my heartfelt appreciation and gratitude to all who have contributed to the completion of my Ph.D. thesis.

First and foremost, I am deeply indebted to my supervisor, Prof. Dr. Elif Uysal, for her exceptional mentorship, and valuable insights. Her dedication to excellence and commitment to pushing me beyond my limits have been instrumental in shaping the quality of my research.

I would like to extend my sincere thanks to Dr. Baran Tan Bacınoğlu for his invaluable contribution in suggesting the main idea of A<sup>3</sup>L-FEC protocol.

To my friends, Dr. Mahdi Shakiba Herfeh, Hakan Saç, Hasan Burhan Beytur, Egemen Sert, Canberk Sönmez, and Tahir Kerem Oğuz, I am grateful for their camaraderie, shared experiences, and invaluable guidance.

Special thanks go to the General Manager of JeoIT Ltd., Mr. Yavuz Eren, and all company staff especially, Koray Aksoy, and Emre Akkaya, for their support during my graduate study. Their understanding and flexibility allowed me to balance work commitments with my academic responsibilities effectively.

Furthermore, I express my gratitude to TÜBİTAK for their financial support under 117E215, which provided the necessary resources for conducting the research.

Lastly, I owe an immeasurable debt of gratitude to my family. To my loving mother, Behjat, my dear father, Ali, and my brother, Behzad, I am eternally thankful for their unwavering belief in me and their constant encouragement throughout this journey. Their love and support have been the bedrock of my accomplishments.

Above all, my heartfelt thanks to my dear wife, Torkan. Her constant love and support fueled my academic journey. Her sacrifices and unwavering faith inspired me to overcome challenges. I'm eternally grateful

In conclusion, I'm grateful for the remarkable support that significantly shaped this thesis. My heartfelt thanks to all who contributed. Carrying valuable lessons, I step into the future.

## TABLE OF CONTENTS

ABSTRACT . . . . .	v
ÖZ . . . . .	vii
ACKNOWLEDGMENTS . . . . .	x
TABLE OF CONTENTS . . . . .	xii
LIST OF TABLES . . . . .	xvi
LIST OF FIGURES . . . . .	xviii
LIST OF ABBREVIATIONS . . . . .	xxii
CHAPTERS	
1 INTRODUCTION . . . . .	1
1.1 Motivation and Problem Definition . . . . .	4
1.2 Definitions: . . . . .	5
1.2.1 Age of Information . . . . .	5
1.2.2 Average Age . . . . .	7
1.2.2.1 Calculating the Average Age with $Q_i$ Trapezoids . . . . .	8
1.2.2.2 Calculating the Average Age with $H_i$ Trapezoids . . . . .	10
1.2.3 Average Peak Age . . . . .	11
1.3 Contributions . . . . .	12
1.4 The Outline of the Dissertation . . . . .	13

2	AGE OF INFORMATION IN REAL LIFE . . . . .	15
2.1	Introduction . . . . .	15
2.2	Age of Information: Definition, Measurement and Behaviour in Queuing Systems . . . . .	18
2.3	Measuring AoI within a Basic Physical Network . . . . .	21
2.4	Clock Bias and Synchronization Process for Age Measurement in Networks . . . . .	24
2.4.1	Measurement of Age Under Synchronization . . . . .	24
2.4.2	Effect of Poor Synchronization on Age-dependent Metrics . . . . .	26
2.4.3	RTT-Based Asynchronous AoI Estimation Method . . . . .	29
2.5	The Impact of the Access Network on AoI within TCP/IP Connections . . . . .	29
2.5.1	Measuring AoI on an Emulation Testbed . . . . .	30
2.5.2	AoI Measurements on a Real-world Testbed . . . . .	31
2.6	Comparing AoI Between UDP and TCP-IP . . . . .	34
2.6.1	Comparing UDP and TCP-IP in a Multi-hop Network Test Environment . . . . .	36
2.6.2	Comparison of UDP and TCP-IP on an IoT Testbed . . . . .	38
2.7	Utilizing Machine Learning Techniques for Enhancing AoI Optimization in Real-life Networks . . . . .	40
2.8	Implementation of Age Control Strategies at the Application Layer Over UDP . . . . .	44
2.9	Practical Testing of Wireless Access Schedulers with Age-aware Characteristics at the Link-Layer . . . . .	45
2.10	Conclusions . . . . .	47
3	A <sup>3</sup> L-FEC PROTOCOL . . . . .	49
3.1	Introduction . . . . .	49

3.2	Related Work . . . . .	50
3.3	A <sup>3</sup> L-FEC Protocol with Fixed Sampling Rate and Fixed Block-length . . . . .	57
3.3.1	Congestion Control Algorithm for the A <sup>3</sup> L-FEC-FSFB Protocol . . . . .	63
3.4	A <sup>3</sup> L-FEC Protocol with Variable Sampling Rate and Variable Block-length . . . . .	71
3.4.1	Congestion Control for the A <sup>3</sup> L-FEC-VSVB Protocol . . . . .	76
3.5	A <sup>3</sup> L-FEC and Theoretical Limits . . . . .	93
3.6	A <sup>3</sup> L-FEC and Multi-Transmitter Multi-Receiver . . . . .	95
3.7	Discussion . . . . .	99
3.8	Conclusion . . . . .	102
4	SIMULATION AND EMULATION RESULTS . . . . .	105
4.1	Simulation-based Evaluation of AoI Under Different TCP Congestion Control Algorithms on ns-3 Network Simulator . . . . .	105
4.1.1	Simulation Testbed Setup on ns-3 for AoI Evaluation of TCP Congestion Control Algorithms . . . . .	107
4.1.2	AoI Evaluation results of TCP Congestion Control Algorithms on ns-3 . . . . .	109
4.2	Simulation-based Evaluation of A <sup>3</sup> L-FEC-FSFB and ACP+ Algorithms on MATLAB . . . . .	113
4.2.1	Overview of the Age Control Protocol (ACP) and ACP+ . . . . .	113
4.2.2	Simulation Testbed Setup on MATLAB for Age Violation Evaluation of A <sup>3</sup> L-FEC-FSFB and ACP+ . . . . .	118
4.2.2.1	Age Violation Evaluation Results of A <sup>3</sup> L-FEC-FSFB and ACP+ Algorithms on MATLAB . . . . .	119
4.2.2.2	Results of Effect of Coding Rate on the Performance of A <sup>3</sup> L-FEC-FSFB on MATLAB . . . . .	126
4.3	Emulation-based Comparison of A <sup>3</sup> L-FEC-VSVB and TCP-BBR congestion control on Mininet-WiFi . . . . .	131

4.3.1	The optimal operating point of TCP . . . . .	132
4.3.2	Overview of the TCP BBR Congestion Control . . . . .	133
4.3.3	Emulation Testbed Setup on Mininet-WiFi for Age Violation Evaluation of A <sup>3</sup> L-FEC-VSVB and TCP-BBR congestion control . . . . .	136
4.3.4	Age Violation Evaluation Results of A <sup>3</sup> L-FEC-VSVB and TCP- BBR Congestion Control on Mininet-WiFi . . . . .	138
4.3.5	AoI Evaluation Results of A <sup>3</sup> L-FEC-VSVB and TCP-BBR Congestion Control on Mininet-WiFi . . . . .	141
4.3.6	Packer Delay Evaluation Results of A <sup>3</sup> L-FEC-VSVB and TCP- BBR Congestion Control on Mininet-WiFi . . . . .	143
4.3.7	Transmission Rate Evaluation Results of A <sup>3</sup> L-FEC-VSVB and TCP-BBR Congestion Control on Mininet-WiFi . . . . .	146
4.3.8	Effect of Sample Chunk Number on the Performance of A <sup>3</sup> L- FEC-VSVB on Mininet-WiFi . . . . .	148
REFERENCES . . . . .		153
CURRICULUM VITAE . . . . .		167

## LIST OF TABLES

### TABLES

Table 4.1	Well-known Network Simulators Overview. . . . .	107
Table 4.2	Decision algorithm of ACP . . . . .	115
Table 4.3	Table of actions and corresponding $b_{k+1}^*$ values . . . . .	116
Table 4.4	MATLAB Simulation Parameters for Age Violation Evaluation of A <sup>3</sup> L-FEC-FSFB and ACP+ . . . . .	118
Table 4.5	Performance comparison of A <sup>3</sup> L-FEC-FSFB and ACP+ algorithms under packet loss probabilities $P_{\text{inLoss}} = 0$ and $P_{\text{outLoss}} = 0$ for the age violation threshold $AVT = 5$ . The results are based on the average per- formance over 50 realizations. . . . .	120
Table 4.6	Performance comparison of A <sup>3</sup> L-FEC-FSFB and ACP+ algorithms under packet loss probabilities $P_{\text{inLoss}} = 0.1$ and $P_{\text{outLoss}} = 0$ for the age violation threshold $AVT = 5$ . The results are based on the average per- formance over 50 realizations. . . . .	120
Table 4.7	Performance comparison of A <sup>3</sup> L-FEC-FSFB and ACP+ algorithms under packet loss probabilities $P_{\text{inLoss}} = 0.2$ and $P_{\text{outLoss}} = 0$ for the age violation threshold $AVT = 5$ . The results are based on the average per- formance over 50 realizations. . . . .	120
Table 4.8	Performance comparison of A <sup>3</sup> L-FEC-FSFB and ACP+ algorithms under packet loss probabilities $P_{\text{inLoss}} = 0$ and $P_{\text{outLoss}} = 0.1$ for the age violation threshold $AVT = 5$ . The results are based on the average per- formance over 50 realizations. . . . .	121



Table 4.9 Performance comparison of A <sup>3</sup> L-FEC-FSFB and ACP+ algorithms under packet loss probabilities $P_{\text{inLoss}} = 0.1$ and $P_{\text{outLoss}} = 0.1$ for the age violation threshold $AVT = 5$ . The results are based on the average performance over 50 realizations. . . . .	121
Table 4.10 Performance comparison of A <sup>3</sup> L-FEC-FSFB and ACP+ algorithms under packet loss probabilities $P_{\text{inLoss}} = 0.2$ and $P_{\text{outLoss}} = 0.1$ for the age violation threshold $AVT = 5$ . The results are based on the average performance over 50 realizations. . . . .	121
Table 4.11 Performance comparison of A <sup>3</sup> L-FEC-FSFB and ACP+ algorithms under packet loss probabilities $P_{\text{inLoss}} = 0$ and $P_{\text{outLoss}} = 0.2$ for the age violation threshold $AVT = 5$ . The results are based on the average per- formance over 50 realizations. . . . .	122
Table 4.12 Performance comparison of A <sup>3</sup> L-FEC-FSFB and ACP+ algorithms under packet loss probabilities $P_{\text{inLoss}} = 0.1$ and $P_{\text{outLoss}} = 0.2$ for the age violation threshold $AVT = 5$ . The results are based on the average performance over 50 realizations. . . . .	122
Table 4.13 Performance comparison of A <sup>3</sup> L-FEC-FSFB and ACP+ algorithms under packet loss probabilities $P_{\text{inLoss}} = 0.2$ and $P_{\text{outLoss}} = 0.2$ for the age violation threshold $AVT = 5$ . The results are based on the average performance over 50 realizations. . . . .	122
Table 4.14 MATLAB Simulation Parameters for Evaluating the Effect of Cod- ing Rate on the Performance of A <sup>3</sup> L-FEC-FSFB . . . . .	127

## LIST OF FIGURES

### FIGURES

Figure 1.1	A simplified communication setup. . . . .	6
Figure 1.2	Sample path of the age process $\Delta(t)$ with $Q_i$ trapezoids. . . . .	8
Figure 1.3	Sample path of the age process $\Delta(t)$ with $H_i$ trapezoids, $\Delta_0 > 0$ . . . . .	10
Figure 2.1	Avg. Delay and AoI attained by an M/M/1 queuing system with Service Rate $\mu = 1$ . . . . .	19
Figure 2.2	A depiction of the practical experimental testbed. . . . .	22
Figure 2.3	AoI vs sampling rate, on the testbed running UDP protocol. . . . .	23
Figure 2.4	Networking Topology in the CORE Emulator. . . . .	30
Figure 2.5	AoI behavior results . . . . .	31
Figure 2.6	A schematic of the physical testbed. . . . .	32
Figure 2.7	AoI measurements of connections over different access networks. . . . .	33
Figure 2.8	AoI measurements on Mobile Networks. . . . .	33
Figure 2.9	Age, delay and packet loss versus rate for UDP over the Internet. . . . .	36
Figure 2.10	Age and packetwise delay for TCP over the Internet. . . . .	38
Figure 2.11	Packet Loss and Average Age TCP and UDP Measured over a Local Wi-Fi Network, using IoT Devices. . . . .	39
Figure 2.12	Results of Q-value and status age vs. iteration. . . . .	43

Figure 3.1	System model of a status update system over an error-prone link with a transmission rate feedback for the single-transmitter case. . . . .	57
Figure 3.2	An illustration of the chunks selected for transmission with respect to their indices and time in A <sup>3</sup> L-FEC-FSFB protocol. . . . .	58
Figure 3.3	An illustration of the chunks that are successfully received with respect to their indices and time in A <sup>3</sup> L-FEC-FSFB protocol. . . . .	59
Figure 3.4	The signaling scheme for the congestion control mechanism of the A <sup>3</sup> L-FEC protocol when the server acts as a transmitter and the client acts as a receiver. . . . .	61
Figure 3.5	Congestion control scheme for the A <sup>3</sup> L-FEC-FSFB protocol. . .	62
Figure 3.6	Update Algorithm for the expected transmission rate in the A <sup>3</sup> L-FEC-FSFB protocol. . . . .	64
Figure 3.7	A illustration of the chunks that are selected for transmission with respect to their indices and time in A <sup>3</sup> L-FEC-VSVB protocol. . . .	72
Figure 3.8	An illustration of the chunks that are successfully received with respect to their indices and time in A <sup>3</sup> L-FEC-VSVB protocol. . . . .	72
Figure 3.9	Congestion control scheme for the A <sup>3</sup> L-FEC-VSVB protocol. . .	75
Figure 3.10	Update algorithm for the expected packet transmission rate in the A <sup>3</sup> L-FEC-VSVB protocol. . . . .	77
Figure 3.11	System model of a status update system over an error-prone point-to-point link in the presence of transmission rate feedback from the receiver for the multi-transmitter multi-receiver case. . . . .	96
Figure 3.12	Rate allocation in the multi-transmitter multi-receiver scenario. .	97
Figure 4.1	End-to-end network topology used for simulation. . . . .	108

Figure 4.2	Average age, average delay, average throughput, average cwnd, average loss, and average queue length for well-known TCP variants under an end-to-end topology which is shown in Figure 4.1. . . . .	112
Figure 4.3	Simulation results for A <sup>3</sup> L-FEC-FSFB protocol with an age violation threshold of 5 ( $AVT = 5$ ) when $P_{inLoss} = P_{outLoss} = 0.2$ . The first figure illustrates the variation in transmission rates, the second shows the average age violation probability, the third displays the average delay of received packets (in chunks), and the fourth depicts the number of packets (in chunks) present in the queue at the end of each monitoring interval. . . . .	123
Figure 4.4	Simulation results for ACP+ protocol with an age violation threshold of 5 ( $AVT = 5$ ) when $P_{inLoss} = P_{outLoss} = 0.2$ . The first figure illustrates the variation in transmission rates, the second shows the average age violation probability, the third displays the average delay of received packets (in samples), and the fourth depicts the number of packets (in samples) present in the queue at the end of each monitoring interval. . . . .	124
Figure 4.5	Simulation results for age violation with threshold $AVT = 2$ under different coding rates when $P_{inLoss} = 0.2$ and $P_{outLoss} = 0.2$ . . .	128
Figure 4.6	Simulation results for age violation with threshold $AVT = 5$ under different coding rates when $P_{inLoss} = 0.2$ and $P_{outLoss} = 0.2$ . . .	128
Figure 4.7	Simulation results for age violation with threshold $AVT = 2$ under different coding rates when $P_{inLoss} = 0.2$ and $P_{outLoss} = 0.1$ . . .	129
Figure 4.8	Simulation results for age violation with threshold $AVT = 5$ under different coding rates when $P_{inLoss} = 0.2$ and $P_{outLoss} = 0.1$ . . .	129
Figure 4.9	Simulation results for age violation with threshold $AVT = 2$ under different coding rates when $P_{inLoss} = 0.1$ and $P_{outLoss} = 0.2$ . . .	130
Figure 4.10	Simulation results for age violation with threshold $AVT = 5$ under different coding rates when $P_{inLoss} = 0.1$ and $P_{outLoss} = 0.2$ . . .	130

Figure 4.11	Emulator testbed. . . . .	131
Figure 4.12	bbbr . . . . .	133
Figure 4.13	End-to-end network topology used for emulation. . . . .	137
Figure 4.14	Age violation probability under AVT = 600 msec. . . . .	139
Figure 4.15	Age violation probability under AVT = 300 msec. . . . .	140
Figure 4.16	Age violation probability under AVT = 150 msec. . . . .	140
Figure 4.17	AoI under AVT = 600 msec. . . . .	142
Figure 4.18	AoI under AVT = 300 msec. . . . .	142
Figure 4.19	AoI under AVT = 150 msec. . . . .	143
Figure 4.20	Average packet delay under AVT = 600 msec. . . . .	144
Figure 4.21	Average packet delay under AVT = 300 msec. . . . .	145
Figure 4.22	Average packet delay under AVT = 150 msec. . . . .	145
Figure 4.23	Average transmission rate under AVT = 600 msec. . . . .	147
Figure 4.24	Average transmission rate under AVT = 300 msec. . . . .	147
Figure 4.25	Average transmission rate under AVT = 150 msec. . . . .	148
Figure 4.26	Effect of Sample Chunk Number and Age Violation Threshold on the Performance of A <sup>3</sup> L-FEC-VSVB . . . . .	150

## LIST OF ABBREVIATIONS

A <sup>3</sup> L-FEC	Age Aware Application Layer Forward Error Correction
A <sup>3</sup> L-FEC-FSFB	A <sup>3</sup> L-FEC - Fixed Sampling Rate Fixed Block Length
A <sup>3</sup> L-FEC-VSVB	A <sup>3</sup> L-FEC - Variable Sampling Rate Variable Block Length
ACK	Acknowledgment
ACP	Age Control Protocol
AIMD	Additive Increase Multiplicative Decrease
AoI	Age of Information
AP	Access Point
AV	Age Violation
AVT	Age Violation Threshold
BBR	Bottleneck Bandwidth and Round-Trip Propagation Time
BDP	Bandwidth Delay Product
BS	Base Station
DEC	additive decrease
DQN	Deep Q Network
EB	Exabytes
EF	Empty Flag
ERP	Expected Received Packet
FCFS	First-Come-First-Serve
FIFO	First-In-First-Out
HARQ	Hybrid Automatic Repeat Request
INC	Additive Increase
IoT	Internet of Things

KPI	Key Performance Indicator
LCFS	Last-Come-First-Serve
LTE	Long Term Evolution
LWIP	Lightweight IP-Stack
MDEC	Multiplicative Decrease
MDP	Markov Decision Processes
MDS	Maximum Distance Separable
MI	Monitoring Interval
mMTC	machine-to-machine communications
NTP	Network Time Protocol
PAoI	Peak Age of Information
PDR	Packet Delivery Ratio
PEC	Packet Erasure Channel
RL	Reinforcement Learning
RTC	Real Time Clock
RTP	Real-time Transport Protocol
RTT	Round Trip Time
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URLLC	Ultra Reliable Low Latency Communications
V2X	Vehicle To Everything
WSN	Wireless Sensor Networks
xURLLC	extreme URLLC
ZB	Zettabyte





## CHAPTER 1

### INTRODUCTION

In the last decade, there has been a rapid advancement in technology, which has led to an increased demand for fast and reliable communication networks. This growing demand has not only led to a surge in global data traffic but is also expected to shape the direction of traffic growth in the forthcoming years. According to the International Telecommunication Union (ITU) report based on statistics of 2015 which is presented in [1], the global mobile data traffic is expected to reach 607 exabytes (EB) per month by 2025 and 5016 EB per month by 2030. While the International Data Corporation (IDC)'s report denotes that the amount of data generated by Internet of Things (IoT) devices was 18.3 zettabytes (ZB) in 2019, which is expected to reach 73.1 zettabytes by 2025, [2].

This exponential growth in data volume indicates a corresponding increase in data flow in communication networks, which is mainly driven by IoT devices located in different geospatial locations. Consequently, the network faces the formidable challenge of identifying and implementing suitable data transfer solutions that can effectively address the demanding needs of real-time monitoring applications. These requirements include critical factors such as low latency, maximum throughput, reliability, and data freshness.

Real-time monitoring applications, such as vehicle-to-everything (V2X) communication, environment monitoring, patient health monitoring, smart home systems, smart cities, and so on heavily rely on the transmission of status update messages. In V2X communication, the availability of timely data related to road conditions, traffic congestion, and potential hazards is essential for ensuring safe and efficient transportation. Similarly, environment monitoring applications require real-time data on air

quality, temperature, humidity, and other environmental factors to facilitate effective decision-making and prompt responses to condition changes. In the context of patient health monitoring, continuous tracking of patient vital signs with up-to-date data enables healthcare experts to quickly notice any abnormalities or emergencies, allowing for timely medical interventions. Likewise, in smart home and smart city applications, real-time status updates play an important role in different scenarios such as energy management, security surveillance, and intelligent automation, where, by having access to fresh data, these applications can make informed decisions to handle the required demands.

It is obvious that the common and fundamental objective of these applications is to consistently deliver fresh and up-to-date information to the endpoint; because the stale information loses its merit and results in malfunctions and failures in system decisions; consequently, the importance of up-to-date information for real-time applications cannot be ignored.

To address the fresh data delivery demand, the Age of Information (AoI) metric was introduced as a Key Performance Indicator (KPI) for time-critical status update systems and applications by [3, 4]. AoI is defined as the time elapsed since the generation time of the last successfully received packet. This metric plays a critical role to measure the freshness and timeliness of information in communication systems. In other words, it presents worthwhile insights into the freshness of received packets, emphasizing how up-to-date it is. Unlike the delay or latency, which concentrates on the time it takes for a packet to journey from the source node to the destination node, AoI precisely depicts the age of the most recently received data by measuring the time difference between the generation of the latest update and its reception.

To understand this difference clearly, consider a First Come First Served (FCFS) queue in a status update system. In a scenario when the update frequency of the source is low, the queuing delay experienced by packets on the bottleneck of the network is minimal. Consequently, it causes a smaller end-to-end delay, but from the perspective of AoI, this condition is not favorable. Because low update frequency leads to an increase in the age at the receiver as the information becomes stale before a new update arrives. In another scenario consider a high-frequency update system.

In this type of update system, the large injection of updates into the network causes a long queue on the bottleneck of the network; in turn, it results in large delays and higher age due to increased waiting time in the queue. However, if the balance between the update frequency and the queuing delays experienced by packets in the network is preserved, the system will have an optimal AoI. Therefore, based on these clarifications it can be concluded that AoI and delay represent distinct reactions to different network conditions.

In recent years, AoI has achieved extra importance in the context of fifth-generation (5G) technologies, [5, 6, 7, 8] such as massive machine-to-machine communications (mMTC), [9, 10] and Ultra Reliable Low Latency Communications (URLLC), [11]. Moreover, with the emergence of extreme URLLC (xURLLC) in sixth-generation (6G) communications, AoI-related metrics are expected to gain more prominence, [12, 13]. It is important to note that, while the average age is commonly used as a KPI in status update systems, it solely does not provide a comprehensive measure of information freshness. To construct a robust infrastructure for analysis, it is crucial to consider other age-related metrics alongside AoI. By applying AoI-based metrics, it is possible to gain deeper insights into the system's performance and make informed decisions to optimize information freshness. For example, in some critical applications such as Ultra-Reliable Low-Latency Communications (URLLC), minimizing the age violation probability is paramount. The age violation probability is known as the probability that the instantaneous age exceeds a predefined threshold value. By keeping this probability low, the system can guarantee that the data received by the receiver stays sufficiently fresh and up-to-date.

This dissertation focuses on addressing data freshness by minimizing the age violation probability at the receiver side. To achieve this goal, an advanced age-aware data flow control algorithm for communication systems is formed. This algorithm dynamically adapts the transmission rate, selects optimal block lengths, and optimizes the sampling rate based on feedback from the receiver. Our simulation and emulation outcomes reveal that implementing this data flow control algorithm enables real-time applications to make more informed and timely decisions. Consequently, this leads to significantly enhanced data freshness, improving overall system performance.

## 1.1 Motivation and Problem Definition

As explained previously, in today's interconnected world, the significance of real-time communication systems lies in a wide variety of applications. In the domain of real-time applications, the concept of fresh data has emerged as a fundamental criterion that requires precise consideration. However, ensuring delivering fresh data to the destination in these communication systems is not an easy task, because several factors can prevent the timely delivery of fresh data. These factors could be transmission errors that can corrupt data packets and lead to delays and the need for re-transmissions; or the existence of network congestion that can cause additional delay and thus jeopardize the rapid arrival of critical information.

In this study, the challenge is to find a solution that optimizes data freshness while reducing the impact of transmission errors and communication network limitations. It is obvious that the traditional algorithms using automatic repeat request (ARQ) and forward error correction (FEC) mainly focus on maximizing throughput and minimizing delay, but do not explicitly consider data freshness. Furthermore, in traditional networks that send long packets, the distortion caused by propagation channels and thermal noise is usually reduced on average. But for short packet communication systems such as age-aware status update system, the distortion effects caused by diffusion channels and thermal noise cannot be effectively mitigated via averaging on the received messages.

The main issue that occurs in short-packet transmission while considering the AoI metric, is the selection of an optimal finite block-length. A larger block-length code that includes more redundancy chunks, effectively reduces the error probability during transmission. But, this benefit comes at the cost of increased transmission duration and higher age value at the receiver side. Conversely, a message with a shorter block length results in a shorter transmission time; but the higher error probability increases the age of valid data at the receiver. Consequently, when choosing the block length, an acceptable balance between error probability and transmission duration should be considered.

Besides dynamically adapting the transmission rate, optimizing the sampling rate for

transmitting data is another challenge of age-aware applications. The purpose of which is the effective production of samples in a way that minimizes the possibility of age violations and thus increases the freshness and timeliness of the received information.

The proposed A<sup>3</sup>L-FEC protocol in this dissertation addresses the mentioned challenge by introducing a novel data flow control for communication systems. This approach aims to optimize data freshness and minimize age violations with the aid of a rule-based algorithm while utilizing forward error correction (FEC) and user datagram protocol (UDP).

In conclusion, the motivation behind the A<sup>3</sup>L-FEC data flow control derives from the crucial need to consistently deliver fresh and up-to-date information in real-time communication systems to the destination. This flow control mechanism aims to increase the performance of real-time applications by handling the challenges associated with data freshness, transmission errors, and varying sampling rates. The successful implementation of A<sup>3</sup>L-FEC on communication systems can have extensive benefits in improving decision-making processes, optimizing system efficiency, and ultimately progressing in various industries and fields.

## 1.2 Definitions:

### 1.2.1 Age of Information

Age of Information (AoI) is known as a metric to quantify the freshness of information that a system observes remotely about one or multiple processes in a timely manner. It measures the time elapsed since the most recent update or observation of certain information.

To define the age metric, let's depict a general communication system with four distinct modules: **Observed Processes**, **Transmitter/Source**, **Network/Channel** and **Receiver/Monitor** as shown in Figure.1.1 which is used in AoI literature:

- **Observed Processes:** There is a set of continuous-time random processes at the

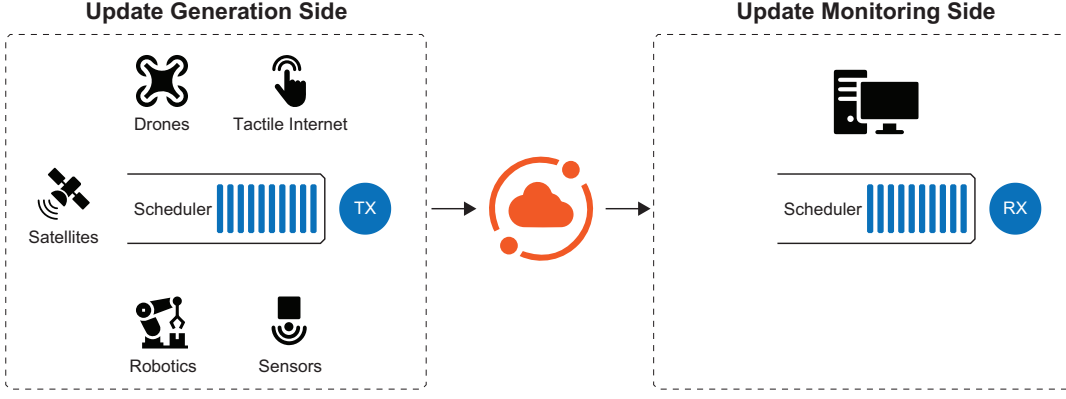


Figure 1.1: A simplified communication setup.

transmitter side which are concerned with the receiver module. It means that the distinct receiver module requires updates of the processes that express the status of the concerned processes at multiple instants in time. Temperature, air quality or carbon dioxide level in a room [14], noise (sound) level in hospitals, the velocity of a drone in the sky, and, location of a car in a parking lot [15, 16] are different examples of the interested processes for IoT based applications.

- **Transmitter/Source:** Transmitter directly observes the process, samples it, and then transmits it to the destination. Transmitted data is named status updates or just updates. Update available at the destination at time  $t$ , where the generation time (i.e. timestamp) of this update is shown by  $U(t)$ .
- **Network/Channel:** In order to make communication between transmitter and receiver a medium or a link is required. This medium can be the cable, the air, or the water. Indeed wire/wireless Channel is the medium that handles the packet transmission from one or multiple transmitters to one or multiple receivers.
- **Receiver/Monitor:** Once the information passes through the communication channel, it is captured at time  $t$  by the receiver/monitoring unit. The receiver follows the evolution of the state of one/several remote processes in a timely manner, such that at any time instant  $t$  the information which the receiver has about the status of a process should be as close as possible to the current status of the process.

After the generation of an update at timestamp  $U(t)$ , till getting complete service on

the monitoring side, that update becomes outdated. Additionally, when the receiver cannot catch any fresh update, the observed process on the receiver side becomes stale. This aging directly impacts the freshness of the receiver/monitor process. Once the receiver gets a fresh update, the age of the process on the monitoring side is updated with the age of the recently arrived update. Age concept can be considered for single or multiple sources and monitors. Without loss of generality, let's define the status age metrics for a single source and a single monitor as follows:

**Definition:** The status age  $\Delta(t)$  is defined as the time that has elapsed since the newest update available at the monitor at time  $t$  while it has the generation time (i.e. timestamp) of  $U(t)$ .

$$\Delta(t) = t - U(t) \quad (1.1)$$

According to the definition of age, it is a continuous-time continuous-valued stochastic process and follows a sawtooth pattern like the sample path presented in Figure 1.2. In this example it is assumed the observation begins at  $t = 0$  and update generation starts at  $t > 0$ , therefore the system has initial age and is shown as  $\Delta(0) = \Delta_0 > 0$ . It is also assumed at  $t = 0$  the queues of the transmitter and receiver are empty. The source generates status updates at  $t_1, t_2, \dots, t_n$ , which are monitored at  $t'_1, t'_2, \dots, t'_n$ , respectively. Note that waiting for an update on the monitor makes information about the observed process outdated. Therefore, in the absence of any updates, the status age of the source at the monitor side increases linearly in time and then decreases to the delay experienced by a recent update packet just after it is monitored.

### 1.2.2 Average Age

**Definition:** The time average of the instantaneous age is called "average age".

Indeed the area under the age graph (Figure 1.2) normalized by time  $T$  gives the time average of AoI which is defined by:

$$\bar{\Delta} = \frac{1}{T} \int_0^T \Delta(t) dt \quad (1.2)$$

In this definition,  $T$  is known as the observation duration. This value can be expressed

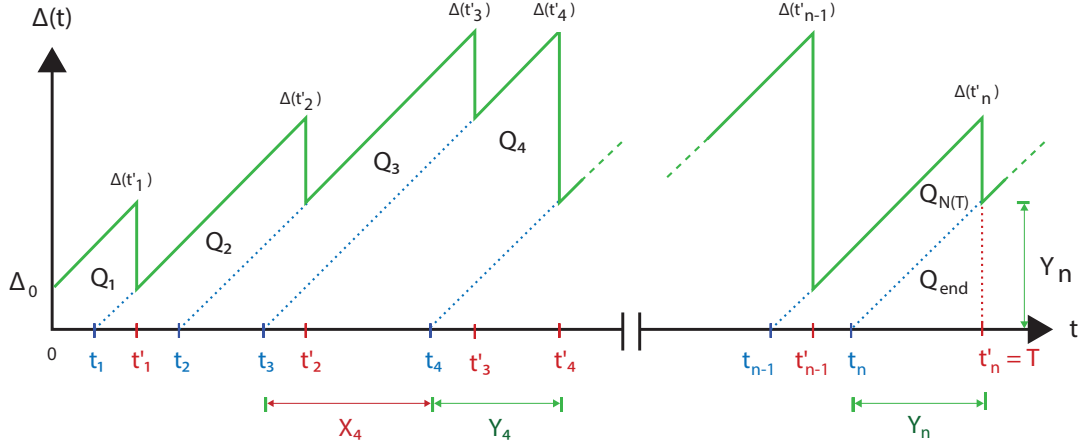


Figure 1.2: Sample path of the age process  $\Delta(t)$  with  $Q_i$  trapezoids.

in terms of update generation intervals (1.3) or update monitoring intervals (1.4).

$$T = \sum_i (t_i - t_{i-1}) \quad (1.3)$$

$$T = \sum_i (t'_i - t'_{i-1}) \quad (1.4)$$

"Average age" is the first metric that is investigated in the AoI literature. As shown in Figure 1.2 and Figure 1.3 the area under the age graph can be calculated as the sum of the areas of trapezoids  $Q_i$ 's or  $H_i$ 's as explained in the rest of this chapter.

### 1.2.2.1 Calculating the Average Age with $Q_i$ Trapezoids

For  $n$  transmitted updates, the area under age graph, (Figure 1.2), is made up of the area of polygon  $Q_1$ , isosceles trapezoids  $Q_i$ 's for  $2 \leq i \leq N(t)$  and the isosceles triangle of length  $Y_n$  located at the bottom of  $Q_{N(t)}$ . This isosceles triangle is shown as  $Q_{end}$  and has an area equal to  $Y_n^2/2$ . Therefore, based on these areas the average age is equal to (1.5), where  $N(t) = \max\{n | t_n \leq T\}$ , the maximum number of arrival updates by time  $T$ .

$$\bar{\Delta} = \frac{Q_1 + \sum_{i=2}^{N(t)} Q_i + Y_n^2/2}{T} \quad (1.5)$$



In equation (1.5) as  $T \rightarrow \infty$ , the  $Q_1$  and  $Y_n^2/2$  can be ignored, due to their small and constant value. Consequently, for  $n$  transmitted updates, the average age is equal to (1.6), where  $Q_i$ 's are obtained in (1.7).

$$\bar{\Delta} = \frac{1}{T} \sum_{i=2}^{N(t)} Q_i \quad (1.6)$$

In a steady state, the area of the isosceles trapezoids becomes important. To calculate these areas, the arrival and departure instances of updates are considered. For instance, the area of  $Q_i$  is obtained by subtracting the area of the isosceles triangle with base connecting the points  $t_i$  and  $t'_i$  from the area of the isosceles triangle whose base connects the points  $t_{i-1}$  and  $t'_i$ . Note that in (1.8),  $(t_i - t_{i-1})$  is the inter-arrival time between successfully transmitted  $i^{th}$  and  $(i-1)^{th}$  updates.

$$Q_i = \frac{1}{2}(t'_i - t_{i-1})^2 - \frac{1}{2}(t'_i - t_i)^2 \quad (1.7)$$

$$Q_i = \frac{(2t'_i - t_i - t_{i-1})(t_i - t_{i-1})}{2} \quad (1.8)$$

To simplify (1.7), let's show the elapsed time between the generation of updates  $i-1$  and  $i$  with  $X_i$ , equation (1.9), and elapsed time between the generation and available time for monitoring in the receiver of update  $i$  with  $Y_i$ , equation (1.10) :

$$X_i = t_i - t_{i-1} \quad (1.9)$$

$$Y_i = t'_i - t_i \quad (1.10)$$

Therefore,  $Q_i$  can be calculated as follows:

$$Q_i = \left(\frac{1}{2}Y_i + X_i\right)^2 - \frac{1}{2}Y_i^2 = X_iY_i + \frac{X_i^2}{2} \quad (1.11)$$

Note that the formulation using  $Q_i$ 's makes the mathematical analysis easier, therefore this type of calculation is used in most of the AoI studies.

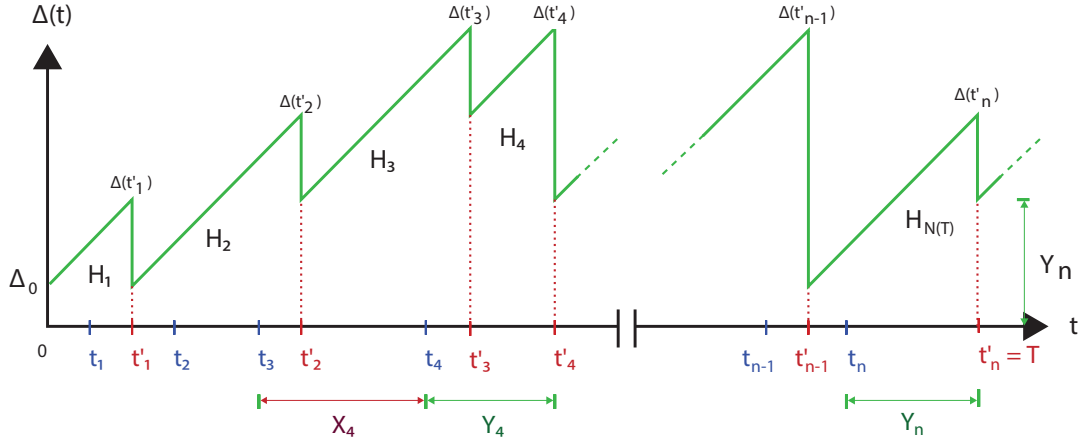


Figure 1.3: Sample path of the age process  $\Delta(t)$  with  $H_i$  trapezoids,  $\Delta_0 > 0$ .

### 1.2.2.2 Calculating the Average Age with $H_i$ Trapezoids

In theoretical AoI studies, since the steady state condition is assumed for calculations, the  $Q_1$  and  $Y_n^2/2$  can be ignored. But in practical applications due to the limited duration of observation, the parameters  $Q_1$  and  $Y_n^2/2$  become significant in (1.5). Consequently, for a practical age calculation, applying  $H_i$ 's trapezoid under age graph, (Figure 1.3) could be an accurate approach, [17].

For  $n$  transmitted updates, the area under age graph, which is illustrated in Figure 1.3, is made up of the area of trapezoids  $H_i$ 's for  $1 \leq i \leq N(t)$ . With this method, the average age is equal to (1.12), where  $H_i$ 's are obtained in (1.13).

$$\bar{\Delta} = \frac{1}{T} \sum_{i=1}^{N(t)} H_i \quad (1.12)$$

$$H_i = (t'_i - t'_{i-1})(t'_{i-1} - t_{i-1}) + \frac{(t'_i - t'_{i-1})^2}{2} \quad (1.13)$$

In (1.13),  $(t'_{i-1} - t_{i-1})$  equals to  $Y_{i-1}$  which is the  $i-1^{th}$  update's delay in the system and  $(t'_i - t'_{i-1})$  defines the inter-departure time between  $i-1^{th}$  and  $i^{th}$  updates or time interval between two fresh updates in the monitor. Note that to calculate the average age with  $H_i$  trapezoids (1.13), it is assumed the system has initial age ( $\Delta_0 > 0$ ).

### 1.2.3 Average Peak Age

The time average of peak ages is another important metric in the context of the AoI society. This metric provides insights into the maximum ages experienced by data and assists in comprehending the system's performance. The behavior exhibited by the average peak age metric is analogous to the average age metric, as both reflect the data freshness. However, tracking and calculating the amount of average peak age typically involve lower complexity compared to the average age metric. This lower complexity feature of the average peak age metric makes it a potentially more efficient and practical option for evaluating the performance of real-time communication systems in terms of information freshness.

**Definition:** The peak age for update  $i$  is defined as the instantaneous age just before the monitoring of that update in the destination, as shown in (1.14).

$$\Delta(t_i'^-) = \lim_{t \rightarrow t_i'} \Delta(t) \quad (1.14)$$

**Definition:** The average peak age [18] is known as the time average of the peak ages as shown in (1.15):

$$\overline{\Delta_{peak}} = \frac{1}{N(t)} \sum_{i=1}^{N(t)} \Delta(t_i'^-) \quad (1.15)$$

To calculate the peak age, it is possible to use transmission and monitoring instances of updates in the system as (1.16), and consequently the average peak age is obtained by (1.17) :

$$\Delta(t_i'^-) = t_i' - t_{i-1} \quad (1.16)$$

$$\overline{\Delta_{peak}} = \frac{1}{N(t)} \sum_{i=1}^{N(t)} (t_i' - t_{i-1}) \quad (1.17)$$

### 1.3 Contributions

This dissertation makes a significant contribution to the understanding of the Age of Information (AoI) in real-world scenarios. Firstly, this study investigates the variation of AoI in communication systems by using emulation-based measurements under different status update generation rates while using the TCP/IP protocol, which is widely used in transport and network communication. Also, the real-world measurements of this dissertation which are conducted on different physical networks including WiFi, Ethernet, LTE, 3G, and 2G, provide worthwhile insights into the impact of various network environments on AoI metric.

For the first time, this research exhibits a non-monotone (specifically, “U-shaped”) AoI vs update rate for the real-world data network in a link with and without buffering, which is in line with theoretical results for FCFS systems with Poisson or Gamma distributed arrivals. Additionally, this study delves into practical concerns such as synchronization, the suitability of congestion control mechanisms, and application layer approaches to control AoI. Besides these, the impact of device-related bottlenecks, such as the limited processing power of IoT nodes, is studied and the consequences of these factors on AoI are demonstrated.

This research study also presents a new data flow control algorithm specially designed for freshness prioritization in data communication systems. This flow control which is called Age-Aware Application Layer Forward Error Correction ( $A^3L$ -FEC), addresses the challenge of enabling freshness over an end-to-end TCP/IP network. The main principle of this protocol is to enhance data freshness, as quantified by the age-violation metric, in real-time communication systems. By comparing the  $A^3L$ -FEC data flow control with related existing algorithms such as BBR and ACP+, the study demonstrates its effectiveness in addressing the freshness challenge in real-world scenarios.

The findings of this research have been disseminated through the publication of several articles. [17],[19], [20], [21], [22], [23] highlight the publications that include the materials covered in this dissertation.

## 1.4 The Outline of the Dissertation

The rest of this thesis is organized as follows. In chapter 2 we present experimental and simulation-based AoI measurement results and practical issues for data flow transmitted over wireless/wired links using different transport layer protocols. Chapter 3 introduces A<sup>3</sup>L-FEC (Age Aware Application Layer Forward Error Correction), a communication solution based on Forward Error Correction (FEC) and User Data Protocol (UDP). ), where A<sup>3</sup>L-FEC improves data freshness at the monitoring end by controlling age violations in the end-to-end path by adapting the state update rate injected into the network. In Chapter 4, we present the emulation results for the proposed age-aware congestion control algorithm, A<sup>3</sup>L-FEC. Additionally, we compare the performance of A<sup>3</sup>L-FEC and ACP+ in MATLAB. Furthermore, building upon the results from Section 4.1, we conduct a comparison of A<sup>3</sup>L-FEC with the TCP-BBR congestion control algorithm on mininet emulator. Moreover, this chapter includes a comprehensive evaluation of the Age of Information (AoI) performance for well-known congestion control algorithms such as TCP BBR, TCP BIC, TCP Cubic, TCP Ldbat, and TCP NewReno over TCP/IP links. These evaluations are performed on the ns-3 simulator.



## CHAPTER 2

### AGE OF INFORMATION IN REAL LIFE

#### 2.1 Introduction

Up to now, theoretical studies have defined the Age of Information as a Key Performance Indicator, which describes the timeliness of information in time-critical systems and applications dealing with status updates. Numerous principles related to age optimization have been developed for network models, assuming simplified conditions where delay statistics and service times are known. However, implementing these optimizations in real-life networks can be challenging due to the complex interactions between different network layers, making the extraction and optimization of these statistics a complex endeavor. Furthermore, theoretical problem formulations that examine the fundamental behavior of age in networks often ignore various practical system issues that are critical in real-world implementations.

Only a limited number of implementation investigations have concentrated on the AoI to date, as evidenced by the references cited: [3, 17, 24, 22, 23, 25, 21, 26, 27]. The purpose of this chapter is to gather and discuss some of the results on the practical aspects of AoI from this growing literature. Our investigation in this chapter will contain a review of experimental and emulation-based measurements of AoI for data flows transmitted over wireless/wired links, employing various transport layer protocols. Moreover, the practical concerns such as synchronization challenges and the effectiveness of different congestion control mechanisms and application layer approach in managing and controlling AoI will be addressed in the following sections. Note that these aspects are important in understanding the real-world implications and feasibility of AoI optimization in network implementations. This chapter also

demonstrates how device-related bottlenecks, such as limited processing power, can significantly affect the AoI in practical scenarios, particularly in the context of simple IoT nodes.

The first emulation investigation of AoI in wireless links was documented in [24]. While, the initial real-life implementation for measuring the AoI variation over TCP/IP links served by various technologies such as WiFi, LTE, 3G, 2G, and Ethernet, was showcased in [22]. The empirical results in [22] show a non-monotone, specifically a “U-shaped” characteristic of AoI vs arrival rate for TCP/IP connections. This observed characteristic aligns with theoretical findings for First-Come-First-Serve systems (FCFS) with Poisson or Gamma distributed arrivals, as discussed in [28].

The U-shaped relation between AoI and the update arrival rate which is shown in [22] for the real-life TCP/IP connection reveals some important insights. Initially, by increasing the update arrival rate, the AoI experiences a rapid decline, which indicates that fresher information is being transmitted and received more frequently. However, after reaching a certain arrival rate, the AoI stays relatively flat, which implies that further increasing the update rate does not improve the data freshness. At this point, by increasing the update rate the queues build up at the bottleneck of the network due to the excessive number of updates. Consequently, a sharp increase in the AoI plot is observed due to an increase in the delay of packets.

These findings underscore the necessity for age-optimal service policies in practical networks due to several reasons: First and foremost, the increasing significance of machine-type communications, including remote monitoring, autonomous systems, and control, highlights the increase of services where data freshness takes priority over throughput as the main performance criterion. In contrast to conventional data transmission, which aims to reliably send an entire data stream, the services like status updates prioritize only the timely delivery of specific data points. Secondly, the objectives of achieving both low AoI and high throughput often align, where the principal reason behind this is that achieving desirable AoI performance necessitates a high throughput of data updates while maintaining sufficiently low delays in some applications. Hence, we believe that AoI optimization should be a focal issue in the future development of network architecture.



Applying the theoretical findings on the control of AoI to practical algorithms poses a challenge, mainly because theoretical work usually assumes complete knowledge related to the network delay statistics (e.g., as seen in [29, 30]). However, it is not easy to obtain such statistics, particularly when the network expands and the number of connections increases. Note that relying on incorrect or insufficient statistics to form decision-making policies can lead to suboptimal results.

In these situations, machine learning methods can be employed to adaptively create AoI-aware sampling, scheduling, and transmission policies. The concept of introducing a delay to the sampling process based on the current AoI and network delay statistics was proposed by [29]. However, this intelligent sampling process was based on knowledge of network delays. In [23, 30], the reinforcement learning method was employed to derive the sampling process, while the used method eliminated the need for prior knowledge of delay statistics. Likewise, in [31, 32, 33, 34], the application of reinforcement learning techniques is investigated for age-aware scheduling.

This chapter is organized as follows: In Section 2.2, a succinct summary of the behavior of steady-state AoI is shown under basic queuing and service disciplines. This foundation serves as context for interpreting practical results discussed in the next sections. In Section 2.3, the methodology utilized for measuring AoI on a basic network testbed is explained. Section 2.4 delves into discussing general issues and solutions related to computing AoI in realistic networks, such that it addresses problems like clock bias, and offers mitigation strategies. The behavior of AoI in TCP connections is studied in Section 2.5. First of all its behavior within an emulation testbed is shown in subsection 2.5.1, and later in subsection 2.5.2 we review AoI in TCP connections on a real-world testbed which operates across diverse physical links like WiFi, Ethernet, LTE, 3G, and 2G. It is followed by a comparative analysis of AoI patterns in UDP and TCP connections in Section 2.6. Later, the statistical learning methods applied to age optimization in practical networks are overviewed in Section 2.7. The application-layer mechanisms for age control over UDP, particularly focusing on the ACP protocol, are summarized in Section 2.8. Then the implementation details of age-aware scheduling in Wi-Fi uplink or downlink scenarios are recapped in Section 2.9. Finally, Section 2.10 concludes the chapter by offering a holistic perspective on potential future advancements in this research field.

## 2.2 Age of Information: Definition, Measurement and Behaviour in Queuing Systems

As mentioned in chapter 1, status age of a flow is defined as  $\Delta(t) = t - U(t)$ , where  $U(t)$  is the generation time (i.e. time stamp) of the newest data packet that has been received by the destination by time time  $t$ .

It should be emphasized that minimizing status age is not equate to minimizing delay. Instead, it requires the joint optimization of delay and throughput [4].

We demonstrate this phenomenon using basic queuing systems in Figure 2.1, considering Poisson packet arrivals. In this figure, the average delay and AoI are plotted versus the load of a server where the server corresponds to a communication channel in this context. We assumed that the transmission duration of each packet in the link is depicted as an exponential random variable, independent of other transmission durations. Load here corresponds to throughput; as long as the queue functions in the stable regime. This abstraction relates to an M/M/1 queuing system<sup>1</sup> that employs a first-come, first-served (FCFS) discipline.

The throughput of the system rises in a straight-line manner with the load, denoted as  $\rho$ , within the range of values plotted, because the network is stable within this range. Keep in mind that the AoI **decreases** as the load increases until reaching the  $\rho^* = 0.53$  which is known as the age optimal operating point of M/M/1. The reason for this decrease is the incoming packets are promptly served without notable wait times in the queue; because the queue of the server is often empty up to the age-optimal operating point of M/M/1. When packages are delivered with almost no delay, they help to a reduction of the age of the system. However, the AoI begins to rise when the load surpasses  $\rho^*$ , due to the impact of the queuing creation event. This phenomenon occurs naturally from the First-Come-First-Serve (FCFS) queuing

---

<sup>1</sup> An M/M/1 queue is an abstraction of a single server queuing system. In this system, the intervals between packet arrivals are exponentially distributed and independent, with a mean of  $1/\lambda$ . Likewise, service times of packets follow independent, identically distributed (i.i.d.) exponential distributions with an average of  $1/\mu$ . At a steady state, the expected delay encountered by packets is  $(1/\mu)/(1-\rho)$ , where  $\rho = \lambda/\mu$  depicts the “load”—the probability that the system is busy in a steady state. Thus, as the load increases, the steady-state delay grows monotonically and escalates towards infinite value as  $\rho$  approaches 1. If  $\rho \geq 1$ , it means the system enters an unstable regime, where the finite delay is not guaranteed. In scenarios where packet discarding is permitted—such as in an M/M/1/k, system capable of holding up to  $k - 1$  packets in the queue, the delay remains finite while sacrificing a proportion  $\max(0, \rho - 1)$  of all incoming packets.

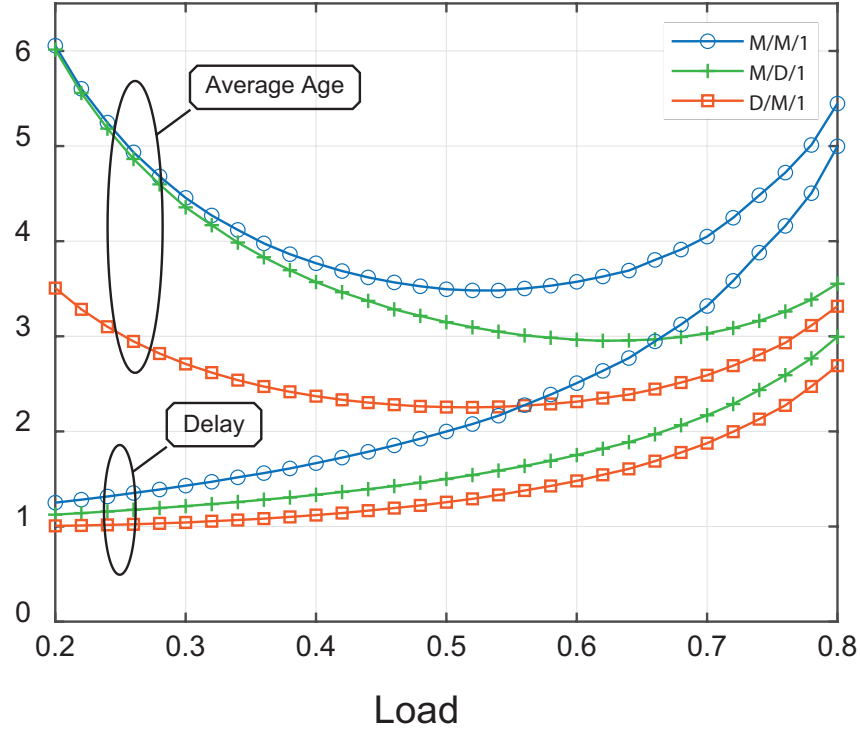


Figure 2.1: average delay and AoI achieved by an M/M/1 queuing system with a service rate of  $\mu = 1$ , [22].

policy, which causes the new arrival packets to become stale as they wait in the queue behind previously arrived packets.

Concentrating on the curves presented in Figure 2.1 reveals that the AoI demonstrates different characteristics in comparison to delay. Increasing the update rate, while potentially improving the volume of throughput, does not inherently enhance AoI. On the contrary, it can lead to a rise in AoI, particularly in the scenario with First-Come-First-Serve (FCFS) discipline.

Improved AoI performance is achieved in models with multiple servers (for example, M/M/2 [35] or M/M/ $\infty$  models) owing to the increased service rate. In an M/M/2 system, the AoI is almost halved compared to that in an M/M/1 system. However, it's important to note that increasing the update rate within these systems might lead to certain packets becoming obsolete upon arrival time at their destination; because there may be other packages before them in the system queue. This underscores the importance of well-designed packet management strategies [36].

The body of work concerning the analysis and enhancement of Age of Information (AoI), as seen in studies like [4, 37, 38, 39, 40, 18, 41, 36]), centered around exogenous packet arrivals prior to the publication of [42]. The concept of the "generate-at-will" framework was initially presented in studies such as [42, 43, 44], wherein a source has the ability to generate and transmit packets intentionally to manage the age of the system. The model introduced in [43] gains significance as we consider the control of rates in an application later on top of various transport layer mechanisms, to achieve favorable AoI performance even in scenarios involving network bottlenecks. Hence, we provide a detailed explanation of this model below.

In [43], updates indicated by indices  $i = 1, 2, \dots$  and originating from a source subject to a stochastic delay process represented by  $Y_i$  within the network. This delay process is arbitrary and not affected by the transmitter's actions. This scenario represents a situation where packets flow into a large network where congestion and delay are caused by various factors, among which the influence of this particular source is minor. The source is assumed to receive immediate feedback when the update is delivered, which corresponds to ignoring the delivery time for packet acknowledgments. The strategy for creating update packets is defined by the sequence of waiting time denoted as  $Z_i$ . One potential work-conserving policy that ensures efficient resource utilization is the zero wait (ZW) policy. This policy maximizes the communication link utilization and minimizes the average delay by setting  $Z_i$  to zero for all  $i$ , leading to the submission of a new update as soon as the server (i.e., communication link) becomes available. [44] raised the query of whether the ZW policy, if it's implementable, exhibits favorable AoI performance. Surprisingly, the response to this question is negative: Across a broad range of delay distributions, the ZW policy cannot achieve the lowest possible AoI.

Strategies that reduce the overall penalty associated with AoI were characterized in [43] and [44]. The results from those investigations revealed that an optimal update policy involves transmitting updates at a rate lower than the maximum allowable rate. These outcomes showed that by prudently adjusting the rate at which updates are generated, the AoI achieved at the destination can be controlled, even when exterior factors like delay are subject to variability. This can be accomplished by using the measured or estimated value of the penalty associated with the age, [42]. From an

overall network perspective, such policies have the potential to prevent unnecessary load on the network, minimize the negative impact of queuing, reduce congestion-induced staleness, and even promote the efficient utilization of resources.

Throughout the remaining sections of this chapter, we will extend the insights gained from studying theoretical queuing and service models, as well as age control strategies documented in the literature, to real-life network scenarios.

### **2.3 Measuring AoI within a Basic Physical Network**

To measure the AoI in an actual network, precise timing information is critical at the node where the computation will be performed. Given that the clocks on client and server nodes might not be necessarily synchronized, it's necessary to either estimate time offsets or establish network-wide synchronization prior to calculating the AoI.

Prior to delving into discussions about synchronization techniques and the impact of potential errors in synchronization or timing estimation on age measurements, let us present the network model that bypasses the necessity for synchronization, allowing us to observe how the sampling rate affects age within a network operating under a realistic transport layer protocol, [17]. The setup consists of a sampler-transceiver node located in Ankara, Turkey, and an echo server located in Istanbul, Turkey (Figure 2.2). The sampler-transceiver node generates UDP packets containing timestamp information indicating the generation time, packet ID, and dummy payload. These packets are sent to the Echo server through the Internet. Once the packet is received, the Echo server immediately sends back each packet. As soon as receiving and decoding packets by the sampler-transceiver node, the time stamp of each received packet is recorded and compared with the time stamp of its generation, allowing the calculation of the instantaneous age. To handle the concurrent tasks of packet sampling, transmission, and reception, the nodes employ multi-threaded and multi-process methods.

The bidirectional link connecting the sampler-transceiver node and the echo server can be understood as a single end-to-end link extending from the sender node to the receiver node, facilitated by the intermediary echo server which acts as one of the intermediate nodes within the network. Since the same device performs the tasks

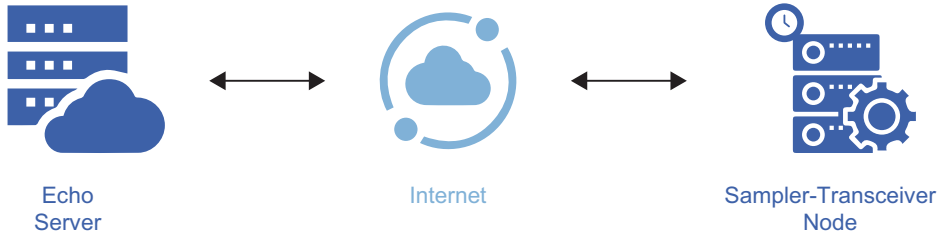


Figure 2.2: A depiction of the practical experimental testbed.

of sending and receiving, the AoI can be measured without facing synchronization challenges. To measure the AoI, UDP packets with a size of 1058 bytes (inclusive of the header) are transmitted at a constant rate. The average age of information is computed by applying the timing data accessible at the transceiver into equation (1.12). The rate in this experiment is linearly increased, from 1 packet per second to 370 packets per second, in order to explore the correlation between age and rate. The relationship of the resultant age rate is exhibited in Figure 2.3. Despite the fact that the routers and switches within the Internet infrastructure utilize First-Come-First-Serve (FCFS) queues, observing results that closely resemble those acquired through the theoretical examination of a basic M/M/1 queuing system is very remarkable. [4].

Another notable conclusion is that contrary to the outcomes shown in Figure 2.1 for M/M/1, M/D/1, and D/M/1 queues, the AoI shows significantly lower sensitivity to sampling rate changes, which covers a large range of rate values. As an example, the average age remains almost unchanged as the packet generation rate increases from 100 packets per second to 340 packets per second. This occurrence can be justified through (i) the determinism in service time among individual links and (ii) load distribution among multiple servers due to routing data through multiple potential paths. However, it is important to emphasize that when the sample injection rate into the network exceeds a certain threshold, the incidence of packet drops sharply increases.

The experimental results shown in Figure 2.3 also confirm the anticipated relation between the minimum average age and the round-trip time (RTT), that packets encounter in the network. In the experimental testbed, the dummy test packets were used to evaluate the hop numbers and determine the average round trip time from the sender to the receiver at the beginning of the test. Observations showed that each packet traveled a total of 18 to 20 hops during its journey from the sender to the echo

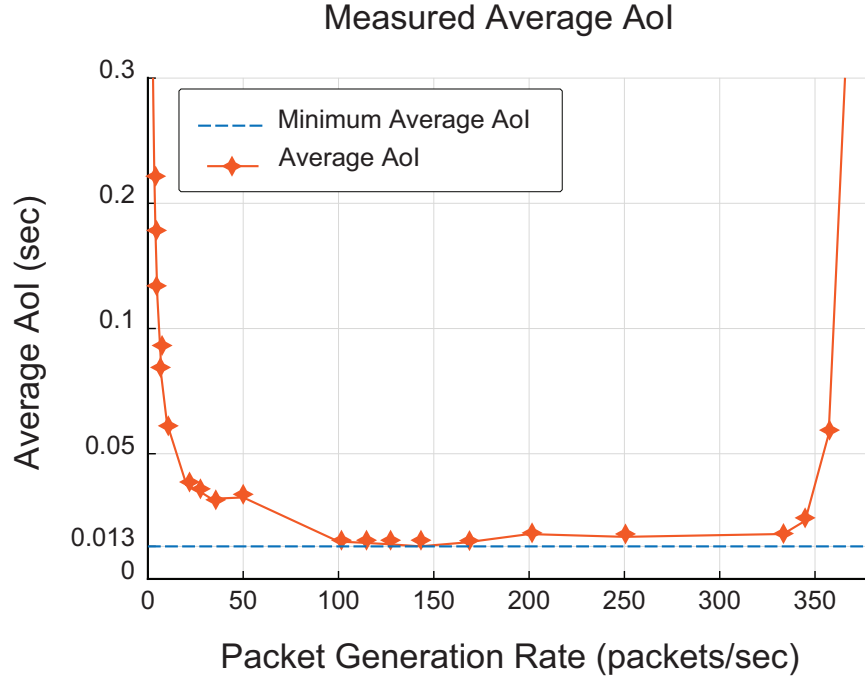


Figure 2.3: AoI plotted against sampling rate, observed on the UDP protocol testbed, [17].

server and back, with an average RTT of 12.5 milliseconds. Since these test packets were sent in no-load network mode, the measured average RTT was near the minimum possible RTT. This minimum RTT is basically the sum of the propagation delay for a round trip and the total transmission time for a packet along its designated path. For a deterministic sequence of packet transmissions at a rate indicated as  $R_p$ , the minimum average age of a packet after receipt at the receiver is given by 2.1 without the existence of delays caused by queuing. The actual measurements obtained from real-world scenarios closely match to this estimate: the age reaches its minimum value at 13 milliseconds, approximately when  $R_p = 140$  Hz. At this particular rate, the equation (2.1) predicts an average age estimation of around 16 milliseconds, derived utilizing the mean RTT. This estimation is expected to be somewhat conservative, given that the actual minimum round-trip time is shorter.

$$\Delta_{\min} = RTT + \frac{1}{2R_p} \quad (2.1)$$

Note that the automatic synchronization facilitated by the simplified network model

utilized in this section, where there exists an echo server, cannot be replicated in real-world scenarios where the sender and receiver are located at different locations. In the following sections of this chapter, we will demonstrate the results of experimental measurements of AoI for such networks. To achieve this goal, we first need to explore potential AoI measurement approaches, synchronization methods, and the impact of synchronization inaccuracy on AoI measurements.

## **2.4 Clock Bias and Synchronization Process for Age Measurement in Networks**

Based on the specific application and network structure, AoI and related metrics may be computed at the receiver, transmitter, or via central processing node. In this section, we will address real-world timing concerns regarding AoI measurement.

### **2.4.1 Measurement of Age Under Synchronization**

Having an accurate method for measuring AoI requires synchronization between the transmitter and receiver clocks. This synchronization can be achieved through several methods, some of which are outlined here:

- **Synchronization via Utilizing a GPS Clock:** Currently, the use of the GPS module is one of the most accurate tools for synchronizing remote devices. According to [45], the GPS clock has an accuracy of tens of nanoseconds. GPS clocks are used in drones, base stations, and software-defined radios [27]. However, this approach may not be suitable for networks consisting of low-power devices such as sensor modules and IoT nodes, because it is not possible to include GPS modules in each node.
- **Synchronization in Advance Through a Common Reference Point:** In certain scenarios, it is possible to physically link two devices before deploying them in specific locations. In such cases, the receiver and transmitter can be synchronized via a common real time clock (RTC). However, a drawback of this approach is that if either device is reset or turned off, the sync will be lost.



- **Synchronization via the Network Time Protocol (NTP):** NTP is utilized to harmonize the clocks of all nodes involved in a network. It provides synchronization accuracy of approximately 0.1 milliseconds in fast LAN environments and within tens of milliseconds on intercontinental Internet connections, [46]. The performance of NTP is influenced by the fluctuations in network jitter, which can lead to a decrease in accuracy.
- **Synchronization Through Preceding Signaling:** In order to measure AoI, synchronization can be established directly at the application layer. This is achieved by measuring the RTT at the transmitter through a simple ping mechanism before starting the main transmission session. The receiver responds to the ping by sending the timestamp of the received packet, and the sender can correct its clock by taking into account a round-trip delay of twice the one-way delay in each direction. This assumption is based on the fact that the packets used in this phase are small, thus incurring minimum transmission times in the queues. As a result, the RTT can be largely attributed to the delay.

After syncing is complete, the measurement of AoI can occur either at the transmitter or the receiver side. To evaluate the AoI at the receiver, the sender must send the timestamps of the generated packets. Note that this data can be extracted from both UDP and TCP headers. If the header does not contain timestamp information, it can be included in the payload. To calculate the average AoI, the equation (1.12) is recommended due to its enhanced accuracy when averaging over short time intervals. On the other hand, to perform AoI measurements at the transmitter side, it is required to send the received timestamp from the receiver to the transmitter. This information can be ingrained in an Acknowledgment (ACK) for each packet, which is usually included in the payload of the ACKs. However, this approach has certain limitations. Due to the need to wait to receive an ACK, it creates an additional delay in the calculations. Additionally, miscalculations may occur if ACKs are lost in the return link. Note that if ACKs are not already part of the protocol and are sent exclusively for AoI measurement, this method consumes more channel resources compared to measurements at the receiver side. However, in scenarios where there is a necessity to dynamically control the age and parameters associated with adapting transmission policies, age measurement at the transmitter becomes a more sensible choice.

### 2.4.2 Effect of Poor Synchronization on Age-dependent Metrics

The techniques discussed in the previous section all come with the limitation that they are prone to introduce certain degrees of synchronization discrepancy, which in turn are assumed as errors in age measurements. This subsection presents how synchronization errors affect age calculation.

To calculate age-related metrics, such as average peak age, timestamps from the receiver and transmitter are necessary. However, since the receiver and transmitter are separate, each has its own separate system clock. Even after synchronization, a constant bias remains because of synchronization errors. Assuming that the drift of each clock is negligible in the corresponding time interval, the bias remains constant between the two clocks.

Let us use  $s'_i$  and  $s_i$  to represent the timestamps that mark the transmission time of the  $i^{th}$  packet from the viewpoint of the receiver and sender, respectively. In other words, the sender has the exact time of transmission, while the receiver's knowledge about the time is affected by a constant bias synchronization error, meaning that:

$$s'_i = s_i + B \quad (2.2)$$

In a similar manner, let  $r'_i$  and  $r_i$  indicate the timestamps of  $i^{th}$  packet which signify the moment of packet reception from the viewpoint of the receiver and sender, respectively. This leads to:

$$r'_i = r_i + B \quad (2.3)$$

The receiver provides the received timestamp  $r'_i$  to the sender and allows the sender to calculate the average age of the information using the equation (1.5). By substituting (2.3) in (1.8) and (1.13), the erroneous areas of the trapezoids are obtained.

$$H'_i = H_i + B(r_i - r_{i-1}) \quad (2.4)$$

Then by inserting (2.4) in (1.12), the average age becomes

$$\overline{\Delta}' = \frac{1}{T} \sum_{i=1}^{N(T)} (H_i + B(r_i - r_{i-1})). \quad (2.5)$$

Ultimately, since the total time interval, denoted by  $T$ , is equal to the sum of the interarrival times at the receiver,  $\sum_{i=1}^{N(T)} (r_i - r_{i-1})$ , calculating the average age in the presence of synchronization errors according to the equation (2.5) is:

$$\overline{\Delta}' = \overline{\Delta} + B. \quad (2.6)$$

This means that due to the synchronization error, the average age is shifted by the same bias value, which exists between the transmitter and receiver clocks. Similarly to calculating the peak AoI via (1.17), the time synchronization error adds the same constant bias to the results.

$$\overline{\Delta_{peak}}' = \overline{\Delta_{peak}} + B \quad (2.7)$$

At this point, an important observation is on the agenda: The effect of a fixed bias or synchronization error between the transmitter and receiver clocks fundamentally changes the AoI and the average peak AoI along the age axis. But for the same bias value, even if there is an unknown bias between the clocks, the optimal operating point in the sampling rate vs. AoI curve remains unaffected. This gives an opportunity to neglect synchronization before the age calculation, especially if the goal is to adjust the transmission rate. This topic will be reconsidered in Section 2.4.3.

It is obvious that a constant bias in clock measurements doesn't influence the optimal sampling rate value when the penalty function, a more general formulation of average age, is linear in relation to age. But there exist penalty functions that are nonlinear with respect to age, as indicated in prior works [43, 44]. In nonlinear penalty functions, the optimal operational points differ for biased and unbiased age. For an age penalty function denoted as  $f(\Delta(t))$ , let  $F(t) = \int_0^t f(\tau) d\tau$ . Consequently, the time-averaged age penalty is expressed as:

$$\overline{\Delta_{Bias}} = \frac{1}{T} \sum_{i=1}^{n(T)} F(r_i + B - s_{i-1}) - F(r_{i-1} + B - s_{i-1}) - F(r_i - s_{i-1}) + F(r_{i-1} - s_{i-1}). \quad (2.8)$$

For the commonly used linear ( $f(t) = \alpha t$ ), exponential ( $f(t) = e^{\alpha t} - 1$ ) and logarithmic ( $f(t) = \log(\alpha t + 1)$ ) penalty functions [47, 48], the age biases are calculated using (2.8) are presented as follows, [21],

$$\overline{\Delta_{Bias, Linear}} = \alpha B; \quad (2.9)$$

$$\overline{\Delta_{Bias, Exp}} = \frac{1}{\alpha \sum_{i=1}^{n(T)} (r_i - r_{i-1})} \left( e^{\alpha(\theta+B)} - e^{\alpha(\beta+B)} - e^{\alpha\theta} + e^{\alpha\beta} \right) \quad (2.10)$$

$$\begin{aligned} \overline{\Delta_{Bias, Log}} = \frac{1}{\sum_{i=1}^{n(T)} (r_i - r_{i-1})} & \left( \frac{1}{\alpha} \left( \log(\alpha\beta + 1) - \right. \right. \\ & \log(\alpha\theta + 1) + \log(\alpha(B + \theta) + 1) - \log(\alpha(B + \beta) + 1) \Big) - \\ & \theta \log(\alpha\theta + 1) + \beta \log(\alpha\beta + 1) + \log(\alpha(B + \theta) + 1)(B + \theta) \\ & \left. \left. - \log(\alpha(B + \beta) + 1)(B + \beta) \right) \right), \quad (2.11) \end{aligned}$$

where  $\beta = r_{i-1} - s_{i-1}$  and  $\theta = r_i - s_{i-1}$ .

Given that non-linear age penalty functions have exhibited advantages in sampling, remote control, and tracking contexts [47, 48, 49], it becomes critical to determine the appropriate age for use in such penalty functions. As a result, synchronization becomes necessary in applications that require non-linear penalty functions. In scenarios where a linear relation between age penalty and measured age exists, the discrepancy in age measurements can be tolerated, and an alternative method to AoI estimation can be utilized instead of synchronization as described in the following sections.

### 2.4.3 RTT-Based Asynchronous AoI Estimation Method

When the exact age value is not critical, the sender can use the RTT of a packet, estimated from receiving an acknowledgment (ACK) from the receiver after sending the packet, to estimate the age, [25]. It is important to note that this approach does not require synchronization between sender and receiver. This is because the ACK, without the timestamp data, has a shorter blocklength. As a result, ACKs are less error-prone and less delayed, and the round-trip time is mainly attributed to the forward link. This approach inherently leads to an overestimation of the AoI. However, it fits well in scenarios where the age penalty function preserves linearity, thus making the bias in the age measurement negligible for the optimal operating point. In section 2.6, we discuss some experimental results using this method for age estimation.

## 2.5 The Impact of the Access Network on AoI within TCP/IP Connections

In this section, an experimental setup was used to evaluate the relationship between AoI and status update generation rate in the context of an end-to-end TCP flow passing through a real network. The first hop of the network after the transmitter was varied between WiFi, Ethernet, LTE, 3G, and 2G access links, providing a comprehensive range of scenarios. In addition, an emulation study was performed to detect the effect of bottlenecks introduced at different points of the network on the correlation between sample generation rate and end-to-end AoI. These findings were detailed in [22], representing an initial effort to explore age dynamics through practical implementation of packet streams in the Internet landscape.

The main motivation behind this empirical investigation stems from a question: whether there is a non-monotonic convex relationship between AoI and sampling rate in real-world scenarios, similar to that observed in simple ideal network settings (e.g., Figur. 2.1)? This query is valid for the following reason: Practical communication networks contain many interdependent protocol components across different layers, each with its own unique queuing mechanisms. Usually, these mechanisms are beyond the control of end-to-end applications. In the following sections, we discuss the results related to this topic, which served as a stepping stone for further research.

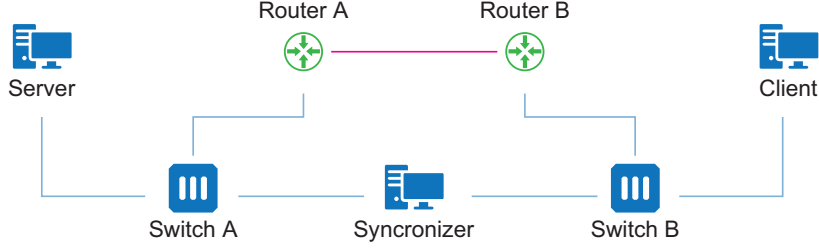


Figure 2.4: Networking topology in the core emulator, [22].

### 2.5.1 Measuring AoI on an Emulation Testbed

For the study of this section, we employed an open-source network emulator called CORE [50], a tool previously utilized in [24] and [22]. The setup within the emulation study is outlined in Figure 2.4. Our emulation setup employs the CORE 5.0 emulator on a PC powered by the Fedora 27 Linux distribution, featuring an Intel® Core™ i7-7700HQ CPU @ 2.80GHz  $\times$  8, 16 GB RAM, and a 1 TB 7200 rpm HDD. This setup contained three virtual nodes: server, client, and time\_synchronizer. The time\_synchronizer node was responsible for synchronizing the clock between the client and the server, which ensures accurate age calculation on the server.

The client generates samples with timestamps at different sampling rates, which are transmitted to the server. After receiving each sample, the server performed calculations to determine the age. The data exchange between the server and the client is done through a 130 kbps BW-limited link, shown in bold line in Figure 2.4. All other links, with the exception of this particular link, had unlimited bandwidth, allowing for instantaneous packet transmission. All links, apart from this particular case, had no inherent path delay, thus indicating that the delays were caused solely by the transport layer queuing. Note that the maximum sampling rate is kept low relative to the maximum transmission rate on the link. This is to ensure that the buffer of the link is not overloaded. Therefore, at the beginning of each test iteration, the buffers were empty, and the sampling rate was gradually increased to create a trajectory that shows the AoI versus rate relationship. Four illustrations of these trajectories are shown in Figure 2.5a. Notably, all replicates show a distinct U-shaped trend in age with respect to the sampling rate. The initial age and the shape of the curve show minimal deviation between iterations, with an AoI of nearly zero at moderate data rates, at-

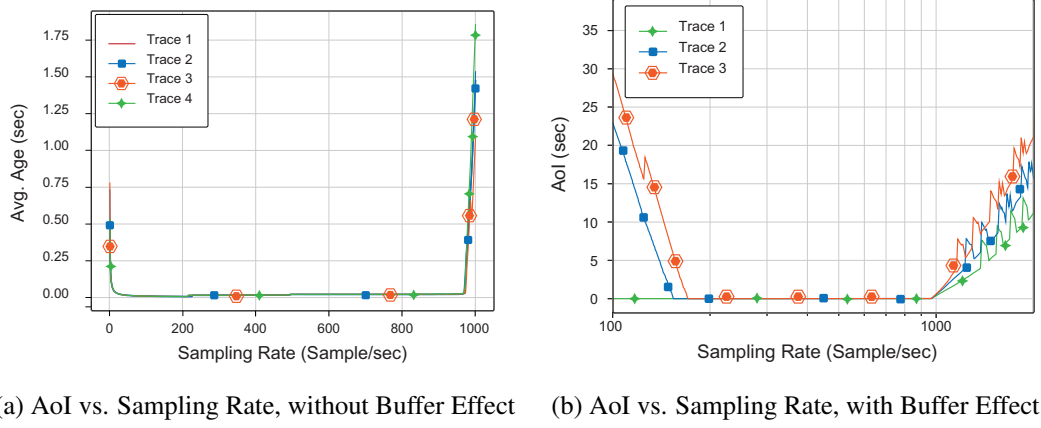


Figure 2.5: AoI behavior in emulation topology, [22].

tributed to the absence of path delay in this emulation setup. However, in the second experiment shown in Figure 2.5b, the sampling rates reach levels of such magnitude that saturate the link's buffer capacity. In this case, the initial iteration starts with the minimum AoI, but subsequent iterations have progressively increased AoI values. As the sampling rate increases to very high levels, the buffer becomes overwhelmed with a notable accumulation of samples from previous iterations. This accumulation leads to the aging of newly received data, which ultimately leads to an increase in the AoI.

### 2.5.2 AoI Measurements on a Real-world Testbed

This section presents an analysis of real-world AoI measurements for various network connections, including 2G, 3G, LTE, WiFi, and Ethernet, as documented in the work by [22]. Measurements used a two-step method: first, timing synchronization between the server and the client was done, then samples were generated at the client end at a variable and increasing rate. The client, located at coordinates  $39^{\circ}53'29.6''\text{N}$   $32^{\circ}46'56.6''\text{E}$ , had access to the Internet through WiFi, Ethernet, or cellular networks interchangeably. For cellular connectivity, the setup uses TURKCELL as the service provider and uses a USB connection to connect to the mobile phone. Known for its relatively high bandwidth capacity of 53 MB/s according to the USB bandwidth specification, the USB 2.0 connection easily manages the data rates arising from the maximum sampling rates. Therefore, the effect of USB buffering on AoI was minor.

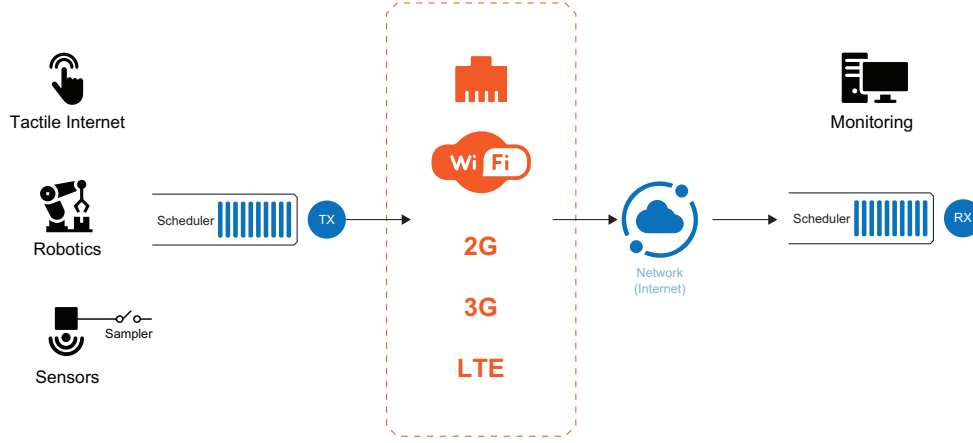


Figure 2.6: Physical testbed setup for AoI measurements across different network connections, [22].

Physical testbed setup for AoI measurements across different network connections is shown in Figure. 2.6. This setup includes a remote server located in France. Ookla Speed Test was used to quantify the upload speed in different network scenarios. Measurements were done for LTE (at 17.84 Mbps ), 3G (at 0.47 Mbps ), 2G (at 0.03 Mbps ), WiFi (at 99.04 Mbps ), and Ethernet (at 93.37 Mbps ) connections. Each individual data sample in this experiment had an approximate size of 35 KB.

Unlike the emulation test which is presented in Section 2.5.1, this experimental investigation does not contain a `time_synchronizer` node. The sender and receiver were located in distant geographical regions. The synchronization process was conducted through preceding signaling, as described in Section 2.4.1. To clarify more, the client would continuously send requests to the server, which would respond by sending its timestamp alongside each response. The client marked the time at the start of each request and again after receiving the corresponding response. This data was used to calculate the round-trip delay. Assuming symmetry in delay in both directions of the communication link, the current time of the server was estimated. By comparing this estimated server time with its own time, the client obtains an offset value for clock synchronization. To improve the accuracy of the offset calculations, the offsets of several independent runs were averaged. For the synchronization process, a reliable Ethernet connection, which is notable for its lower latency compared to a cellular connection, was used.



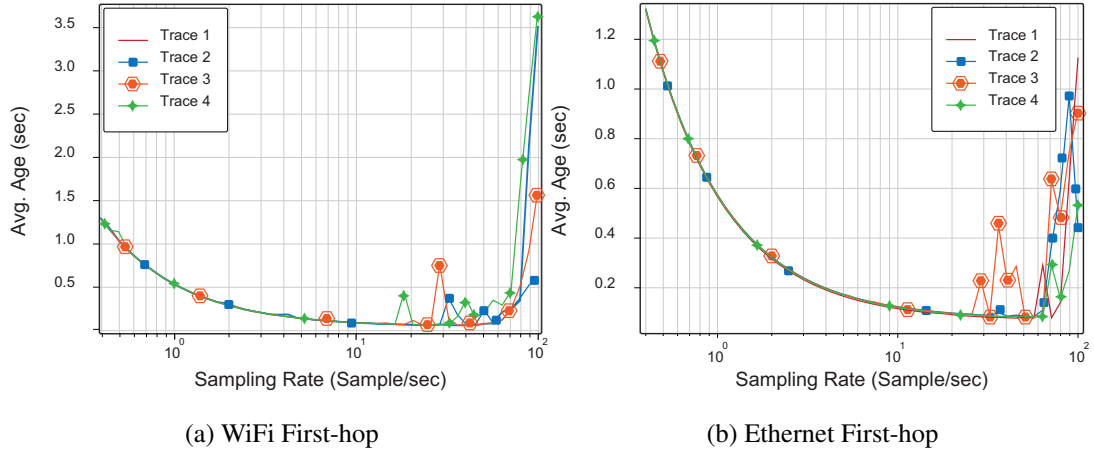


Figure 2.7: AoI measurements across various access networks, [22].

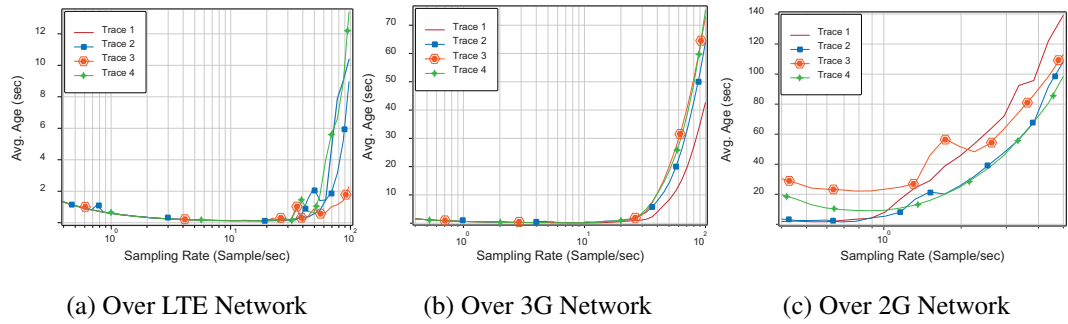


Figure 2.8: AoI measurements on mobile networks, [22].

Note that in this experiment the transport layer queues function on a first-in-first-out (FIFO) principle. It means the packet at the front of the queue is selected and served when a transport channel becomes available. Moreover, the frequency at which fresh data was entered into the queue was dictated by the sampling rate.

Figures 2.7a, 2.8b and, 2.8c show results for individual mobile communication network types. In these graphs, a non-monotonic pattern is revealed. AoI decreases with increasing sampling rate but begins to increase beyond a certain threshold. Although these figures initially show a similar pattern, the sampling rate at which the AoI rise begins is different in each case. This variation can be attributed to the various bandwidth limitations inherent in these communication network links. It can be noted that the AoI experiences a considerably more rapid rise in cases involving 2G networks.

## 2.6 Comparing AoI Between UDP and TCP-IP

Sophisticated mechanisms in TCP variants facilitate high-speed data transmission while reducing packet loss by using features such as adaptive window size, congestion control, and re-transmissions. Although these features enhance the protocol, unfortunately, they lead to increased computational demands and somewhat hinder its suitability for real-time applications. For example, the TCP re-transmission mechanism is inefficient in terms of age management because old re-transmitted loss packets do not contribute to age reduction.

Many real-time applications use RTP over UDP, but an important query arises: Are existing protocols designed for real-time applications compatible with age objectives? The answer to this question emphasizes the need to carefully examine the aging performance of UDP traffic. In summary, while protocols such as RTP and their configurations at the application layer, as seen in technologies such as cloud gaming (e.g., [51]), offer relatively timely data delivery for streaming applications, they do not inherently address age objective. In such protocols, the conventional approach involves transmitting all the data, while in state update applications it becomes unnecessary to transmit the entire data set, especially when a fresh packet is in the sender's buffer. Because this newer package can effectively make all previous packages obsolete. Hence, our view supports the idea that it is crucial to integrate age-aware sampling at the application layer and use the Last-Come-First-Serve (LCFS) queuing discipline on top of the UDP framework.

As evidenced in references [4, 37, 52], it has been observed that in First-Come-First-Serve (FCFS) systems operating without buffer management or strict constraints, the average age experiences an initial decrease as throughput rises, however, as the communication system struggles to service the increased throughput, the queuing delay becomes significant and causing the average age to escalate. As a result, a U-shaped correlation between the AoI and throughput emerges. Therefore, FCFS buffers in practical network infrastructure components such as routers, switches, and access points can be expected to create a similar non-monotonic relationship in real-world networks.

The specific pattern of the relationship between age and throughput remains under the impact of factors like transport layer protocol characteristics, CPU capacity, as well as transmit and receive capabilities. These aspects will be explained through the forthcoming results within the context of the IoT setup discussed in this section.

To achieve satisfactory AoI performance in a low-power IoT node, certain prerequisites must be met. Initially, the CPU should possess the appropriate capability to generate packets at a sufficient rate. Subsequently, the transceiver modules deployed at both sender and receiver nodes must be able to process the generated packets promptly. Any shortage in processing power in any of these elements can lead to a performance bottleneck for the network. This problem is particularly relevant for low-power IoT devices, that are prone to such bottlenecks. Even with typical transport protocols like TCP, known for their substantial computational demands, and similarly, with UDP, IoT devices can face challenges with AoI minimization issues. This emphasizes the necessity for innovative approaches to address these concerns.

In this section, our primary focus is to investigate the age characteristics exhibited by TCP and UDP flows traversing the Internet. Then, we assess the age behavior of the same types of flows among IoT nodes. However, this evaluation is performed in a local IoT setup and highlights the impact of the limited computing capabilities inherent in IoT devices on the AoI. The findings stemming from these investigations are substantiated in the work cited as [21].

In these conducted experiments, the RTT information was employed to constrain the synchronization error, as elucidated in Section 2.4.3. Before taking AoI measurements, RTT estimation was conducted by dispatching multiple packets and subsequently receiving the corresponding acknowledgments (ACKs). Given the relatively small size of ACK packets, the estimated RTT was assumed as an upper limit for transmission time. This estimated value was used to approximate the synchronization error. Furthermore, the remaining age bias was limited by the RTT value. Note that, based on equation (2.9), even if the approximations still have a constant bias, this does not affect the operating point of the optimal sampling rate for obtaining optimal age due to the linearity of the penalty function, which in this case is the age itself.

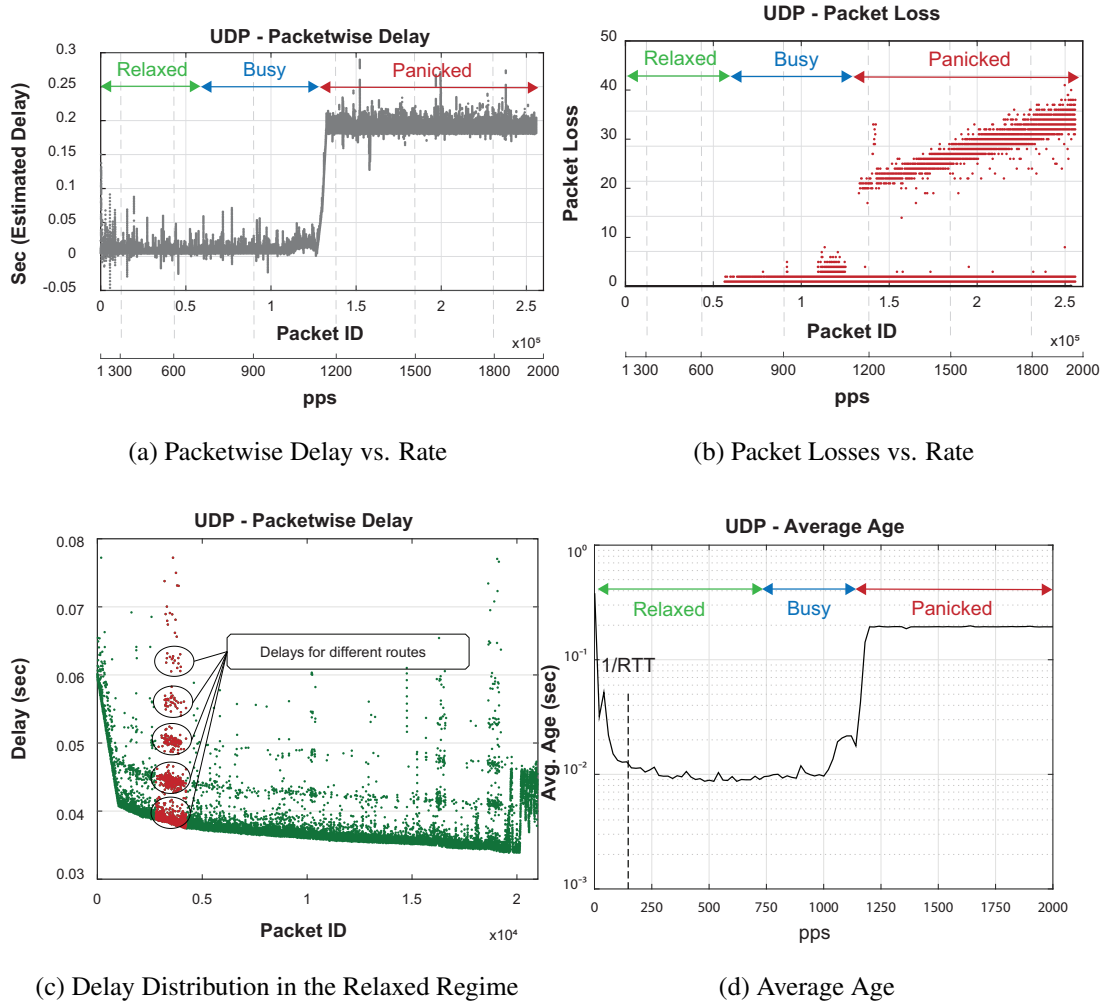


Figure 2.9: Delay, packet loss and age versus rate for UDP over Internet, [21].

### 2.6.1 Comparing UDP and TCP-IP in a Multi-hop Network Test Environment

In this test setup, powerful desktop computers were used to send and receive TCP or UDP packets over a conventional Internet/IP infrastructure. This involves the deployment of three PCs located in remote locations within a geographical area of 450 km. This arrangement resulted in a network characterized by paths with different delays. Among these PCs, one was designated as the receiver node, while the remaining PCs served as senders of packets to the receiver side. Notably, the communication path between one of the sender's and receiver's PCs contained around 7 hops, which is associated with an RTT of approximately 6 milliseconds. On the other hand, the second path, consisting of about 12 hops, showed an RTT of 80 ms.

The results of the UDP transmission tests on the aforementioned Internet testbed are illustrated in Figure 2.9. In these results, the primary reason for the AoI increase was attributed to a noticeable packet loss rate, while the delays incurred within the transport layer queues were observed to be minimal. Figure 2.9a depicts the delays experienced by successfully transmitted packets, while Figure 2.9b displays the count of lost packets situated between consecutive successful transmissions. The horizontal axes in both figures correspond to packet IDs and the increasing sample generation rate. In Figure 2.9c, delays experienced by all packets are plotted against the rising rate, revealing a delay distribution that indicates the existence of multiple routes, a factor contributing to the sustained constancy of age despite the escalating rate. Figure 2.9d displays the average age variations for this experiment.

The outcomes of the UDP tests conducted on the Internet testbed disclose the existence of three operational modes: relaxed, busy, and panicked. The classification of these modes is predicated on the distinctive characteristics of packet loss.

In the relaxed regime, the number of generated samples remains notably lower than the threshold that would lead to network congestion. Increasing the transmission rate during this state decreases the average age because the heightened frequency of sampling delivers more prompt data. By increasing the samples in the network, congestion gradually occurs, prompting certain nodes to randomly discard packets, thereby triggering the onset of the busy phase. This change is evident in Figure 2.9b, where the initiation of the busy domain is marked by a noticeable uptick in lost packets, ranging from zero to a moderate number between 1 and 3. Throughout this busy mode, the count of lost packets typically adheres to this range. Despite intermittent packet loss during the busy mode, the delay remains similar to that experienced in the relaxed mode (refer to Figure 2.9a). This phenomenon arises due to the intermediate nodes' capacity to handle elevated traffic without introducing additional queuing delay. Consequently, the predominant reason for end-to-end delay is the propagation delay along the network links.

As the rate continues to rise, we move into Panic mode, characterized by amplified occurrences of both packet losses and delays. In this case, queuing delays in intermediate nodes play an important role. It is crucial to note that since there is no resending

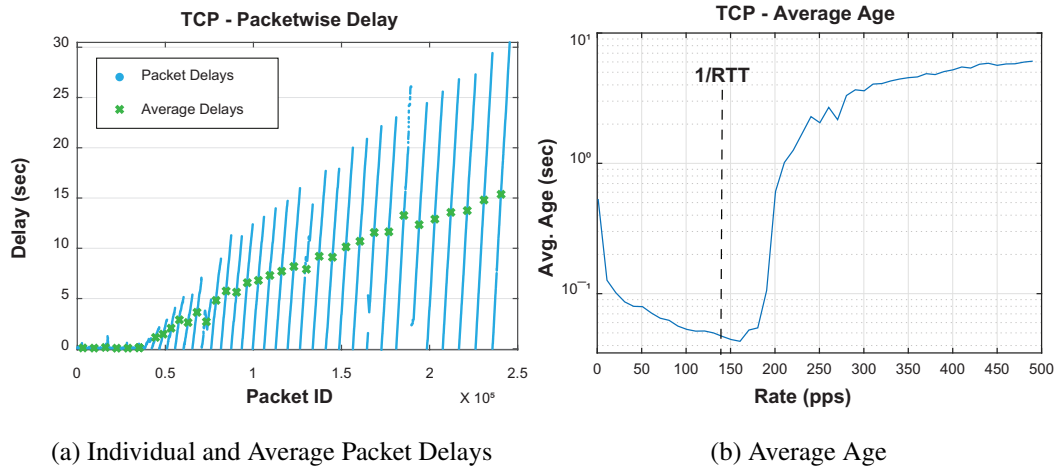


Figure 2.10: Age and packetwise delay for TCP over the Internet, [21].

mechanism for loss packets in UDP, packet delay does not show a continuous increase with the increase of the rate. Instead, they stabilize at a level subject to the highest queue delay. Consequently, as the rate increases, the AoI maintains a constant value.

In contrast to UDP, TCP employs a packet retransmission mechanism to compensate packet loss in the network which leads to an increase in delay as shown in Figure 2.10a. Therefore, in the case of TCP, if the transmission rate escalates, the system's age increases due to the increase in both packet loss and delay as illustrated in Figure 2.10b. Furthermore, the empirical results demonstrate a resemblance to the U-shaped pattern observed in the age-throughput curve of FCFS queues. Finally, based on the test results, it can be seen that TCP flows often experience a rapid increase in AoI due to congestion when the rate exceeds a certain threshold. While for UDP scenario abrupt growth in age is not observed as the rate rises.

## 2.6.2 Comparison of UDP and TCP-IP on an IoT Testbed

In this subsection, a Wi-Fi network is developed, which consists of two NodeMCU ESP32 IoT nodes equipped with Xtensa® LX6 processors (600 MIPS), and an 802.11n mode Wi-Fi router. These nodes act as a sender and receiver pair, transmitting TCP/UDP packets through the central router. The results of this experiment are presented in [21]. Since the IoT nodes of this testbed are not capable of running

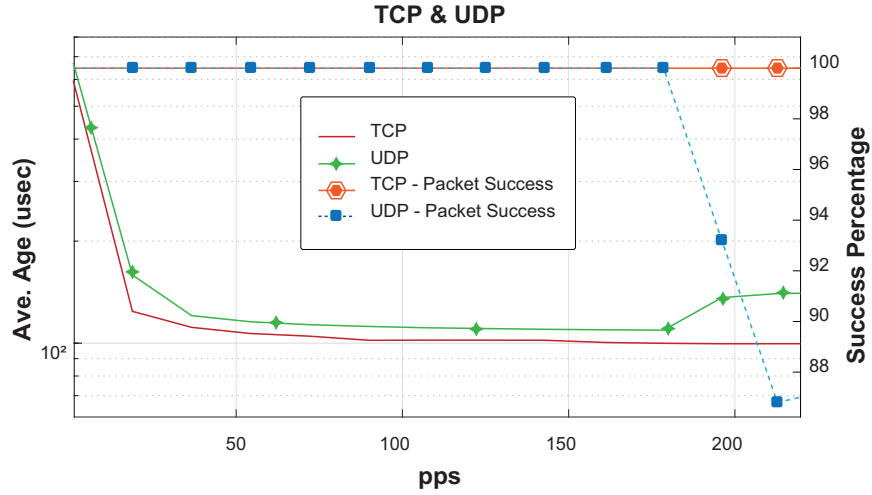


Figure 2.11: Packet loss and average age evaluation of TCP and UDP on a local Wi-Fi network utilizing IoT devices, [21].

standalone operating systems, the implementation of TCP/UDP was facilitated by Lightweight IP-Stack (LWIP), open-source software used by various IoT devices.

Our test results emphasize that the AoI behavior in IoT devices is significantly limited by the available memory and processing capabilities of the device. The maximum packet generation rate achievable by these devices is remarkably lower than that of conventional PCs, and the size of the TX/RX buffer is insufficient to effectively reflect the queuing delay introduced at high transmission rates. As depicted in Figure 2.11, device limitations prevent a clear U-shaped AoI behavior for both UDP and TCP. It has been observed that the Wi-Fi module integrated with the ESP32 has a higher throughput than the processing unit of the device itself. Consequently, the absence of a U-shaped AoI trend is not surprising.

It is worth noting that in numerous IoT applications, given that IoT widgets are often small and battery-operated, energy appears as a limiting factor affecting operational effectiveness. Energy constraints directly affect on device's attributes such as transfer rate, duty cycle (as shown in [53, 54, 55, 56, 57]), and processing capability. In this subsection, we will show that in simple IoT devices, the primary limiting factor for aging performance is processing power. This observation is consistent with previous theoretical efforts that have aimed to address this issue (e.g., [58]).

## **2.7 Utilizing Machine Learning Techniques for Enhancing AoI Optimization in Real-life Networks**

The optimization techniques for AoI presented in various prior studies (e.g., [4, 59, 40, 60, 61, 62]) relied heavily on having knowledge about the statistical characteristics of the underlying processes (such as network delays and energy arrivals). This dependency limits the practicality of these methods, especially in scenarios where these statistics are unknown. Consequently, utilizing machine learning techniques becomes a promising avenue for optimizing AoI in real-life networks.

As far as we are aware, [63, 23] were the pioneering works to apply machine learning, particularly reinforcement learning, for AoI minimization. In [63], the problem of AoI minimization on a point-to-point link was formulated using Hybrid Automatic Repeat reQuest (HARQ) and cast as a constrained Markov Decision Process. The issue was addressed through the application of algorithms like Value Iteration and SARSA on a discrete state space with ages defined as integers. More recently, a series of studies ([64, 65, 66, 67, 68]) have investigated the application of machine learning techniques for age optimization under different network configurations.

In this section, our attention is directed toward the practical deployment of deep reinforcement learning (RL) algorithm with the aim of sample rate optimization. This algorithm operates in a continuous state-space and serves to regulate the arrival rate of a packet flow across the Internet, as presented in [23]. This concept is similar to the rate control models found in previous studies of TCP and UDP, which were covered in the previous sections.

By modeling a stream of packets originating from an application running between a server and a client over the Internet, we introduced a new technique based on deep reinforcement learning in [23]. This approach is designed to obtain the ability to adjust the update generation rate, aiming to minimize the AoI, while ignoring the need for a priori knowledge about the network topology. By assessing the learning model in an emulated network environment, it was argued that this method has the potential to be extended to larger and more complex networks with unknown delay distribution.



First of all, let us briefly review the approach presented in [23]. In this approach, the sender initiates generating update packets at distinct times  $t_1, t_1, \dots, t_n$ , which are subsequently received at times  $U(t_1), U(t_2), \dots, U(t_n)$ . It is important to note that the order of packets may be changed in the network. The AoI optimization problem is transformed into a Markov Decision Process (MDP) characterized by unknown transition probabilities; where, the age at time  $t$  serves as the state at that specific moment:  $s_t = \Delta_t$  and at any given time  $t$ , the available set of actions consists of two options (pause or resume):  $a_t = \{p, r\}$ .

Therefore, the next state at any time relies on the present state and the chosen action, as determined by the transition probabilities denoted by  $p(s_{t+1}|s_t, a_t)$ . The learning process of these transition probabilities was achieved through the use of reinforcement learning (RL). In this context, the main goal of RL is to identify the action that maximizes the expected cumulative reward  $r(s, a)$  across the distribution of trajectories, denoted by  $p_\theta(\tau)$ . A trajectory is defined as a state-action pairing, denoted as  $\tau = (s, a)$ . In line with the goal of minimizing AoI, a cost function was introduced, which was shown as  $c(s_t, a_t) = -r(s_t, a_t) = -\Delta_t$ , and the goal was formulated precisely in [23], intended to address the equation (2.12).

$$\theta^* = \arg \min_{\theta} \left( \limsup_{T \rightarrow \infty} \frac{1}{T} \mathbb{E}_{(s_t, a_t) \sim p_\theta(s_t, a_t)} \left[ \sum_{t=1}^T \Delta_t \right] \right) \quad (2.12)$$

For solving the equation (2.12), the widely recognized deep Q-network (DQN) algorithm first introduced in [69] was employed. The main objective of this learning algorithm involved estimating the expected reward (Q-Value) of potential actions within each state, a process facilitated through neural networks. Given that the intention was to minimize costs rather than maximize rewards, changes were made to the algorithm. In this context, it was modified to minimize the cost function, denoted in (2.13).

$$c(s_t, a_t) = 1 - \exp\{-\Delta_t\} \quad (2.13)$$

In the algorithm outlined in [69], a neural network was utilized to establish a mapping from states to the corresponding Q-value for each feasible action. These Q-values essentially form a vector denoting the expected costs when each action  $a$  is executed

within state  $s$ . The term  $\arg \min_a \hat{Q}_\phi(s)$  shows the action in state  $s$  that would minimize the expected cost  $\mathbb{E}[c(s, a)]$ . Instead of selecting the action linked to the minimum Q-value at each step with a probability of 1, the agent is prompted to engage in exploration via the utilization of the  $\epsilon_{greedy}$  strategy [70]. This strategy presents randomness by selecting a random action with a probability of  $\epsilon$ . In addition, techniques aimed at increasing stability, such as replay [71] and Double DQN [72], were also integrated into the process.

The previously expressed DQN algorithm was tested using the topology shown in Figure. 2.4 on CORE, an open-source network emulator. In this topology, the delay along the direct link between routers A and B was chosen from a truncated Gaussian distribution with adjustable mean and variance, while the remaining links have zero delay. The routing design guarantees that there is only one path between the server and the client, which includes routerA and routerB.

A synchronizer node was used to ensure synchronization between server and client. Synchronization accuracy in this topology is guaranteed due to no delay in the links connecting the synchronizer, server, and client. The client acts in three distinct modes: (1) generating time-stamped samples at a constant rate, (2) transmitting generated samples to the server, and (3) responding to pause and resume instructions that are received from the server. Upon receiving the pause control command, the client ceased generating new samples and continued this activity upon receiving the resume control command.

The initial test included a link with unlimited bandwidth. In this setting, due to the absence of queues, aging was attributed only to the path delay ( $1 \pm 0.5s$ ). Therefore, in this setting, the pause command only had the potential to increase age. The age calculation was based on the timestamps of the received samples. This age value was then employed by a deep reinforcement learning (RL) algorithm to determine an appropriate age reduction command. This experiment effectively served as a performance evaluation of the RL algorithm.

Notably, in this particular case, the optimal decision was always “resume”. This result was due to the fact that the network buffer was able to accommodate the data transfer at any sampling rate without overloading.

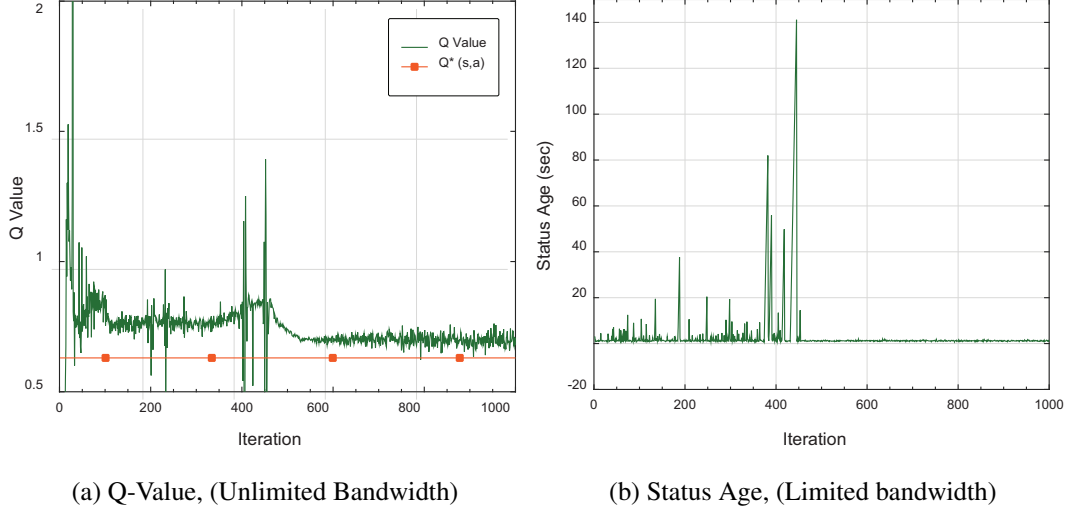


Figure 2.12: Results of Q-Value and status age vs. iteration per 10 iterations, [23].

During this experiment, a dual deep Q network (DQN) configuration was implemented, which used two 24-unit hidden layers and experience replay. The initial training consisted of 10,000 iterations. The original network was updated after each iteration, while the target layer received updates every 100 iterations. According to the boundary conditions set in the Bellman equation, the Q-value for the action *resume* is expected to approach  $Q(s, a) \approx 1 - \exp\{-\Delta_{min\_age}\}$ :

$$Q(s, a) = c(s, a) + \gamma \min_{a'} Q(s', a'); \quad (2.14)$$

After reaching the terminal state, the optimal Q-value becomes  $Q^*(s, a) = c(s, a) = 1 - \exp\{-\Delta_{min\_age}\}$ . Given the fixed delay of 1 seconds, it is expected that the Q-value related to the *resume* action will gradually approach the value of  $Q(s, a) = 1 - \exp\{-1\} \approx 0.63$ . This convergence was observed after approximately 5000 iterations. The evolution of the instantaneous Q-value and its optimal value are shown in Figure 2.12a. Meanwhile, Figure 2.12b shows the state's age progression over time.

Drawing upon the collected experimental results described earlier and the Universal Approximation Theorem [73], it was claimed in [23] that this strategy holds promise for extension to larger networks with general delay distributions.

## 2.8 Implementation of Age Control Strategies at the Application Layer Over UDP

Age Control Protocol (ACP) [25] is a transmission control strategy designed to operate on top of UDP, with the aim of reducing the AoI in dynamic networks. This protocol consists of two distinct phases: the initial phase and the epoch phase. In the initialization phase, the source (sender) sends a specified number of packets to the monitor (receiver) while simultaneously calculating the round-trip time (RTT). Subsequently, the protocol moves to the epochs phase, which consists of discrete intervals called epochs. In each interval, the ACP calculates the average AoI, the exponentially weighted moving average RTT, the exponentially weighted moving average of inter-ACK arrival times, and the average backlog, which represents the average number of packets sent to the receiver but not yet received. At the end of each epoch phase, the transmission rate of the transmitter is updated in response to variations in backlog and average AoI. This update is done by using three distinct choices of action: additive increase (INC), additive decrease (DEC), and multiplicative decrease (MDEC). By choosing either "INC" or "DEC" actions, the protocol attempts to increase or decrease the average delay by a specified step size, denoted as  $\kappa$ . This parameter, step size, is important in shaping the subsequent age dynamics.

Research in [25] has shown that in an intercontinental link, ACP has the ability to reduce the average age by approximately 33% compared to *Lazy*. *Lazy* is a simple policy that keeps the total number of in-flight packets around 1 (corresponding to the bandwidth-delay product [74]), which is performed by setting the packet transmission rate to  $1/\text{RTT}$ . Elaborated insights into this ACP and its enhanced version, ACP+ have been provided in Section 4.2.1.

In a study conducted by [75], we evaluated the performance of ACP and ACP+ [76] in the context of IoT devices, specifically concentrating on ESP32 devices. Our research led us to recognize several challenges related to the application of ACP and ACP+ in these particular IoT devices. To handle these challenges, we offered solutions aimed at solving the problems. Moreover, a feedback mechanism was designed to facilitate loop termination that ACP and ACP+ often get caught in due to buffer management errors in ESP32 devices.

Our research in [75] includes AoI measurements performed under ACP for different values of  $\kappa$ , as well as *Lazy Policy* [25] and ACP+ evaluations. Results from these evaluations show that the original ACP+ presents excessive greediness when deployed in networks characterized by small delays. To address this inherent problem of the original ACP+, we developed a compatible version of ACP+. Outcomes revealed that the modified ACP+ produced better results compared to the original ACP+.

## 2.9 Practical Testing of Wireless Access Schedulers with Age-aware Characteristics at the Link-Layer

Building on the theoretical findings in the field of multiaccess and broadcast scheduling, as shown in studies such as [77, 78], real testbed implementations are presented in [27, 79]. The WiFresh RT approach, [27], offers an innovative deployment of Wi-Fi uplink communication using hardware-level scheduler programming in a network of FPGA-based software-defined radios. In addition, it introduces an application layer method that facilitates the integration of age adaptation and stops the necessity for changes to the lower layer protocol stack.

In the WiFresh RT framework, each source appends a timestamp in each newly generated packet before depositing it into a Last-Come-First-Served (LCFS) queue of size 1. This LCFS queue only keeps the last packet and throws away the elder ones. Note that the coordination of network communications is facilitated by WiFresh RT on the access point (AP) side. WiFresh defines two operational modes for the AP: (1) waiting for a data packet, and (2) sending a polling packet. The WiFresh application, which works on the standard Wi-Fi protocol using UDP, includes all the attributes of WiFresh RT and introduces additional features. These additional features include the fragmentation of large updates and a simple internal synchronization algorithm.

Experimental findings presented in [27] show that severe congestion in wireless networks causes significant deterioration in the Age of Information (AoI), resulting in outdated data arriving at the destination. WiFresh emerges as a practical solution to this issue, showing the capacity to achieve nearly optimal information freshness in wireless networks of various sizes, even in network overload scenarios. A compar-

ative evaluation was performed involving WiFresh, UDP over Wi-Fi, and the Wi-Fi Age Control Protocol (ACP) [25]. It was done on a Raspberry Pi wireless base station which received data from  $N$  sources.

The results show that ACP combined with Wi-Fi UDP enhances the average age by a factor of four. In contrast, WiFresh is forty times better than UDP Wi-Fi in terms of average age improvement. This obvious advantage of WiFresh is attributed to the integration of a polling-based multiple access mechanism, Max-Weight (MW) policy, and Last-Come First-Served (LCFS) queues in the Application layer.

Unlike ACP, neither the WiFresh nor Wi-Fi UDP architectures attempts to control the packet generation rate at the source level. Therefore, when the number of sources ( $N$ ) increases to the point that the cumulative packet generation rate exceeds the network capacity, both Wi-Fi UDP and WiFresh experience overload. This situation leads to a rapid accumulation of backlogged packets in resources. While Wi-Fi UDP uses a First-Come-First-Served (FCFS) policy, which results in a significant queuing backlog and increased packet delay, WiFresh exhibits graceful scalability, allowing The average age to increase linearly with  $N$ .

The outcomes of an experimental implementation of a wireless downlink scenario are detailed in [79]. Using a multiuser testbed supplied with multiple software-defined USRP radios, the investigators implemented and evaluated scheduling policies such as Round-Robin, Max-Weight, Whittle's Index, and Greedy policies presented in [78]. Implementations were realized by utilizing Labview and subjected to different node formation and power constraints.

In this downlink scenario, in each time frame, the access point (AP) generates a status update, which is then sent to the specified receiver as defined by the employed scheduling policy. The transmission consists of QPSK modulation and BCH coding. Notably, both the sender and receiver nodes are in clock synchronization owing to their interconnection with a shared computer. A comparison of the performance of these scheduling policies, based on the measured age values, shows that the Max-Weight and Whittle's Index policies perform better than the Round-Robin and Greedy policies, especially when dealing with signal strength power constraints and asymmetric channel conditions i.e., observing different channel gains.

## 2.10 Conclusions

In summary, this chapter has investigated the practical measurements of the Information Age (AoI) by reviewing the existing literature. We have studied experiments and emulations related to AoI in wireless and wired data flow with different transport layer protocols. Additionally, we address practical concerns such as synchronization, congestion control, and the impact of application layer strategies on AoI. This insight sheds light on the real implications of AoI optimization in the network. The impact of device-related limitations, especially in simple IoT nodes, is also highlighted.

Moreover, fundamental studies were recapped, including the first emulation of AoI in wireless links and the setup of real-world AoI measurements. The outcomes showed a U-shaped trend of AoI, which mirrors the theoretical findings for certain systems. This trend shows that increasing the update arrival rate initially reduces the AoI, but after a certain threshold, further increase leads to congestion and increased delay and finally leads high AoI value.

Based on the explanations of this chapter, it can be concluded that simple settings, such as sampling rate optimization at the application layer, can significantly improve the AoI and meet the needs of modern applications, including the Internet of Things.





## CHAPTER 3

### A<sup>3</sup>L-FEC PROTOCOL

#### 3.1 Introduction

In recent years, the demand to transmit the state of variable systems through high-fidelity communication channels has increased significantly. This need arises in various fields, including industrial automation, network control, and remote monitoring, where accurate and reliable monitoring of system status at the destination is of great importance. In order to ensure that the system's status is monitored correctly and in a timely manner, it is imperative that the system's response time remains within an acceptable range. However, when a network is overloaded and carries more data than it can handle, the network/queue response time is increased, leading to a decrease in performance and quality. This is because the high response time is manifested through an increase in queuing delay and packet losses of the network, leading to congestion in the network. In essence, congestion poses a significant challenge to the efficient and timely transmission of variable system status packets over networks. It is therefore necessary to develop strategies to manage congestion in networks, especially in scenarios where the system's status must be accurately monitored in real-time or with high data freshness requirements. Such strategies may include variable sampling rate, traffic shaping, load balancing, or prioritization of traffic, among others, aimed at ensuring that the network capacity is not exceeded and the system's response time remains within an acceptable range while data freshness is preserved.

This chapter introduces the A<sup>3</sup>L-FEC (Age Aware Application Layer Forward Error Correction), a practical age-aware congestion control algorithm for communication networks. A<sup>3</sup>L-FEC is a communication solution based on "forward error correction"

(FEC) and "user datagram protocol" (UDP) to improve the freshness of data at the monitoring end. This algorithm controls the age violation over the end-to-end path by adapting the rate of status updates injected into the network. The idea is to ensure that there are enough status updates during transit to provide fresh data at the destination, while also avoiding a large backlog of excess update packets at the bottleneck. This implies that congestion control algorithms should adopt a "keeping the pipe just full, but no fuller" strategy, [80].

### **3.2 Related Work**

Several congestion control algorithms have been proposed and employed to improve network performance across different categories. The initial category of congestion controls, such as Reno [81] and Cubic [82], rely mainly on packet loss as a primary indicator of congestion. They tend to fill network buffers, leading to eventual losses due to excessive queuing at the bottleneck. While these congestion control algorithms aim to achieve high throughput, they do so at the expense of increased queuing delay.

To tackle the issue of queuing delay in networks, a second category of congestion control algorithms has emerged. These algorithms, such as Vegas [83] and FAST [84], focus on utilizing the delay rather than the loss as the congestion signal. By doing so, they attempt to maintain the network link capacity at an optimal level. However, a challenge with these congestion control algorithms is that they tend to overestimate the network delay due to factors such as ACK compression and network jitter. As a result, they operate at a lower level than the maximum network link capacity, which may not be optimal for efficient data transmission.

To address this issue, researchers have been working on developing the third category of congestion control algorithms that can accurately estimate the network delay while maintaining high link capacity utilization. These algorithms aim to strike a balance between minimizing queuing delay and maximizing link capacity utilization, resulting in efficient data transmission over the network. The development of such congestion control algorithms is crucial to ensure that the network operates optimally, and data is transmitted efficiently without significant delays.

The third category of congestion control algorithms, like BBR, uses the delay and bottleneck bandwidth rate as a signal to avoid congestion. This category is known as hybrid congestion control algorithms. The recently proposed Bottleneck Bandwidth and Round-trip propagation time (BBR) protocol [85] tries to run a TCP connection at the bottleneck bandwidth rate with minimal delay. This situation is achieved when the total data in flight is equal to the bandwidth-delay product, (BDP). Indeed BBR uses the round trip time, (RTT), signal to detect congestion before a loss occurs due to buffer unavailability at a specific queue along the path between source and destination. This approach of BBR can be referred to as "proactive queue management" and is effective in improving network performance and reducing queuing delays.

It is evident that none of the congestion control categories mentioned in this study take into account the freshness of updates. Because their ultimate aim is to provide high throughput by having as many packets as possible in a queue without exceeding the available queue occupancy. However, they have their challenges and applications. The absence of a freshness metric in congestion control algorithms can cause a significant issue in scenarios where the timeliness of data is critical. These scenarios could include real-time applications, trading markets, and other time-sensitive systems. Failure to consider the freshness of updates in congestion control can result in stale or outdated data being presented to end-users, potentially leading to incorrect decisions or actions and can have severe consequences. In such cases, it is crucial to develop congestion control mechanisms that prioritize the freshness of updates, along with ensuring efficient use of network resources.

Fortunately in recent years to provide a fresh update on the destination, various aspects of the age of information were discussed and analyzed by researchers. Recent surveys [86], [87], [88] list several studies that have been conducted in this area. Among them, there has been a relatively small number of studies such as [24], [22], [17], [21], [23], [27], [89], [90] and [20] focus on the age of information in real-world networks and its optimization.

The first emulation study of the Age of Information (AoI) metric in wireless links was investigated in [24] by applying open-source network emulation tools CORE [50] and EMANE. Researchers in [24] evaluated the age for different theoretical models under

different system settings such as arrival and departure processes, average age, peak average age, number of flows, and packet loss. Study [22] exhibits the first real-world AoI measurement over TCP/IP links, where it also used the CORE emulator to understand the effect of bottlenecks on end-to-end AoI for different sample rates at various links. Their experimental AoI measurement results for TCP/IP connections served by WiFi, LTE, 3G, 2G, and Ethernet links, report a non-monotone (“U-shaped”) AoI vs. transmission rate. These results are in line with theoretical results for FCFS systems with Poisson or Gamma distributed arrivals which are reported in [28]. Authors in [17] consider a two-way connection between a sampler-transceiver node in Ankara, Turkey, and an echo server in Istanbul, Turkey. The sampler-transceiver produces and sends UDP packets (1058 bytes), containing the timestamp marking the generation time, packet ID, and a dummy payload, and transmits the packet to the echo server through the Internet. The echo server echoes the received packet immediately after receiving it. Upon reception and decoding, the receive timestamp of each packet is noted and compared with the generation timestamp. Then the instantaneous age and average AoI are calculated. The sampling rate is linearly increased, from 1 packet per second up to 370 packets per second, to investigate the age and transmission rate relation. Their findings indicate that the results are consistent with the theoretical analysis of a simple M/M/1 queuing system [4]. In order to explore the behaviors of the age of information (AoI) for several transport protocols, including TCP, UDP, and web-socket, on both wired and wireless links, the authors of [21] established two testbeds, which they named them "Internet Testbed" and "IoT Testbed". Internet Testbed contains three high-power desktop PCs to send TCP/UDP packets through regular Internet/IP infrastructure. These nodes are located in Istanbul (Anatolian side), Istanbul (European side), and Ankara to test the behaviors of the age of information (AoI) on different paths with different path delays. The route between PCs in Istanbul/Anatolia and Istanbul/Europe contains approximately 7 hops with 6 ms RTT. The path between Ankara and Istanbul/Europe has around 12 hops with 80 ms RTT. Their IoT Testbed contains two NodeMCU ESP32 as IoT nodes and a Wi-Fi router as the central node to set up a local wireless IP network. In their study, IoT devices - with one serving as a transmitter (TX) and the other as a receiver (RX) - send TCP/UDP packets via a central router node using 802.11n IEEE standard. To facilitate this process, the researchers utilized Lightweight IP-Stack (LWIP), an open-

source software that supports IP, TCP, UDP, DHCP, DNS, and HTTP, to establish communication between a transmitter and receiver over the Internet. Their study not only demonstrates the practical computation of age but also highlights the potential bias that can arise from synchronization errors between the transmitter and receiver. By conducting experiments on the Internet and IoT testbeds, they were able to gain valuable insights into how TCP, UDP, and web-socket protocols perform in terms of optimizing data freshness and reducing the impact of congestion on network performance. These efforts serve as a roadmap for improving the overall performance and reliability of the system in handling the increasing data flow and maintaining optimal information freshness. Building upon these insights, we have launched the study of refining and redesigning congestion control mechanisms. Our aim is to ensure the delivery of fresh and up-to-date data to the destination, even in complex scenarios characterized by high congestion and dynamic data communication environments.

As explained, the cited studies on the age of information (AoI) measurement hold considerable promise for understanding the dynamics of information freshness in real-world communication systems. However, a significant issue that has not been addressed in most of these studies is the development of an AoI optimizer that can adjust the transmission rate according to changes in network topology, delay, and packet loss based on the age metric.

Fortunately, the existing body of literature on the topic of AoI shows that researchers have proposed various methods in recent years to integrate AoI optimizers into communication systems and increase data freshness. Real-life implementation of a deep reinforcement learning-based algorithm for transmission rate optimization to control the arrival rate of a flow of packets over the Internet has been introduced in [23]. The AoI optimization problem in their study is modeled as a Markov Decision Process (MDP) with unknown transition probabilities. They evaluated the learning model on an emulated network and demonstrated that the method can be scaled up to any realistic network with unknown delay distribution. In [27], the authors propose WiFresh, an unconventional network architecture that scales gracefully, achieving near-optimal information freshness in wireless networks of any size. Indeed WiFresh is an innovative scheduling solution designed to improve the freshness performance of WiFi networks. It is implemented and tested on two different platforms. The first imple-

mentation, WiFresh Real-Time, was done at the MAC layer in a network of eleven FPGA-based software-defined radios using hardware-level programming. The second implementation, WiFresh App, runs over UDP and standard WiFi at the Application layer of a network of twenty-five Raspberry Pis using Python 3, without modifications to lower layers of the networking protocol stack. Their experimental results show that WiFresh can improve information freshness by two orders of magnitude compared to an equivalent standard WiFi network under a high load. To recap the [27] study: WiFresh is a scheduling approach designed for dense WiFi networks to improve data freshness.

Authors in [89] showed efforts to carry the intuition from theory to practice to control congestion with an age objective by keeping the “pipe just full, but not fuller” as suggested in [80]. They designed the Age Control Protocol (ACP), a transport-layer solution for multi-hop IP networks, enabling timely delivery of updates by adapting the sending rate. ACP works on top of UDP in an application-independent and network-transparent manner without specific knowledge about the network condition. Indeed the Age Control Protocol aims to minimize the age of information of a source update at a monitor connected over an end-to-end path on the Internet. [89] also introduced the Lazy policy, a transmission policy, which transmits data with an interval around Round Trip Time (RTT). The lazy policy was used as a benchmark to evaluate the performance of the ACP. Since ACP had some flaws and shortcomings, an enhanced version of ACP called ACP+ was presented in [90]. They showed ACP+ has superior performance for timely delivery of updates over fat pipes and long paths. Recently to evaluate the ACP and ACP+ performances on IoT devices, such as ESP32, a short-haul, and low-delay connection has been set up by [20]. Several issues related to the implementation of ACP and ACP+ in this setting have been reported and solutions to solve these issues have been offered in their study. According to their results, the original ACP+ is too greedy for small-delay networks, therefore they suggested the modified ACP+ for these types of networks.

Our aim in this study is to control the age of Information in a communication system that transmits samples of variable processes from the transmitter side to the receiver side with high data freshness over the network, which is similar to ACP protocol but with a different approach. Our AoI controller application, A<sup>3</sup>L-FEC, will monitor

the packets to minimize the frequency of AoI violations, the number of times that age exceeds a threshold value, on the receiver side. A<sup>3</sup>L-FEC is a communication solution based on "forward error correction" (FEC) and "user datagram protocol" (UDP) to optimize data freshness. The FEC mechanism that we examine for this study is based on the available AL-FEC (Application Layer FEC) methods. The existing AL-FEC mechanisms mainly concentrate on achieving objectives related to throughput and latency, which may not necessarily align with the purposes of optimizing the Age of Information (AoI). It is important to note that our proposed protocol stands apart by precisely targeting a metric based on AoI, prioritizing the freshness of data as a key objective. This significant difference distinguishes our protocol, as it places a direct emphasis on keeping optimal data freshness rather than solely optimizing traditional performance metrics such as throughput and latency.

Recently, the study in [91] proposed an adaptive packet coding scheme that uses random linear network codes. This scheme includes closed-form expressions of AoI as a function of tunable parameters, admission rate, and coding bucket size. Our suggested approach, which is presented in the following sections, displays several distinct differences compared to [91]. These primary differences can be summarized as follows:

- The proposal in [91] considers the network coding of a sequence of packets (i.e., the packets in the coding bucket) for in-order delivery of packets without data loss while our proposal considers sample-based coding, i.e. samples are coded individually, and loss of samples are allowed. Note that the state of the observed process can be recovered from any successfully decoded sample.
- The study in [91] idealizes the network, supposing that congestion is solely due to packets transmitted for end-to-end communication and that intermediate nodes only affect the packet loss rate. In contrast, our proposal includes a congestion control mechanism that dynamically adapts to the network based on received feedback.

Throughout the rest of this dissertation, we assumed a networked application for which a receiver monitors a process based on data received from a remote transmitter over an IP network. On the transmitter side, an optimal erasure code, i.e., a

maximum distance separable (MDS) erasure code, encodes each sample into  $n$  coded chunks such that correctly receiving any  $k$  coded chunks is necessary and sufficient for decoding the sample. On the receiver side, the successfully received coded chunks are stored and used to recover the corresponding samples. It follows by checking the age violation and delays of received chunks, then defining the transmission rate of the transmitter at the receiver side based on measured values. The receiver reports the transmission rate value to the transmitter periodically, at a defined time interval,  $\tilde{T}$  time slot, known as the "monitoring interval". This allows the transmitter to adjust the transmission rate accordingly.

In this study, two different flavors and implementations of A<sup>3</sup>L-FEC have been raised and tested, each with unique features and performance characteristics. For the scenarios where the block-length and the sampling rate are fixed, the A<sup>3</sup>L-FEC-FSFB, (Age Aware Application Layer Forward Error Correction - Fixed Sampling Rate Fixed Block-length), and for the scenarios with the variable sampling rate and variable block-length, the A<sup>3</sup>L-FEC-VSVB, (Age Aware Application Layer Forward Error Correction - Variable Sampling Rate Variable Block-length) are introduced. The primary approach of these two variations is the same, although they differ slightly in terms of sample generation and transmission intervals. The following sections provide further details on these differences.

The rest of this chapter is organized as follows: In Section 3.3, we begin by defining the system model and introducing the main concept behind A<sup>3</sup>L-FEC-FSFB. This is followed by explicitly explaining each block within the data flow control unit of A<sup>3</sup>L-FEC-FSFB in Section 3.3.1. Then the system model and data flow control algorithm of A<sup>3</sup>L-FEC-VSVB are presented in Section 3.4 and 3.4.1 respectively. Furthermore, theoretical limits of A<sup>3</sup>L-FEC flow control algorithm are discussed and manifested in Section 3.5. Section 3.6 delves into the innovative application of A<sup>3</sup>L-FEC in a Multi-Transmitter Multi-Receiver scenario. This section meticulously discusses the intricacies of its implementation, while this section also highlights the remarkable potential benefits it brings. Next, in the 3.7 section, we discussed the questions related to A<sup>3</sup>L-FEC and provided explanations to answer these questions. Finally, Section 3.8, outline potential areas for future work, summarizes the key insights, and concludes this chapter.



### 3.3 A<sup>3</sup>L-FEC Protocol with Fixed Sampling Rate and Fixed Block-length

From an application layer perspective for A<sup>3</sup>L-FEC-FSFB, the system under consideration is modeled as a time-slotted status update system, over an error-prone communication link (Figure 3.1). A receiver monitors a time-varying source process based on data received from a remote transmitter over a Packet Erasure Channel (PEC) with limited link availability. Let  $S$  represent a source process that consists of samples,  $\{s_\tau\}_{\tau \geq 0}$ , generated at regular intervals, i.e., equally separated time slots since the beginning of the communication session. During the course of this study, all time durations will be normalized using the duration of a single time slot as the reference.

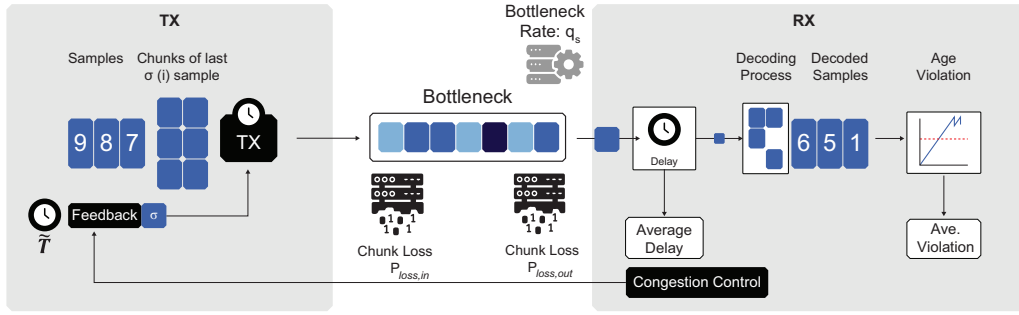


Figure 3.1: System model of a status update system over an error-prone link with a transmission rate feedback for the single-transmitter case.

At the transmitter side after sampling, the transmitter splits the sample of length  $K$  bits into  $k$  equal length data chunks and encodes the sample data chunks to  $n$  coded chunks of equal length  $\frac{K}{k}$  bits. The encoding is based on a maximum distance separable (MDS) code so the receiver can decode the codeword/sample whenever at least  $k$  coded chunks out of the  $n$  coded chunks are available. Let  $c_{i,\tau}$  denotes the  $i$ th coded chunk of the sample  $s_\tau$  which is generated at time  $\tau$ . The transmitter sends coded chunks to the receiver via UDP Protocol where one UDP packet is used per coded chunk, i.e., the payload of a UDP packet carries a coded chunk<sup>1</sup>. We assume that the transmitter can transmit multiple UDP packets in a time slot. Let  $\mathcal{T}_t$  denotes the set of chunks transmitted at time slot  $t$ , i.e.,  $c_{i,\tau} \in \mathcal{T}_t$ , where the  $i^{th}$  coded chunk of sample  $s_\tau$  is transmitted at time slot  $t$ . We assume that the transmitter buffers only the coded

<sup>1</sup> Together with the coded chunk, the payload also contains the index/timing information for identifying the chunk and cyclic redundancy check (CRC) data for error detection.

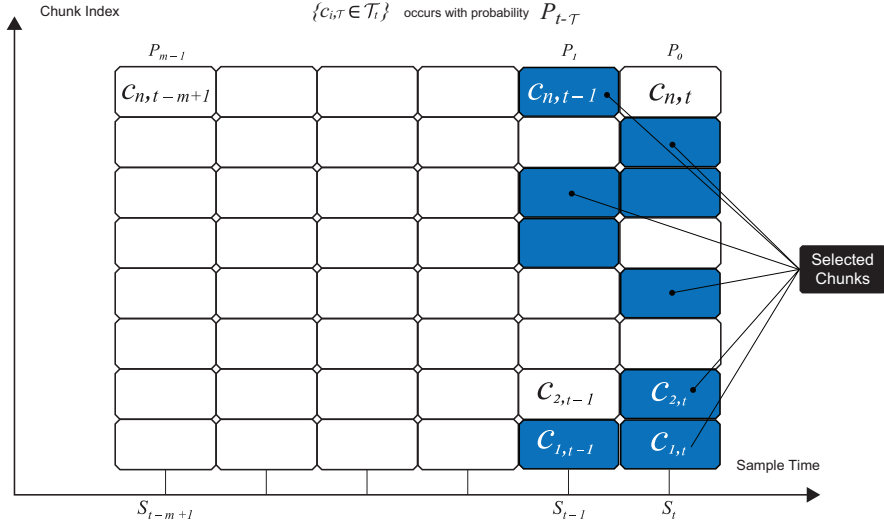


Figure 3.2: An illustration of the chunks selected for transmission with respect to their indices and time in A<sup>3</sup>L-FEC-FSFB protocol.

chunks from  $m$  freshest samples from the available coded chunks at a given time slot, i.e.,  $\mathcal{T}_t \subset \{c_{i,\tau} : i = 1, \dots, n, \tau = t - m + 1, \dots, t\}$ , Figure 3.2.

In this study in order to illustrate the performance of A<sup>3</sup>L-FEC-FSFB, the M/M/1 queuing model with first come first serve (FCFS) service discipline is used. The transmitter sends UDP packets at a rate  $\tilde{\sigma}$  [codeword/timeslot] to the receiver over a single infinite buffer bottleneck (queue), where the service rate of the queue is  $q_s$  [chunk/timeslot]. Each transmission of any chunk takes a constant time, which is equal to a duration of  $\frac{1}{q_s}$  time slot if there is no chunk ahead in the queue. It is assumed the UDP packet is dropped before entering the bottleneck with probability  $P_{loss,in}$  and also dropped before reaching the receiver with probability  $P_{loss,out}$ . Consequently, the probability that a UDP packet vanishes in the network is  $P_{loss,c}$  as follows:

$$P_{loss,c} = P_{loss,in} + (1 - P_{loss,in})P_{loss,out}. \quad (3.1)$$

Due to the bottleneck in the network, a random duration elapses before a UDP packet reaches the receiver from the transmitter. Let  $D_{t;i,\tau}$  denote the delay of the coded chunk  $c_{i,\tau}$  between its transmission and successful reception in case it is transmitted at time slot  $t$ . As mentioned previously, a UDP packet may be lost or corrupted. In that case, the delay will be considered as infinite, i.e.,  $D_{t;i,\tau} = \infty$ .

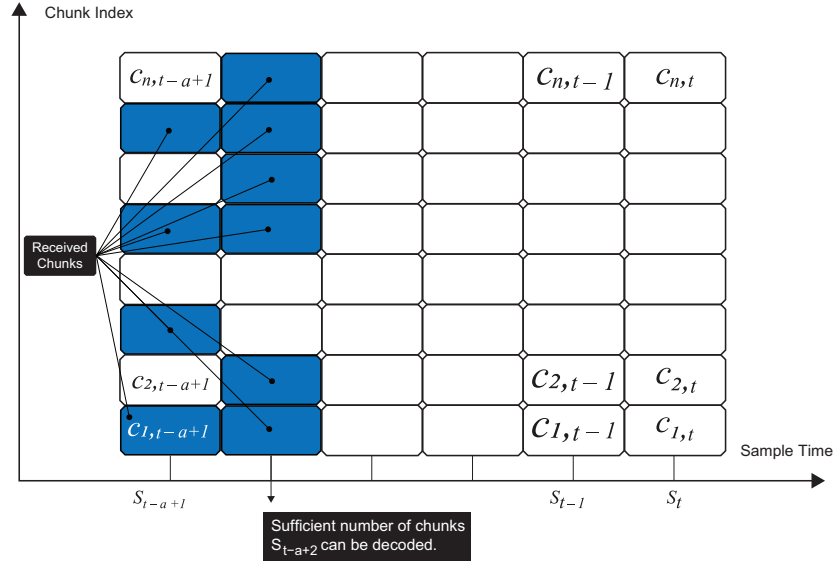


Figure 3.3: An illustration of the chunks that are successfully received with respect to their indices and time in A<sup>3</sup>L-FEC-FSFB protocol.

At the receiver side, the successfully received coded chunks are stored and used to recover the corresponding codeword, Figure 3.3. Let  $\mathcal{C}_t$  and  $\mathcal{S}_t$  denote the set of coded chunks that are successfully received by time  $t$  and the set of samples that can be decoded at time  $t$ , respectively. The event that a sample cannot be decoded at time  $t$  occurs if and only if there are more than  $n - k$  missing coded chunks for that sample:

$$\{s_\tau \notin \mathcal{S}_t\} = \left\{ \sum_{i=1}^n \mathbb{1}_{\{c_{i,\tau} \notin \mathcal{C}_t\}} > n - k \right\}, \quad (3.2)$$

where  $\mathbb{1}$  is the indicator of a missing coded chunk event and computed as follows:

$$\mathbb{1}_{\{c_{i,\tau} \notin \mathcal{C}_t\}} = \prod_{j=\tau}^t \left( \mathbb{1}_{\{c_{i,\tau} \notin \mathcal{T}_j\}} + \mathbb{1}_{\{c_{i,\tau} \in \mathcal{T}_j\}} \mathbb{1}_{\{D_{t;i,\tau} > t-j\}} \right). \quad (3.3)$$

The timeliness of the information at the receiver side is measured by AoI metric. AoI is defined as the number of time slots elapsed since the generation of the most up-to-date sample successfully decoded at the receiver side. This study uses the Age of Information to quantify the quality of monitoring experienced at the receiver side. The AoI at the receiver side is expressed as follows:

$$\Delta(t) = t - \max\{\tau \in \{1, 2, \dots, t\} : s_\tau \in \mathcal{S}_t\}. \quad (3.4)$$

Note that for a given age violation threshold value,  $AVT$  [timeslot], the objective of

the A<sup>3</sup>L-FEC protocol is to minimize the age violation metric,  $AV$ . Where (3.5) is a measure for the quality of monitoring the process  $S$ ; and  $T$  is the monitoring duration.

$$AV = \frac{1}{T} \sum_{t=1}^T \mathbb{1}_{\{\Delta(t) \geq AVT\}}, \quad (3.5)$$

To improve the quality of monitoring, we will consider a specific type of transmission policy known as *Stationary Independent Selection* (SIS) policies. This policy is designed to optimize the selection of samples for transmission in a way that ensures effective monitoring by providing fresh information at the receiver side.

**Definition 3.1** *A transmission policy is said to be a stationary independent selection policy if the events  $\{c_{i,\tau} \in \mathcal{T}_t\}$  are independent for all  $i, \tau$  and  $t$  such that  $\{c_{i,\tau} \in \mathcal{T}_t\}$  occurs with probability  $p_{t-\tau}$  where  $p_v = 0$  for  $v < 0$  and  $v > m - 1$ .*

According to this definition, the selection of samples to be transmitted is based on the principle of stationarity and independence. It means the happening of events, denoted as  $c_{i,\tau} \in \mathcal{T}_t$ , is considered independent for all  $i, \tau$ , and  $t$ . Moreover, in line with this definition a sequence of probabilities  $p_0, \dots, p_{m-1}$  identifies a SIS policy.

In our threshold-based AoI framework, it is necessary to highlight that transmitting samples containing obsolete status updates is unnecessary. As part of the framework, the transmitter actively filters out samples with generation times greater than  $AVT$  and exclusively transmits fresh status samples. Indeed by implementing SIS policies, A<sup>3</sup>L-FEC aims to optimize the selection of samples for transmission, enabling more effective and up-to-date data at the receiver side of the system. This policy offers a systematic approach to ensure the quality and accuracy of transmitted information, ultimately enhancing the overall freshness and reliability of the monitoring process.

Based on the provided explanations, when the expected transmission rate is a constant value denoted by  $\tilde{\sigma}$ , the following relationship can be established:

$$\sum_{j=0}^{m-1} p_j = \min(\tilde{\sigma}, AVT). \quad (3.6)$$

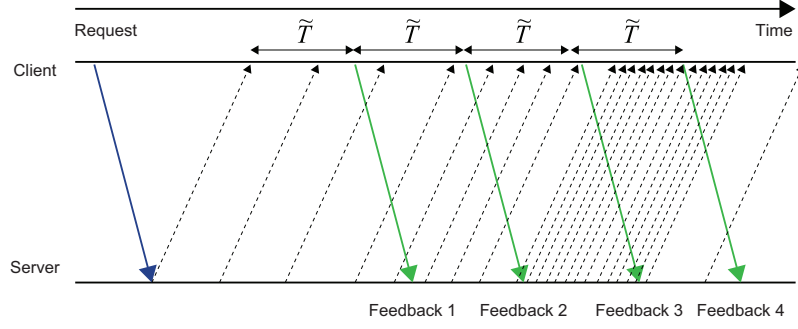


Figure 3.4: The signaling scheme for the congestion control mechanism of the A<sup>3</sup>L-FEC protocol when the server acts as a transmitter and the client acts as a receiver.

Let  $\Pi_{SIS}$  represent the set of all possible SIS policies. In our analysis, we will focus on addressing the following problem:

$$\min_{\pi \in \Pi_{SIS}} \limsup_{t \rightarrow \infty} \frac{1}{T} E \left[ \sum_{t=1}^T I_{\Delta(t) > AVT} \right] \quad (3.7)$$

In particular, for a given expected transmission rate  $\tilde{\sigma} = \min(\tilde{\sigma}, AVT)$ , the A<sup>3</sup>L-FEC-FSFB protocol chooses the  $\mathbf{p}^* = [p_0^*(\tilde{\sigma}) \dots p_{m-1}^*(\tilde{\sigma})]^T$  such that:

$$p_j^*(\tilde{\sigma}) = \begin{cases} 1 & \text{if } j \leq \lfloor \tilde{\sigma} \rfloor, \\ \tilde{\sigma} - \lfloor \tilde{\sigma} \rfloor & \text{if } j = \lfloor \tilde{\sigma} \rfloor + 1, \\ 0 & \text{if } j > \lfloor \tilde{\sigma} \rfloor + 1. \end{cases} \quad (3.8)$$

Where, at any time slot  $t$  the value  $p_j^*(\tilde{\sigma})$  indicates the probability of selecting any encoded chunks of the sample which is generated at time slot  $t - j$  for sending in time slot  $t$ . The  $\tilde{\sigma}$  value represents the number of codewords sent per time slot, while each codeword consists of  $n$  chunks.

As a part of the A<sup>3</sup>L-FEC, the transmission rate is computed at the receiver side and then reported to the transmitter regularly at every fixed  $\tilde{T}$  time slot as feedback. This feedback is generated based on the age violation, (3.9), and average chunk delays, (3.10). Details of this process is summarized in Figures 3.4 and 3.5. Note that  $\tilde{T}$  is known as the "monitoring interval" and  $MI$  is the monitoring interval number.

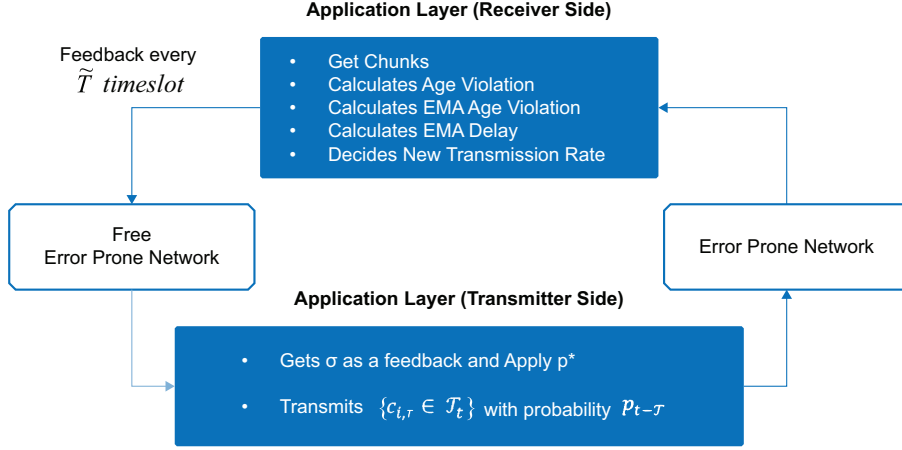


Figure 3.5: Congestion control scheme for the A<sup>3</sup>L-FEC-FSFB protocol.

The age violation and average chunk delays are evaluated based on the equations, 3.9 and 3.10 respectively. The age violation is represented by the  $AV_{MI}$ , which describes the average number of age violations over the  $MI^{th}$  monitoring interval. It is computed as the sum of indicator function  $\mathbb{1}_{\Delta(t) > AVT}$  for each time slot  $t$  within the monitoring interval, divided by the duration of the monitoring interval,  $\tilde{T}$ .

On the other hand,  $\bar{W}_{MI}$  represented the average chunk delay, which is calculated as the ratio between the  $L_1$  norm of the delay set,  $\mathcal{D}_{MI}$ , and the cardinality of  $\mathcal{D}_{MI}$ . This provides an average measure of the delay experienced by the transmitted chunks during the  $MI^{th}$  monitoring interval.

$$AV_{MI} = \frac{1}{\tilde{T}} \sum_{t=(MI-1) \times \tilde{T}}^{MI \times \tilde{T}} \mathbb{1}_{\{\Delta(t) > AVT\}}. \quad (3.9)$$

$$\bar{W}_{MI} = \frac{\|\mathcal{D}_{MI}\|}{|\mathcal{D}_{MI}|} \quad (3.10)$$

In (3.9) and (3.10),  $MI \in \{1, 2, 3, \dots, \lceil \frac{T}{\tilde{T}} \rceil\}$  and  $\mathcal{D}_{MI}$  denotes the set of delays of all chunks of all codewords that are successfully received by the end of  $MI^{th}$  monitoring interval.

### 3.3.1 Congestion Control Algorithm for the A<sup>3</sup>L-FEC-FSFB Protocol

In communication systems, each encoded packet injected into the network potentially improves data freshness at the receiver side, so it is in the receiver's interest that the transmitter increases its expected transmission rate. But when the number of packets injected into the network is a little high, the delay and the age characteristics start to become affected due to the congestion in the network or buffer bloats. Therefore, to deal with such consequences, the transmitter should adapt its expected transmission rate for filling the network just enough.

Based on Problem (3.7), the optimal solution (3.8) depends only on the  $\tilde{\sigma}$ , and as expected transmitter tries to send as new samples as possible. While, according to the solution, the packet delay statistics are not considered in the analyses. However, in practice, when the transmission rate is high, it affects the delay statistics due to the density of the packets in the network. For this reason, it is necessary to make decisions to update the transmission rate, but this update should be done adaptively as network statistics vary. This decision is based on the understanding of the relationship between transmission rate and changes in system age statistics, while it is influenced by several time-varying factors of the network such as bandwidth, packet loss ratio, and the traffic of other transmission streams.

Note that, in our algorithm, this adaptive update process is performed on the receiver side, because the receiver has access to these factors and can present more accurate decisions based on these factors.

The congestion control algorithm of A<sup>3</sup>L-FEC-FSFB is illustrated in Figure 3.6, includes an adaptive rate mechanism. This mechanism includes several blocks, each designed to perform a specific task in the algorithm. Some of these blocks are activated under urgent conditions. Conversely, the blocks within the green and red zones which represent the "Good state" and "Bad state," respectively, are triggered when the system is in regular condition.

In the following sections, the functioning of each block is elucidated in detail, providing a comprehensive understanding of their roles and contributions within the A<sup>3</sup>L-FEC-FSFB congestion control algorithm.

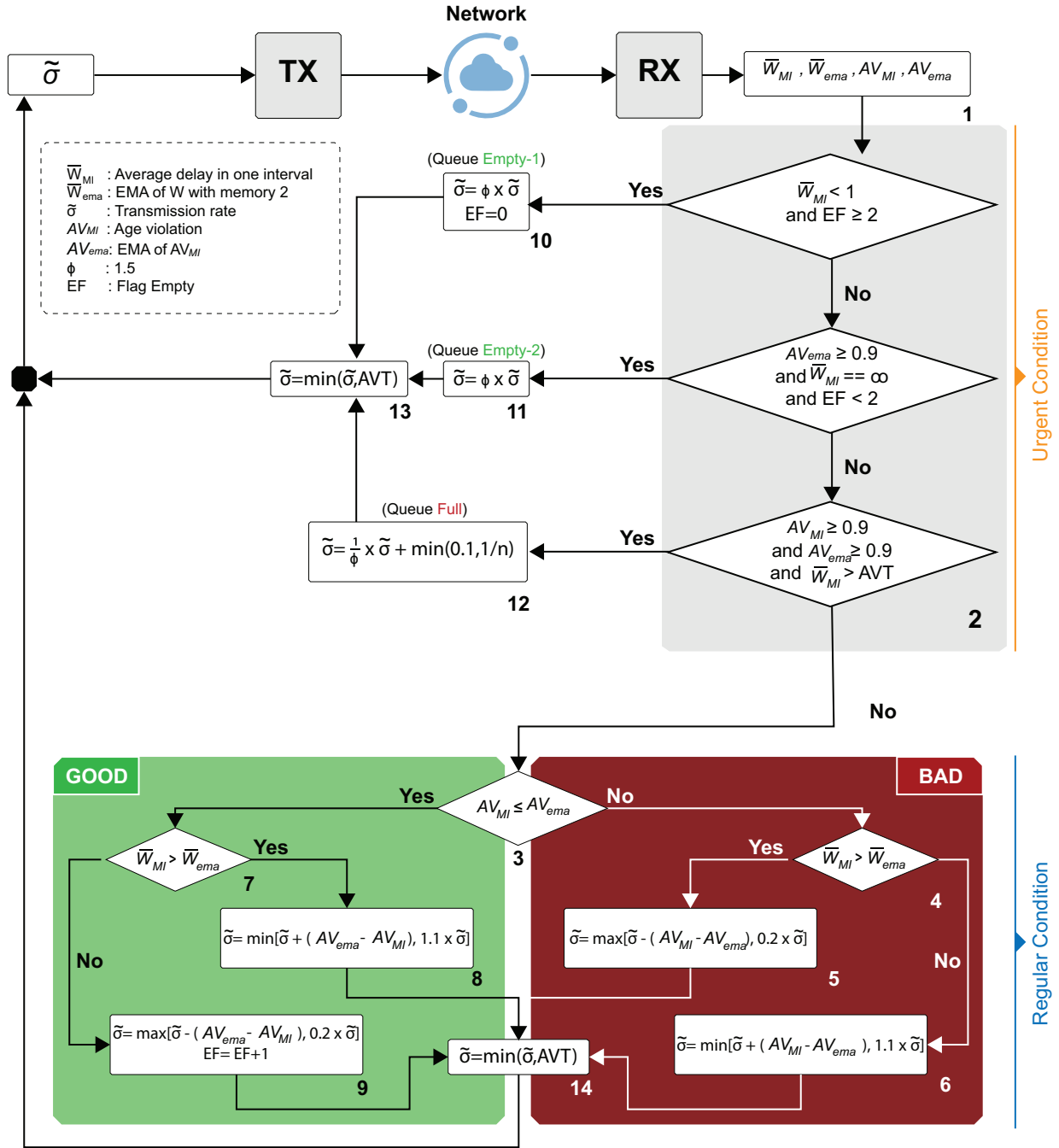


Figure 3.6: Update Algorithm for the expected transmission rate in the A<sup>3</sup>L-FEC-FSFB protocol.



- **Block 1**

This block has two tasks:

- The first task of this block is calculating the average delay (3.10),  $\bar{W}_{MI}$ , based on the received chunks in each monitoring interval and computing the exponential moving average (EMA) of delay (3.11),  $\bar{W}_{ema}$ , given the history of the last two monitoring intervals.

In (3.10), EMA is a moving average that places weight and significance on the most recent data points. The only decision a user of the EMA must make is the selection of the parameter  $\Psi$ . This parameter selection allows A<sup>3</sup>L-FEC-FSFB to adapt the EMA to its specific needs, striking a balance between the weight given to recently received data and the overall trend of the data.

For our case, we set it to 0.8 to contribute %80 of the last monitoring interval's chunk delays and %20 of the last two previous monitoring interval's chunk delays to generate  $\bar{W}_{ema}$  at the end of each monitoring interval. It means that the last monitoring interval's delay has a significant effect on  $\bar{W}_{ema}$ , while the system has a memory for just two last monitoring intervals.

$$\bar{W}_{ema} = \Psi \times \bar{W}_{MI} + (1 - \Psi) \times \bar{W}_{ema} \quad (3.11)$$

- The second task of this block is to calculate the age violation  $AV_{MI}$  by using 3.9 or (3.17), and the EMA of age violation ( $AV_{ema}$ ) based on the decoded samples in each monitoring interval, (3.12).

We set  $\Omega$  to 0.8 to contribute %80 of the last monitoring interval's age violation and %20 of the previous monitoring interval's age violations to generate ( $AV_{ema}$ ) at the end of each interval. It means that the last monitoring interval's age violation has a significant effect on ( $AV_{ema}$ ) while the ( $AV_{ema}$ ) has a memory of the age violation for whole times from the beginning of the transmission.

$$AV_{ema} = \Omega \times AV_{MI} + (1 - \Omega) \times AV_{ema} \quad (3.12)$$

## • Block 2

Based on “average delay” and “age violation,” three different conditions determine the state of this block of congestion control algorithm:

- If the average delay of the current interval is less than a one-time slot, ( $\bar{W}_{MI} < 1$ ), and the Empty Flag is greater or equal to two, ( $EF \geq 2$ ), it means at least in two consecutive previous intervals, the congestion control reduced the transmission rate in block 9, and average delay of the received chunks becomes less than a one-time slot. It implies that the queue in the bottleneck has been drained recently. Consequently, the age violation will increase soon. The best decision for congestion control is to increase the transmission rate immediately by going to block 10.
- If the age violation ratio at the last monitoring interval is above %90 of the times, ( $AV_{ema} \geq 0.9$ ), and the average delay of current monitoring interval is infinite, ( $\bar{W}_{MI} == \infty$ ), and the Empty Flag is less than two, ( $EF < 2$ ), it means that the violation happens in the system due to the lack of chunks in the network. The reason for that is no chunks in the system, therefore the average delay goes to infinite. As a result, the best strategy for the algorithm is to maximize network utilization by rapidly increasing the transmission rate. This objective can be accomplished by proceeding directly to block 11, where the network will be efficiently filled with fresh chunks.
- If for the current monitoring interval and for all duration from the beginning of the transmission the age violation happens in more than %90 of the times, ( $AV_{MI} \geq 0.9$ ) and ( $AV_{ema} \geq 0.9$ ), and furthermore if the average delay of current monitoring interval greater than  $AVT$  time slot, ( $\bar{W}_{MI} > AVT$ ), it means that the queue is full. Therefore the system should reduce the number of chunks by decreasing the transmission rate immediately by going to block 12.
- Else go to block 3

If none of the previous conditions are met, the flow proceeds to Block 3, which is responsible for comparing the average age violation. This block ensures that the transmission rate will be regulated properly.

- **Block 3**

The congestion control flow goes over this block most of the time. This block compares the portion of the time that age violations,  $(AV_{MI})$ , occur in the recent monitoring interval with respect to the EMA of age violations of the system from the start time of the transmission,  $(AV_{ema})$ . If the congestion control unit observes that the violation raised in the system, it pushes the flow toward block 4; otherwise, the flow will be pushed toward block 7 and in both scenarios based on average delay it will increase or decrease the transmission rate.

- **Block 4**

Since the age violation,  $(AV_{MI})$ , on the current monitoring interval is greater than the EMA of age violations,  $(AV_{ema})$ , the decision flow has come to this block. It means the system operates outside desirable limits. Now it has two choices:

- If the average delay,  $(\bar{W}_{MI})$ , in the current monitoring interval is greater than the EMA delay of the system,  $(\bar{W}_{ema})$ , it means there is congestion in the network, and the number of chunks in the bottleneck queue has increased. This is the reason for age violation in the system. Therefore the congestion control will decrease the transmission rate by pushing the decision flow toward block 5.
- If the average delay in the current monitoring interval,  $(\bar{W}_{MI})$ , is less than the EMA delay of the system,  $(\bar{W}_{ema})$ , it means the congestion in the network is reduced, and the number of chunks in the bottleneck queue has reduced. This is the reason for age violation in the system. Therefore the congestion control will increase the transmission rate by pushing the flow decision toward block 6 to provide fresh data at the receiver side.

- **Block 5**

This block reduces the transmission rate. First, the absolute value of the difference between the current interval's age violation and EMA of age violations is computed. Then this value is subtracted from the current transmission rate. Note that, in order to prevent the system from assigning a transmission rate of

zero, the minimum transmission rate for the next monitoring interval will be set at a value greater than or equal to 20% of the recent transmission rate.

- **Block 6**

This block is for increasing the transmission rate. First, the absolute value of the difference between the current interval's age violation and EMA of age violations is computed. This value is added to the current transmission rate. Note that to prevent sharp changes in the transmission rate, the maximum transmission rate for the next monitoring interval will be set to a value less than or equal to 110% of the current transmission rate.

- **Block 7**

Since the age violation, ( $AV_{MI}$ ), on the current interval is less than the EMA of age violations of the system, ( $AV_{ema}$ ), the flow of congestion control decision has been pushed to this block. It means the system is in the desirable operating regime or good condition. At this point, there are two choices:

- Suppose the average delay, ( $\bar{W}_{MI}$ ), in the current monitoring interval is greater than the EMA delay of the system, ( $\bar{W}_{ema}$ ). It means that although the transmission rate increased in the last monitoring interval and the number of chunks increased in the bottleneck queue, but this rate could not utilize the network capacity sufficiently. Therefore the system can probe for fewer age violations by increasing the transmission rate for the next monitoring interval by pushing the control flow toward block 8.
- Suppose the average delay, ( $\bar{W}_{MI}$ ), in the current monitoring interval is less than the EMA delay of the system, ( $\bar{W}_{ema}$ ). It means that reducing the transmission rate in the last monitoring interval had the desired effect on reducing the age violation. Therefore, in the following monitoring interval the congestion control unit will reduce the transmission rate again by pushing the control flow toward block 9.

- **Block 8**

This block increases the transmission rate. First, it calculates the absolute difference between the current interval's age violation and EMA of age violations.

Then this value is added to the recent rate. To prevent sudden fluctuations, the maximum rate for the next monitoring interval will be set to a value that does not exceed 110% of the current transmission rate.

- **Block 9**

This block reduces the rate. First, it calculates the absolute value difference between the current interval's age violation and EMA of age violations. This value is subtracted from the current transmission rate. To avoid assigning a transmission rate of zero, the minimum acceptable transmission rate for the next monitoring interval will be selected as 20% of the current transmission rate. In this block after each rate reduction, the EF increases by one unit. The EF shows how many times the rate is decreased for probing a better age violation quantity when the system is in "Good Condition". The EF is used in block 2.

- **Block 10**

This block is used in urgent situations when there is no chunk in the network (queue empty), and the congestion control algorithm reduced the transmission rate more than one time in the previous monitoring intervals, ( $EF \geq 2$ ). Therefore it is concluded that there are near zero chunks in the bottleneck. Under the operation of this block, the new transmission rate will be " $\Phi$ " times of previous monitoring interval's transmission rate, where we select  $\Phi = 1.5$ . It means the rate is increased by 50% for the next monitoring interval. Furthermore,  $EF$  is reset to 0 after the rate increase.

- **Block 11**

This block is used in urgent situations when the queue is empty, the average delay of chunks in the current interval is infinite, and  $EF$  is less than 2; while the age violation has occurred more than 90% of the times in the current monitoring interval. This implies that there is a lack of chunks in the network. Under the operation of this block, in the following monitoring interval, the new transmission rate will be " $\Phi$ " times of previous monitoring interval's transmission rate, where we select  $\Phi = 1.5$ . It means the rate is increased by 50% for the next monitoring interval. This block usually is used at the beginning of the transmissions to have a multiplicative increase in the transmission rate.

- **Block 12**

This block is used in urgent situations when the queue is Full. If age violation occurs in the current monitoring interval %90 of the time, the EMA of age violation be also greater than 0.9 and the average delay be greater than a value such as "AVT", which means that the queue is full, therefore this block is activated.

This block reduces the number of chunks by decreasing the transmission rate. The new transmission rate will be " $\frac{1}{\Phi}$ " times the previous monitoring interval transmission rate, plus a small value like  $\min(0.1, \frac{1}{n})$ .

The presence of the expression  $\min(0.1, \frac{1}{n})$  may raise questions for the reader. To address these questions, the answer is as follows: If  $(\frac{1}{\Phi} \times \tilde{\sigma})$  goes to zero and  $\min(0.1, \frac{1}{n})$  does not exist, the algorithm can not detach the transmission rate from small values near zero easily. Therefore reaching to suitable transmission rate after visiting the near zero transmission rate, will take a long time, or it will remain near zero in certain cases. Hence,  $\min(0.1, \frac{1}{n})$  acts as the minimum transmission rate when the  $(\frac{1}{\Phi} \times \tilde{\sigma})$  goes to zero. It should be noted that in scenarios when A<sup>3</sup>L-FEC-FSFB selects a high transmission rate, the term  $\min(0.1, \frac{1}{n})$  has minimal impact on the overall transmission rate.

- **Block 13 and 14**

Note that in the A<sup>3</sup>L-FEC-FSFB protocol, in each time slot a sample is generated. The age of the generated sample increases by one with each subsequent time slot. As explained previously, the transmission rate, denoted as  $\tilde{\sigma}$ , defines the set of selected samples for transmission. Based on this knowledge, it is not reasonable to transmit the codeword of a sample whose generation time exceeds the predefined age violation threshold (AVT).

Therefore, in blocks 13 and 14 of the protocol, a simple comparison is made between the age violation threshold value and the transmission rate which is determined by one of the blocks, specifically blocks 6, 9, 10, 11, or 12. Then the minimum of the age violation threshold and the transmission rate is selected as the new transmission rate for the next monitoring interval. This mechanism ensures that only fresh and up-to-date samples are transmitted, avoiding the transmission of outdated information.

### 3.4 A<sup>3</sup>L-FEC Protocol with Variable Sampling Rate and Variable Block-length

Just like A<sup>3</sup>L-FEC-FSFB, which is introduced in Section 3.3, A<sup>3</sup>L-FEC-VSVB also considers a time-slotted status update system over an error-prone communication link. The receiver monitors a time-varying process by collecting data transmitted from a remote transmitter. However, this transmission occurs over a Packet Erasure Channel (PEC) with limited link availability, which may result in incomplete or missing data.

The A<sup>3</sup>L-FEC-VSVB is designed with the assumption that the transmitter operates under the "**Generate at will**" model, which means the transmitter generates packets as needed and then sends them. This model allows for greater flexibility in the transmission process, as the source can adjust the rate of packet generation based on the current network conditions. Thus, the network is not overloaded with data; otherwise, it will increase the delay, packet loss, and ultimately the unfavorable AoI status. Let  $S$  denote a source process consisting of samples  $\{s_\tau\}_{\tau \geq 0}$ , which are generated at regular intervals,  $T_s$ , during each monitoring interval. However, it should be noted that in this flavor of the A<sup>3</sup>L-FEC, the sample generation interval can vary from one monitoring interval to the next. This protocol also splits the sample of length  $K$  bits into  $k$  equal length chunks and encodes the chunks by using the maximum distance separable (MDS) code to  $n$  chunks of equal length  $\frac{K}{k}$  bits. Indeed this coding enables the receiver to decode the codeword by receiving any combination of at least  $k$  chunks out of the  $n$  total chunks that make up the code. It means that even if some of the chunks are lost or corrupted during transmission, the receiver can still reconstruct the original message using the remaining chunks. This approach provides a high level of reliability and robustness in data transmission, making it an ideal choice for applications where data loss or corruption could have serious consequences.

As shown in Figure 3.7, during each monitoring interval, the transmitter generates a sample every  $T_s$  time slot, encodes it, and sends all  $n$  chunks of that sample immediately after generation. Therefore the transmission rate in every monitoring interval is  $\frac{n}{T_s}$ . Note that the A<sup>3</sup>L-FEC-VSVB communication solution is equipped with a unique variable code-length feature, such that the block-length may change from one monitoring interval to the next. This ensures that the system can adapt to changes in the transmission environment, allowing for optimized data freshness at the receiver side.

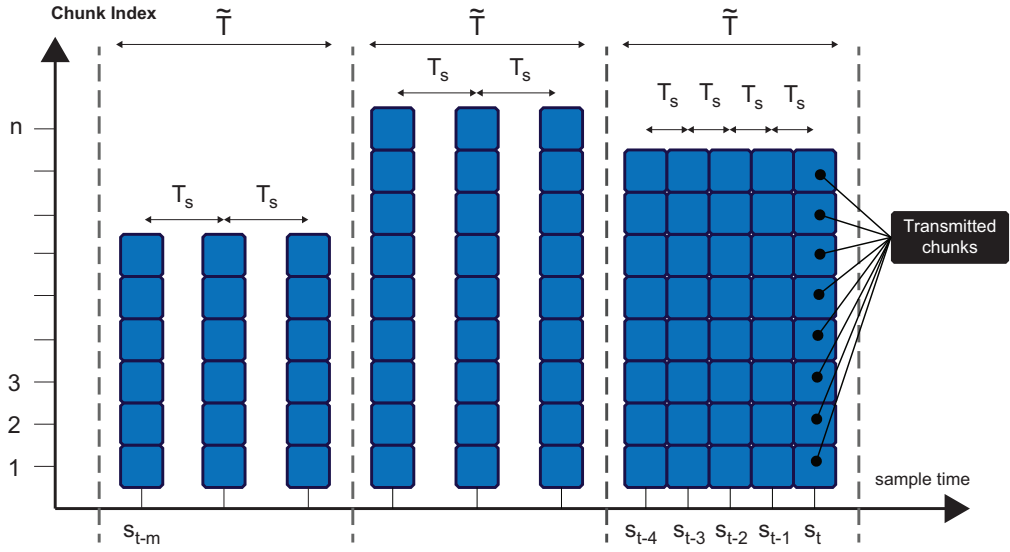


Figure 3.7: A illustration of the chunks that are selected for transmission with respect to their indices and time in A<sup>3</sup>L-FEC-VSVB protocol.

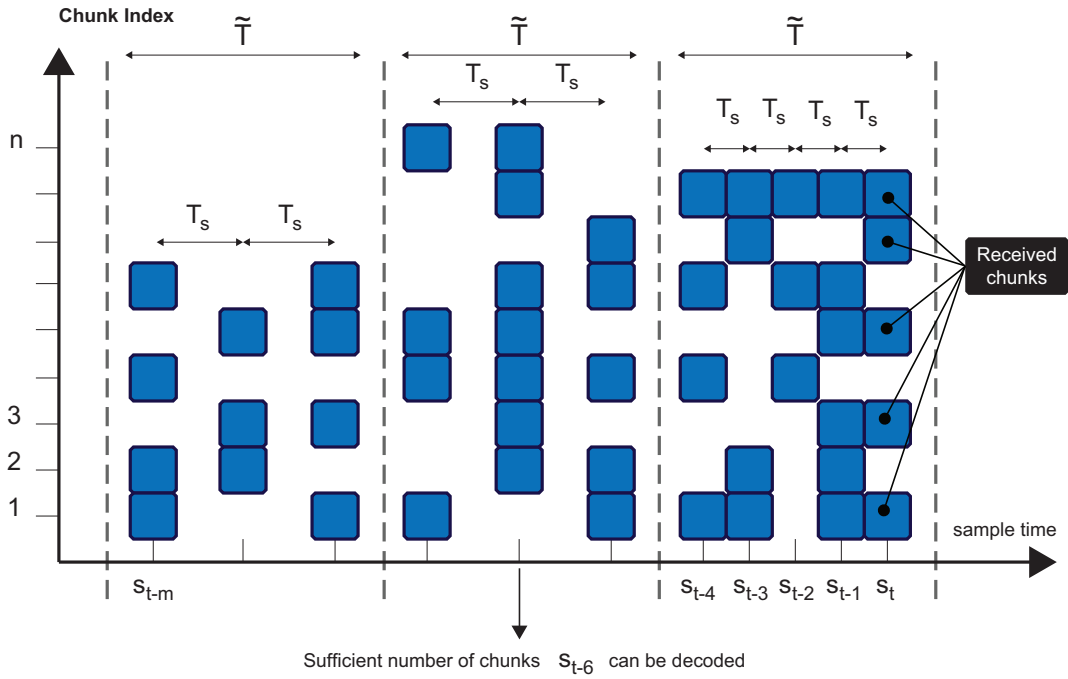


Figure 3.8: An illustration of the chunks that are successfully received with respect to their indices and time in A<sup>3</sup>L-FEC-VSVB protocol.



The transmitter sends coded chunks to the receiver via User Datagram Protocol (UDP) where one UDP packet is used per coded chunk, i.e., the payload of a UDP packet carries a coded chunk. We assume that the transmitter can transmit  $n$  UDP coded chunks in a time slot. Let  $c_{i,\tau}$  denotes the  $i$ th coded chunk of the sample  $s_\tau$  and  $\mathcal{T}_t$  denotes the set of packets transmitted at time slot  $t$ , i.e.,  $c_{i,\tau} \in \mathcal{T}_t$  the  $i^{th}$  coded chunk of sample  $s_\tau$  is transmitted at time slot  $t$ . The coded chunks that were received at the receiver side are stored and used to recover the corresponding codeword, Figure 3.8.

**At the beginning of each monitoring interval**, the A<sup>3</sup>L-FEC-VSVB creates two lists to store the decode time and the generation time of decoded samples. These lists are initialized with the last sample's decode and generation time from the previous monitoring interval. As new samples are decoded during the current monitoring interval, their decode time and generation time are appended to the respective lists. As a result, each list will have a length of  $C_{MI} + 1$  by the end of the monitoring interval;  $C_{MI}$ , (3.13) is the total number of received chunks in the  $MI^{th}$  monitoring interval:

$$C_{MI} \in \{0, 1, \dots, n \times \lceil \frac{\tilde{T}}{T_s} \rceil\} \quad (3.13)$$

**At the end of each monitoring interval**, the following five main tasks are performed:

**Task 1:** By entering the sample's generation and decode time lists as the inputs of the equations (3.14), (3.15) and (3.16), then substituting their outputs in equations (3.17), the age violation for the  $MI^{th}$  monitoring interval which is represented by  $AV_{MI}$  is calculated. This value allows the system to determine whether the age of the data being transmitted violates the AoI threshold for the given application or not. Indeed, by keeping track of the decode time and generation time of the decoded samples, the system is able to accurately and precisely calculate the age violation by using (3.17) instead of employing (3.9) which is introduced in the Section 3.3.

In the following equations which are employed for computing the  $MI^{th}$  monitoring interval's age violation,  $ST_{MI}$  denotes the start time of the monitoring interval,  $AVT$  represents the age violation threshold of the system, while  $G_i$  and  $D_i$  correspond to the  $i^{th}$  element in the recorded generation and decode time lists, respectively.

let:

$$\alpha = -\left| \min\left\{(ST_{MI} - G_1), \max\{(ST_{MI} - G_1 - AVT), 0\}\right\} \right| \quad (3.14)$$

$$\beta_i = D_i - D_{i-1} \quad (3.15)$$

$$\gamma_i = AVT - (D_{i-1} - G_{i-1}) \quad (3.16)$$

By applying  $\alpha$ ,  $\beta_i$ , and  $\gamma_i$  in (3.17), the  $MI^{th}$  monitoring interval's age violation is computed:

$$AV_{MI} = \alpha + \sum_{i=2}^{C_{MI}} \min\left\{\beta_i - \max\left\{0, \min\{\beta_i, \gamma_i\}\right\}, \beta_i\right\}. \quad (3.17)$$

**Task 2:** Receiver computes the average chunk delay ( $\bar{W}_{MI}$ ) and the Packet Delivery Ratio ( $PDR_{MI}$ ) of transmitted chunks using the equations (3.18) and (3.19), where in (3.18)  $W_i$  is the delay of  $i^{th}$  received chunks and in (3.19)  $\tilde{T}$  is the length of the monitoring interval.  $\bar{W}_{MI}$  and  $PDR_{MI}$  metrics provide important information about the performance of the algorithm for the  $MI^{th}$  monitoring interval, such as the average delay experienced by chunks and the percentage of successfully delivered chunks. Indeed by computing these metrics, the A<sup>3</sup>L-FEC-VSVB can gain insights into its own performance and make suitable adjustments on the transmission rate as necessary to improve efficiency and minimize age violations for the next monitoring interval. The adjustment of the transmission rate is done through the use of the congestion control algorithm, which is thoroughly explained in Section 3.4.1.

$$\bar{W}_{MI} = \begin{cases} \infty & \text{if } C_{MI} = 0 \\ \frac{\sum_{i=1}^{C_{MI}} W_i}{C_{MI}} & \text{if } C_{MI} \neq 0 \end{cases} \quad (3.18)$$

$$PDR_{MI} = \frac{C_{MI}}{n \times \left\lceil \frac{\tilde{T}}{T_s} \right\rceil} \quad (3.19)$$

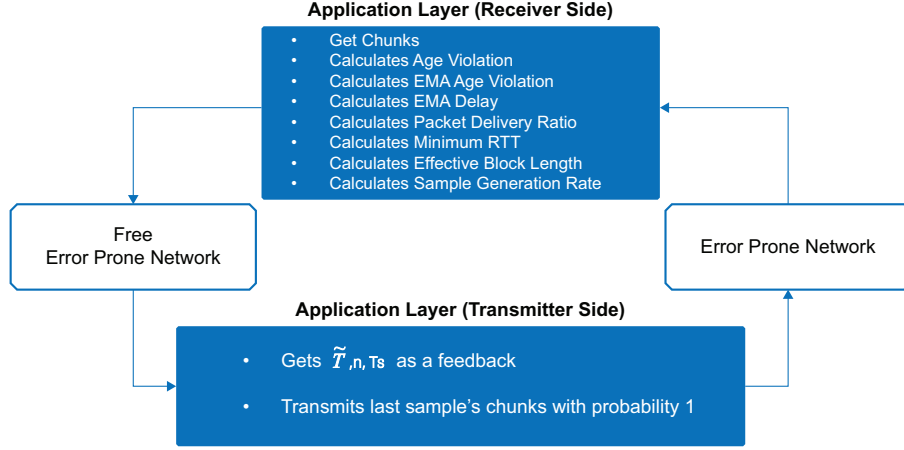


Figure 3.9: Congestion control scheme for the A³L-FEC-VSVB protocol.

**Task 3:** In this task it is assumed that the end-to-end packet delay and loss characteristics in the network will remain relatively unchanged between two adjacent monitoring intervals. In light of this assumption, A³L-FEC-VSVB calculates all possible age violation values for different possible block-lengths based on the number of received chunks and their corresponding reception times. Then, Among the possible block-lengths selects the one which could result in the minimum age violation as the effective block-length,  $n$ , for the subsequent monitoring interval. A detailed explanation of this concept is provided in section 3.4.1.

**Task 4:** The fourth task is performed at the end of each monitoring interval. In this task the algorithm determines the optimal sample generation interval,  $T_s$ , for the next monitoring interval using the expression (3.20). Where for the best transmission interval, the next monitoring interval's optimal transmission rate and the effective block-length are considered in equation (3.20).

$$T_s = \frac{n}{\sigma} \quad (3.20)$$

**Task 5:** Once the effective block-length ( $n$ ) and sample generation interval ( $T_s$ ) have been computed, they are conveyed as feedback to the transmitter. It is assumed that feedback is transmitted over an error-free and delay-free channel, as illustrated in Figure 3.9. By transmitting this information in a reliable and timely manner, the transmitter can adjust its operations to maintain optimal performance.

### 3.4.1 Congestion Control for the A<sup>3</sup>L-FEC-VSVB Protocol

The congestion control algorithm in A<sup>3</sup>L-FEC-VSVB is similar to A<sup>3</sup>L-FEC-FSFB. However, there are some additional metrics that the A<sup>3</sup>L-FEC-VSVB uses to determine the transmission rate at the end of each monitoring interval. The A<sup>3</sup>L-FEC-VSVB's adaptive flow control algorithm, as shown in Figure 3.10, is designed to use an appropriate portion of the communication network's capacity while avoiding congestion and have minimum age violation. To achieve this purpose, the algorithm employs a calculation of the effective block-length, denoted as  $n$ , along with the determination of the optimal transmission rate and an appropriate sample generation interval, denoted as  $\sigma$  and  $T_s$  respectively.

Note that in this study the length of each monitoring interval is  $\tilde{T}$  time slot and propagation delay is assumed negligible compared to the service time of the queue for each packet.

The A<sup>3</sup>L-FEC-VSVB's congestion control algorithm consists of several distinct blocks, each serving a specific purpose in regulating network congestion. By employing these interconnected blocks, the algorithm ensures that the network can effectively manage congestion, maintain efficient data transmission, and ultimately deliver fresh data to the destination. These blocks, as depicted in Figure 3.10, are explained as follows:

- **Block 1**

This block has the following tasks:

- **The first task** of this block is calculating the average delay,  $\bar{W}_{MI}$ , (3.10), and the EMA of delay,  $\bar{W}_{ema}$ , (3.11), with a history of the last two monitoring interval, based on the received chunks in each monitoring interval. EMA is a type of moving average that puts more weight and importance on the latest data points by choosing an appropriate value for the  $\Psi$  parameter. For our case, we set it to 0.8 to contribute %80 of the last monitoring interval's chunk delays and %20 of the last two previous monitoring interval's chunk delays to generate  $\bar{W}_{ema}$ . It implies that the last monitoring interval's delay has a significant effect on  $\bar{W}_{ema}$  while the system has a memory for just two last monitoring intervals.

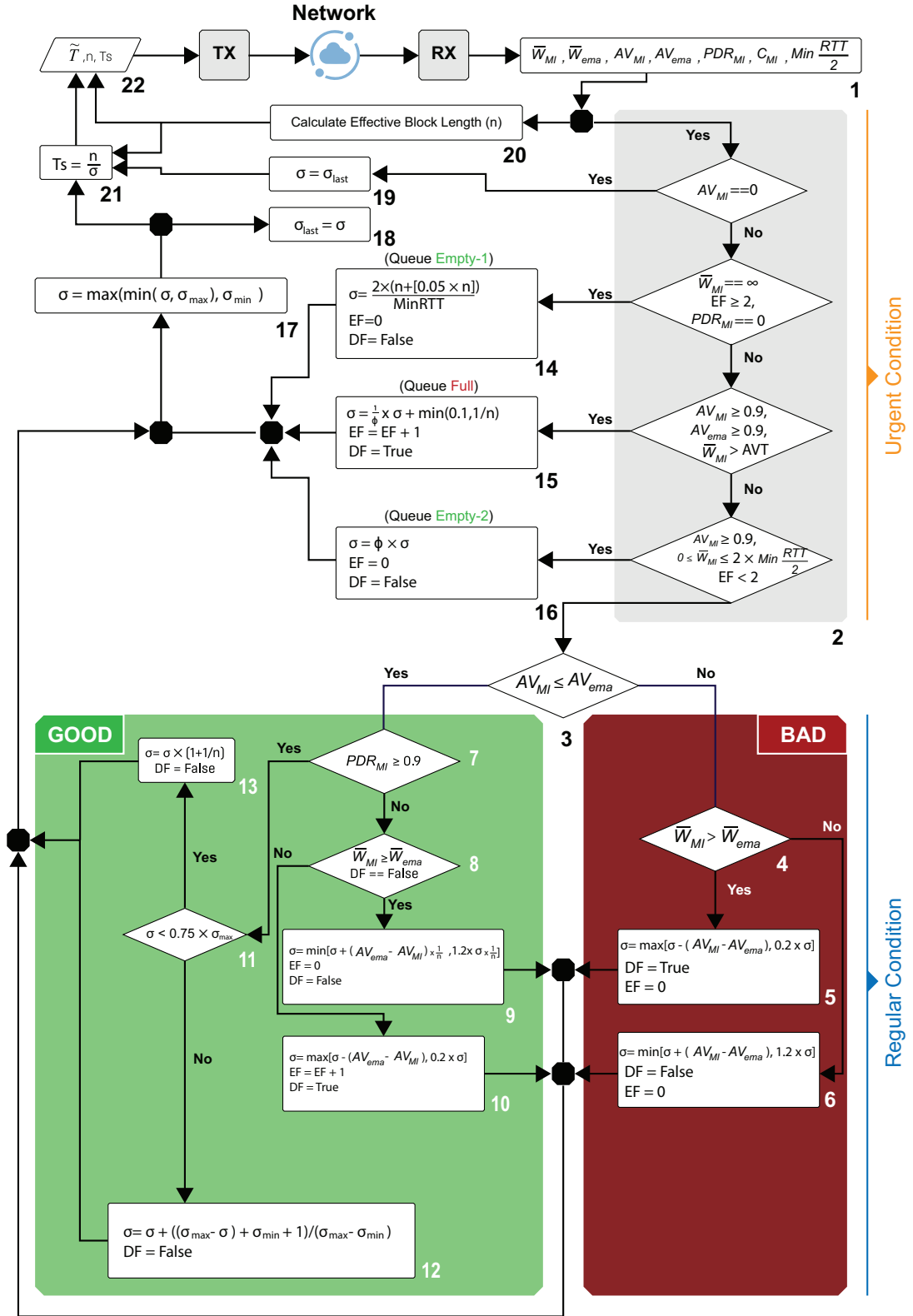


Figure 3.10: Update algorithm for the expected packet transmission rate in the A<sup>3</sup>L-FEC-VSVB protocol.

- **The second task** of this block is calculating the age violation,  $AV_{MI}$ , by using (3.17) and the EMA of age violation,  $AV_{ema}$ , (3.12), based on the decoded samples in each monitoring interval. We set  $\Omega$  to 0.8 to contribute %80 of the last monitoring interval's age violation and %20 of all the previous monitoring interval's age violations to generate ( $AV_{ema}$ ) at the end of each interval. It means that the last monitoring interval's age violation has a significant effect on ( $AV_{ema}$ ) while the ( $AV_{ema}$ ) has a memory of the age violation for whole times from the beginning of the transmission.
- **The third task** of this block is counting and storing the number of received packets in each monitoring interval. By implementing this counting and storage mechanism, the A<sup>3</sup>L-FEC-VSVB's algorithm gains valuable insights into the reception performance and overall data integrity. It allows for a comprehensive analysis of the received packets, assisting in determining any potential issues or anomalies during the transmission session. In other words, the number of stored received packets serves as a vital identifier for estimating the reliability of the communication channel.
- **The fourth task** of this block involves the measurement and storage of the minimum  $\frac{RTT}{2}$ . The purpose of this task is to capture and track the  $\frac{RTT}{2}$ , which represents the time taken for a packet to travel from the transmitter to the receiver. This is accomplished by examining the received packets throughout the entire transmission session. Storing the minimum  $\frac{RTT}{2}$  value is critical as it supplies valuable insights into the efficiency and performance of the network. It acts as a benchmark for evaluating the responsiveness and latency of the communication channel.
- **The fifth task** within this block is estimating the packet delivery ratio ( $PDR_{MI}$ ), as depicted in (3.19). The  $PDR_{MI}$  is defined by comparing the number of received packets within  $MI^{th}$  monitoring interval with the expected received packet (ERP), which is outlined in equation (3.21).

To compute the ERP, three key parameters come into play:

- . The block length, denoted by  $n$ .
- . The duration of the monitoring interval, represented by  $\tilde{T}$ .
- . The sample generation interval, is indicated as  $T_s$ .

By using these parameters and equation (3.21), the algorithm calculates the expected number of packets that have to be received during a specific monitoring interval.

$$ERP = n \times \left\lceil \frac{\tilde{T}}{T_s} \right\rceil \quad (3.21)$$

- **Block 2**

This block uses *average delay*, *age violation*, *PDR*, *AVT*, *minimum  $\frac{RTT}{2}$* , and *Empty Flag* values. Based on these parameters four different decisions define the state of the congestion control block as follows. Note that the Empty Flag, (EF), shows how many times the transmission rate was decreased for probing a better age violation quantity when the system was in Good Condition.

- **Decision one:** For the situations when the current monitoring interval does not result in an age violation ( $AV_{MI} == 0$ ), it implies that the selected block-length and sampling period have fulfilled the age-aware application's requirements. This achievement is noteworthy since it indicates that the algorithm has successfully maintained the freshness of the transmitted data at the destination. Thus, by assuming that the end-to-end packet delay and loss behavior remain relatively unchanged between two consecutive monitoring intervals, the transmission rate for the next interval can be set to the recent transmission rate,  $\sigma$ . This approach ensures consistency in the system's transmission rate, allowing the algorithm to maintain optimum performance while avoiding unnecessary fluctuations. In summary, if there are no instances of age violation during the current monitoring interval, there is no necessity to alter the transmission rate in the subsequent monitoring interval.
- **Decision two:** In the scenario when the average delay of the current interval is infinite ( $\bar{W}_{MI} == \infty$ ), the Empty Flag (EF) is greater than or equal to two ( $EF \geq 2$ ), and the packet delivery ratio is zero ( $PDR_{MI} == 0$ ), the status indicates that the queue at the bottleneck is empty and there are no chunks waiting to be processed; because the algorithm has reduced the transmission rate in the last two consecutive intervals by passing through block 15 or 10 of the algorithm. As a result, the average age and average

delay tend towards large and infinite values respectively. Note that when the algorithm does not capture any packet during a specific monitoring interval, it sets the average delay to infinity, ( $\bar{W}_{MI} = \infty$ ). This increase leads to a subsequent rise in age violation of the system. At this point the best strategy for congestion control is to increase the transmission rate by going to block 14.

- **Decision three:** If for the current monitoring interval, the following events are observed simultaneously, it means the queue is full and the age of the system is greater than the threshold value ( $AVT$ ). Therefore the system should reduce the number of packets by decreasing the transmission rate immediately by going to block 15.

- . If the age violation happens in more than %90 of the times in current monitoring interval, ( $AV_{MI} \geq 0.9$ ).
- . If EMA of the age violation shows that the violation happened in more than %90 of the times ( $AV_{ema} \geq 0.9$ ).
- . If the average delay of the current monitoring interval is greater than the age violation threshold, ( $\bar{W}_{MI} > AVT$ ).

Hence, as previously mentioned, the system needs to reduce the number of packets by promptly decreasing the transmission rate, effectively pushing the decision toward block 15.

- **Decision four:** If for the current monitoring interval, the age violation happens in more than %90 of the observation times, ( $AV_{MI} \geq 0.9$ ), and the average delay of the current monitoring interval is greater than or equal to twice of the minimum  $\frac{RTT}{2}$ , ( $\bar{W}_{MI} \geq 2 \times \text{Min} \frac{RTT}{2}$ ), while the Empty Flag (EF) is less than or equal to 2, ( $EF < 2$ ), it represents that the violation happens in the system due to the lack of the sufficient number of packets in the network. Consequently, the system should inject additional packets into the network by increasing the transmission rate immediately, which can be achieved by visiting block 16.

- **Decision five:** Else go to block 3

If none of the previous conditions occurs, the flow goes over this block to block 3 to control the transmission rate.



- **Block 3**

The congestion control flow goes over this block most of the time to control the transmission rate. This block compares the portion of the time that age violations,  $(AV_{MI})$ , occur in the last recent monitoring interval with respect to the EMA of age violations of the system,  $(AV_{ema})$ , from the beginning of the transmission session.

When the A<sup>3</sup>L-FEC-VSVB's congestion control algorithm detects an increase in age violation event, it directs the decision flow towards block 4, representing a "Bad Condition." Conversely, if the age violation event is not observed to have increased, the decision flow is directed toward block 7, indicating a "Good Condition." In both possibilities, the algorithm modifies the transmission rate based on the average delay; either increasing or decreasing it accordingly.

- **Block 4**

Since the age violation,  $(AV_{MI})$ , on the current monitoring interval is greater than the EMA of age violations of the system,  $(AV_{ema})$ , the flow of congestion control decision has been pushed to this block. It means the system is in "Bad Condition". Now, the system is faced with two options:

- **Option 1:** If the average delay,  $(\bar{W}_{MI})$ , in the current monitoring interval is greater than the EMA delay of the system,  $(\bar{W}_{ema})$ , it means there is congestion in the network, and the number of packets in the bottleneck has been increased. Based on this analysis the congestion control mechanism will take action to handle the age violation issue by reducing the transmission rate. This adjustment pushes the decision flow towards block 5, where appropriate modifications on the transmission rate can be taken to maintain more optimal system performance.
- **Option 2:** If the average delay in the current monitoring interval,  $(\bar{W}_{MI})$ , is less than the EMA delay of the system,  $(\bar{W}_{ema})$ , it means the congestion in the network is reduced, and the number of packets in the bottleneck has been reduced. This is the reason for age violation in the system. Therefore the algorithm will increase the transmission rate by pushing the flow decision toward block 6.

- **Block 5**

This block is responsible for reducing the transmission rate. To achieve this, it executes the following steps:

First, it calculates the absolute value of the difference between the age violation of the current monitoring interval and the EMA of age violations. Next, the calculated value is subtracted from the current transmission rate, and the resulting value becomes the transmission rate for the following monitoring interval.

It is worth noting that to avoid sudden and drastic changes to the transmission rate, a minimum transmission rate for the next monitoring interval is enforced. This minimum rate is set to be greater than or equal to %80 of the recent transmission rate. By implementing this method, the system guarantees a gradual and controlled reduction in the transmission rate, preventing abrupt fluctuations and maintaining a more stable and consistent flow of data on the network.

- **Block 6**

This block has the duty of increasing the transmission rate. It executes the following steps:

First, it computes the absolute value of the difference between the age violation of the current monitoring interval and the EMA of age violations. Next, the calculated value is added to the recent transmission rate.

It is noteworthy to know that to provide smooth and controlled adjustments to the transmission rate, a limit-imposed feature is considered in the algorithm. Such that, the maximum transmission rate for the subsequent monitoring interval is set to be less than or equal to %120 of the current transmission rate. In summary, by implementing this feature, the system allows for gradual and controlled increments in the transmission rate, preventing sudden spikes and maintaining a more stable and consistent flow of data.

- **Block 7**

If the age violation, ( $AV_{MI}$ ), on the current interval is less than the EMA of age violations of the system, ( $AV_{ema}$ ), the flow of congestion control decision is pushed toward this block. It indicates that the system is in a favorable state or "Good Condition". At this point, the system has two options:

- **Option 1:** If the Packet Delivery Ratio ( $PDR_{MI}$ ) exceeds 0.90, meaning that at least 90% of the expected packets were received within the current interval; therefore it offers an opportunity to probe improved conditions. In such cases, the increase in transmission rate becomes applicable and it is possible to use more network capacity. This, in turn, propels the decision flow towards block 11, facilitating the potential for boosted performance and efficiency.
- **Option 2:** If the Packet Delivery Ratio ( $PDR_{MI}$ ) is less than 0.90, it means that there have been packet losses during the current monitoring interval. It is possible that the system may experience "Bad Conditions" in the upcoming monitoring interval. Therefore, it is important to re-evaluate and modify the transmission rate by comparing the average delay,  $\bar{W}_{MI}$ , and the EMA delay,  $\bar{W}_{ema}$ . Where this valuable comparison provides insight into the trend of packet delays within the network. Therefore, when option 2 is activated, the control flow moves to block 8.

#### • Block 8

In this block, based on the average delay level two scenarios can be evaluated:

- Suppose a scenario where the average delay, ( $\bar{W}_{MI}$ ), in the current monitoring interval, is higher than the EMA delay of the system, ( $\bar{W}_{ema}$ ), and there was no rate decrease action in previous monitoring interval ( $DF == False$ ). This scenario indicates that despite the increase in the transmission rate during the last monitoring interval, resulting in a larger number of packets in the bottleneck queue, the chosen transmission rate was unable to sufficiently utilize the network capacity. Consequently, the A<sup>3</sup>L-FEC-VSVB can take action to reduce age violations by increasing the transmission rate for the next monitoring interval, thus shifting the control flow towards block 9. By doing so, the system aims to optimize the utilization of network capacity and reduce the occurrence of age violations by delivering more fresh data to the destination.
- Suppose a scenario where the average delay ( $\bar{W}_{MI}$ ) during the current monitoring interval is lower than the EMA delay of the system ( $\bar{W}_{ema}$ ), and a rate decrease happened in the previous monitoring interval ( $DF ==$

*True*). In such a situation, it implies that reducing the transmission rate during the last monitoring interval successfully achieved the desired result of reducing the age violation. Consequently, in the next monitoring interval, the A<sup>3</sup>L-FEC-VSVB will further decrease the transmission rate. This action is taken to maintain ongoing efforts in mitigating buffer bloat (a condition of excessive buffering) and to sustain optimal performance. This is done by directing the control decision flow to block 10.

- **Block 9**

The responsibility of this block is to increase the rate of transmission, which it achieves through the following steps:

First, it calculates the difference between the current age violation of the monitoring interval and the EMA of age violations. This value is computed by taking the absolute difference. Next, this computed difference is added to the current transmission rate after being scaled by a factor of  $\frac{1}{n}$ , where this scaling operation helps in fine-tuning the adjustment based on the specific coding characteristics of the system. It is important to note that to ensure a gradual and controlled increase in the transmission rate, the algorithm includes a feature that limits the maximum transmission rate for the next monitoring interval to no more than %120 of the current transmission rate which is scaled by  $\frac{1}{n}$ . In summary, by implementing this feature, the system allows for gradual and controlled increments in the transmission rate, preventing sudden spikes and maintaining a more stable and consistent flow of data.

In addition, this block resets the Empty Flag value to 0 and the Decrease Flag value to False, ( $EF = 0$  and  $DF = False$ ).

- **Block 10**

This block reduces the transmission rate.

First, it calculates the absolute value of the difference between the current monitoring interval's age violation and EMA of age violations. Then subtracts it from the current transmission rate. Note that to prevent the system from assigning sharp changes to the transmission rate, the minimum rate for the following monitoring interval will be greater than or equal to %20 of the current rate.

Note that in this block, after each transmission rate reduction, the Empty Flag (EF) increases by one unit. The EF shows how many times the transmission rate is decreased for probing a better age violation quantity when the system is in "Good Condition". The EF is used in block 2. Moreover, in this block, the Decrease Flag (DF) is set to *True*, which shows a rate reduction occurrence in the current monitoring interval. The DF flag serves a crucial role in block 8.

- **Block 11**

Being in the Block 11 means that the control flow is in "Good Condition" and more than %90 of expected packets are received ( $PDR_{MI} \geq 0.9$ ). Therefore this block can probe for the new transmission rates and quickly converge to the desired transmission rate. The decision mechanism of this block is as follows: If the current transmission rate is less than %75 of the maximum transmission rate, ( $\sigma \leq \sigma_{max}$ ), the control flow is directed towards block 13. Otherwise, it is pushed toward block 12.

- **Block 12**

If the current transmission rate is lower than %75 of the maximum transmission rate, i.e.,  $\sigma \leq \sigma_{max} \times 0.75$ , the decision flow comes from block 11 to this block. In block 12 to elevate the transmission rate to a possible maximum rate a dynamic adjustment mechanism is implemented. This mechanism takes into account the proximity of the current transmission rate to the maximum transmission rate. Indeed the adjustment mechanism ensures that the amount of increased rate that is applied to the current transmission rate inversely correlates with the distance to the maximum rate. If the current transmission rate is already close to %75 of the maximum rate, a more substantial increase is required to further enhance the rate for the next monitoring interval. On the other hand, if the current transmission rate is relatively distant from the %75 of  $\sigma_{max}$  and close to maximum rate  $\sigma_{max}$ , the rate is increased by a more diminutive increment, enabling more room for slow growth without immediate concern for reaching the maximum rate limit. This method promotes a cautious and controlled increase while avoiding potential congestion or performance degradation in the network. This block also resets the value of the Decrease Flag to the *False*.

- **Block 13**

If the current transmission rate is greater than %75 of the maximum transmission rate, i.e.,  $\sigma \geq \sigma_{max} \times 0.75$ , the decision flow is moved from block 11 to the current block.

The primary function of block 13 is to increase the rate, such that the extent of growth is directly proportional to a factor of  $(1 + \frac{1}{n})$ . This block effectively controls the pace at which the rate is increased for the next monitoring interval. It takes into account the trade-off between a smoother, gradual rate change for large-size block-lengths and a more abrupt, significant rate change for small-size block-lengths. This feature of A<sup>3</sup>L-FEC-VSVB provides to it the flexibility of rate management based on the specific requirements and conditions of the network environment.

This block also resets the value of the Decrease Flag to the *False*.

- **Block 14**

This block is used in urgent situations when there is no packet in the network (**Queue Empty-1 state**), and more than two times in the previous monitoring intervals the congestion control reduced the transmission rate, ( $EF \geq 2$ ) by passing through block 10 or 15. Therefore, it is concluded that there are near zero packets in the bottleneck. Under the operation of this block,  $\frac{2 \times (n + [0.05 \times n])}{MinRTT}$  is assigned as the new transmission rate for the upcoming monitoring interval.

To explain how this value is obtained, consider equation (3.22) which is derived from equation (3.20) where in these equations  $n$  is block-length,  $T_s$  is sample generation period and  $\sigma$  is transmission rate. Since  $MinRTT$  could be a benchmark for sample generation without causing a queue in the network, the sample generation period,  $T_s$ , is set to  $MinRTT$ . Additionally, to compensate for the lost packets in the network,  $0.05 \times n$  number of chunks are added to the current block-length amount,  $(n + [0.05 \times n])$ , because it is assumed the loss rate for most Internet traffic is below %5. By substituting the mentioned values in (3.22), the transmission rate for the following monitoring interval is figured.

This block resets the value of Empty Flag (EF) and Decrease Flag (DF) to 0 and *False*, respectively.

$$\sigma = \frac{n}{T_s} \quad (3.22)$$

• **Block 15**

When the following events occur simultaneously, it represents a special state of the system characterized by full queue and buffer bloat in the network, which is defined as a **Queue Full state** in our study:

- For the current monitoring interval, the age violation happens in more than %90 of the times, ( $AV_{MI} \geq 0.9$ )
- EMA of age violation, shows that the age violation happens more than %90 of the times in previous monitoring intervals, ( $AV_{ema} \geq 0.9$ ).
- The average delay is greater than a value such as “AVT”.

The occurrence of these events collectively depicts a critical situation where the system is operating in an "overutilized network capacity" state and requires immediate attention. In response, the algorithm must manage the operational requirements of the system to reduce the transmission rate. At this point, block 15 comes into play to address the aforementioned system state.

This block's primary objective is to reduce the number of transmitted packets by decreasing the transmission rate. To compute a suitable transmission rate for the next monitoring interval, the block modifies the current monitoring interval's transmission rate by multiplying it with the inverse of a decreasing factor,  $\Phi$ , represented as  $\frac{1}{\Phi}$ . Additionally, a diminutive value is added to this product, which is determined as  $\min(0.1, \frac{1}{n})$ . The objective of including this small addition is to guarantee that the transmission rate does not approach or reach zero, and to facilitate a smooth shift to a suitable transmission rate. In other words, if the product ( $\frac{1}{\Phi} \times \sigma$ ) tends to zero, it becomes challenging for the algorithm to detach the transmission rate from small values after encountering a near-zero transmission rate. And for the conditions without the inclusion of  $\min(0.1, \frac{1}{n})$ , it takes a long time to reach a suitable transmission rate, or the transmission rate remains close to zero transmission rate, especially in unfavorable scenarios.

Therefore, to mitigate this problem, the term  $\min(0.1, \frac{1}{n})$  acts as a safeguard

when  $(\frac{1}{\Phi} \times \sigma)$  approaches zero. In summary, by setting this lower limit, block 15 guarantees that the transmission rate does not go toward a very small value. Conversely, when  $(\frac{1}{\Phi} \times \sigma)$  is immense, the effect of term  $\min(0.1, \frac{1}{n})$  on the transmission rate becomes negligible. This is because in such cases, the effect of this small value on the overall transmission rate is minimal.

To recap, we can say the new transmission rate under the operation of block 15 will be " $\frac{1}{\Phi}$ " times the previous monitoring interval transmission rate, plus a small value like  $\min(0.1, \frac{1}{n})$ .

In this block, after each transmission rate reduction, the Empty Flag (EF) increases by one unit. The EF shows how many times the transmission rate is decreased for probing a lower age violation quantity. Note that the EF is used in block 2.

Moreover, in the next monitoring interval, there is a necessity to determine whether a rate reduction has happened in the current interval or not. If the DF flag is set to *True*, it implies that there has been a decrease in the rate during the current monitoring interval. This important information is conveyed by setting the Decrease Flag (DF) to *True* within this block. The DF flag serves as an indicator in block 8 of the algorithm.

#### • Block 16

When the following events happen simultaneously, it implies a specific system state characterized by an empty queue and insufficient packets in the network which is defined in our study as a **Queue Empty-2 state**:

- The packets' average delay in the current monitoring interval is negligible.
- The Empty Flag (*EF*) is less than 2.
- Age violation, when the age of the system exceeds the *AVT* threshold value, occurs more than %90 of the time in the current monitoring interval.

The occurrence of these events collectively represents a critical situation where the system is operating in an "underutilized network capacity" state and requires immediate attention. In response, the algorithm must manage the operational



requirements of the system to increase the transmission rate. At this point, block 16 comes into play to address the aforementioned system state.

This block starts adjusting the transmission rate by multiplying the last monitoring interval's rate by a factor which is shown via " $\Phi$ ." Where the value of  $\Phi$  in this study is 1.5, ( $\Phi = 1.5$ ). This action is translated as a %50 growth in the transmission rate for the following monitoring interval. This block is typically utilized at the beginning of the transmission to enable a multiplicative increase in the transmission rate, thereby assisting in utilizing the maximum capacity of the network.

Actually, in **Queue Empty-2 state** by increasing the transmission rate, the algorithm wants to tackle the problems associated with an empty queue, and insufficient packets in the network. This adjustment allows the system to replenish the network with a sufficient number of packets, and maintain a flow of data, thus mitigating the critical state it is experiencing.

In summary, the simultaneous occurrence of certain events indicates a Queue Empty-2 system state characterized by an empty queue, insufficient packets, and the need for immediate action. The algorithm responds by increasing the transmission rate by an appropriate factor for the next monitoring interval. The purpose of this preventive action is to address data shortages and restore the system to a stable operational state.

Note that, this block resets the value of Empty Flag (EF) and Decrease Flag (DF) to 0 and *False*, respectively.

- **Block 17**

Under different network conditions, one of the following blocks (5, 6, 9, 10, 12, 13, 14, 15, or 16) generates a new transmission rate. To ensure that the generated transmission rate falls within an acceptable range, it is compared against two reference values,  $\sigma_{max}$  and  $\sigma_{min}$  as follows:

Since injecting a large number of packets into the network may cause bufferbloat in the network, a phenomenon where extreme packet buffering happens in the network; hence to mitigate or even prevent this issue, The A<sup>3</sup>L-FEC-VSVB protocol selects the lowest value between  $\sigma$  and  $\sigma_{max}$  as the rate candidate

for the next monitoring interval. This approach ensures that the transmission rate stays within a proper range, considering the potential impact of excessive packet injection on system performance.

Subsequently, the candidate rate value is compared against  $\sigma_{min}$ . This comparison's aim is to prevent injecting an excessively low or even zero number of packets into the network, which could lead to a lack of update packets. Such a situation may cause an increase in age when the system actually ready to handle the incoming packets.

For the A<sup>3</sup>L-FEC-VSVB protocol, the values of  $\sigma_{max}$  and  $\sigma_{min}$  are defined in 3.23 and 3.24, respectively. In these equations, the variable  $k$  represents the minimum number of chunks required at the receiver side to successfully decode a codeword.

These regulations ensure that transmission rates remain within appropriate boundaries and strike a balance between network resource utilization and system performance while mitigating network congestion and data aging issues.

$$\sigma_{max} = \frac{[4.4 \times k]}{AVT} \quad (3.23)$$

$$\sigma_{min} = 0.99 \quad (3.24)$$

- **Block 18**

The function of this block involves keeping the last calculated transmission rate. This stored value,  $\sigma_{last}$ , is subsequently utilized by the algorithm to make adjustments to the transmission rate during the next monitoring interval.

- **Block 19**

If the age violation probability becomes zero, it means that the selected transmission rate for the current monitoring interval fulfilled the requirements of the age-aware application. This indicates that the age of the system is lower than the threshold value, which implies that the system is operating at the desired performance. Therefore, if the system operated desirably in the last monitoring interval, there is no need to change the transmission rate for the upcoming

monitoring interval. The reason for this decision is as follows: Because it is assumed the end-to-end packet delay and loss behavior in the network are not changed remarkably over two adjacent monitoring intervals, therefore it is expected that the age violation probability becomes zero again in the next monitoring interval. Consequently, the last transmission rate, denoted by  $\sigma_{last}$ , is assigned to  $\sigma$  and used in the next monitoring interval.

- **Block 20**

This block is fed from block 1 by the total number of received chunks in the  $MI^{th}$  monitoring interval, denoted as  $C_{MI}$ , (3.13), generation time, and receive time of each of them, and also the minimum number of the chunks,  $k$ , that are required at the receiver side to decode a codeword. Based on these inputs, under different block-lengths, (3.25), possible sample decode scenarios are recorded, then their related age violations by using (3.17) are calculated.

Among the all possible block-lengths the one with the minimum age violation is selected as the effective block-length for the subsequent monitoring interval. This value is injected into block 21 to calculate the optimal sample generation rate and then reported to block 22 to convey to the transmitter as a part of the feedback.

Note that in (3.25), the symbols  $PDR_{MI}$  is the packet delivery ratio in the  $MI^{th}$  monitoring interval, which is documented in (3.19).

$$n \in \left\{ k + [k * (1 - PDR_{MI})] + 1 - [1 - PDR_{MI}], \dots, \min\{(2 * k) + 1, C_{MI} + 1\} \right\} \quad (3.25)$$

- **Block 21**

This particular block serves to calculate the optimal sample generation period, which is denoted by  $T_s$ , employing the formula referenced as (3.20). The calculation of  $T_s$  relies on two key inputs: the effective block-length,  $n$ , and the transmission rate,  $\sigma$ . The effective block-length parameter which is fed from block 20 provides the necessary information regarding the size of the codeword; and the transmission rate which is obtained from either blocks 17 or 19, furnishing the data regarding the rate at which codewords are transmitted.

## • Block 22

This block has the duty of calculating the monitoring interval length which is represented by  $\tilde{T}$ .

At the beginning of each monitoring interval, block 22 receives the effective block-length and optimal sample generation period as input from blocks 20 and 21. Then by using the formula which is represented in (3.26), it computes the monitoring interval length.

To define an appropriate monitoring interval length, the following considerations are taken into account while proposing equation (3.26):

- Under ideal network conditions, with no propagation delays, no queues, or no packet losses, it is possible to achieve a zero value for the age violation. This desire can be attained when the maximum sample generation period is equal to the age violation threshold (AVT). Therefore, the AVT parameter is considered within (3.26).
- To ensure a reliable estimation for Packet Delivery Ratio ( $PDR_{MI}$ ) which is presented in equation (3.19), it is essential to have a sufficient number of packets in each monitoring interval. Therefore, the coefficient of 100 is incorporated into (3.26).
- The monitoring interval length should be compatible with the coding parameter. Thus,  $n$  which represents the total number of chunks generated by the maximum distance separable erasure code is included in (3.26).

Overall, equation (3.26) provides benefits such as optimizing the monitoring interval duration, controlling age violations, facilitating reliable  $PDR_{MI}$  calculations, and adaptability to coding parameters. These advantages contribute to better monitoring and management of the network, leading to improved performance and enhanced understanding of the system's behavior. Therefore based on this explanation, the multiplication of the age violation threshold, AVT, and  $\frac{100}{n}$  offers a suitable monitoring interval duration.

$$\tilde{T} = \left[ AVT \times \frac{100}{n} \right] \quad (3.26)$$

### 3.5 A<sup>3</sup>L-FEC and Theoretical Limits

- **The upper bound for transmission rate:**

In the A<sup>3</sup>L-FEC protocol, it is assumed that UDP packets experience a certain probability of being dropped before entering the bottleneck. This probability is represented as  $P_{loss,in}$ . Moreover, it is considered that a packet is processed at the bottleneck of the network with the rate  $q_s$ . Based on this assumption, it is possible to determine a rate for transmitting codewords, which guarantees the stability of the queue for data transmission.

The upper rate, or upper bound,  $\sigma^{up}$ , denotes the maximum transmission rate that can be supported without causing instability in the bottleneck of the network. It acts as a theoretical limit or reference point to determine the optimal transmission rate in the A<sup>3</sup>L-FEC protocol.

In the upper bound rate formula, represented in equation (3.27), the following three parameters play critical roles:  $q_s$ ,  $P_{loss,in}$ , and  $n$

- $q_s$  denotes the service rate of the bottleneck, indicating the maximum rate at which packets can be processed at the bottleneck of the network.
- $P_{loss,in}$  represents the packet loss probability at the bottleneck input, which reflects the probability that packets are lost or discarded during transmission before entering the bottleneck.
- $n$  represents the number of encoded chunks into which a sample is divided. Indeed each sample is encoded into  $n$  number of chunks.

$$\sigma^{up} = q_s \times \frac{1}{n \times (1 - P_{loss,in})}. \quad (3.27)$$

- **The lower bound for age violation:**

To establish a lower bound for age violation, we assume the presence of an ideal queue, (i.e.,  $q_s = 0$ ) and evaluate the average age violation for a given transmission rate  $\sigma$ . Under this assumption, the sample  $s_\tau$  which is generated

at time slot  $\tau$  is decoded with probability  $P_{d_{s\tau}}(z)$  until time slot  $z$  as follows:

$$P_{d_{s\tau}}(z) = \sum_{i=0}^{n-k+1} \left( \frac{n!}{(n-i)!i!} \right) (P_l(z-\tau))^i (1 - P_l(z-\tau))^{n-i} \quad (3.28)$$

Here, the symbol "!" represents the factorial function, and  $P_l(z-\tau)$  is given by:

$$P_l(z-\tau) = (P_{loss,c})^{z-\tau+1}, z \geq \tau \quad (3.29)$$

The value of  $P_{loss,c}$  in equation (3.29) can be computed using the expression provided in equation (3.1).

Note that the probability of successfully decoding a codeword is affected by several factors, including the number of encoded chunks, the erasure code used, the decoding algorithm employed, and the number of available chunks during the decoding process. In addition, the channel conditions also play a critical role in defining the decoding probability. However, further research and investigation of these parameters are considered for future work.

- **Outage probability:**

Consider the event  $E_{e,t}$ , illustrating a scenario where the age of the system at time  $t$  is  $e$  [time slot]. The probability of observing the event  $E_{e,t}$  is expressed as follows:

$$P(E_{e,t}) = P_{d_{se}}(t) \times \prod_{j=t-e+1}^t (1 - P_{d_{sj}}(t)) \quad (3.30)$$

Based on (3.30), the probability of observing age be greater than  $e$  is derived in (3.31) and called in the rest of this study as the outage probability. Indeed, the outage probability quantifies the likelihood of the age of the system exceeding a certain threshold.

$$P_{Outage}(Age > e) = 1 - \sum_{i=0}^e P(E_{i,t}) \quad (3.31)$$

### 3.6 A<sup>3</sup>L-FEC and Multi-Transmitter Multi-Receiver

To capture a comprehensive representation of real-world scenarios where multiple flows compete for shared resources, it is necessary to extend the system model which is represented in Section 3.3. This extension not only allows for a more accurate depiction of the complexities involved when multiple entities compete to utilize the same resources; but also helps to gain a deep understanding of the dynamics related to potential bottlenecks in this context.

In this regard, the system model illustrated in Figure 3.1 is expanded to accommodate the multi-transmitter multi-receiver scenario, as depicted in Figure 3.11. In this extended configuration, all network flows traverse through a shared bottleneck. This shared bottleneck creates a centralized point of congestion for the whole system which represents a potential constraint in the system's overall performance. As a result, the bottleneck becomes the focal point for data flow management. Therefore, strategies such as load balancing, traffic shaping, or resource allocation require to be employed to ensure the appropriate distribution of resources among the flows passing through the bottleneck to satisfy the receiver's requirements. The receiver may have different requirements that need to be fulfilled in order to achieve optimal performance. These requirements encompass metrics such as maximum throughput, minimum latency, minimum average age, and operation under an age violation threshold, among others. Failure to effectively address this congestion point may result in reduced system performance to meet the receiver's needs or even system instability.

In the system model which is investigated in this section, the congestion control unit applies a policy for handling the packet transmission rate subject that resembles the approach used in the single-transmitter case. The first step in this policy involves computing the possible total rate for the whole system, represented as  $\sigma_{Total}$ , just as in the single transmitter case. Once the total rate for the next monitoring interval is defined, the rates are allocated to each transmitter according to the respective age violation level. Figure 3.12 graphically depicts this rate allocation process.

In the transmission rate allocation process, the age violation level acts as a criterion for distributing the rates among the transmitters. Essentially, transmitters with higher

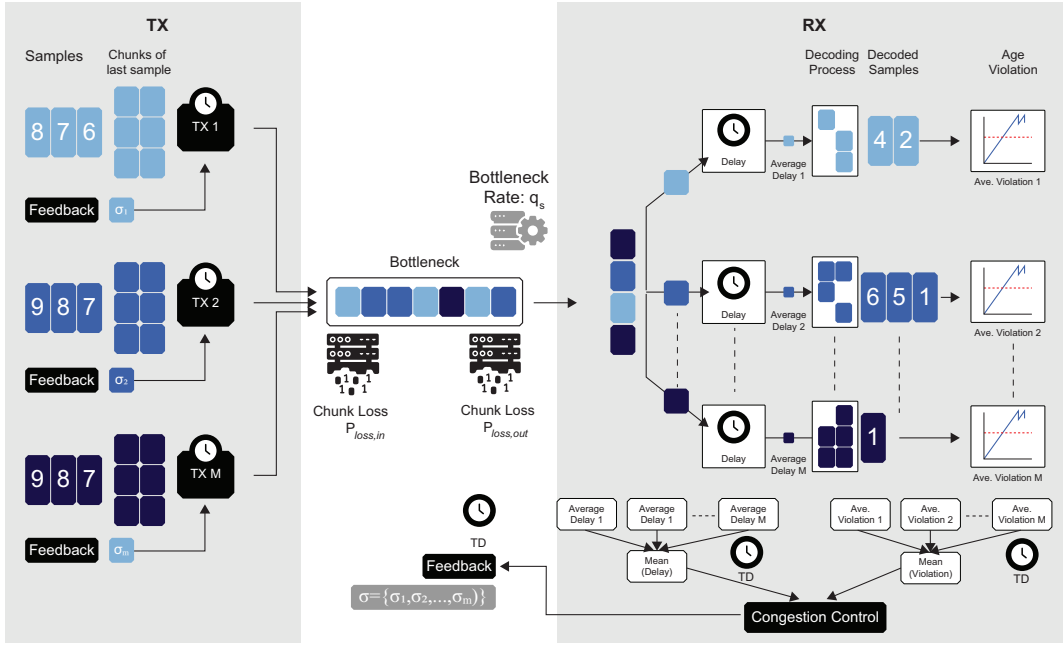


Figure 3.11: System model of a status update system over an error-prone point-to-point link in the presence of transmission rate feedback from the receiver for the multi-transmitter multi-receiver case.

age violation levels are assigned higher data rates. The goal of this rate allocation strategy is to address age concerns and ensure that transmitters with high age violation levels have the opportunity to disseminate their packets promptly, as a result of that reducing overall age violations in the system. Indeed, by taking the age violation level of individuals as the basis for rate allocation, the multi-transmitter multi-receiver system model calculates different levels of urgency among the transmitters. This approach helps maintain fairness and efficiency in the system, as it dynamically adjusts the transmission rates of each transmitter based on the age violation level of that transmitter in the network.

By considering these explanations, in this study a formula that effectively computes the transmission rate of transmitter  $i$  for the following monitoring interval with duration of  $\tilde{T}$  is derived. This formula, represented as equation (3.32), takes into account the current and total transmission rate, the age violation level of transmitter  $i$ , and the mean age violation level of the system to determine the proper rate allocation for transmitter  $i$  within a multi-transmitter multi-receiver system.



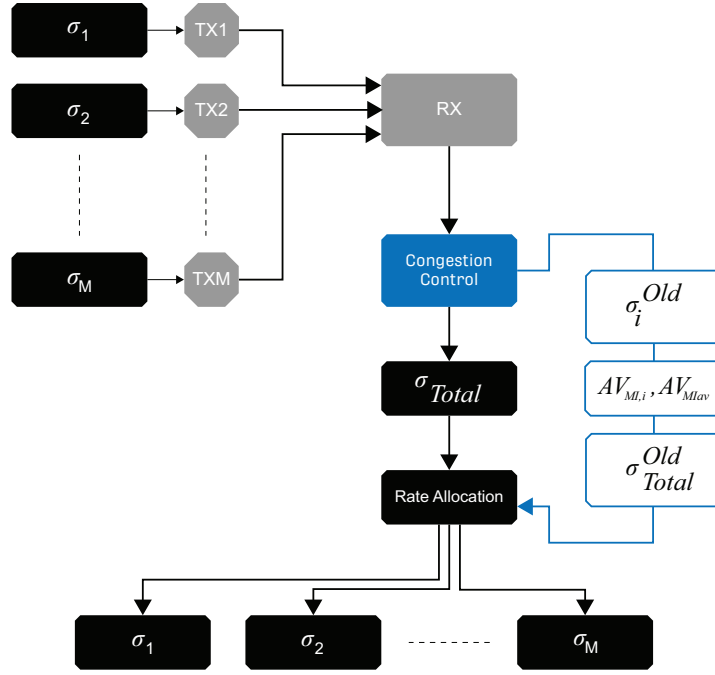


Figure 3.12: Rate allocation in the multi-transmitter multi-receiver scenario.

$$\sigma_i = (\sigma_i^{Old} + (AV_{MI,i} - AV_{MLav})) \times \left( \frac{\sigma_{Total}}{\sigma_{Total}^{Old}} \right) \quad (3.32)$$

In equation (3.32),

- $\sigma_i$  is the rate of the transmitter  $i$  at the next monitoring interval,
- $\sigma_i^{Old}$  is the rate of the transmitter  $i$  at the current monitoring interval,
- $AV_{MI,i}$  is the  $i^{th}$  transmitter's age violation level at current monitoring interval,
- $AV_{MLav}$  is the mean of all transmitters' age violation level or the system's age violation level at the current monitoring interval,
- $\sigma_{Total}$  is the congestion control's decision of total transmission rates for the next monitoring interval,
- $\sigma_{Total}^{Old}$  is the congestion control's decision of total transmission rates for the current monitoring interval,
- In this equation  $i \in 1, 2, \dots, M$ , where  $M$  is the total number of transmitters.

The term  $(AV_{MI,i} - AV_{MIav})$  in equation (3.32) provides valuable insights into the relationship between the age violation level of transmitter  $i$  and the average age violation level of all transmitters in the system. This phrase reveals whether transmitter  $i$  experiences a higher or lower age violation level in relation to the overall average age violation level of the system. Specifically, If the age violation level of transmitter  $i$  is higher than the average of the system, the algorithm infers that transmitter  $i$ 's data decoding process has been prolonged more than others or its last transmission rate was not adequate to satisfy the freshness requirement of the system. Therefore based on this observation, the transmitter  $i$ 's rate must be increased in the next monitoring interval to deliver more fresh data to the receiver. The amount of this increment is represented by  $(AV_{MI,i} - AV_{MIav})$ . On the other hand, If the age violation level of transmitter  $i$  is lower than the system average, it means that transmitter  $i$  experienced relatively fewer outdated status updates than other transmitters. This can be interpreted that its last transmission rate was sufficient to meet the system's freshness requirements. Therefore, in light of these facts, the algorithm can satisfy the fairness between transmitters by reducing the rate of transmitter  $i$  in the next monitoring interval with the amount of  $(AV_{MI,i} - AV_{MIav})$ ; and devote this portion to other transmitters who have exceeded the age violation threshold.

The term  $\left(\frac{\sigma_{Total}}{\sigma_{Total}^{Old}}\right)$  in (3.32) acts as a scaling factor for amplifying or attenuating the transmission rate of transmitter  $i$  based on the relative change in the total transmission rate between the current and the next monitoring interval. This factor ensures that the new rate is allocated proportionally among transmitters in such a way that balanced and coordinated behavior is maintained among all transmitters in the system.

In summary, the equation (3.32) plays a critical role in dynamically altering the transmitter's transmission rate to align with the congestion control decisions and the age violation levels. Indeed it calculates the updated rate for transmitter  $i$  for the next monitoring interval. The intuition behind this rate allocation is regarding fairness, making the transmitters have the same age violation distance to their respective age violation threshold meanwhile keep the sum rate equal to the  $\sigma_{Total}$ . It is worth noting that the suggested method used for rate allocation may change depending on the system design and study goals. However, the fundamental principle of using age violation values to conduct the rate allocation procedure remains the same.

### 3.7 Discussion

**At this point, the reader may raise the following questions regarding the proposed congestion control protocol:**

- **Question: What innovations are included in the A<sup>3</sup>L-FEC?**

**Answer:** Present AL-FEC schemes and congestion control protocols aim for network optimization based on first-order statistics of throughput and delay. On the other hand, emerging applications of interactive streaming and networked control suggest the optimization of data freshness which may be based on higher-order statistics such as the age of information (AoI). Hence, schemes and protocols aiming for average throughput and delay optimization can be inefficient for such applications.

These may be even due to some design choices which seem to be logical when data freshness is not the primary goal. For example, AL-FEC schemes for multimedia streaming encapsulate sample data and parity data in the same network packet in order to save the overheads required for extra packets. However, when the network packet is successfully decoded, the parity data can be only used to decode an earlier sample than the new sample that is already contained in the network packet and it is not meaningful to decode an earlier sample if only the freshest samples are useful which is the case for objectives based on AoI.

- **Question: What is the technical problem that the A<sup>3</sup>L-FEC can solve?**

**Answer:** The proposed scheme and protocol provide a solution to the technical problem of maintaining the freshness of samples that are interactively generated from a source and sent to a destination even under extreme network conditions where network packets are prone to high packet loss and propagation delay.

- **Question: What factors and methods have been preserved by A<sup>3</sup>L-FEC?**

**Answer:** The proposed congestion control protocol that is engineered for age-aware optimization and the proposed A<sup>3</sup>L-FEC mechanism that trade-offs sampling rate and data redundancy are the new elements of the invention.

The proposed A<sup>3</sup>L-FEC mechanism relies on UDP as the transport layer pro-

protocol and popular methods such as Reed-Solomon codes for encoding samples, hence these methods are preserved with the intention as they are used in present AL-FEC mechanisms.

- **Question: How important are the benefits of A<sup>3</sup>L-FEC to the exist internet-based applications?**

**Answer:** The proposed communication protocol is precisely designed to prioritize data freshness, which meets the requirements of various applications such as interactive streaming, network control, and remote monitoring.

One of the key benefits of the proposed protocol is its goal-oriented approach, which distinguishes it from conventional congestion control protocols that rely on loss- or delay-based mechanisms. Instead, the proposed protocol operates at a lower average transmission rate while still achieving optimal network performance. This leads to more efficient use of network resources and improved overall system performance.

- **Question: What does the industry/Internet need in relation to the proposed A<sup>3</sup>L-FEC protocol?**

**Answer:** Indeed the available internet infrastructure does not need any modifications to adapt itself to the A<sup>3</sup>L-FEC protocol. The proposed communication protocol can use the available and well-known facilities such as the UDP protocol and Reed-Solomon error correction method to provide freshness for age-aware applications.

- **Question: What is the market size for the proposed A<sup>3</sup>L-FEC protocol?**

**Answer:** The market size of the proposed A<sup>3</sup>L-FEC protocol is expected to be significant. Because: based on a report which is provided by International Data Corporation (IDC), [2], the amount of generated data from connected IoT devices was 18.3 zettabytes in 2019, while it estimates that this value will grow to 73.1 zettabytes by 2025. A large share of this amount will arise from security and video surveillance, where industrial IoT applications will also represent a significant share.

Additionally, [92] reports that the Industrial IoT connections will lead the overall growth of total IoT connections between 2017 and 2025, with an annual

average of 21%. Consequently, based on this report more than half of the worldwide connections will belong to Industrial IoT connections by 2025. This increase in overall data will imply rising data flows over the communication networks across borders, as the different connected devices can be located all around the world. To date, analysis of the relationship between IoT development and the freshness of information in cross-border data streams is scarce. In addition, there is no comprehensive communication protocol that considers information freshness.

According to the report published by “Fortune Business Insights” in 2022, [93], the size of the global IoT market was \$308.97 billion in 2020. It is expected that the IoT market experience a huge growth from \$381.30 billion in 2021 to \$1.85 trillion in 2028, which depicts an annual growth rate of 25.4 percent over this time period.

Moreover, the forecast of IDC, [94], for the period 2020–2024 shows the negative impact of the pandemic on the worldwide spending on IoT, while it estimates a double-digit growth for the worldwide spending on IoT in the mid-to-long term. This means that worldwide spending on IoT will represent an annual growth rate of 11.3 percent for the period from 2020 to 2024.

Based on this report, about three-quarters of all IoT spending will be done in China, the United States, and Western Europe. Although these three regions will have similar spending totals initially; but over time China’s spending annual growth rate will be 13.4 percent and the other two regions will be 9.0 percent and 11.4 percent respectively. Consequently, based on this difference China will be the leading country in IoT spending. This report also mentioned that the fastest annual IoT spending growth will be in the Middle East and North Africa (19.0 percent), Central and Eastern Europe (17.6 percent), and Latin America (15.8 percent).

In summary, drawing from the findings documented in the aforementioned reports, it is obvious that the A<sup>3</sup>L-FEC protocol holds substantial promise within the market landscape. This promising potential can be attributed to two main factors: The growing need for Internet of Things (IoT) connectivity and the concurrent necessity for communication protocols that prioritize data freshness.

### 3.8 Conclusion

In this study, we proposed the A<sup>3</sup>L-FEC protocol, which is designed to control the age of information in a communication system transmitting samples of variable processes from the transmitter to the receiver with high data freshness. The protocol utilizes forward error correction (FEC) and user datagram protocol (UDP) to optimize data freshness and minimize age violations.

The A<sup>3</sup>L-FEC protocol differs from existing automatic repeat request/forward error correction (ARQ/FEC) mechanisms by specifically targeting the age of information as a criterion for optimization. Unlike traditional ARQ/FEC mechanisms that focus on throughput and latency, A<sup>3</sup>L-FEC prioritizes data freshness.

Two flavors of A<sup>3</sup>L-FEC were introduced in this study: A<sup>3</sup>L-FEC-FSFB (Age Aware Application Layer Forward Error Correction - Fixed Sampling Rate Fixed Block-length) and A<sup>3</sup>L-FEC-VSVB (Age Aware Application Layer Forward Error Correction - Variable Sampling Rate Variable Block-length). While the primary approach of these variations is identical, they differ in terms of sample generation and transmission intervals.

The system model that is considered for A<sup>3</sup>L-FEC-FSFB includes a time-slotted status update system over an error-prone communication link. Maximum distance separable (MDS) erasure code is used at the transmitter side to encode each sample into coded chunks. Moreover, the receiver stores and utilizes the received coded packets to reconstruct the corresponding codeword. Additionally, a congestion control algorithm dynamically adapts the transmission rate based on feedback from the receiver in every monitoring interval. By dynamically adjusting the sampling rate, A<sup>3</sup>L-FEC can decrease unnecessary data transmissions, leading to lower energy consumption in the communication process.

A<sup>3</sup>L-FEC-VSVB extends the system model by equating it to a variable sampling rate and a varying block length. The transmitter modifies the transmission rate based on the measured values of age violation and delays of received chunks. The receiver notifies the transmission rate value periodically to the transmitter.

The A<sup>3</sup>L-FEC-VSVB protocol similar to the A<sup>3</sup>L-FEC-FSFB protocol operates on the application layer of the network and uses a fire-and-forget pattern technique for data transmission, but it employs a generate-at-will model for generating packets. The fire-and-forget pattern technique eliminates the need for establishing and maintaining reliable connections, reducing the overhead associated with establishing connections and consequently minimizing the energy consumption of the communication process.

The proposed A<sup>3</sup>L-FEC protocol presents various benefits in terms of controlling the age of information and optimizing data freshness. This technique allows real-time applications to receive fresh data and make timely decisions. The protocol can be used in different types of systems, including single-transmitter systems and multi-transmitter multi-receiver systems.

In summary, the proposed A<sup>3</sup>L-FEC protocol integrates an adaptive data flow control algorithm, optimal SIS transmission policies, and the coding rates and transmission rate adjustment feature to optimize data freshness and reduce age violation levels in the network. Therefore, by dynamically adapting to network conditions and incorporating effective transmission policies, the protocol presents a comprehensive solution for age-aware communication scenarios. With these compatibility, efficiency, and compatibility features, the A<sup>3</sup>L-FEC protocol promises to improve the performance and reliability of interactive streaming, network control, and age-aware applications.





## CHAPTER 4

### SIMULATION AND EMULATION RESULTS

#### 4.1 Simulation-based Evaluation of AoI Under Different TCP Congestion Control Algorithms on ns-3 Network Simulator

To comprehensively evaluate the Age of Information (AoI) performance of well-known congestion control algorithms such as TCP BBR, TCP BIC, TCP Cubic, TCP Ldbat, and TCP NewReno over TCP/IP links, we meticulously designed and conducted a series of simulations. By performing these simulations, we aimed to investigate and compare the performance of various congestion control algorithms in minimizing network congestion and optimizing the freshness of information. Our simulation testbed involved carefully configuring the ns-3 simulator to replicate real-world network conditions and accurately modeling the behavior of TCP/IP links. Simulation results related to the performance of different flavors of TCP congestion control algorithms gave us valuable insights into their respective impacts on the AoI metric.

A network simulator is a network analytical modeling tool. Researchers and developers frequently use these types of tools to diagnose and interpret network performance for current network behavior or to predict the consequences of changes in the network by testing new scenarios. In addition, network simulators reduce costs and facilitate analyzing network behavior when the network scale is increased or new technology such as congestion control is developed. For real-world network applications, preparing an under-control environment to evaluate the congestion control algorithms to ensure their robustness before using them in real-world traffic is critical.

Network Simulation is categorized based on different execution techniques: Discrete event simulation (DES), Parallel discrete event simulation (PDES), and Ultra large

scale simulation framework (USSF) [95]. A discrete event simulation models the performance of a system as a (discrete) sequence of events in time[96]. Each event happens at a specific instant in time and indicates a state change in the system. For discrete event simulation between successive events, it is assumed that there is no change in the system. Therefore, the simulation time can directly hop to the subsequent event's occurrence time, which is named the next-event time progression. Parallel Discrete Event Simulation (PDES) refers to running a discrete event simulation program on a parallel computer that is sometimes called distributed simulation[97]. This type of approach decomposes the simulation application into a set of concurrently executing processes, thus in general, increasing in performance of network simulators. An Ultra large scale simulation framework (USSF) is an optimistic simulation system that has been suggested for large-scale systems requiring very long simulation times. USSF dramatically reduces model run-time state by logical processes (LP) aggregation, and swapping LPs out of core[98]. USSF uses "NOTIME", [99], to execute simulations unsynchronized. Moreover, The features of USSF make it possible to exploit on top of a discrete event simulator with only minor changes to the application modules, [100].

There are several different network simulator software that utilizes DES, PDES, or USSF techniques. Table 4.1 demonstrates the well-known ones with their unique features. In [101], the authors compare memory usage, CPU usage, scalability, and computation time for different network simulators. Based on [101] results, the ns-3 network simulator performs better than other simulators such as OMNET ++, GloMoSim, and ns-2. Therefore, in this study to comprehensively evaluate the AoI performance of well-known congestion control algorithms over TCP/IP links we applied ns-3 network simulator.

ns-3 is a discrete event network simulator, with the evolution of a network system modeled through discrete events in time. This tool simulates the real-world network on one computer by writing scripts in C++ or Python. ns-3 also has the ability to utilize the parallelism feature to speed up the overall performance of simulations. This open-source discrete-event network simulator by creating various virtual nodes (i.e., computers in real life) and utilizing various types of Helper classes permit the user to install devices, internet stacks, applications, etc on each node. Moreover,

Table 4.1: Well-known Network Simulators Overview.

Simulator	Method	Feature
OPNET	(Parallel) Discrete event	GUI support, object-oriented, large scalability [102]
OMNET++	Discrete event	GUI support, component based (modular) analysis [103]
GloMoSim	(Parallel) Discrete event	Large scale wireless networks, library-based [104]
QualNet	(Parallel) Discrete event	Commercialised GloMoSim, protocol design environment [105][106]
NetSim	(Stochastic) Discrete event	Development environment, feature rich (layer abstraction) [107][106]
ns-2	Discrete event	Object-oriented (TCL), protocol simulation in virtual time [108]
ns-3	(Parallel) Discrete event	Numerous networks types and protocol detail emphasis [109]

Creating point-to-point, Wireless, CSMA, etc connections between nodes is possible in the ns-3 network simulator. A point-to-point connection is the same as a LAN connection between two computers and a wireless connection is the same as a WiFi connection between various computers and routers in a real network. To simulate a bus topology between computers, ns-3 offers the CSMA connection.

In this simulator, the applications have the duty of generating traffic and sending the packets. ns-3 also simulates the communication channel and its related parameters such as data rate, packet size, packet loss ratio and etc.

#### 4.1.1 Simulation Testbed Setup on ns-3 for AoI Evaluation of TCP Congestion Control Algorithms

This section outlines the simulation parameters utilized to construct the testbed for evaluating the AoI performance of well-known congestion control algorithms. Through simulation result assessment, we desired to determine which algorithm provides superior AoI performance by considering network throughput, latency, packet loss, and congestion avoidance factors. Then based on these results, we will use the superior one as the benchmark for comparing with the A<sup>3</sup>L-FEC protocol. As mentioned pre-

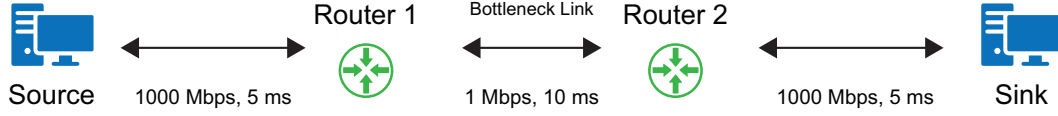


Figure 4.1: End-to-end network topology used for simulation.

viously due to the outstanding performance of ns-3, this simulator was employed to simulate an end-to-end network topology, which includes one source (transmitter) node, two routers, and one sink (receiver) node, as illustrated in Figure 4.1.

To present network heterogeneity, the link (channel) between Router 1 and Router 2 was designated as a bottleneck link, limiting its capacity to 10 *Mbps*, while all other links were set to have a capacity of 1000 *Mbps*. The propagation delay (Link delay) between the two routers was configured to 10 *ms*, while the propagation delay for the links connecting Router 1 and the source, as well as Router 2 and the sink, was set to 5 *ms* for each one on our testbed.

To emulate a realistic network scenario, a queue was introduced at the bottleneck of this simulated network. The ns-3's `FifoQueueDisc` module was utilized to implement the FIFO policy, where data packets were placed in a distinct internal queue, modeled as a DropTail queue. Note that the capacity of the queue discipline can be defined using either packets or bytes as units of measurement. For this particular simulation, the capacity of the queue was specified in terms of packets, and it was set to a maximum of 100 packets. This means that the network queue could hold up to 100 packets before reaching its capacity, after which other arriving packets would be dropped according to the implemented queue management policy.

In AoI evaluation of various TCP variants, we considered BBR, BIC, Cubic, Leaky bucket, and NewReno. Each simulation run for these TCP variants was conducted for a duration of 30 seconds. Note that these congestion control algorithms are mainly concerned with handling network congestion and optimizing throughput rather than directly impacting the AoI. However, we have carefully selected the simulation parameters to investigate and compare the data freshness performance of different TCP congestion control algorithms in real network conditions.

#### 4.1.2 AoI Evaluation results of TCP Congestion Control Algorithms on ns-3

The simulation outcomes provide valuable insights into the performance of different TCP congestion control algorithms in terms of AoI. While TCP BBR stands out with its minimum average AoI according to our experiments, it is necessary to examine the performance of other well-known flavors of TCP to gain a comprehensive understanding of their strengths and weaknesses in this field.

TCP BBR, TCP BIC, TCP Cubic, TCP Leaky, and TCP NewReno are congestion control algorithms assessed in this study. Since each of these congestion controls possesses different characteristics in terms of delay, packet loss, and throughput, consequently, these algorithms represent distinct AoI performances for sending the same data under the same conditions.

The simulation results, (Figure 4.2a), indicate that among the evaluated TCP congestion controls, the TCP BBR transmitted data with the lowest average age of information. This result can be attributed to the lowest average delay shown by TCP BBR, (Figure 4.2c). Also, in addition to low AoI, TCP BBR presents a competitive average throughput compared to other types of TCP, (Figure 4.2b).

A fundamental factor for TCP BBR's low AoI is its conservative packet injection behavior. Unlike some of the other variants, TCP BBR does not aggressively inject excessive packets into the network. This behavior results in fewer packets being queued in the bottleneck of the network.

Moreover, the TCP BBR implementation experiences the lowest packet loss rate among all TCP variants investigated in the testbed; as a result, it needs less packet retransmission when any packet loss event occurs in the network.

Overall, the superior performance of TCP BBR in terms of AoI is attributed to the efficient usage of network resources, which reduces both delay and packet loss. By keeping a small number of packets in the queue, (Figure 4.2d), and avoiding congestion, TCP BBR has been able to achieve a significant reduction in the average age of information, ensuring that information is sent across the network in a timely and up-to-date manner.

TCP BIC (Binary Increase Congestion Control) exhibits a higher average AoI value when compared to the other variants, (Figure 4.2a). This can be attributed to the relatively higher average delay and a high level of packet loss, (Figure 4.2c and 4.2e). The explanation for the high average delay could be as follows: TCP BIC is a loss-based congestion control algorithm. It basically relies on packet loss detection as an indication of network congestion. Based on the results, it can be observed that TCP BIC has a higher average congestion window (CWND) and a larger average queue length. These characteristics imply that TCP BIC aims to inject a higher number of packets into the network until it receives a packet loss signal. Therefore it will have a high number of packets in the queue, which causes a high average delay and loss and, in turn, yields a high AoI. Moreover, The aggressive nature of TCP BIC increases losses in the network, [110], as observed in our results, (Figure 4.2e). Therefore, it is not suitable for low-speed networks. Because more loss means: more retransmissions, more queues, more delay, and eventually more AoI.

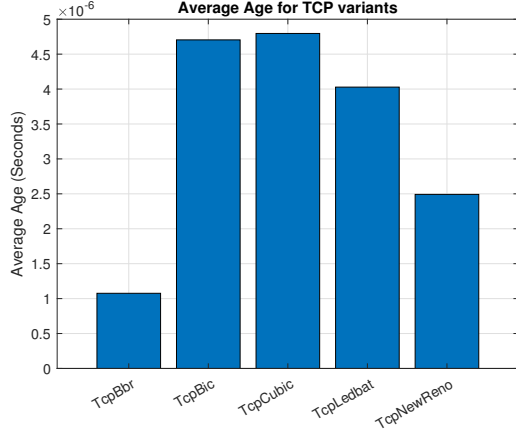
TCP Cubic, another popular TCP congestion control algorithm which is the default congestion control algorithm in Linux, shows a higher average AoI compared to TCP BBR, (Figure 4.2a). This is because TCP Cubic does not have a specific mechanism to reduce packet delay. Its primary focus is on fairness and throughput by using the maximum congestion window (CWND) from the previous congestion period as a reference, and may not prioritize low latency as much as TCP BBR. Note that TCP Cubic decreases the CWND only to 0.7 times its current value when it senses congestion, [111]. This approach helps efficient link utilization when the buffer size of the bottleneck is smaller than the bandwidth-delay product (BDP). However, this feature can cause a larger standing queue and subsequently increase queuing delay, (Figure 4.2d), and finally a higher average AoI as observed in our simulation results, which are presented in Figure 4.2a.

The congestion control algorithm of TCP Lebat relies on one-way delay estimation. It estimates the queue delay by computing the difference between the current delay and the reference delay, which is the minimum delay observed in previous measurements. When the transmitter notices an increase in one-way delay, it translates it as a signal of increasing congestion in the network and responds by reducing its transmission rate, [112]. Our simulation outcome shows that TCP Lebat has a lower average

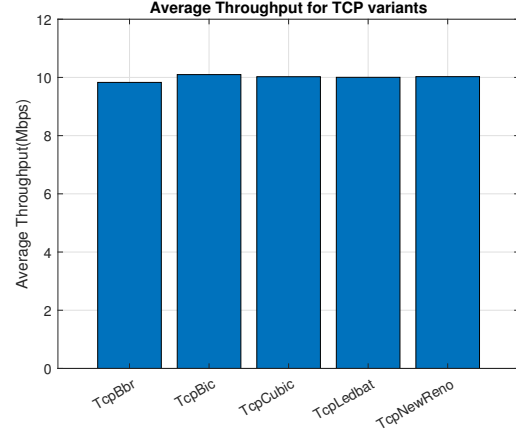
delay, (Figure 4.2c) and average queue length compared to TCP BIC, and TCP Cubic, (Figure 4.2d). As a result, it shows better age of information (AoI) performance than these congestion control algorithms. However, when comparing the packet loss ratio results of TCP Ldbat with TCP BBR and TCP NewReno, it is obvious that TCP Ldbat performs worse in terms of packet loss, (Figure 4.2e). Therefore, it can be concluded that the higher loss ratio experienced by TCP Ldbat may lead to higher AoI compared to TCP BBR and TCP NewReno.

Lastly, TCP NewReno is a traditional congestion control algorithm based on packet loss detection. It follows a non-scalable approach and uses an Additive increase and multiplicative decrease (AIMD) mechanism. This congestion control algorithm introduces changes to the fast recovery mechanism used in TCP Reno to handle scenarios where multiple losses occur in a single window. In TCP NewReno, when a sender receives a partial acknowledgment (ACK), it does not immediately exit the fast recovery phase as in TCP Reno. Instead, it supposes that the packet following the last acknowledged packet has been lost, and so resends the lost segment. As a result, TCP NewReno does not rely on waiting for the retransmission timeout in scenarios involving multi-segment losses. Instead, it keeps retransmitting the missing pieces every time it receives a partial ACK, [113]. The retransmission mechanism used in TCP NewReno, which avoids waiting for the retransmission timeout, not only helps decrease the loss ratio but also has a positive impact on the AoI in our simulation as seen in Figures 4.2e and 4.2a. However, it is necessary to note that this approach has a drawback: it can lead to an increase in the queue length. According to our results, TCP NewReno exhibits a lower average AoI compared to TCP BIC, TCP, and Cubic, while it has a higher average AoI compared to TCP BBR as shown in Figure 4.2a.

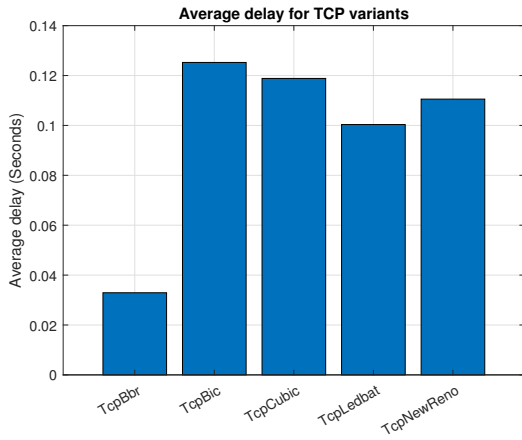
In conclusion, while TCP BBR showcases the lowest average AoI among the evaluated TCP variants, other congestion control algorithms such as TCP BIC, TCP Cubic, TCP Ldbat, and TCP NewReno also demonstrate their own strengths and weaknesses, (Figure 4.2a). These algorithms exhibit varying levels of average delay, packet loss, and throughput, which ultimately impact their performance in terms of AoI. Understanding the characteristics and trade-offs of different TCP variants enables network administrators and researchers to select the most suitable congestion control algorithm based on their specific requirements and network conditions.



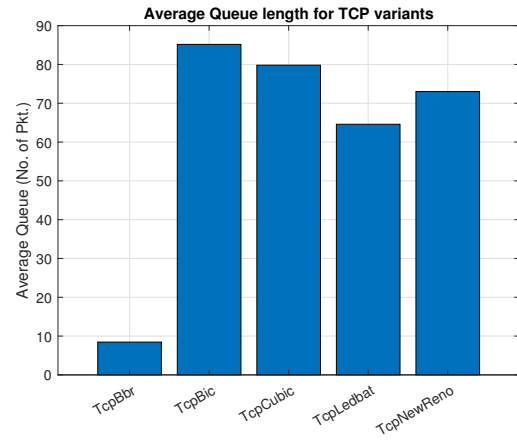
(a) Average age for TCP variant



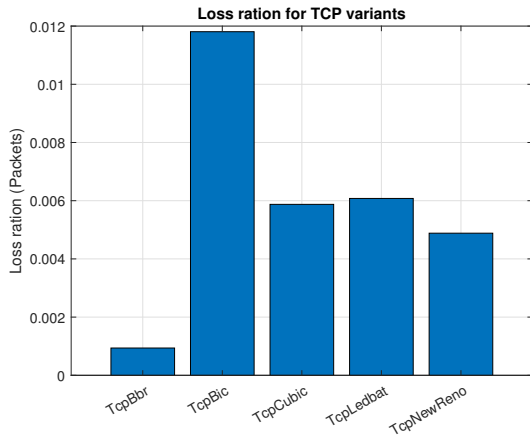
(b) Average throughput for TCP variant



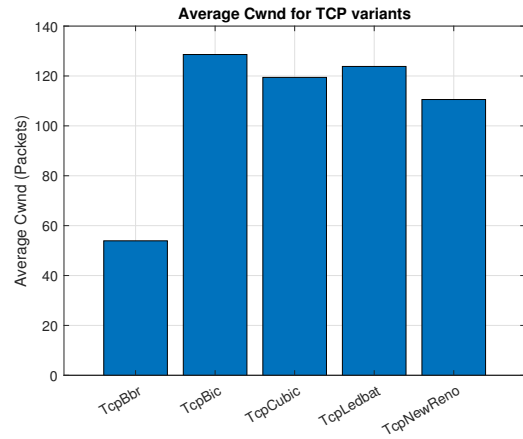
(c) Average delay for TCP variant



(d) Average queue length for TCP variant



(e) Average loss for TCP variant



(f) Average Cwnd for TCP variant

Figure 4.2: Average age, average delay, average throughput, average cwnd, average loss, and average queue length for well-known TCP variants under an end-to-end topology which is shown in Figure 4.1.



## **4.2 Simulation-based Evaluation of A<sup>3</sup>L-FEC-FSFB and ACP+ Algorithms on MATLAB**

In this section, we have designed and conducted a series of simulations to evaluate the performance of our proposed age-aware data flow control algorithm, A<sup>3</sup>L-FEC-FSFB, and compare it with the existing age-aware congestion control algorithm, ACP+, as summarized in the subsequent section, 4.2.1. The simulation testbed was meticulously prepared, employing MATLAB to precisely replicate the functionalities of both A<sup>3</sup>L-FEC and ACP+ algorithms. The main focus of the simulation is to provide a comprehensive evaluation of the effectiveness of these algorithms, especially in terms of age violations through analysis of simulation results. By intently examining the results, it is possible to gain valuable insights into the advantages and limitations of each algorithm and how these age-aware algorithms manage the data freshness.

Our MATLAB simulations are carefully planned to contain various aspects and scenarios related to age-aware congestion control, such as packet losses, different service rates, and bottleneck bandwidth. By conducting detailed assessments, meaningful conclusions about the performance and suitability of A<sup>3</sup>L-FEC and ACP in different network conditions and practical settings are obtained. To perform the simulation, we use the system model introduced in Section 3.3, which includes transmission through a queue using a FCFS discipline with deterministic service rates. This choice of system model assures that the simulation closely resembles the real-world behavior of the communication system we are analyzing. Before going into the details of this simulation, let us provide some information about the ACP+ algorithm in the next section, 4.2.1. It is worth noting that the comprehensive description of A<sup>3</sup>L-FEC-FSFB has already been presented in Section 3.3.1.

### **4.2.1 Overview of the Age Control Protocol (ACP) and ACP+**

To the best of our knowledge, the Age Control Protocol (ACP) [89] and its enhanced version, ACP+ [76], are the only suggested communication protocols specifically aiming to control the age of information and addressing the data freshness in communication systems.

In order to make a comprehensive comparison between the A<sup>3</sup>L-FEC and ACP+ protocols, it is important to know the Age Control Protocol (ACP) in detail. In this section, we take a closer look at ACP and its enhanced version, ACP+, and highlight their key features and functionality. This knowledge helps us to make a more informed comparison between ACP and A<sup>3</sup>L-FEC algorithms by evaluating their respective strengths and limitations.

In this section, we detail two transport layer protocols, ACP and ACP+, which are specifically designed to minimize the age at the receiver side. To achieve this goal these protocols dynamically adjust the update (send) rate of the transmitter based on estimated network conditions. In this section also we overview the differences between ACP and ACP+ in terms of their strategies for minimizing age.

ACP is built on top of User Datagram Protocol (UDP). This protocol operates at both ends of a communication system and consists of two distinct phases: the “initial” phase and the “epochs” phase. In the initialization phase, the sender sends a predetermined number of packets to the receiver and waits for the corresponding acknowledgment packets. Based on the received ACKs, the ACP algorithm calculates an estimated round-trip time (RTT). The main purpose of this step is to configure the initial update rate of the transmitter, which sets the stage for subsequent operations. After the initialization phase, the system enters the epoch phase. This phase is divided into discrete periods, which are called epochs. Each epoch is denoted by  $k$ . The task of ACP in these epochs is the calculation of two important metrics: the average age of information ( $\bar{\Delta}_k$ ) and the average backlog ( $\bar{B}_k$ ). The average age of information describes the average time elapsed since the last update was successfully received, while the average backlog depicts the average number of packets sent to the receiver but awaiting acknowledgment.

This protocol by analyzing the average age of information and average backlog metrics at each epoch provides valuable insights into overall system performance and makes informed decisions regarding backlog adjustments. The decision-making algorithm of the protocol is summarized in Table 4.2, with  $\delta_k$  representing the difference between the consecutive average age of information measurements ( $\bar{\Delta}_k - \bar{\Delta}_{k-1}$ ) and  $b_k$  reveals the difference between consecutive average backlog values ( $\bar{B}_k - \bar{B}_{k-1}$ ).

Table 4.2: Decision algorithm of ACP

$\delta_k$	$b_k$	ACP targets to:
$> 0$	$> 0$	Decrease the backlog
$> 0$	$< 0$	Increase the backlog
$< 0$	$> 0$	Increase the backlog
$< 0$	$< 0$	Decrease the backlog

The table summarizes four possible combinations of  $\delta_k$  and  $b_k$  and the corresponding target actions. If  $\delta_k$  and  $b_k$  are both positive, revealing an increase in both average age and backlog, consequently ACP targets to decrease the backlog. This action helps to alleviate congestion in the network and reduce the age. Likewise, if  $\delta_k$  and  $b_k$  are both negative, indicating a decrease event in both average age and backlog, therefore ACP targets to decrease the backlog further. On the other hand, if  $\delta_k$  is positive but  $b_k$  is negative, which means an increase in average age and a decrease in the backlog, ACP targets to increase the backlog. The reason for this action is to ensure sufficient buffer space and minimize age violations. Similarly, if  $\delta_k$  is negative and  $b_k$  is positive, revealing a decrease in average age but an increase in the backlog, ACP targets to increase the backlog to deliver more updates to the receiver.

Indeed this decision algorithm serves as a guideline to determine the proper course of rate variation actions based on observed changes in the average age of information and backlog. By tracking the changes in these metrics in each epoch, ACP dynamically adjusts the backlog size to optimize the transmission and reception of updates, ultimately aiming to minimize the age of information for the next epoch. The structured approach of the algorithm ensures that the congestion control decisions meet the specific requirements and characteristics of age-aware communication systems.

To determine an appropriate transmission rate ( $\lambda_k$ ) at the transmitter side for use in the next epoch, the ACP computes the exponentially weighted moving average (EWMA), round-trip time (RTT) ( $\overline{RTT}$ ), and the EWMA of the arrival times between ACKs ( $\overline{Z}$ ) Upon receiving an ACK packet. By updating  $\lambda_k$ , ACP can alter the backlog size and thereby control the age of information. To achieve this, ACP adjusts the update rate

Table 4.3: Table of actions and corresponding  $b_{k+1}^*$  values

The Action	$b_{k+1}^*$
Increase (INC)	$\kappa$
Decrease (DEC)	$-\kappa$
Multiplicative Decrease (MDEC)	$-(1 - 2^{-\gamma})B_k$

by determining the value of  $b_{k+1}^*$  in Equation (4.1). Note that  $\min(\overline{RTT}, \overline{Z})$  in this equation ensures that the update rate is not larger than the minimum of these two values. The specific value of  $b_{k+1}^*$  depends on the chosen action, which is defined by the algorithm. The relationship between the actions and their corresponding  $b_{k+1}^*$  values is outlined in Table 4.3.

$$\lambda_k = \frac{1}{\overline{Z}} + \frac{b_{k+1}^*}{\min(\overline{RTT}, \overline{Z})} \quad (4.1)$$

Table 4.3 illustrates the actions which are used by ACP and their corresponding  $b_{k+1}^*$  values. When the action is "Increase (INC)," the  $b_{k+1}^*$  will be  $\kappa$ . Similarly, for the "Decrease (DEC)" action,  $b_{k+1}^*$  will get  $-\kappa$  value. In the case of "Multiplicative Decrease (MDEC)," the value of  $b_{k+1}^*$  is computed as  $-(1 - 2^{-\gamma})B_k$ , where  $\gamma$  is a parameter that defines the degree of multiplicative decrease, while  $B_k$  represents the average backlog. ACP uses the "MDEC" action when the "DEC" (Decrease) action alone is not effective in reducing the average backlog ( $\overline{B}_k$ ) at a sufficient rate. The aim of the "MDEC" action is to achieve a multifold reduction in the backlog, which allows for a faster reduction in congestion levels. By utilizing this method, ACP can respond more dynamically to network conditions and ensure effective congestion control in scenarios where simple decrease action may not be sufficient. Note that the value of  $\kappa$  plays a crucial role in the ACP protocol as it directly impacts the update rate calculation in Equation (4.1). Consequently, it is essential to define  $\kappa$  precisely to ensure the effectiveness and stability of the congestion control mechanism.

Moreover, ACP specifies the length of each epoch ( $\overline{T}$ ) using the Equation (4.2) at the end of the previous epoch. The length of each epoch has been carefully chosen to

strike a balance between being short enough to respond promptly to changes in network conditions and long enough to provide a fine estimate of prevailing conditions. This ensures that the duration is optimized to effectively capture network dynamics and facilitate efficient congestion control.

$$\bar{T} = 10 \times \min(\overline{RTT}, \bar{Z}) \quad (4.2)$$

ACP+ is an advanced version of ACP that builds on the ACP logic while incorporating several fundamental changes to improve its performance and effectiveness. These key modifications are as follows:

- **Computation of  $\lambda_k$ :** In contrast to ACP, which uses the Equation (4.1) to update the transmission rate ( $\lambda_k$ ), ACP+ makes slight changes by substituting  $\min(\overline{RTT}, \bar{Z})$  with  $\overline{RTT}$  in the equation. The goal of this change is to improve the accuracy and effectiveness of the transmission rate adjustment process in ACP+.
- **Setting  $\kappa = 1$  and clamping  $\lambda_k$ :** Another important modification is that in ACP+, the value of  $\kappa$  is set to 1 and the update rate ( $\lambda_k$ ) is clamped to avoid drastic changes. In a study by [76], the bounds of the clamp were defined as 1.25 times the transmission rate of the previous epoch ( $\lambda_{k-1}$ ) for the maximum transmission rate and 0.75 times  $\lambda_{k-1}$  for the minimum transmission rate. These clamping bounds guarantee that the update rate stays within a reasonable range and avoids extreme oscillations.
- **Modification in Equation (4.2):** The final distinction lies in the equation used to specify the length of each epoch ( $\bar{T}$ ). ACP+ protocol utilizes (4.3) to compute  $\bar{T}$ , which replaces  $\min(\overline{RTT}, \bar{Z})$  in (4.2) with  $\frac{1}{\lambda_k}$ . This change reflects ACP+'s objective of aiming to transmit approximately 10 packets within each epoch duration.

$$\bar{T} = \frac{10}{\lambda_k} \quad (4.3)$$

Table 4.4: MATLAB Simulation Parameters for Age Violation Evaluation of A<sup>3</sup>L-FEC-FSFB and ACP+

<b>Age Violation Threshold</b>	5 [time slot]
<b>Bottleneck Queue Size</b>	5000 [chunk]
<b>Queue service rate</b>	$k \times 1.4706 [\frac{packets}{timeslot}]$
$P_{inLoss}$	0, 0.1, 0.2
$P_{outLoss}$	0, 0.1, 0.2
<b>Propagation Delay</b>	1 [time slot]
<b>Monitor Intervals (MI) Duration</b>	$10^2$ [time slot]
<b>Simulation Duration</b>	$10^5$ [time slot]
<b>k for A<sup>3</sup>L-FEC-FSFB</b>	3, 4
<b>n for A<sup>3</sup>L-FEC-FSFB</b>	4, 5, 6

#### 4.2.2 Simulation Testbed Setup on MATLAB for Age Violation Evaluation of A<sup>3</sup>L-FEC-FSFB and ACP+

This section provides the details of the simulation setup used to compare the age violation performance of A<sup>3</sup>L-FEC-FSFB and ACP protocols and to determine the algorithm that causes the minimum age violation while sending the same number of packets. By investigating the simulation results, we can achieve insights into the effectiveness of each algorithm and draw conclusions about their performance in terms of age violations. The simulations were performed based on a system model similar to that described in Section 3.3. Transmissions are done through a queue using a FCFS discipline with deterministic service rates,  $k \times 1.4706$  packets per time slot. The simulation duration was set to  $10^5$  time slots, while the monitor intervals (MI) had a duration of  $\tilde{T} = 10^2$  time slots. The simulation parameters, presented in Table 4.4, were carefully selected to provide meaningful insight into the performance of the algorithms and cover different possibilities. Note that  $P_{inLoss}$  is the packet loss probability before entering the bottleneck and  $P_{outLoss}$  is the packet loss probability after leaving the bottleneck.

#### 4.2.2.1 Age Violation Evaluation Results of A<sup>3</sup>L-FEC-FSFB and ACP+ Algorithms on MATLAB

In this section, simulation results for the proposed age-aware flow control algorithm, A<sup>3</sup>L-FEC-FSFB, are represented and compared with ACP+, the only existing age-aware congestion control algorithm in the literature. We compared the A<sup>3</sup>L-FEC and ACP+ algorithms for different system parameters, represented in Table 4.4. Tables 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, and, 4.13 show the age violation results of A<sup>3</sup>L-FEC and ACP+ algorithms for different values of packet loss probability.

For most of the simulations, we notice the outperformance of our proposed algorithm over the ACP+ in terms of minimizing the age violation. To gain insights into the superior performance of the A<sup>3</sup>L-FEC-FSFB data flow control over the ACP+ in different scenarios, several key factors need to be investigated. These factors contain variations in transmission rates, average age violation, average age, average delay, and the number of packets in the bottleneck at the end of each monitoring interval.

In the rest of this section, we focus on Figures 4.3 and 4.4, where we compare the performance of the A<sup>3</sup>L-FEC-FSFB and ACP+ algorithms based on the aforementioned factors. The assessment is conducted under the following conditions: packet loss probability of  $P_{\text{inLoss}} = 0.2$ ,  $P_{\text{outLoss}} = 0.2$ , and coding parameters of  $k = 4$  and  $n = 5$ . For both algorithms, the age violation threshold is set to 5. It is important to note that the coding parameter is utilized exclusively in the A<sup>3</sup>L-FEC-FSFB.

Based on the results shown in Fig. 4.3, it is seen that the proposed algorithm consistently keeps transmission rates ( $\sigma$ ) lower than the capacity of the bottleneck service rate (i.e., the optimal transmission rate). However, the selected transmission rates still remain close to the optimal transmission rate which is calculated based on (3.27). This behavior guarantees that the algorithm avoids considerable queuing in the buffer, while simultaneously providing fresh data to the client.

For the ACP+ protocol, we notice that the transmission rate chosen by ACP+ also rarely exceeds the queue capacity. However, there are cases where the algorithm sends at very low rates, leading to a high value for age violations, Figure. 4.4.

Table 4.5: Performance comparison of A<sup>3</sup>L-FEC-FSFB and ACP+ algorithms under packet loss probabilities  $P_{\text{inLoss}} = 0$  and  $P_{\text{outLoss}} = 0$  for the age violation threshold  $AVT = 5$ . The results are based on the average performance over 50 realizations.

<b>A<sup>3</sup>L-FEC-FSFB</b>				
k = 3, n = 4	k = 3, n = 5	k = 3, n = 6	k = 4, n = 5	k = 4, n = 6
AV = 0.0075	AV = 0.0168	AV = 0.0224	AV = 0.0052	AV = 0.0093
<b>ACP+</b>				
AV = 0.6580				

Table 4.6: Performance comparison of A<sup>3</sup>L-FEC-FSFB and ACP+ algorithms under packet loss probabilities  $P_{\text{inLoss}} = 0.1$  and  $P_{\text{outLoss}} = 0$  for the age violation threshold  $AVT = 5$ . The results are based on the average performance over 50 realizations.

<b>A<sup>3</sup>L-FEC-FSFB</b>				
k = 3, n = 4	k = 3, n = 5	k = 3, n = 6	k = 4, n = 5	k = 4, n = 6
AV = 0.0025	AV = 0.0087	AV = 0.0163	AV = 0.0036	AV = 0.0096
<b>ACP+</b>				
AV = 0.0041				

Table 4.7: Performance comparison of A<sup>3</sup>L-FEC-FSFB and ACP+ algorithms under packet loss probabilities  $P_{\text{inLoss}} = 0.2$  and  $P_{\text{outLoss}} = 0$  for the age violation threshold  $AVT = 5$ . The results are based on the average performance over 50 realizations.

<b>A<sup>3</sup>L-FEC-FSFB</b>				
k = 3, n = 4	k = 3, n = 5	k = 3, n = 6	k = 4, n = 5	k = 4, n = 6
AV = 0.0023	AV = 0.0089	AV = 0.0140	AV = 0.0008	AV = 0.0070
<b>ACP+</b>				
AV = 0.0140				



Table 4.8: Performance comparison of A<sup>3</sup>L-FEC-FSFB and ACP+ algorithms under packet loss probabilities  $P_{\text{inLoss}} = 0$  and  $P_{\text{outLoss}} = 0.1$  for the age violation threshold  $AVT = 5$ . The results are based on the average performance over 50 realizations.

<b>A<sup>3</sup>L-FEC-FSFB</b>				
k = 3, n = 4	k = 3, n = 5	k = 3, n = 6	k = 4, n = 5	k = 4, n = 6
AV = 0.0083	AV = 0.0164	AV = 0.0274	AV = 0.0073	AV = 0.0103
<b>ACP+</b>				
AV = 0.0218				

Table 4.9: Performance comparison of A<sup>3</sup>L-FEC-FSFB and ACP+ algorithms under packet loss probabilities  $P_{\text{inLoss}} = 0.1$  and  $P_{\text{outLoss}} = 0.1$  for the age violation threshold  $AVT = 5$ . The results are based on the average performance over 50 realizations.

<b>A<sup>3</sup>L-FEC-FSFB</b>				
k = 3, n = 4	k = 3, n = 5	k = 3, n = 6	k = 4, n = 5	k = 4, n = 6
AV = 0.0025	AV = 0.0095	AV = 0.0200	AV = 0.0031	AV = 0.0101
<b>ACP+</b>				
AV = 0.0143				

Table 4.10: Performance comparison of A<sup>3</sup>L-FEC-FSFB and ACP+ algorithms under packet loss probabilities  $P_{\text{inLoss}} = 0.2$  and  $P_{\text{outLoss}} = 0.1$  for the age violation threshold  $AVT = 5$ . The results are based on the average performance over 50 realizations.

<b>A<sup>3</sup>L-FEC-FSFB</b>				
k = 3, n = 4	k = 3, n = 5	k = 3, n = 6	k = 4, n = 5	k = 4, n = 6
AV = 0.0017	AV = 0.0087	AV = 0.0153	AV = 0.0030	AV = 0.0084
<b>ACP+</b>				
AV = 0.0459				

Table 4.11: Performance comparison of A<sup>3</sup>L-FEC-FSFB and ACP+ algorithms under packet loss probabilities  $P_{\text{inLoss}} = 0$  and  $P_{\text{outLoss}} = 0.2$  for the age violation threshold  $AVT = 5$ . The results are based on the average performance over 50 realizations.

<b>A<sup>3</sup>L-FEC-FSFB</b>				
k = 3, n = 4	k = 3, n = 5	k = 3, n = 6	k = 4, n = 5	k = 4, n = 6
AV = 0.0076	AV = 0.0146	AV = 0.0314	AV = 0.0074	AV = 0.0133
<b>ACP+</b>				
AV = 0.0189				

Table 4.12: Performance comparison of A<sup>3</sup>L-FEC-FSFB and ACP+ algorithms under packet loss probabilities  $P_{\text{inLoss}} = 0.1$  and  $P_{\text{outLoss}} = 0.2$  for the age violation threshold  $AVT = 5$ . The results are based on the average performance over 50 realizations.

<b>A<sup>3</sup>L-FEC-FSFB</b>				
k = 3, n = 4	k = 3, n = 5	k = 3, n = 6	k = 4, n = 5	k = 4, n = 6
AV = 0.0034	AV = 0.0113	AV = 0.0288	AV = 0.0096	AV = 0.0119
<b>ACP+</b>				
AV = 0.0476				

Table 4.13: Performance comparison of A<sup>3</sup>L-FEC-FSFB and ACP+ algorithms under packet loss probabilities  $P_{\text{inLoss}} = 0.2$  and  $P_{\text{outLoss}} = 0.2$  for the age violation threshold  $AVT = 5$ . The results are based on the average performance over 50 realizations.

<b>A<sup>3</sup>L-FEC-FSFB</b>				
k = 3, n = 4	k = 3, n = 5	k = 3, n = 6	k = 4, n = 5	k = 4, n = 6
AV = 0.0070	AV = 0.0118	AV = 0.0164	AV = 0.0279	AV = 0.0154
<b>ACP+</b>				
AV = 0.0518				

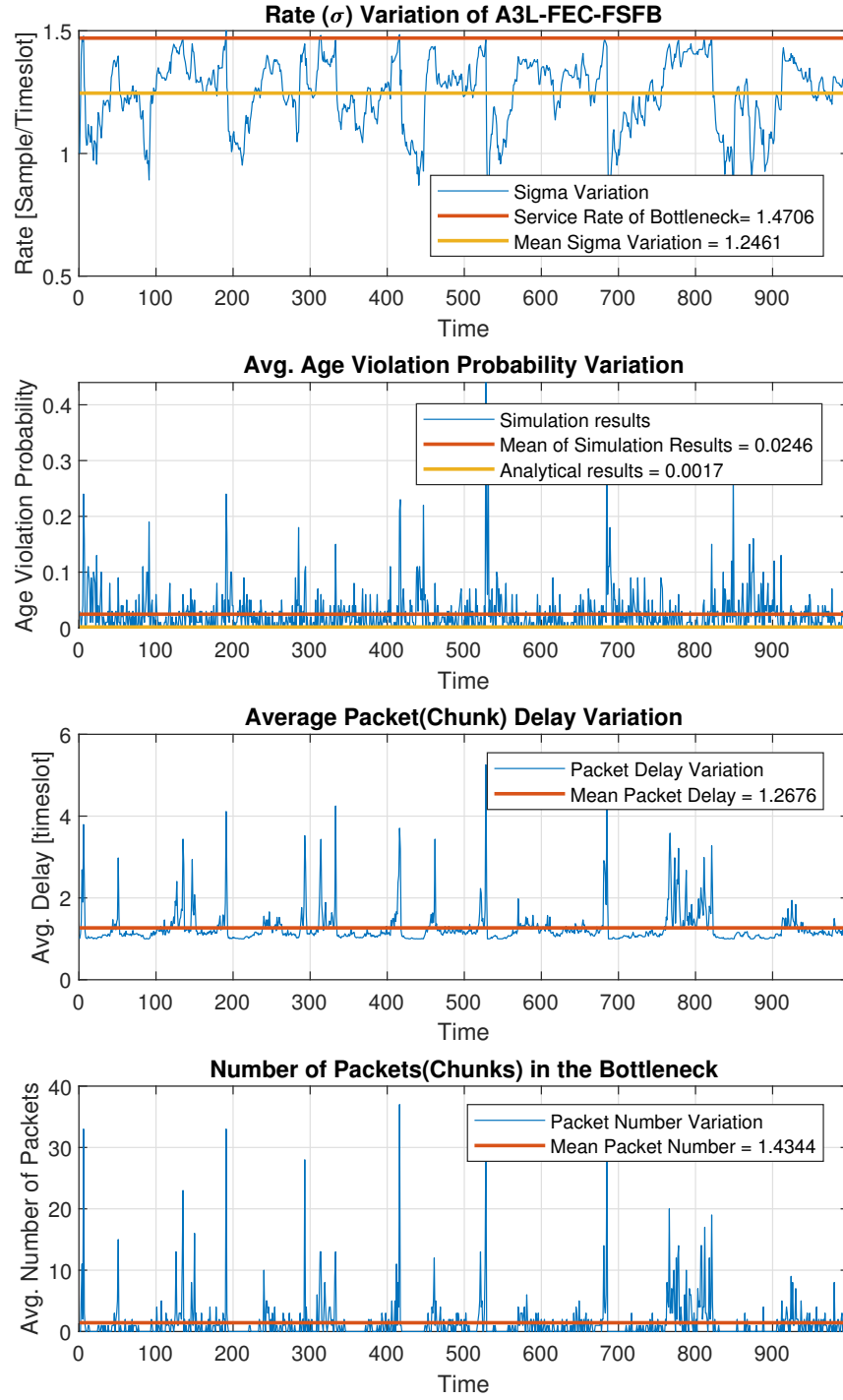


Figure 4.3: Simulation results for A<sup>3</sup>L-FEC-FSFB protocol with an age violation threshold of 5 ( $AVT = 5$ ) when  $P_{inLoss} = P_{outLoss} = 0.2$ . The first figure illustrates the variation in transmission rates, the second shows the average age violation probability, the third displays the average delay of received packets (in chunks), and the fourth depicts the number of packets (in chunks) present in the queue at the end of each monitoring interval.

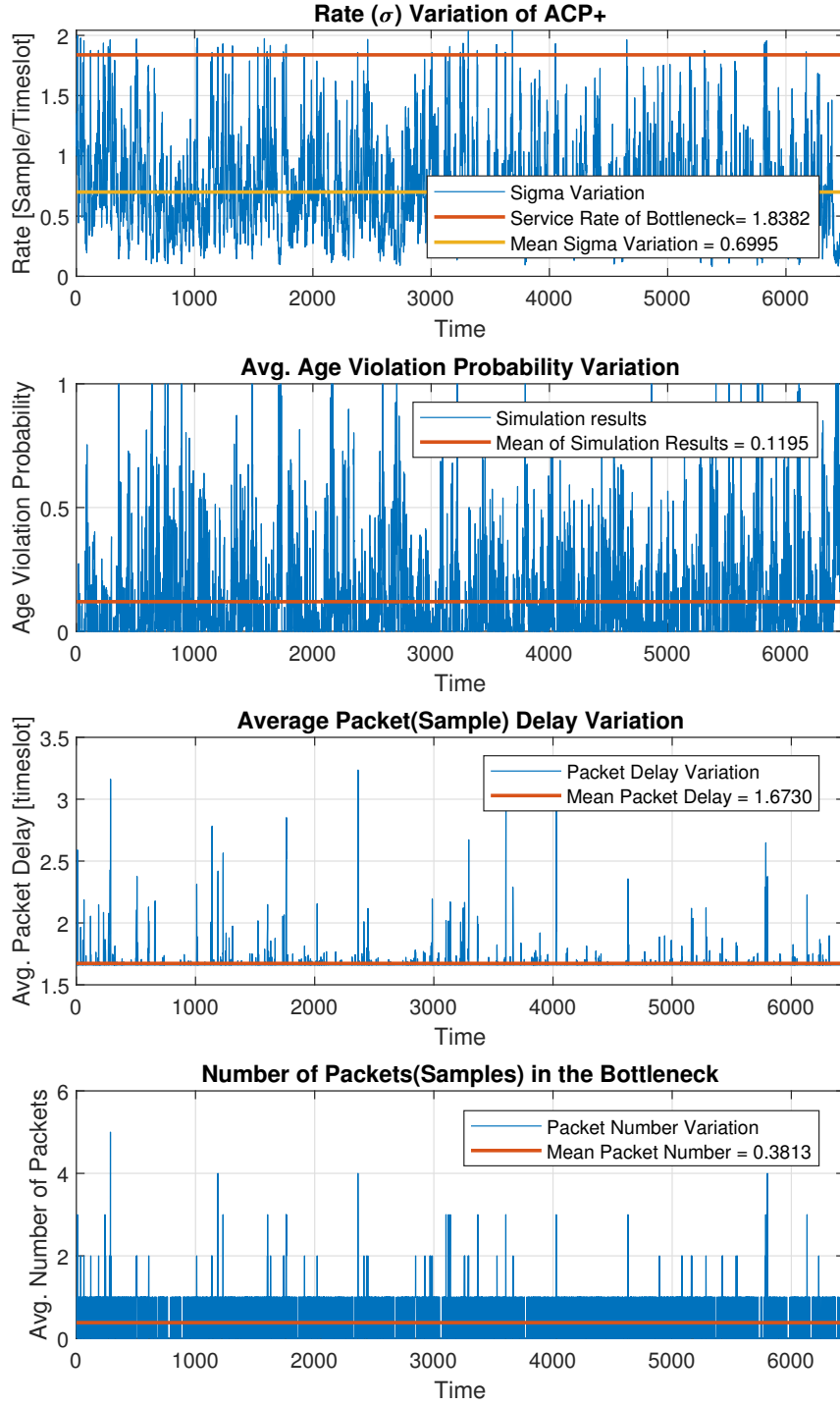


Figure 4.4: Simulation results for ACP+ protocol with an age violation threshold of 5 ( $AVT = 5$ ) when  $P_{inLoss} = P_{outLoss} = 0.2$ . The first figure illustrates the variation in transmission rates, the second shows the average age violation probability, the third displays the average delay of received packets (in samples), and the fourth depicts the number of packets (in samples) present in the queue at the end of each monitoring interval.

We conducted further research to comprehend why the ACP+ protocol performs better than A<sup>3</sup>L-FEC-FSFB in some scenarios. Our analysis focused on investigating the rate changes and the average number of packets in the network bottleneck. In these particular cases, we found that the transfer rate of A<sup>3</sup>L-FEC-FSFB at the beginning of the session was often higher than the bottleneck service rate. As a result, a considerable number of packets became queued in the bottleneck in the early times of the transmission session. As a result, the packet delay increased and led to age violations.

In response to these unpleasant age violation events, the A<sup>3</sup>L-FEC-FSFB endeavors to alleviate congestion by reducing the transmission rate in the next monitoring interval. Although A<sup>3</sup>L-FEC-FSFB is successful in reducing the number of packets in the queue and reducing the packet delay through rapid rate reduction, the initial high level of age violations has a negative impact on the overall value of age violations.

A possible solution to address this problem is to explore adaptive mechanisms in the A<sup>3</sup>L-FEC-FSFB algorithm that can actively handle the initial high level of age violation and improve its overall performance. This mechanism aims to ensure that the initial rate is nearer to the bottleneck capacity and not more than it, and consequently reduce initial congestion and minimize subsequent age violations. One strategy that can be employed is utilizing a slow start feature during the rate initialization phase. In the slow-start mechanism, the transfer rate starts at a low value and gradually increases over time until it reaches the desired level. This method permits the algorithm to measure network conditions and adjust the transmission rate accordingly, avoiding sudden bursts of traffic that can lead to congestion in the network.

In the A<sup>3</sup>L-FEC-FSFB context, the rate initialization process can be revised to accommodate the slow start feature. Instead of initiating at a fixed high transmission rate, the algorithm would initially send packets at a lower rate, allowing it to assess network capacity and the presence of any congestion. As the algorithm monitors network conditions and observes lower age violations and low queue length, it can gradually increase the transmission rate to improve the data freshness at the receiver side.

Another solution to handle the issue of high initial age violation and subsequent performance degradation is to decrease the duration of the monitor interval and sub-

sequently increase the number of feedbacks in the A<sup>3</sup>L-FEC-FSFB algorithm. By reducing the duration of the monitor interval, the algorithm can receive more frequent measurements of network conditions. This feature allows faster detection and response to changes in congestion levels and enables the A<sup>3</sup>L-FEC-FSFB algorithm to adjust transmission rates more quickly. A shorter monitor interval facilitates faster feedback and adaptation, as it diminishes the time required for the algorithm to evaluate the network condition and make transmission rate adjustments. With frequent updates, the A<sup>3</sup>L-FEC-FSFB algorithm can maintain an optimal transmission rate that is close to the bottleneck capacity.

By taking these ideas into account, we have made updates to the algorithm and introduced a new variant called A<sup>3</sup>L-FEC-VSVB as described in Section 3.4.

#### **4.2.2.2 Results of Effect of Coding Rate on the Performance of A<sup>3</sup>L-FEC-FSFB on MATLAB**

To evaluate the effect of the coding rate on the performance of the proposed algorithm we set up a series of simulations. The simulations were performed based on a system model similar to that described in Section 3.3. Transmissions are done through a queue using a FCFS discipline with deterministic service rates,  $k \times 1.4706$  packets per time slot. The simulation duration was set to  $10^5$  time slots, while the monitor intervals (MI) had a duration of  $\tilde{T} = 10^2$  time slots. The simulation parameters, presented in Table 4.14, were carefully selected to provide meaningful insight into the performance of the algorithms. Moreover, this simulation is done for two different age violation thresholds,  $AVT = 2$  and  $AVT = 5$ , under different packet loss probabilities. The packet loss probabilities, denoted by  $P_{\text{inLoss}}$  and  $P_{\text{outLoss}}$ , represent the probability of packet loss before and after entering the bottleneck, respectively. The experimental setup consisted of performing each scenario 50 times and averaging them to ensure statistical significance and reduce the effect of random variation. By repeating and averaging, it is possible to catch a more reliable and representative understanding of the system's performance. Note that for each realization, the testbed was subjected to identical test conditions.

The effect of different coding rates on the system performance is illustrated in Figures

Table 4.14: MATLAB Simulation Parameters for Evaluating the Effect of Coding Rate on the Performance of A<sup>3</sup>L-FEC-FSFB

<b>Age Violation Threshold</b>	2, 5 [time slot]
<b>Bottleneck Queue Size</b>	5000 [chunk]
<b>Queue service rate</b>	$k \times 1.4706 \left[ \frac{\text{packets}}{\text{timeslot}} \right]$
$P_{inLoss}$	0, 0.1, 0.2
$P_{outLoss}$	0, 0.1, 0.2
<b>Propagation Delay</b>	1 [time slot]
<b>Monitor Intervals (MI) Duration</b>	$10^2$ [time slot]
<b>Simulation Duration</b>	$10^5$ [time slot]
<b>k for A<sup>3</sup>L-FEC-FSFB</b>	2, 3, 4
<b>n for A<sup>3</sup>L-FEC-FSFB</b>	2, 3, 4, 5, 6, 7, 8, 9

4.5 to 4.10. Results show that picking a suitable coding rate, which is shown by the green point, significantly enhances the age violation performance of the system. This effect is especially evident when the system faces a high packet loss probability.

The importance of coding rates can be attributed to their ability to provide redundancy in error-prone networks. By introducing redundancy, the receiver can recover the message without having to completely resend the entire data by the transmitter. However, including additional data adds additional load to the network and results in buffer queue growth. This congestion in the network leads to an increase in packet delays and consequently age violations on the client side. Hence, optimizing the coding rate is very important to minimize age violations and increase system performance.

For the scenarios with low packet loss probability, the need for redundancy may seem less critical. However, it is necessary to note that even in such cases, there is still the potential for occasional packet loss due to network congestion or transmission errors. By choosing an appropriate encoding rate, the system can proactively handle these rare cases of packet loss and provide fresh data delivery.

According to these explanations, we have made updates to the algorithm and introduced a new variant called A<sup>3</sup>L-FEC-VSVB as described in Section 3.4.

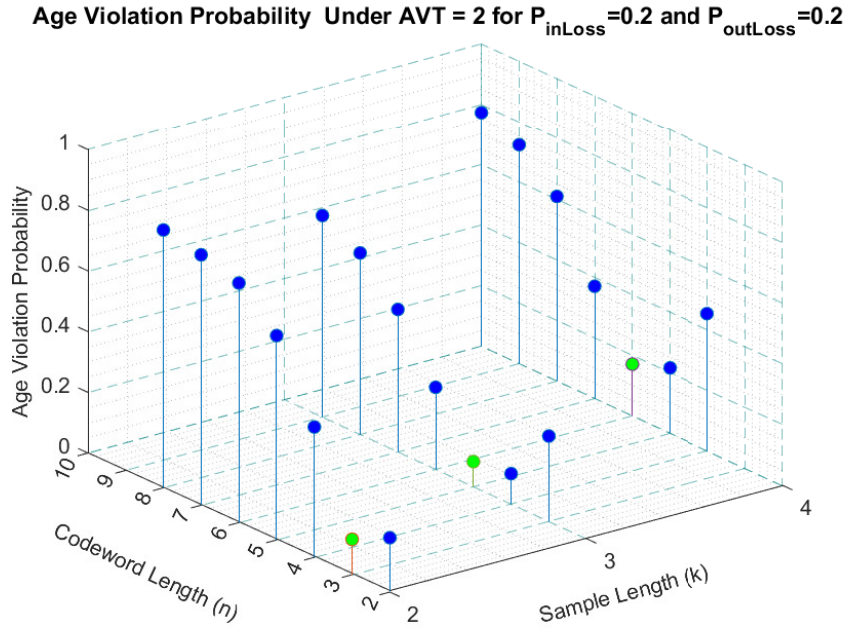


Figure 4.5: Simulation results for age violation with threshold  $AVT = 2$  under different coding rates when  $P_{inLoss} = 0.2$  and  $P_{outLoss} = 0.2$ .

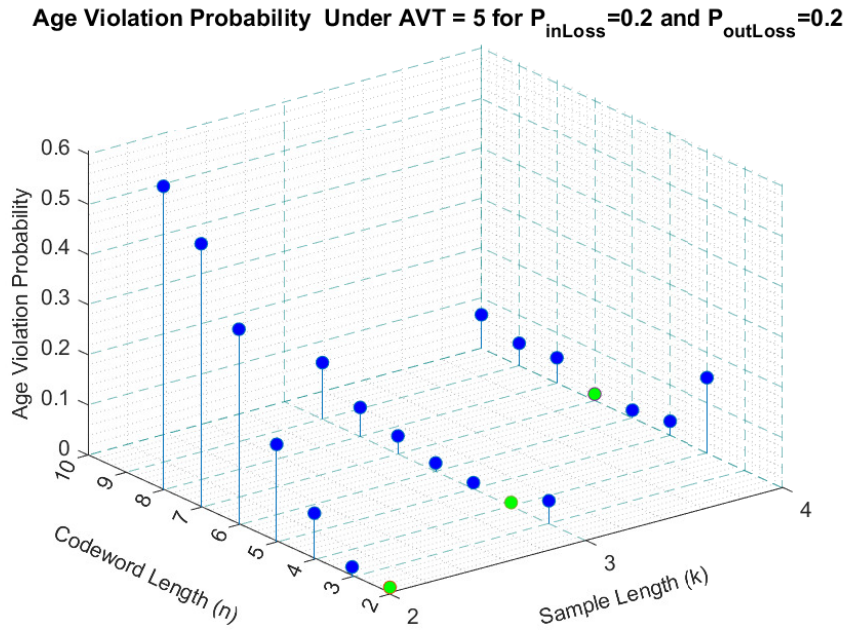


Figure 4.6: Simulation results for age violation with threshold  $AVT = 5$  under different coding rates when  $P_{inLoss} = 0.2$  and  $P_{outLoss} = 0.2$ .



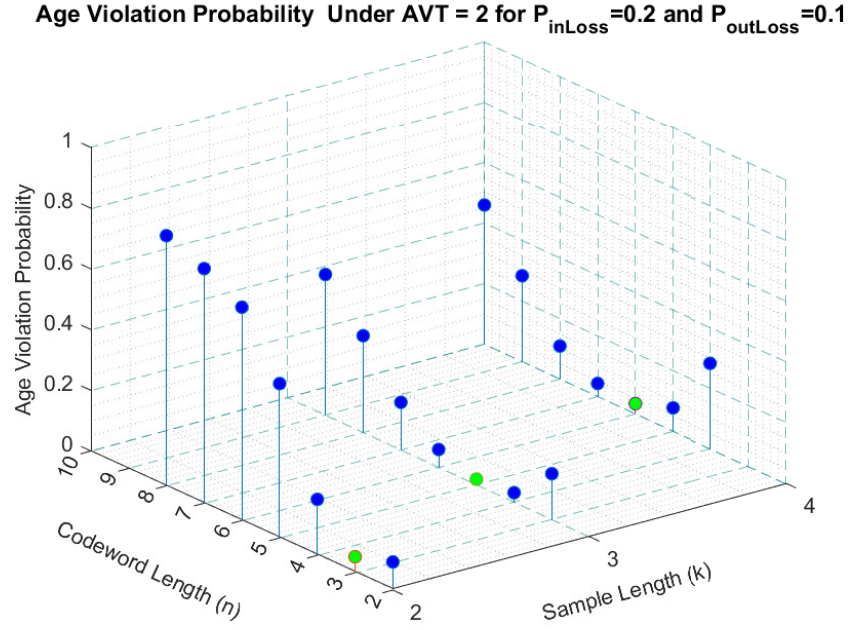


Figure 4.7: Simulation results for age violation with threshold  $AVT = 2$  under different coding rates when  $P_{inLoss} = 0.2$  and  $P_{outLoss} = 0.1$ .

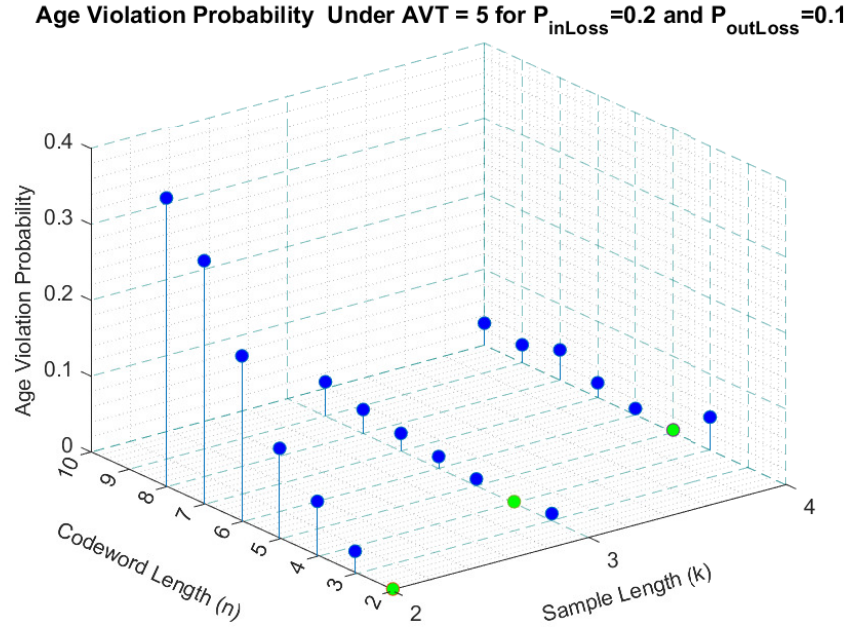


Figure 4.8: Simulation results for age violation with threshold  $AVT = 5$  under different coding rates when  $P_{inLoss} = 0.2$  and  $P_{outLoss} = 0.1$ .

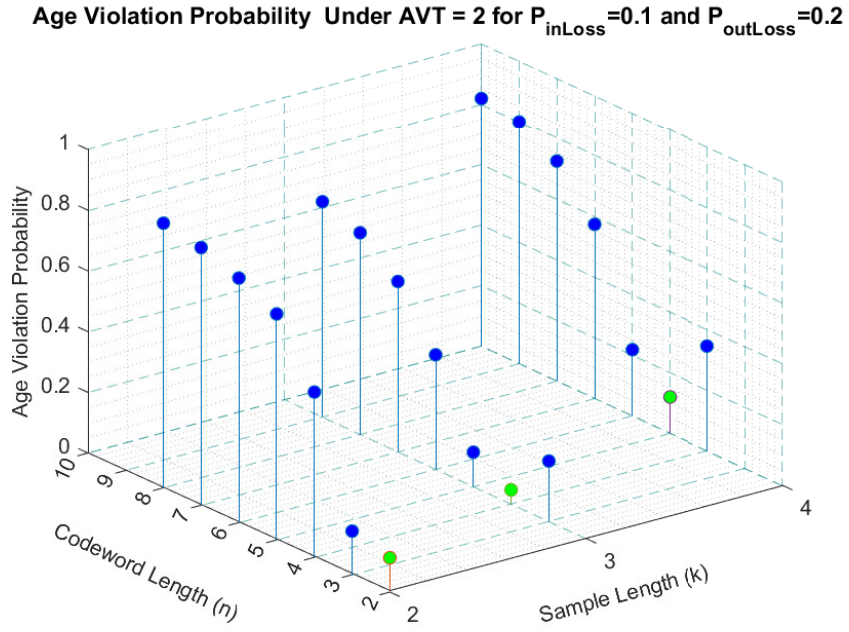


Figure 4.9: Simulation results for age violation with threshold  $AVT = 2$  under different coding rates when  $P_{inLoss} = 0.1$  and  $P_{outLoss} = 0.2$ .

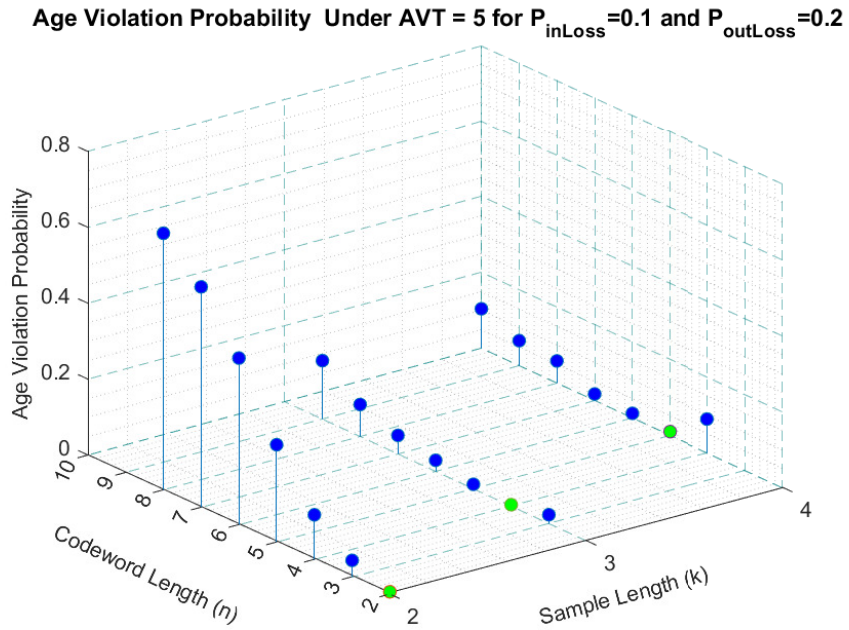


Figure 4.10: Simulation results for age violation with threshold  $AVT = 5$  under different coding rates when  $P_{inLoss} = 0.1$  and  $P_{outLoss} = 0.2$ .

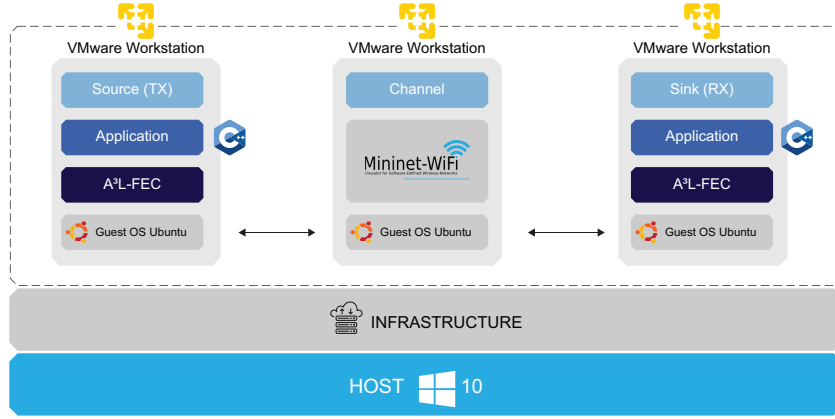


Figure 4.11: Emulator testbed.

### 4.3 Emulation-based Comparison of A<sup>3</sup>L-FEC-VSVB and TCP-BBR congestion control on Mininet-WiFi

This section provides emulation results for the proposed age-aware congestion control algorithm, A<sup>3</sup>L-FEC-VSVB, which is presented in 3.4. Moreover, based on the results of Section 4.1 since the BRR congestion control algorithm presents better performance than other TCP variants in terms of age violation probability, we compare A<sup>3</sup>L-FEC-VSVB results with TCP BBR congestion control algorithm.

In this research, we used the Mininet-WiFi, an extension of the Mininet emulator, to emulate an end-to-end network topology, Figure 4.11. Mininet is a versatile open-source network emulator that can create virtual networks on a single computer while providing a lightweight and scalable method for simulating complicated network topologies. It allows researchers to validate network configurations, applications, and services, supports SDN development, enables simulation and emulation of real-world scenarios, facilitates prototyping and proof-of-concept, and supports network troubleshooting and debugging. This emulator uses Linux container technology, known as network namespaces, to create virtual network nodes, while each node can run its own operating system instance and network stack. This feature allows the user to simulate a variety of network devices such as switches, routers, and hosts. Moreover, it is possible to connect these virtual nodes together to form arbitrary network topology and then emulate complex network scenarios. Overall, Mininet presents a cost-effective and flexible platform for a wide range of network tasks.

The emulations of this section are carefully planned to contain various aspects and scenarios related to age-aware congestion control, such as different age violation thresholds and different packet sizes. By conducting detailed inspections, meaningful conclusions about the performance and suitability of A<sup>3</sup>L-FEC-VSVB and TCP-BBR in different network conditions are obtained. The experimental setup consisted of performing each scenario 3 times and averaging them to ensure statistical significance and reduce the effect of random variation. By repeating and averaging, it is possible to catch a more reliable and representative understanding of the system's performance. Note that for each realization, the testbed was subjected to identical test conditions. Before going into the details of this emulation, let us provide some information about TCP optimal operation point and the TCP-BBR in Sections 4.3.1 and 4.3.2 respectively. It is worth noting that the comprehensive description of A<sup>3</sup>L-FEC-VSVB has already been presented in Section 3.4.1.

#### **4.3.1 The optimal operating point of TCP**

The total capacity of any data network is limited by the part with the lowest available bandwidth along the communication path, which is usually called a bottleneck. This bottleneck constrains the overall throughput of the connection. Assuming a link connecting the transmitter and receiver, the delay of this link is determined by summing all propagation delays, while the bandwidth is adjusted according to the bottleneck capacity, (BtlBw). The time it takes for a packet to travel from the sender to the receiver and back to the sender is named Round Trip Time (RTT) and the round trip propagation delay is shown by RTprop. For these types of networks, the bandwidth delay (BDP) is calculated by multiplying the available bandwidth of the network link by the round-trip time (RTT) of the connection and refers to a measure of the amount of data that can be in transit on the network path at any given time. In other words, BDP defines the maximum amount of unacknowledged data that can be in flight without experiencing congestion. BDP is also cited as Kleinrock's optimum performance point, [114].

Figure 4.12 shows the impacts of increasing inflight data on the delivery rate and RTT. If transmitted data is less than BDP inflight, it means there is no congestion and RTT

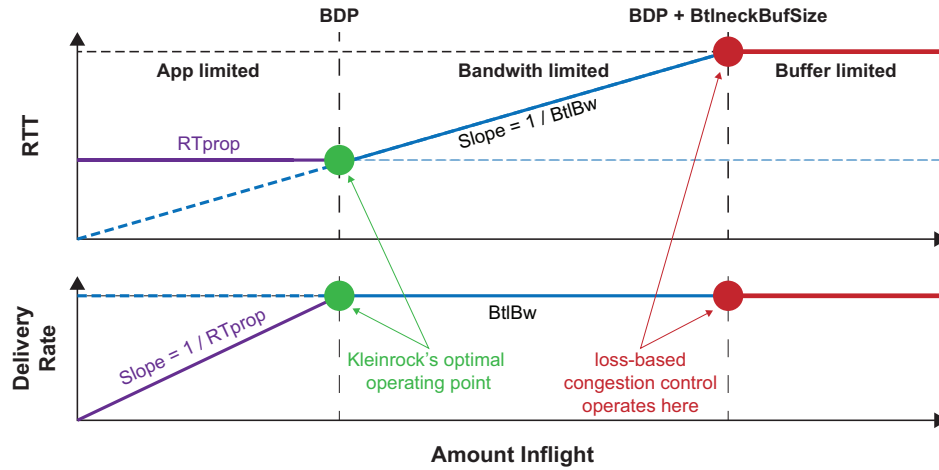


Figure 4.12: Delivery rate and RTT vs. inflight, [85].

is equal to  $RT_{prop}$  (application limited). The delivery rate is directly proportional to the send rate but is maximized when the inflight data reaches the BDP at the Kleinrock point. Persisting to increase the inflight further causes packets to reach the bottleneck faster than they can be forwarded to the destination. This condition leads to a queue in bottleneck being filled, causing an additional delay that grows proportionally with the amount of inflight data. This behavior is recognized and controlled by delay-based algorithms.

When the number of packets in flight matches the sum of the BDP and the Bottleneck Buffer Size, the bottleneck buffer starts to discard packets. This action of the system limits the Round Trip Time (RTT) to a certain threshold. Note that, this approach is identified and managed by loss-based algorithms. This explanation shows that both delay-based and loss-based algorithms perform beyond the optimal operating point proposed by Kleinrock.

### 4.3.2 Overview of the TCP BBR Congestion Control

TCP BBR (Bottleneck Bandwidth and Round-trip time), [85], is a congestion control algorithm introduced by Google. The design idea of this protocol is to improve network congestion control and bandwidth utilization for TCP connections. The ultimate objective of BBR is achieving high network throughput and low latency by dynamically estimating the network capacity and adjusting the transmission rate accordingly.

Traditional TCP congestion control algorithms rely on packet loss or packet delays as an indicator of congestion in the network, explained in Section 4.3.1. When packet loss occurs or packet delay increases, these algorithms assume that the network is congested and try to mitigate this event by reducing sending rate. While BBR by using measurements of available bandwidth and round-trip time (RTT), estimates network congestion and builds a model of the network bottleneck, and then adjusts the transmission rate based on this model. Indeed, BBR uses the BDP as a key reference point for optimizing network performance and utilizes a multi-phase strategy to optimize network performance: Startup, Drain, Probe Bandwidth, and Probe RTT phases.

In the Startup phase, this algorithm begins with a conservative sending rate, then gradually increases the rate, doubling the sending rate with each round-trip, until it reaches a BDP-based threshold to estimate available bandwidth and RTT. Once the BDP-based threshold is reached it is assumed it has reached the bottleneck bandwidth. At this step the drain phase of BBR begins. Because due to the one round-trip time delay (RTT) in making this observation, a queue is already formed at the bottleneck of the network. Therefore, BBR attempts to alleviate this situation by temporarily reducing the pacing gain to drain the queue. Later, BBR goes into the Probe Bandwidth phase, during which it conducts probing to discover more available bandwidth. The BBR continuously measures the bandwidth and takes the highest sampled value as the bottleneck bandwidth estimator, which remains valid for ten round-trip time propagation (RTprop) delays. Once the BBR does not measure a new round trip time propagation delay (RTprop) value for ten seconds, it stops probing for bandwidth and starts the Probe RTT phase. At this step, the bandwidth is reduced to four packets to flush any potential queues and facilitate accurate RTT estimation. The Probe RTT phase takes 200 ms plus one RTT. If a new minimum RTprop measurement is recorded in this phase, it will be updated and remain valid for ten seconds.

BBR has the ability to detect different network modes and adjust its sending rate accordingly helping to optimize TCP traffic and improve overall network performance. These modes are represented in Figure 4.12 and named as "app-limited," "bandwidth-limited," and "buffer-limited". Each of these modes describes a specific state in the network and is described as follows:

- **App-Limited mode:**

In the app-limited mode, the application itself limits its sending rate, rather than by the network capacity. This means that the application is not generating data fast enough to fully utilize the available network bandwidth. The absence of packet loss or increased round-trip time gives the signal of no congestion in the network to the BBR and this algorithm detects the App-Limited mode.

Under this condition, the BBR doesn't try to further increase the sending rate and keeps the sending rate at the level allowed by the application's data generation rate. As a result, it prevents unnecessary congestion on the network and ensures the application gets the best possible performance given its limitations.

- **Bandwidth-Limited:**

If the network path between the sender and receiver has limited capacity, the network's available bandwidth becomes the bottleneck for data transmission. Therefore, the sending rate is constrained by this capacity.

BBR detects bandwidth-limited mode by observing that the round-trip time (RTT) is relatively stable, therefore it indicates that the network buffer has not filled up yet. In this situation, the BBR tries to make full use of the available bandwidth without causing excessive queuing delay or congestion by steadily increasing the sending rate until it reaches the maximum bandwidth allowed by the network. Indeed BBR by checking and adjusting the transmission rate based on the Bandwidth Delay Product (BDP), ensures efficient use of available bandwidth while maximizing the delivery rate and maintaining a low RTT.

- **buffer-Limited:**

In this case, the receiver or router buffer acts as the bottleneck of the network, because it has limited room to store the received packets. When the buffer is completely filled with packets, it causes queuing delays and can negatively affect network performance.

BBR detects buffer-limited conditions by observing an increase in round-trip time (RTT) and a decrease in delivery rate, indicating congestion in the receiver's buffer. In this condition, BBR decreases the sending rate to avoid overfilling the buffer, while maintaining low queuing delay and stable RTT.

#### 4.3.3 Emulation Testbed Setup on Mininet-WiFi for Age Violation Evaluation of A<sup>3</sup>L-FEC-VSVB and TCP-BBR congestion control

To evaluate the A<sup>3</sup>L-FEC algorithm, a network emulation was set up using a WiFi-Mininet emulator. To facilitate data transmission between a client and a server, custom C++ code was designed and developed for both ends. The heart of this experiment lies in integrating the A<sup>3</sup>L-FEC algorithm into custom C++ code. A<sup>3</sup>L-FEC data flow control algorithm operates at the application layer and is designed to optimize data freshness by minimizing age violation occurrence. To achieve this goal, the algorithm operates on top of the User Datagram Protocol (UDP).

The reason to operate A<sup>3</sup>L-FEC at the application layer of the network was driven by specific design considerations and requirements. The application layer is the topmost layer of the seven layers of the OSI (Open Systems Interconnection) model. This layer is responsible for providing network services directly to end-user or applications. Operating at this layer provides several advantages to A<sup>3</sup>L-FEC, such as:

- **Control over data packets:** Operation at the application layer authorizes A<sup>3</sup>L-FEC to have more precise control over the data packets being transmitted from the transmitter to the receiver. This control can process, encode, and decode data packets in a way that is optimized for the needs of specific applications.
- **Independence from lower layers:** A<sup>3</sup>L-FEC is independent of lower network layers such as transport and physical layers in the OSI model. This independence allows for easier integration into different applications without the need for modifications in the lower layers of the network.
- **Application-Specific Optimization:** Different users of the programs may have different necessities and constraints. Functionality at the application layer allows A<sup>3</sup>L-FEC to be tailored and optimized according to specific application needs, providing a more flexible and adaptable solution.
- **Simplified Deployment:** Since A<sup>3</sup>L-FEC runs independently of lower layers in the OSI model, its deployment becomes less complicated. This data flow control algorithm can be easily integrated into available applications without major changes to the network infrastructure.



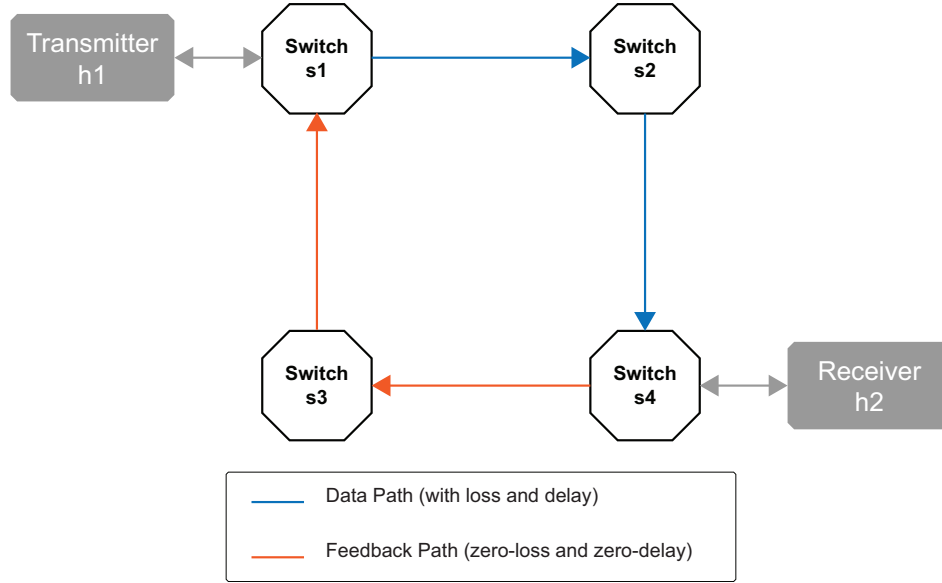


Figure 4.13: End-to-end network topology used for emulation.

- **End-to-End Protection:** A<sup>3</sup>L-FEC can provide end-to-end protection for data transmission between the client and the server. By operating at the application layer of OSI, this algorithm not only can encrypt the packets but also can offer custom forward error correction techniques to provide error resilience, and error correction characteristics to the application; and finally, minimize the need for retransmissions within the application.

In general, by implementing the A<sup>3</sup>L-FEC algorithm in the application layer, the system benefits from advanced control, end-to-end protection, and application-specific optimization, making it a suitable and efficient solution for age-aware applications in communication systems.

The network topology employed to assess the proposed age-aware congestion control algorithm, A<sup>3</sup>L-FEC Protocol, consists of one source node (transmitter), named h1, four switches, and one sink node (receiver), named h2, as depicted in Figure 4.13. This testbed provides an environment to fully evaluate the performance and effectiveness of the A<sup>3</sup>L-FEC algorithm. The selected topology and parameter settings enable us to analyze the behavior of the protocol under various conditions, including bottleneck links, propagation delays, and packet loss scenarios. Moreover, it facilitates the comparison of A<sup>3</sup>L-FEC with TCP-BBR under a controlled environment.

Within this topology, the link between s1 and s4 over s2 serves as a bottleneck of the network, with a bandwidth limited to 1 *Mbps* and a queue size capable of keeping 1000 packets. The remaining links have a higher capacity of 1000 *Mbps*. The propagation delay from the transmitter, (h1), to the receiver, (h2), which traverses through switches s1, s2, and s4, is set to 30 *ms*, while the same parameter for the links from h2 to h1, passing through switches s4, s3, and s1, is set to a 0 *ms* value.

In this testbed, a 5% packet loss rate is experienced on the link between s1 and s4 on s2, while no packet loss occurs on the other links. The codeword transmissions originate from h1 and travel to h2 through switches s1, s2, and s4, using a FCFS discipline. Note that the feedback generated by the A<sup>3</sup>L-FEC data flow controller is returned from h2 to h1 through switches s4, s3, and s1.

#### **4.3.4 Age Violation Evaluation Results of A<sup>3</sup>L-FEC-VSVB and TCP-BBR Congestion Control on Mininet-WiFi**

Figures 4.14, 4.15, and 4.16 present the age violation probability results for the transmission of 500 samples using A<sup>3</sup>L-FEC-VSVB and TCP-BBR. The emulation were conducted with age violation thresholds of 600 ms, 300 ms, and 150 ms, respectively. It is important to note that the sampling period of A<sup>3</sup>L-FEC-VSVB is adaptive, enabling the algorithm to dynamically adjust its sampling rate for optimal performance. On the other hand, the age violation performance of TCP-BBR was evaluated under various fixed sampling rates, including 100 ms, 200 ms, 300 ms, and 600 ms.

The results clearly demonstrate the superiority of A<sup>3</sup>L-FEC-VSVB over TCP-BBR across various age violation threshold levels and sampling periods, especially under different coding rate scenarios. Notably, when the sampling rate of TCP-BBR is set to 100 ms, its performance significantly deteriorates in terms of age violation. Further analysis reveals that the worse performance of TCP-BBR with 100 ms sampling rate can be attributed to the way samples are generated independently of the transmission. As a consequence, these samples are queued in the transmitter's buffer, leading to increased age violations. While this observation holds true for other TCP-BBR experiments as well, the magnitude of aging and age violation is relatively lower compared to the 100 ms sampling rate case.

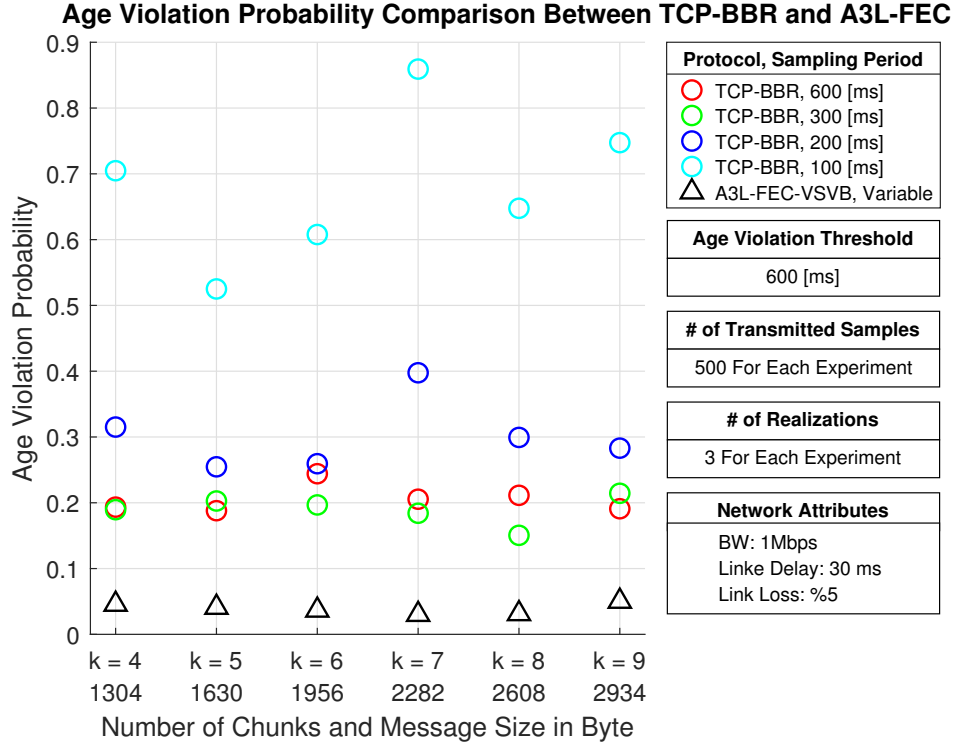


Figure 4.14: Age violation probability under AVT = 600 msec.

In contrast, the A<sup>3</sup>L-FEC-VSVB algorithm ensures more efficient management of data freshness by dynamically adjusting the sampling intervals and blocklengths. By doing so, A<sup>3</sup>L-FEC-VSVB minimizes the number of age violations, providing consistently fresher information at the receiver's end.

Additionally, our observations reveal that by decreasing the age violation threshold, TCP-BBR with a sampling rate of 200 ms exhibits relatively improved performance compared to other TCP-BBR scenarios. However, it is important to emphasize that even in this scenario, A<sup>3</sup>L-FEC-VSVB maintains its superior performance in terms of age violation. Because the adaptability and efficiency of the A<sup>3</sup>L-FEC-VSVB protocol allow it to continually optimize data freshness and mitigate age violations more effectively than TCP-BBR. This capability remains consistent across various age violation threshold levels and different coding rate scenarios.

Overall, these findings highlight the clear advantage of the A<sup>3</sup>L-FEC-VSVB protocol in optimizing age violation performance and reinforcing the importance of its adaptive approach in maintaining data freshness during transmissions.

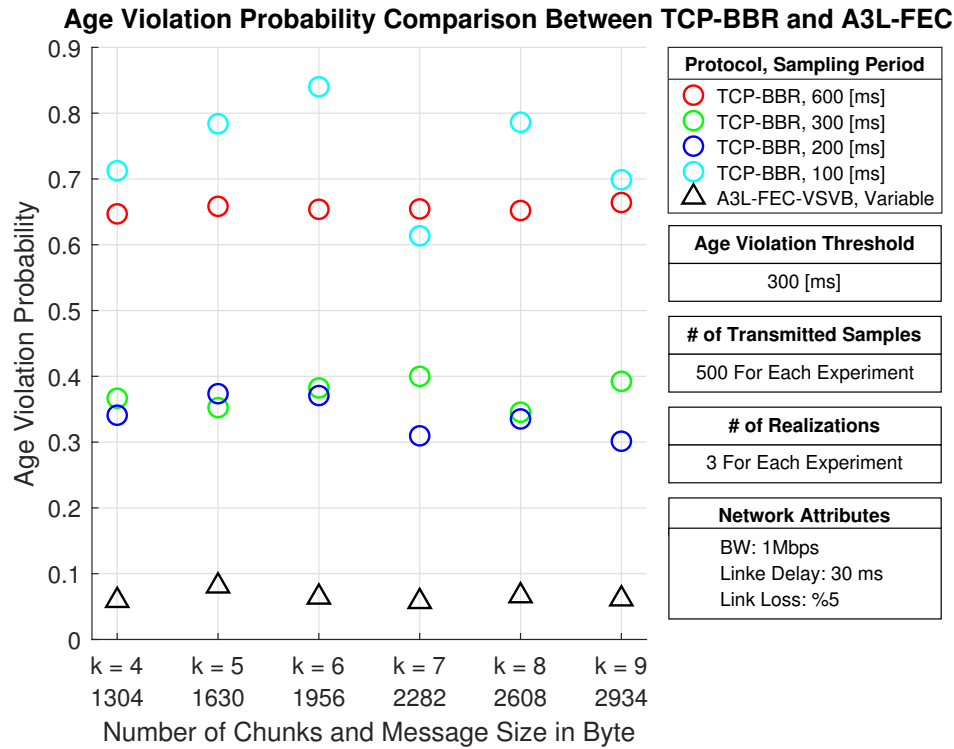


Figure 4.15: Age violation probability under AVT = 300 msec.

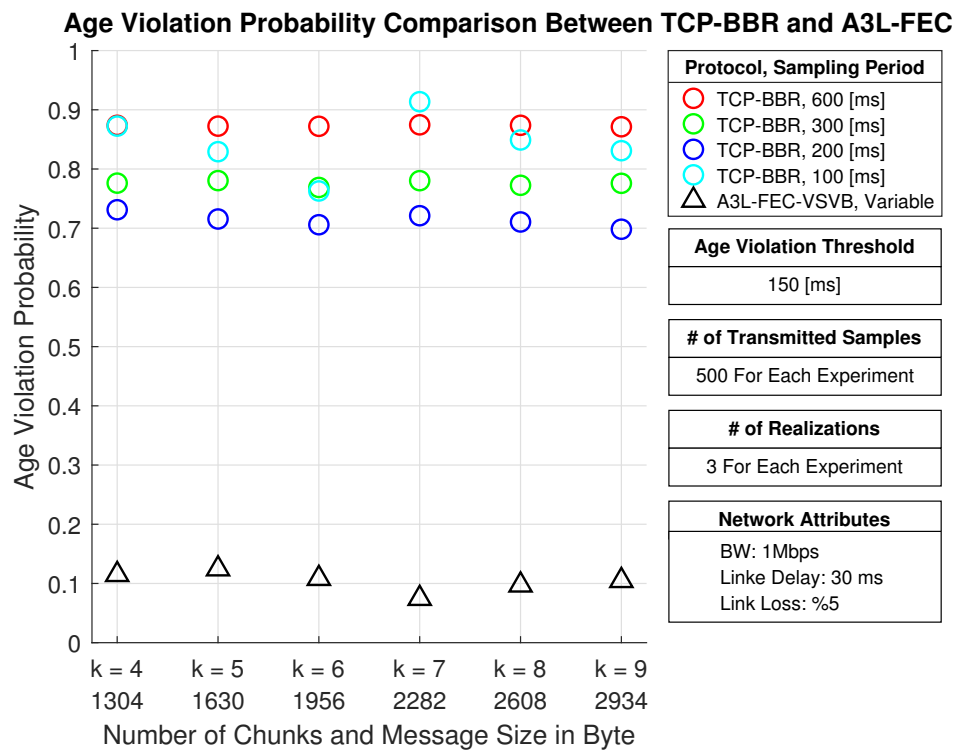


Figure 4.16: Age violation probability under AVT = 150 msec.

#### **4.3.5 AoI Evaluation Results of A<sup>3</sup>L-FEC-VSVB and TCP-BBR Congestion Control on Mininet-WiFi**

The AoI results which are presented in Figures 4.17, 4.18, and 4.19 are based on the series of emulation experiments, that are introduced in Section 4.3.4. These figures provide valuable insights into the performance of A<sup>3</sup>L-FEC-VSVB, and TCP-BBR, under various network parameters and application settings.

The superior AoI performance of A<sup>3</sup>L-FEC-VSVB compared and TCP-BBR is evident across all experiments. This can be attributed to the fundamental differences in the transmission mechanisms utilized by the protocols.

In the case of TCP-BBR, when a sample is generated, it is first located in the transmission buffer. TCP-BBR continuously watches the buffer and starts sending packets to the destination as soon as they become available. However, during this process, new packets may join the transmission queue buffer, leading to a potential increase in packet aging and age violation events.

On the other hand, A<sup>3</sup>L-FEC-VSVB adopts a "generate-at-will" model and employs the "fire-and-forget" pattern technique. Consequently, when a sample is generated under the A<sup>3</sup>L-FEC-VSVB application, it has a low possibility of waiting in the transmitter buffer. Moreover, since A<sup>3</sup>L-FEC-VSVB uses the UDP protocol for transmitting data, it initiates sending the data without the need to establish a reliable connection, thus bypassing unnecessary delays.

The utilization of UDP protocol in A<sup>3</sup>L-FEC-VSVB further enhances its efficiency in delivering fresh and up-to-date information. The absence of connection setup overheads allows A<sup>3</sup>L-FEC-VSVB to minimize transmission delays and reduce the chances of packet aging, ultimately leading to improved AoI performance.

An interesting observation that we observed from the results is that as we increase the sample length (i.e., increasing the chunk number) in A<sup>3</sup>L-FEC-VSVB experiments, the AoI also increases. This behavior is expected, because longer samples take more time to transmit, leading to a higher AoI. However, in contrast, the AoI changes in TCP-BBR experiments exhibit more random behavior.

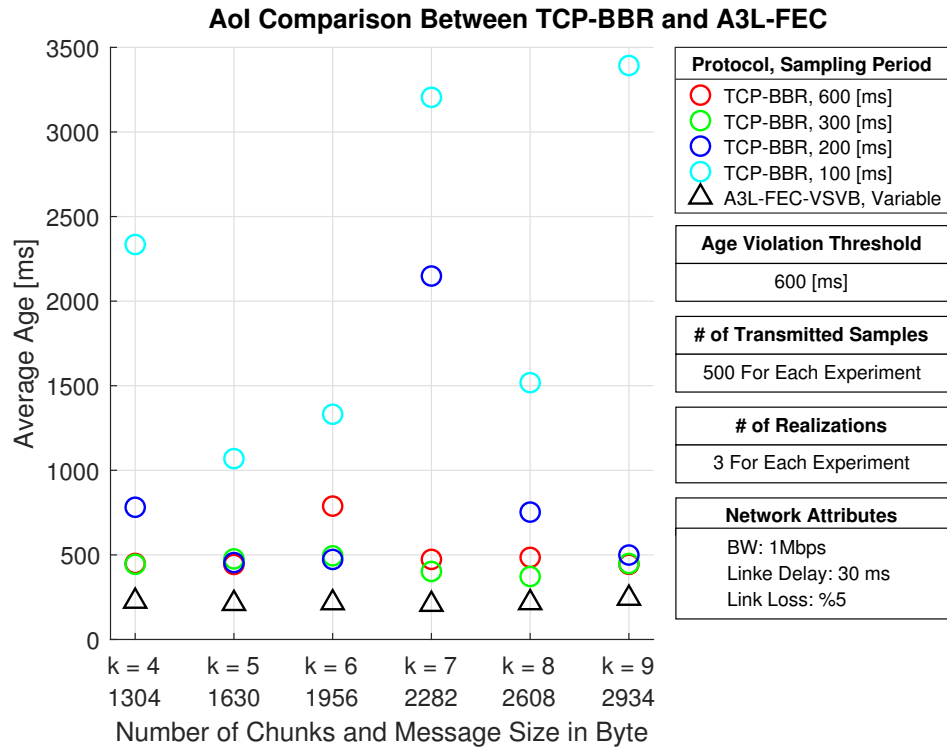


Figure 4.17: AoI under AVT = 600 msec.

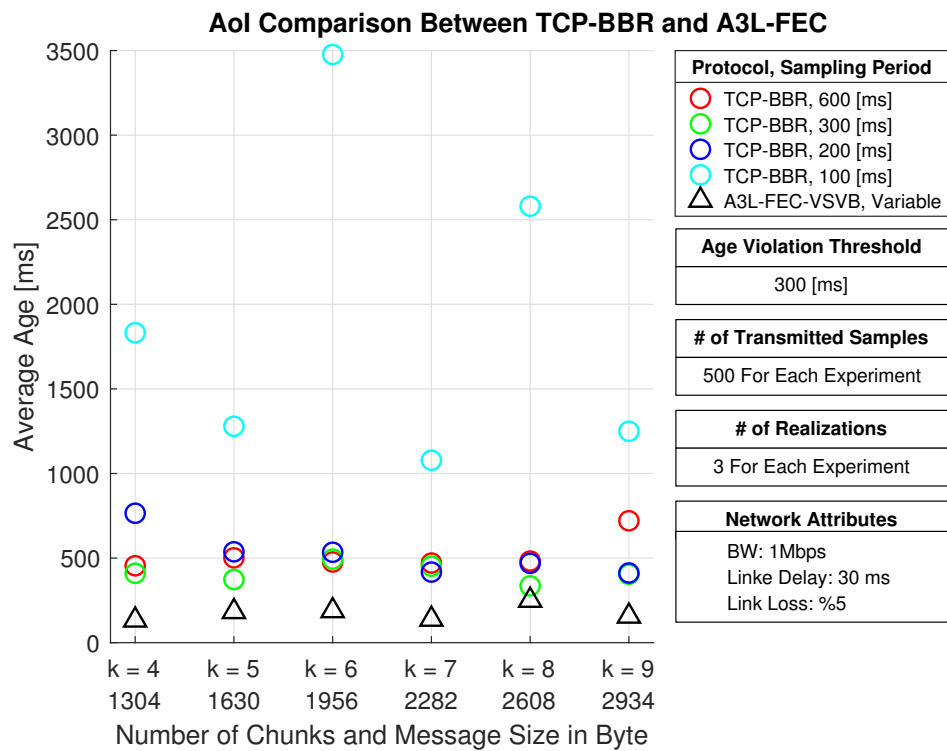


Figure 4.18: AoI under AVT = 300 msec.

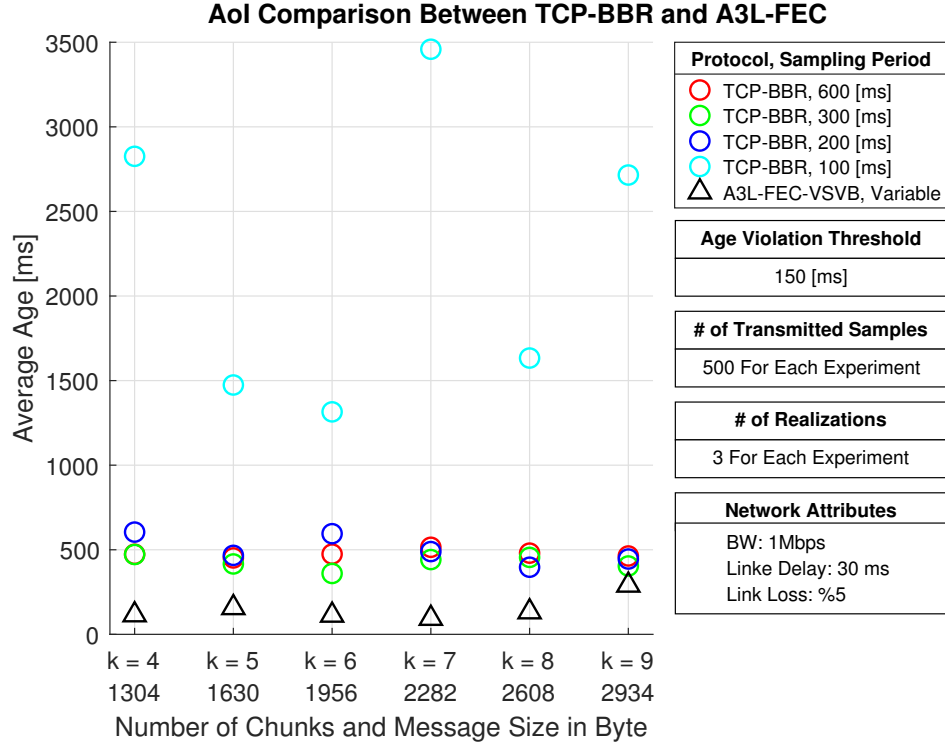


Figure 4.19: AoI under AVT = 150 msec.

This distinction in AoI behavior can be attributed to the underlying mechanisms of the protocols. In A<sup>3</sup>L-FEC-VSVB, as we increase the sample length, more packets need to be transmitted, causing a proportional increase in AoI. This is a predictable outcome as the protocol aims to deliver fresh information while accounting for the transmission time required for longer samples.

On the other hand, TCP-BBR is a classic transport layer protocol that does not consider age awareness. Its behavior is affected by factors such as network congestion, packet losses, and retransmissions. As a result, the AoI variation in this protocol tend to exhibit a more erratic pattern, as it is not explicitly optimized for minimizing AoI.

#### 4.3.6 Packer Delay Evaluation Results of A<sup>3</sup>L-FEC-VSVB and TCP-BBR Congestion Control on Mininet-WiFi

The results illustrated in Figures 4.20, 4.21, and 4.22 provide insights into the average packet delay for transmitting 500 samples under various scenarios.

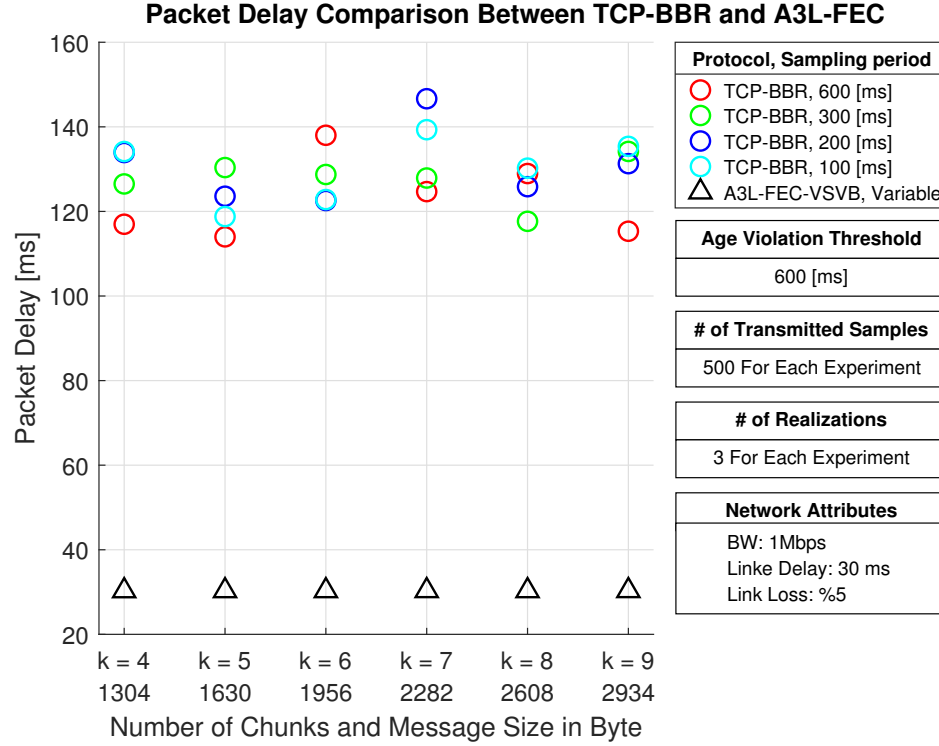


Figure 4.20: Average packet delay under AVT = 600 msec.

It is clear from the results that the average packet delay is various in different congestion control algorithms and age violation thresholds. In the case of A<sup>3</sup>L-FEC-VSVB, the average packet delay tries to exhibit a more consistent behavior compared to TCP-BBR. This consistency can be attributed to the age-aware nature of A<sup>3</sup>L-FEC-VSVB, which optimizes the sample transmission rate to minimize age violation.

The obtained results show that the average packet delay in the application using A<sup>3</sup>L-FEC-VSVB for data transmission is approximately 30ms, which is comparable to the link delay in all scenarios. In contrast, the application using TCP-BBR shows an average packet delay of about 130 ms. This significant difference can be attributed to the underlying mechanisms and algorithms used by the two protocols.

Note that, A<sup>3</sup>L-FEC-VSVB works on top of User Datagram Protocol (UDP) and does not engage in packet retransmission in case of loss. This approach results in faster data delivery, and consequently in less latency. On the other hand, TCP-BBR, which is a reliable transport protocol, uses retransmission mechanisms to recover lost packets. As a result, the retransmission process introduces more delay, which leads



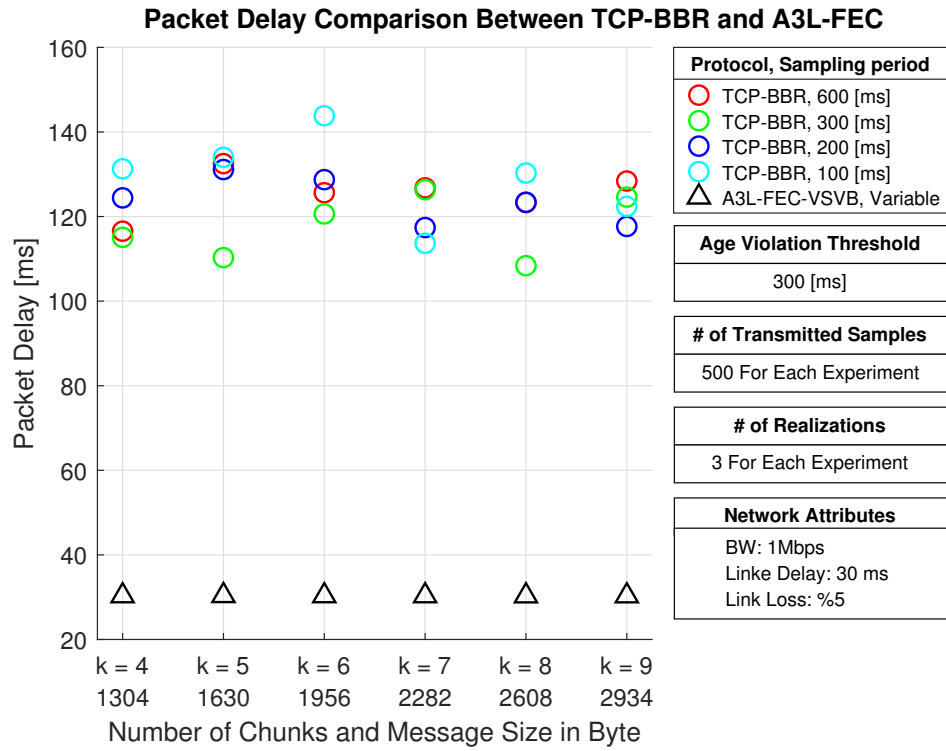


Figure 4.21: Average packet delay under AVT = 300 msec.

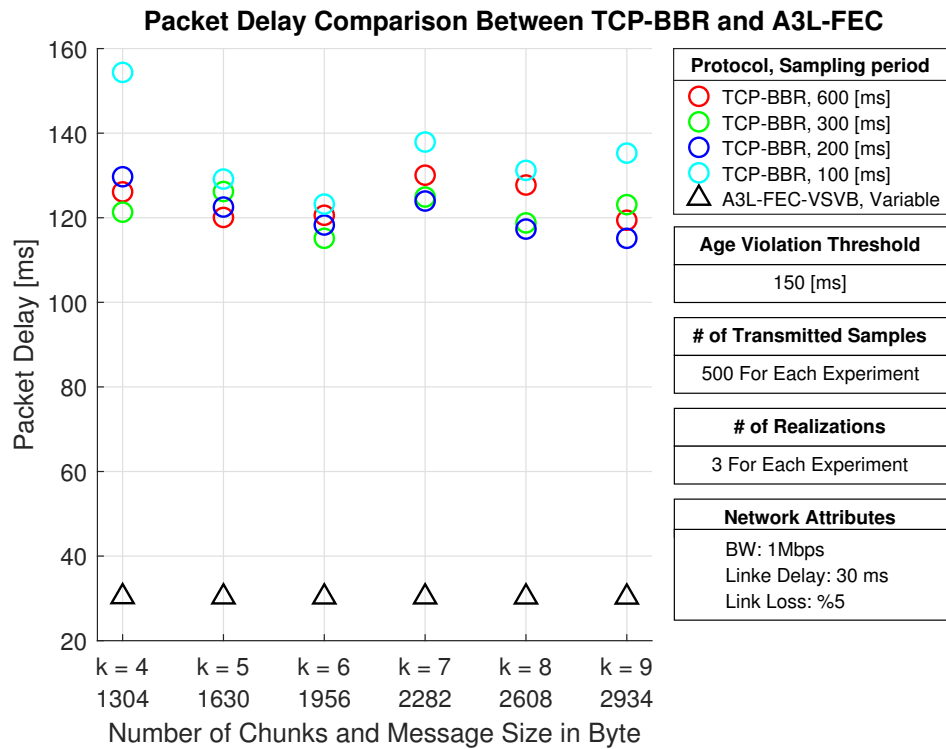


Figure 4.22: Average packet delay under AVT = 150 msec.

to higher packet delays compared to A<sup>3</sup>L-FEC-VSVB.

Moreover, TCP-BBR employs a three-way handshake approach to establish a reliable connection before initiating transmission, which in turn it adds additional setup time to the process. In contrast, A<sup>3</sup>L-FEC-VSVB adopts the fire-and-forget pattern, eliminating the need for connection establishment and diminishing unnecessary delays.

The ability of A<sup>3</sup>L-FEC-VSVB to provide reliable data transmission with a lower average packet delay can be beneficial for time-sensitive applications, where minimizing delay is necessary. Its efficient utilization of UDP and avoidance of retransmissions make it a promising choice for delay-sensitive communication applications.

#### **4.3.7 Transmission Rate Evaluation Results of A<sup>3</sup>L-FEC-VSVB and TCP-BBR Congestion Control on Mininet-WiFi**

Figures 4.23, 4.24, and 4.25 depict the average transmission rate results acquired from the emulation of 500 samples using two distinct congestion control algorithms: A<sup>3</sup>L-FEC-VSVB and TCP-BBR. The emulations of this experiment also were conducted under age violation thresholds of 600 ms, 300 ms, and 150 ms, respectively.

The outcomes reveal that the average transmission rate of TCP-BBR is somewhat fixed under different age violation thresholds, but it directly correlates with the sampling period. For instance, when the sampling period of the TCP-BBR application is set to 100 ms, the transmission rate for all scenarios remains around 10 packets per second. Similarly, when the sampling period is set to 200 ms, 300 ms, and 600 ms the transmission rate for all scenarios remains around 5, 3.2, and 1.8 packets per second respectively. On the other hand, the behavior of A<sup>3</sup>L-FEC-VSVB becomes more dynamic as the age violation threshold is reduced. For example, when the age violation threshold is set to 600 ms, the transmission rate is about 3.5 packets per second. However, when the age violation threshold is decreased to 300 ms and 150 ms, the transmission rates increase to approximately 7.5 packets per second and 16.5 packets per second, respectively in our experiments.

These observations indicate that A<sup>3</sup>L-FEC-VSVB adapts its transmission rate more aggressively to minimize age violation, providing higher transmission rates as the age

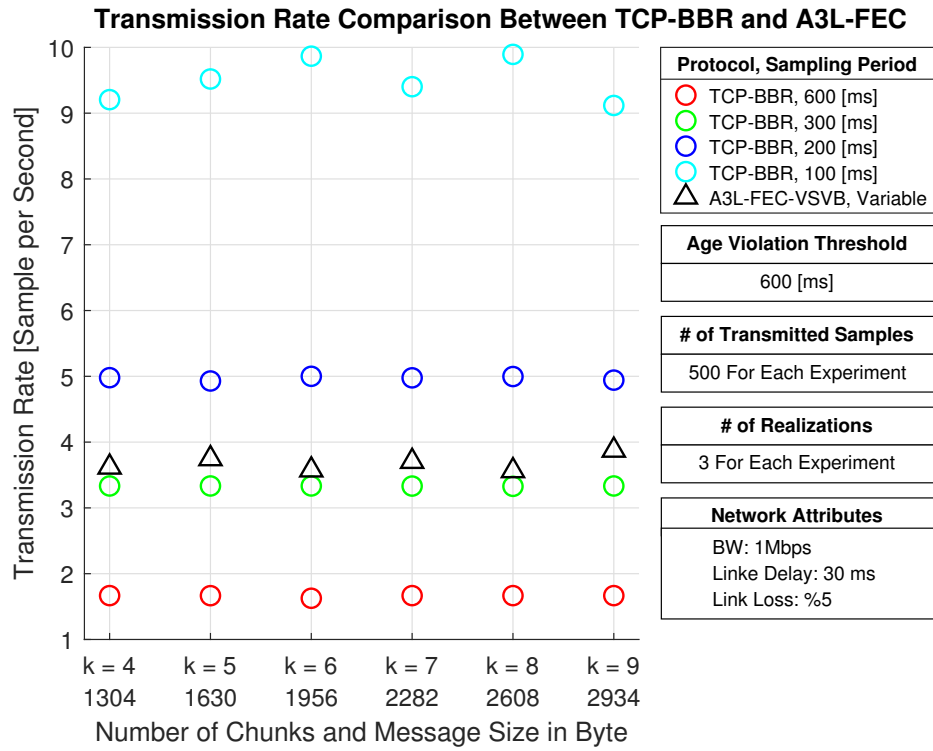


Figure 4.23: Average transmission rate under AVT = 600 msec.

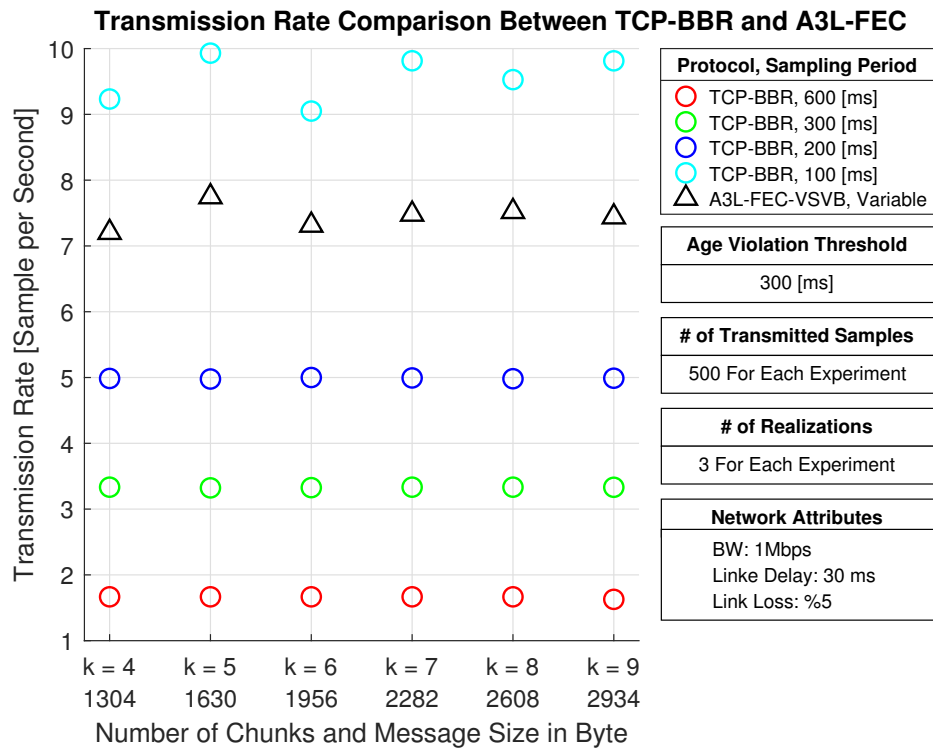


Figure 4.24: Average transmission rate under AVT = 300 msec.

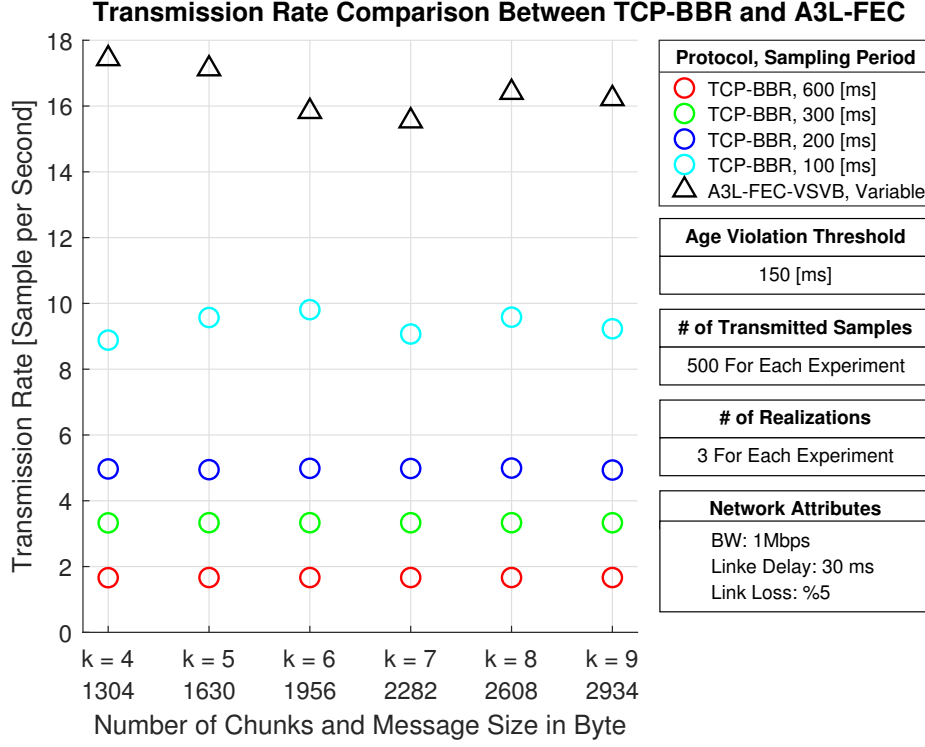


Figure 4.25: Average transmission rate under AVT = 150 msec.

violation threshold decreases. In contrast, the transmission rate of TCP-BBR remains more stable regardless of the age violation threshold. This conclusion shows the superior adaptability and responsiveness of A<sup>3</sup>L-FEC-VSVB in achieving low-age violation transmission rates in different scenarios.

#### 4.3.8 Effect of Sample Chunk Number on the Performance of A<sup>3</sup>L-FEC-VSVB on Mininet-WiFi

To comprehensively examine the impact of various numbers of chunks on the performance of A<sup>3</sup>L-FEC-VSVB, we designed and conducted an emulation using Mininet-WiFi. Our experiment consisted of communicating 2000 samples. In this experiment, we carefully varied the age violation threshold to investigate its effect on the system. Specifically, we considered threshold values of 70, 60, 50, and 40 *ms*. By meticulously selecting these values, we aimed to gain valuable insights into how different numbers of sample chunks impact the overall performance of the A<sup>3</sup>L-FEC-VSVB. Our systematic approach permitted us to draw meaningful conclusions regarding our

data flow control algorithm's efficiency and robustness.

As the sample chunk number ( $k$ ) increases, the age violation occurrence also increases, leading to a greater possibility of encountering aged data during transmission. This trend can be attributed to the nature of the A<sup>3</sup>L-FEC algorithm. Because for each sample, the transmitter sends  $n$  chunks, where  $n$  is greater than or equal to  $k$ . While the receiver needs at least  $k$  out of  $n$  chunks to successfully decode a sample, the presence of additional chunks increases the data transmission process time and contributes to the overall congestion in the network.

To clarify more, we can say that with the larger chunk number for each sample, the number of packets in the network increases, leading to a higher likelihood of experiencing congested paths, buffering delays, and potential packet losses. These factors collectively contribute to age violations, as aged data may encounter additional delays or even be dropped due to congestion. As a result, the probability of age violations rises as the chunk number increases.

For instance, when considering violation thresholds of 70 ms and 60 ms, we see that even with relatively low chunk numbers, the age violation probability remains relatively low. This indicates that A<sup>3</sup>L-FEC performs well in keeping fresh data under these circumstances. However, as the chunk number grows, the age violation probability also increases, indicating that the algorithm faces challenges in effectively reducing age violations when chunk sizes become larger.

It is important to consider the trade-off between the number of chunks and the possibility of age violations. While larger chunk numbers may provide better performance in data reconstruction, they also pose a higher risk of age violations. On the other hand, smaller chunk numbers may decrease age violation occurrence but may come at the cost of increased overhead and higher data reconstruction complexity.

Moreover, the results show that as the age violation threshold decreases, the age violation probability raised. For scenarios with the age violation threshold equal to 70 ms and 60 ms, the age violation probability is relatively low for most coding rates, indicating that A<sup>3</sup>L-FEC-VSVB performs well in maintaining fresh data under these circumstances. However, for age violation thresholds of 50 ms, 40 ms, and 30 ms,

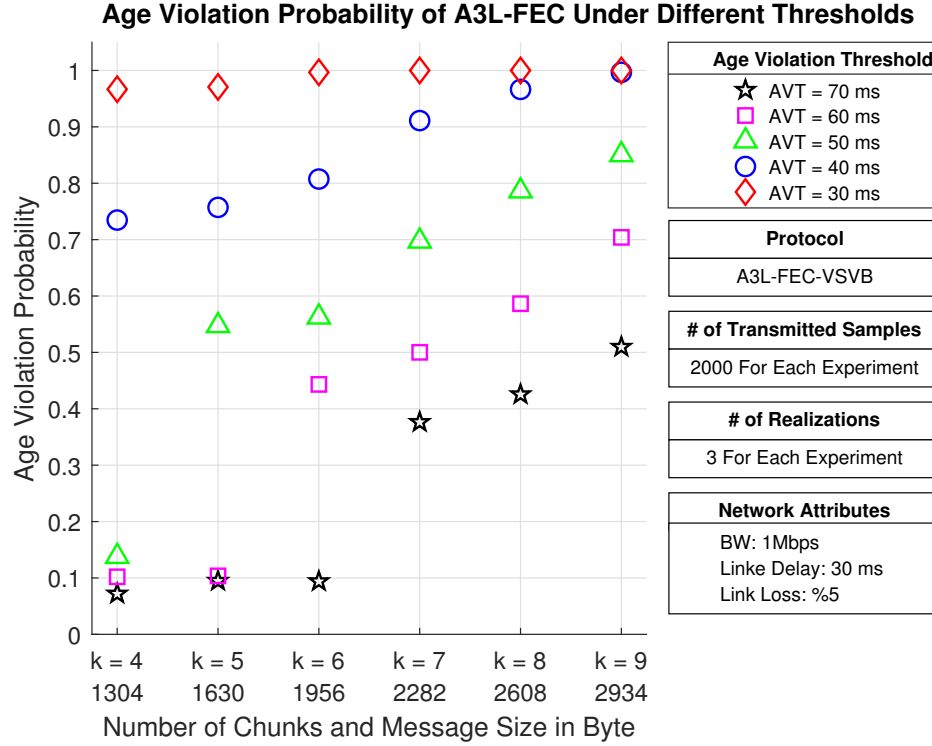


Figure 4.26: Effect of Sample Chunk Number and Age Violation Threshold on the Performance of A<sup>3</sup>L-FEC-VSVB

the age violation probability grows substantially, reaching 1 for some coding rates. This implies that the A<sup>3</sup>L-FEC-VSVB algorithm cannot effectively minimize age violations, because the physical condition of the network such as link delay does not allow for minimizing the age.

Through measurements performed on the emulated testbed, we determined that the network round-trip time (RTT) was approximately 60 ms. According to these observations, we can make an important inference regarding the lower bound of age violation thresholds for the A<sup>3</sup>L-FEC algorithm. Since RTT measures the time it takes for a packet to travel from sender to receiver and back, it effectively represents the minimum time required for data to propagate through the network. Therefore, to have a logical setting on A3L-FEC, the lower bound for the age violation threshold should be set to a level that is greater than half of the minimum RTT time. Because the age violation threshold defines the maximum time allowed for a data sample to be considered fresh before it becomes outdated. Therefore, based on our measured RTT value

which was 60 ms, we can infer that the lower bound of age violation thresholds for the A<sup>3</sup>L-FEC algorithm in our testbed should be set to at least half of the RTT, which corresponds to 30 ms. By doing so, we guarantee that data samples are delivered and received within a reasonable time frame, minimizing the risk of encountering aged data during transmission.





## REFERENCES

- [1] IMT, “Imt traffic estimates for the years 2020 to 2030..” <https://www.itu.int/pub/R-REP-M.2370-2015>, 2015. [Online; accessed 22-October-2022].
- [2] International Data Corporation, “Iot growth demands rethink of long-term storage strategies.” <https://www.eetasia.com/iot-growth-demands-rethink-of-long-term-storage-strategies/>, 2020. Accessed on 29 July 2020.
- [3] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, “Minimizing age of information in vehicular networks,” in *2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pp. 350–358, 2011.
- [4] S. Kaul, R. Yates, and M. Gruteser, “Real-time status: How often should one update?,” in *2012 Proceedings IEEE INFOCOM*, pp. 2731–2735, March 2012.
- [5] C. Li, Y. Huang, S. Li, Y. Chen, B. A. Jalaian, Y. T. Hou, W. Lou, J. H. Reed, and S. Kompella, “Minimizing aoi in a 5g-based iot network under varying channel conditions,” *IEEE Internet of Things Journal*, vol. 8, no. 19, pp. 14543–14558, 2021.
- [6] C.-C. Wu, P. Popovski, Z.-H. Tan, and C. Stefanovic, “Design of aoi-aware 5g uplink scheduler using reinforcement learning,” in *2021 IEEE 4th 5G World Forum (5GWF)*, pp. 176–181, 2021.
- [7] N. Modina, R. El Azouzi, F. De Pellegrini, D. S. Menasche, and R. Figueiredo, “Joint traffic offloading and aging control in 5g iot networks,” *IEEE Transactions on Mobile Computing*, pp. 1–1, 2022.
- [8] B. Soret, I. Leyva-Mayorga, S. Cioni, and P. Popovski, “5g satellite networks

- for internet of things: Offloading and backhauling,” *Int. J. Satell. Commun. Netw.*, vol. 39, p. 431–444, jun 2021.
- [9] B. Yu, Y. Cai, and D. Wu, “Joint access control and resource allocation for short-packet-based mmhc in status update systems,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 3, pp. 851–865, 2021.
  - [10] S. Saha, V. B. Sukumaran, and C. R. Murthy, “On the minimum average age of information in irsa for grant-free mmhc,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1441–1455, 2021.
  - [11] C. M. Wijerathna Basnayaka, D. N. K. Jayakody, T. D. Ponnimbaduge Perera, and M. Vidal Ribeiro, “Age of information in an urlhc-enabled decode-and-forward wireless communication system,” in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, pp. 1–6, 2021.
  - [12] X. Zhang, J. Wang, and H. V. Poor, “Statistical delay and error-rate bounded qos provisioning for 6g murlhc over aoi-driven and uav-enabled wireless networks,” in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pp. 1–10, 2021.
  - [13] X. Zhang, Q. Zhu, and H. V. Poor, “Age-of-information for murlhc over 6g multimedia wireless networks,” in *2021 55th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6, 2021.
  - [14] S. Baghaee and I. Ulusoy, “User comfort and energy efficiency in hvac systems by q-learning,” in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, May 2018.
  - [15] S. BAGHAEI, S. Z. GURBUZ, and E. UYSAL-BIYIKOGLU, “Implementation of an enhanced target localization and identification algorithm on a magnetic wsn,” *IEICE Transactions on Communications*, vol. E98.B, no. 10, pp. 2022–2032, 2015.
  - [16] S. Baghaee, S. Zubeyde Gurbuz, and E. Uysal-Biyikoglu, “Implementation of an enhanced target localization and identification algorithm on a magnetic wsn,” *IEICE Transactions on Communications*, vol. E98.B, p. 2022–2032, 10 2015.

- [17] H. B. Beytur, S. Baghaee, and E. Uysal, “Measuring age of information on real-life connections,” in *2019 27th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2019.
- [18] M. Costa, M. Codreanu, and A. Ephremides, “Age of information with packet management,” in *2014 IEEE International Symposium on Information Theory*, pp. 1583–1587, June 2014.
- [19] E. Uysal, O. Kaya, S. Baghaee, and H. B. Beytur, “Age of information in practice,” 2021.
- [20] U. Guloglu, S. Baghaee, and E. Uysal, “Evaluation of age control protocol (acp) and acp+ on esp32,” in *2021 17th International Symposium on Wireless Communication Systems (ISWCS)*, pp. 1–6, 2021.
- [21] H. B. Beytur, S. Baghaee, and E. Uysal, “Towards aoi-aware smart iot systems,” in *2020 International Conference on Computing, Networking and Communications (ICNC)*, pp. 353–357, 2020.
- [22] C. Sönmez, S. Baghaee, A. Ergişi, and E. Uysal-Biyikoglu, “Age-of-Information in practice: Status age measured over TCP/IP connections through WiFi, Ethernet and LTE,” in *2018 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pp. 1–5, June 2018.
- [23] E. Sert, C. Sönmez, S. Baghaee, and E. Uysal-Biyikoglu, “Optimizing age of information on real-life TCP/IP connections through reinforcement learning,” in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, May 2018.
- [24] C. Kam, S. Kompella, and A. Ephremides, “Experimental evaluation of the age of information via emulation,” in *MILCOM 2015 - 2015 IEEE Military Communications Conference*, pp. 1070–1075, Oct 2015.
- [25] T. Shreedhar, S. K. Kaul, and R. D. Yates, “An age control transport protocol for delivering fresh updates in the Internet-of-Things,” in *2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pp. 1–7, 2019.

- [26] I. Kadota, M. S. Rahman, and E. Modiano, “Age of information in wireless networks: From theory to implementation,” in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking, MobiCom '20*, (New York, NY, USA), Association for Computing Machinery, 2020.
- [27] I. Kadota, M. S. Rahman, and E. Modiano, “WiFresh: Age-of-information from theory to implementation,” 2020.
- [28] E. Najm and R. Nasser, “Age of information: The gamma awakening,” in *Proc. IEEE ISIT*, pp. 2574–2578, July 2016.
- [29] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, “Update or wait: How to keep your data fresh,” *IEEE Transactions on Information Theory*, vol. 63, pp. 7492–7508, Nov 2017.
- [30] C. Kam, S. Kompella, and A. Ephremides, “Learning to sample a signal through an unknown system for minimum AoI,” in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 177–182, 2019.
- [31] H. B. Beytur and E. Uysal, “Age minimization of multiple flows using reinforcement learning,” in *2019 International Conference on Computing, Networking and Communications (ICNC)*, pp. 339–343, Feb 2019.
- [32] E. T. Ceran, D. Gündüz, and A. György, “Reinforcement learning to minimize age of information with an energy harvesting sensor with HARQ and sensing cost,” in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 656–661, 2019.
- [33] 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*, vol. 68, no. 8, pp. 4747–4760, 2020.
- [34] M. A. Abd-Elmagid, H. S. Dhillon, and N. Pappas, “Aoi-optimal joint sampling and updating for wireless powered communication systems,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 14110–14115, 2020.

- [35] C. Kam, S. Kompella, G. D. Nguyen, and A. Ephremides, “Effect of message transmission path diversity on status age,” *IEEE Trans. Inf. Theory*, vol. 62, pp. 1360–1374, March 2016.
- [36] M. Costa, M. Codreanu, and A. Ephremides, “On the age of information in status update systems with packet management,” *IEEE Transactions on Information Theory*, vol. 62, pp. 1897–1910, April 2016.
- [37] S. K. Kaul, R. D. Yates, and M. Gruteser, “Status updates through queues,” in *2012 46th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6, Mar 2012.
- [38] R. D. Yates and S. Kaul, “Real-time status updating: Multiple sources,” in *IEEE International Symposium on Information Theory (ISIT)*, pp. 2666–2670, July 2012.
- [39] C. Kam, S. Kompella, and A. Ephremides, “Age of information under random updates,” in *2013 IEEE International Symposium on Information Theory*, pp. 66–70, July 2013.
- [40] C. Kam, S. Kompella, and A. Ephremides, “Effect of message transmission diversity on status age,” in *2014 IEEE International Symposium on Information Theory*, pp. 2411–2415, June 2014.
- [41] L. Huang and E. Modiano, “Optimizing age-of-information in a multi-class queueing system,” in *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 1681–1685, June 2015.
- [42] B. T. Bacinoglu, E. T. Ceran, and E. Uysal-Biyikoglu, “Age of information under energy replenishment constraints,” in *2015 Information Theory and Applications Workshop (ITA)*, pp. 25–31, Feb 2015.
- [43] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, “Update or wait: How to keep your data fresh,” *IEEE Transactions on Information Theory*, vol. 63, pp. 7492–7508, Nov 2017.
- [44] Y. Sun, E. Uysal-Biyikoglu, R. Yates, C. E. Koksal, and N. B. Shroff, “Update or wait: How to keep your data fresh,” in *IEEE INFOCOM 2016 - The 35th*

*Annual IEEE International Conference on Computer Communications*, pp. 1–9, April 2016.

- [45] Official U.S. government information about the Global Positioning System (GPS) and related topics, *GPS Accuracy*, Accessed January 10, 2021.
- [46] D. L. Mills, *Computer Network Time Synchronization: The Network Time Protocol on Earth and in Space, Second Edition*. USA: CRC Press, Inc., 2nd ed., 2010.
- [47] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis, “Age and value of information: Non-linear age case,” in *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 326–330, 2017.
- [48] M. Klügel, M. H. Mamduhi, S. Hirche, and W. Kellerer, “AoI-Penalty minimization for networked control systems with packet loss,” in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 189–196, 2019.
- [49] Y. Sun and B. Cyr, “Sampling for data freshness optimization: Non-linear age functions,” *Journal of Communications and Networks*, vol. 21, no. 3, pp. 204–219, 2019.
- [50] J. Ahrenholz, C. Danilov, T. R. Henderson, and J. H. Kim, “Core: A real-time network emulator,” in *MILCOM 2008 - 2008 IEEE Military Communications Conference*, pp. 1–7, Nov 2008.
- [51] A. Alós, F. Morán, P. Carballeira, D. Berjón, and N. García, “Congestion control for cloud gaming over udp based on round-trip video latency,” *IEEE Access*, vol. 7, pp. 78882–78897, 2019.
- [52] Y. Inoue, H. Masuyama, T. Takine, and T. Tanaka, “A general formula for the stationary distribution of the age of information and its application to single-server queues,” *IEEE Transactions on Information Theory*, vol. 65, no. 12, pp. 8305–8324, 2019.
- [53] H. Erkal, F. M. Ozcelik, and E. Uysal-Biyikoglu, “Optimal offline broadcast scheduling with an energy harvesting transmitter,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, no. 1, p. 197, 2013.

- [54] S. Chamanian, S. Baghaee, H. Uluhan, O. Zorlu, H. Kulah, and E. Uysal-Biyikoglu, “Powering-up wireless sensor nodes utilizing rechargeable batteries and an electromagnetic vibration energy harvesting system,” *Energies*, vol. 7, no. 10, pp. 6323–6339, 2014.
- [55] S. Baghaee, S. Chamanian, H. Uluhan, and O. Zorlu, “Wirelessenergysim: A discrete event simulator for an energy-neutral operation of iot nodes,” in *2018 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pp. 1–5, June 2018.
- [56] F. M. Ozcelik, G. Uctu, and E. Uysal-Biyikoglu, “Minimization of transmission duration of data packets over an energy harvesting fading channel,” *IEEE Communications Letters*, vol. 16, pp. 1968–1971, December 2012.
- [57] E. T. Ceran, D. Gündüz, and A. György, “Average age of information with hybrid arq under a resource constraint,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1900–1913, 2019.
- [58] B. T. Bacinoglu, Y. Sun, E. Uysal, and V. Mutlu, “Optimal status updating with a finite-battery energy harvesting source,” *Journal of Communications and Networks*, vol. 21, no. 3, pp. 280–294, 2019.
- [59] R. D. Yates and S. K. Kaul, “Status updates over unreliable multiaccess channels,” in *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 331–335, 2017.
- [60] Y. Sun, E. Uysal-Biyikoglu, and S. Kompella, “Age-optimal updates of multiple information flows,” in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 136–141, 2018.
- [61] B. T. Bacinoglu, E. Uysal-Biyikoglu, and C. E. Koksall, “Finite-horizon energy-efficient scheduling with energy harvesting transmitters over fading channels,” *IEEE Transactions on Wireless Communications*, vol. 16, pp. 6105–6118, Sept 2017.
- [62] B. T. Bacinoglu, O. Kaya, and E. Uysal-Biyikoglu, “Energy efficient transmission scheduling for channel-adaptive wireless energy transfer,” in *2018*

*IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2018.

- [63] E. T. Ceran, D. Gündüz, and A. György, “Average age of information with hybrid ARQ under a resource constraint,” in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2018.
- [64] A. Elgabli, H. Khan, M. Krouka, and M. Bennis, “Reinforcement learning based scheduling algorithm for optimizing age of information in ultra reliable low latency networks,” in *2019 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–6, IEEE, 2019.
- [65] M. A. Abd-Elmagid, A. Ferdowsi, H. S. Dhillon, and W. Saad, “Deep reinforcement learning for minimizing age-of-information in uav-assisted networks,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2019.
- [66] M. Ma and V. W. Wong, “A deep reinforcement learning approach for dynamic contents caching in hetnets,” in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2020.
- [67] M. Hatami, M. Jahandideh, M. Leinonen, and M. Codreanu, “Age-aware status update control for energy harvesting iot sensors via reinforcement learning,” in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1–6, IEEE, 2020.
- [68] A. Ferdowsi, M. A. Abd-Elmagid, W. Saad, and H. S. Dhillon, “Neural combinatorial deep reinforcement learning for age-optimal joint trajectory and scheduling design in uav-assisted networks,” *arXiv preprint arXiv:2006.15863*, 2020.
- [69] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, pp. 529–533, Feb. 2015.



- [70] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 1998.
- [71] L.-J. Lin, *Reinforcement Learning for Robots Using Neural Networks*. PhD thesis, Carnegie Mellon University, USA, 1992.
- [72] H. v. Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double q-learning,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI’16, p. 2094–2100, AAAI Press, 2016.
- [73] K. Hornik, M. Stinchcombe, and H. White, “Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks,” *Neural Netw.*, vol. 3, p. 551–560, Oct. 1990.
- [74] L. Kleinrock, “Internet congestion control using the power metric: Keep the pipe just full, but no fuller,” *Ad Hoc Networks*, vol. 80, pp. 142 – 157, 2018.
- [75] U. Guloglu, “Implementation and evaluation of ACP on an IoT testbed,” technical report, metu, Middle East Technical University, 2021.
- [76] T. Shreedhar, S. K. Kaul, and R. D. Yates, “An empirical study of ageing in the cloud,” *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, 2021.
- [77] I. Kadota, A. Sinha, and E. Modiano, “Scheduling algorithms for optimizing age of information in wireless networks with throughput constraints,” *IEEE/ACM Transactions on Networking*, p. 1–14, 2019.
- [78] 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*, vol. 26, no. 6, pp. 2637–2650, 2018.
- [79] T. K. Oğuz, E. Uysal, and T. Girici, “Implementation and evaluation of MAC layer age-aware scheduling algorithms in a wireless uplink,” technical report, metu, Middle East Technical University, 2021.
- [80] L. Kleinrock, “Internet congestion control using the power metric: Keep the pipe just full, but no fuller,” *Ad Hoc Networks*, vol. 80, pp. 142–157, 2018.

- [81] V. Jacobson, “Congestion avoidance and control,” in *SIGCOMM '88*, 1988.
- [82] S. Ha, I. Rhee, and L. Xu, “Cubic: A new tcp-friendly high-speed tcp variant,” *SIGOPS Oper. Syst. Rev.*, vol. 42, p. 64–74, jul 2008.
- [83] L. Brakmo, S. O'Malley, and L. Peterson, “Tcp vegas: New techniques for congestion detection and avoidance,” in *Proceedings of the Conference on Communications Architectures, Protocols and Applications, SIGCOMM 1994*, Proceedings of the Conference on Communications Architectures, Protocols and Applications, SIGCOMM 1994, pp. 24–35, Association for Computing Machinery, Inc, Oct. 1994.
- [84] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, “Fast tcp: Motivation, architecture, algorithms, performance,” *IEEE/ACM Trans. Netw.*, vol. 14, p. 1246–1259, dec 2006.
- [85] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, “Bbr: Congestion-based congestion control,” *ACM Queue*, vol. 14, September-October, pp. 20 – 53, 2016.
- [86] A. Kosta, N. Pappas, and V. Angelakis, “Age of information: A new concept, metric, and tool,” *Foundations and Trends® in Networking*, vol. 12, no. 3, pp. 162–259, 2017.
- [87] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. H. Modiano, and S. Ulukus, “Age of information: An introduction and survey,” *IEEE Journal on Selected Areas in Communications*, vol. 39, pp. 1183–1210, 2021.
- [88] E. Uysal, O. Kaya, S. Baghaee, and H. B. Beytur, “Age of information in practice,” *CoRR*, vol. abs/2106.02491, 2021.
- [89] T. Shreedhar, S. K. Kaul, and R. D. Yates, “Acp: Age control protocol for minimizing age of information over the internet,” in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, MobiCom '18*, (New York, NY, USA), ACM, 2018.
- [90] T. Shreedhar, S. K. Kaul, and R. D. Yates, “An age control transport protocol for delivering fresh updates in the internet-of-things,” in *2019 IEEE 20th In-*

*ternational Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pp. 1–7, June 2019.

- [91] M. Costa, Y. Sagduyu, T. Erpek, and M. Médard, “Robust improvement of the age of information by adaptive packet coding,” 2020.
- [92] GSMA, “The gsma guide to the internet of things. global system for mobile communications association, london, july..” <https://www.gsma.com/iot/resources/the-gsma-guide-to-the-internet-of-things-2/>, 2022. [Online; accessed 28-June-2022].
- [93] F. B. Insights, “Internet of things (iot) market size, share & covid-19 impact analysis, by component (platform, solution & services), by end use industry (bfsi, retail, government, healthcare, manufacturing, agriculture, sustainable energy, transportation, it & telecom, others), and regional forecast, 2022–2029. report id: Fbi100307, 22 march..” <https://www.fortunebusinessinsights.com/industry-reports/internet-of-things-iot-market-100307>, 2022. [Online; accessed 28-June-2022].
- [94] Business Wire, “Worldwide spending on the internet of things will slow in 2020, then return to double-digit growth, according to a new idc spending guide.” <https://www.businesswire.com/news/home/20200618005125/en/Worldwide-Spending-on-the-Internet-of-Things-Will-Slow-in-2020-Then-Return-to-Double-Digit-Growth-According-to-a-New-IDC-Spending-Guide>, June 2020. [Online; accessed 06-June-2023].
- [95] G. Borboruah and G. Nandi, “A study on large scale network simulators,” 2014.
- [96] S. Robinson, *Simulation: The Practice of Model Development and Use*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2004.
- [97] R. M. Fujimoto, “Parallel discrete event simulation,” *Commun. ACM*, vol. 33, p. 30–53, oct 1990.

- [98] G. R. Yaun, D. Bauer, H. L. Bhutada, C. D. Carothers, M. Yuksel, and S. Kalyanaraman, "Large-scale network simulation techniques: Examples of tcp and ospf models," *SIGCOMM Comput. Commun. Rev.*, vol. 33, p. 27–41, jul 2003.
- [99] D. M. Rao, N. V. Thondugulam, R. Radhakrishnan, and P. A. Wilsey, "Unsynchronized parallel discrete event simulation," in *Proceedings of the 30th Conference on Winter Simulation, WSC '98*, (Washington, DC, USA), p. 1563–1570, IEEE Computer Society Press, 1998.
- [100] D. M. Rao and P. A. Wilsey, "An ultra-large-scale simulation framework," *Journal of Parallel and Distributed Computing*, vol. 62, no. 11, pp. 1670–1693, 2002.
- [101] A. ur Rehman Khan, S. M. Bilal, and M. Othman, "A performance comparison of network simulators for wireless networks," *ArXiv*, vol. abs/1307.4129, 2013.
- [102] X. Chang, "Network simulations with opnet," in *Proceedings of the 31st Conference on Winter Simulation: Simulation—a Bridge to the Future - Volume I*, WSC '99, (New York, NY, USA), p. 307–314, Association for Computing Machinery, 1999.
- [103] A. Varga, "The omnet++ discrete event simulation system," *Proceedings of the European Simulation Multiconference (ESM'2001)*, June 2001.
- [104] X. Zeng, R. Bagrodia, and M. Gerla, "Glomosim: a library for parallel simulation of large-scale wireless networks," in *Proceedings. Twelfth Workshop on Parallel and Distributed Simulation PADS '98 (Cat. No.98TB100233)*, pp. 154–161, 1998.
- [105] S. Networks., "Qualnet - network simulation software." <https://www.scalable-networks.com/products/qualnet-network-simulation-software>, 2022. [Online; accessed 27-June-2022].
- [106] M. H. Kabir, S. Islam, and S. Hossain, "Detail comparison of network simulators," 2014.
- [107] Tetcos., "Netsim,." <https://www.tetcos.com/index.html>, 2022. [Online; accessed 27-June-2022].

- [108] “Ns-2,” <https://www.isi.edu/nsnam/ns/>, 2022. [Online; accessed 27-June-2022].
- [109] G. F. Riley and T. R. Henderson, “The ns-3 network simulator,” in *Modeling and Tools for Network Simulation*, 2010.
- [110] H. H. Hasan and Z. T. Alisa, “Effective iot congestion control algorithm,” *Future Internet*, vol. 15, no. 4, 2023.
- [111] M. Kühlewind, “Tcp siad: Congestion control supporting high speed and low latency,” *ArXiv*, vol. abs/1612.07947, 2016.
- [112] D. Rossi, C. Testa, S. Valenti, and L. Muscariello, “Ledbat: The new bittorrent congestion control protocol,” in *2010 Proceedings of 19th International Conference on Computer Communications and Networks*, pp. 1–6, 2010.
- [113] E. Atxutegi, F. Liberal, H. K. Haile, K.-J. Grinnemo, A. Brunstrom, and A. Arvidsson, “On the use of tcp bbr in cellular networks,” *IEEE Communications Magazine*, vol. 56, no. 3, pp. 172–179, 2018.
- [114] L. Kleinrock, “Power and deterministic rules of thumb for probabilistic problems in computer communications,” in *Proceedings of the International Conference on Communications*, vol. 43, 1979.



## **CURRICULUM VITAE**

Surname, Name: Baghaee, Sajjad

### **EDUCATION**

<b>Degree</b>	<b>Institution</b>	<b>Year of Graduation</b>
MS	METU Electrical and Electronics Engineering	2012
BS	IAURMIA Electrical Eng./Telecommunications	2005
High School	Dehkoda, Urmia	2001

### **FOREIGN LANGUAGES**

English, Persian, Turkish, Azeri

### **PUBLICATIONS**

1. M Mitev, M Shakiba-Herfeh, A Chorti, M Reed, S Baghaee, "A physical layer, zero-round-trip-time, multifactor authentication protocol," IEEE Access, vol. 10, pp. 74 555–74 571, 2022
2. U. Guloglu, S. Baghaee and E. Uysal, " Evaluation of Age Control Protocol (ACP) and ACP+ on ESP32," ISWCS'21 - 17th International Symposium on Wireless Communication Systems, Berlin, Germany, 2021.
3. E. Uysal, O. Kaya, S. Baghaee, H. B. Beytur, "Age of Information in Practice", to appear as a chapter in "Age of Information", Cambridge University Press, 2021.
4. H. B. Beytur, S. Baghaee and E. Uysal, "Towards AoI-aware Smart IoT Systems," 2020 International Conference on Computing, Networking and Communications (ICNC), Big Island, HI, USA, 2020, pp.
5. Sajjad Baghaee, Hasan Burhan Beytur, Elif Uysal, "Towards AoI-aware Smart IoT

Systems", IEEE INFOCOM 2019-IEEE Conference on Computer Communications, Paris, France, 29 April - 2 May 2019. (Poster Presentation)

6. H. B. Beytur, S. Baghaee and E. Uysal, "Measuring Age of Information on Real-Life Connections," 2019 27th Signal Processing and Communications Applications Conference (SIU), Sivas, Turkey, 2019, pp..

7. S. Chamanian, S. Baghaee, H. Uluşan, Ö. Zorlu, E. Uysal-Biyikoglu and H. Külâh, "Implementation of Energy-Neutral Operation on Vibration Energy Harvesting WSN," in IEEE Sensors Journal, vol. 19, no. 8, pp. 3092-3099, 15 April 2019.

8. S. Baghaee, Ilkay Ulusoy, "User Comfort and Energy Efficiency in HVAC Systems by Q-learning", IEEE SIU 2018, 26th Signal Processing and Communications Applications Conference, Izmir, Turkey, 2-5 May 2018.

9. Canberk Sonmez, Sajjad Baghaee, Abdussamed Ergisi, Elif Uysal-Biyikoglu, "Age-of-Information in Practice: Status Age Measured over TCP/IP Connections through WiFi, Ethernet and LTE", IEEE BlackSeaCom., Batumi, Georgia, 4-7 June 2018.

10. S. Baghaee, S. Chamanian, H. Uluşan, Ö. Zorlu, "WirelessEnergySim: A Discrete Event Simulator for an Energy-Neutral Operation of IoT Nodes ", IEEE BlackSeaCom., Batumi, Georgia, 4-7 June 2018.

11. Egemen Sert, Canberk Sönmez, Sajjad Baghaee, Elif Uysal-Biyikoglu, " Optimizing Age of Information on Real-Life TCP/IP Connections through Reinforcement Learning", IEEE SIU 2018, 26th Signal Processing and Communications Applications Conference, Izmir, Turkey, 2-5 May 2018.

12. S. Chamanian, H Uluşan, Ö Zorlu, S. Baghaee, E Uysal-Biyikoglu, H Külâh, "Wearable battery-less wireless sensor network with electromagnetic energy harvesting system" Sens. Actuators A: Phys., 249 October 2016.

13. S. Baghaee, S.Z. Gürbüz, E. Uysal-Bıyıkoglu, "Implementation of an Enhanced Target Localization and Identification Algorithm on a Magnetic WSN," IEICE Trans. on Communications, June 2015.



14. Chamanian, S.; Baghaee, S.; Ulsan, H.; Zorlu, Ö.; Kùlah, H.; Uysal-Biyikoglu, E. Powering-up Wireless Sensor Nodes Utilizing Rechargeable Batteries and an Electromagnetic Vibration Energy Harvesting System. *Energies* 2014, 7, 6323-6339.
15. S. Baghaee, H. Ulsan, S. Chamanian, O. Zorlu, E. Uysal-Biyikoglu, and H. Kùlah, "Demonstration of Energy-Neutral Operation on a WSN Testbed Using Vibration Energy Harvesting," *European Wireless 2014 (EW2014)*, Barcelona, Spain, 14-16th May 2014.
16. S. Baghaee, O. Nazim Onak, I. Ulusoy , "Inferring Brain Effective Connectivity via DBN and EEG Time Series Data", *The 4th International Scientific Conference of Iranian Academics in Turkey*, March 2014.
17. S. Baghaee, H. Ulsan, S. Chamanian, O. Zorlu, H. Kùlah and E. Uysal-Biyikoglu, "Towards a Vibration Energy Harvesting WSN Demonstration Testbed", *IEEE TI-WDC 2013 "Green ICT"*, *The 24th Tyrrhenian International Workshop on Digital Communications* ,Genoa, Italy, September 2013.
18. S. Baghaee, S.Z. Gùrbüz, E. Uysal-Biyikoglu, "Application and Modeling of a Magnetic WSN for Target Localization", *IEEE UKSim2013, 15th International Conference on modelling and simulation*, April 2013.
19. S. Baghaee, S.Z. Gùrbüz, E. Uysal-Biyikoglu, "Identification and Localization on a Wireless Magnetic Sensor Network with MED method", *The Third International Scientific Conference of Iranian Academics in Turkey*, February 2013.
20. S. Baghaee, S. Z. Gùrbüz and E. Uysal-Biyikoğlu, "Orthogonal Matching Pursuit for ferromagnetic target localization and identification," *2012 20th Signal Processing and Communications Applications Conference (SIU)*, Mugla, Turkey, 2012, pp. 1-4, doi: 10.1109/SIU.2012.6204588.