Abstract

Real-Time Communication (RTC) have become an integral part of modern networked systems, where timely and accurate data transmission is crucial. The Age of Information (AoI) has emerged as a key metric for assessing information freshness, particularly in applications such as the Internet of Things (IoT), and autonomous systems. Unlike conventional metrics such as delay and throughput, AoI provides a direct measure of the time elapsed since the most recent successful update, making it a critical tool for optimizing network performance.

This thesis investigates the impact of random-access mechanisms on AoI, focusing on Gated Access and Free Access protocols. It examines how factors such as contention, pre-emption, and system load influence AoI across finite and infinite user scenarios. Through statistical modeling and comparative analysis, key performance trends are identified, providing insight into the trade-offs between controlled and open access strategies.

The findings offer valuable guidance for designing efficient communication systems where maintaining up-to-date information is essential.

Resumen

Las comunicaciones en tiempo real se han convertido en una parte fundamental de los sistemas de redes modernos, donde la transmisión de datos de manera oportuna y precisa es crucial. La Edad de la Información ha surgido como una métrica clave para evaluar la frescura de la información, especialmente en aplicaciones como Internet de las Cosas y sistemas autónomos. A diferencia de métricas convencionales como el retraso y el rendimiento, la Edad de la Información proporciona una medida directa del tiempo transcurrido desde la última actualización recibida con éxito, lo que la convierte en una herramienta esencial para optimizar el rendimiento de las redes.

Este trabajo estudia el impacto de los mecanismos de acceso aleatorio en la Edad de la Información, centrándose en los protocolos de Acceso Restringido y los protocolos de Acceso Libre. Se analiza cómo factores como el retraso, la preferencia de paquetes y la carga del sistema influyen en la Edad de la Información en escenarios con un número finito de usuarios. A través del modelado estadístico y análisis comparativo, se identifican tendencias clave de rendimiento, proporcionando información sobre los compromisos entre estrategias de acceso controlado y abierto.

Los hallazgos de esta investigación ofrecen orientación valiosa para el diseño de sistemas de comunicación eficientes, donde mantener la información actualizada es esencial.

Resum

Les comunicacions en temps real s'han convertit en una part fonamental dels sistemes de xarxes moderns, on la transmissió de dades de manera oportuna i precisa és crucial. L'Edat de la Informació ha sorgit com una mètrica clau per a avaluar la frescor de la informació, especialment en aplicacions com l'Internet de les Coses i els sistemes autònoms. A diferència de mètriques convencionals com el retard i el rendiment, l'Edat de la Informació proporciona una mesura directa del temps transcorregut des de l'última actualització rebuda amb èxit, fet que la converteix en una eina essencial per a optimitzar el rendiment de les xarxes.

Aquest treball estudia l'impacte dels mecanismes d'accés aleatori en l'Edat de la Informació, centrant-se en els protocols d'Accés Restringit i els protocols d'Accés Lliure. S'analitza com factors com el retard, la preferència de paquets i la càrrega del sistema influeixen en l'Edat de la Informació en escenaris amb un nombre finit d'usuaris. A través del modelatge estadístic i l'anàlisi comparativa, s'identifiquen tendències clau de rendiment, proporcionant informació sobre els compromisos entre estratègies d'accés controlat i obert.

Les troballes d'aquesta investigació ofereixen una orientació valuosa per al disseny de sistemes de comunicació eficients, on mantindre la informació actualitzada és essencial.

RESUMEN EJECUTIVO

La memoria del TFM del Máster Universitario en Ingeniería de Telecomunicación debe desarrollar en el texto los siguientes conceptos, debidamente justificados y discutidos, centrados en el ámbito de la ingeniería

CONCEPT (ABET)	**¿Cumple? (S/N)**	**¿Dónde? (páginas)**
1. IDENTIFY:		
1.1. Problem statement and opportunity	S	1-3, 25-32
1.2. Constraints (standards, codes, needs, requirements & specifications)	S	6-25
1.3. Setting of goals	S	3
2. FORMULATE:		
2.1. Creative solution generation (analysis)	S	32-51
2.2. Evaluation of multiple solutions and decision- making (synthesis)	S	32-51
3. SOLVE:		
3.1. Fulfilment of goals	S	51,52
3.2. Overall impact and significance (contributions and practical recommendations)	S	51,52

Contents

List of Acronyms

1 Introduction 1
- 1.1 Introduction . . . 1
- 1.2 Motivation . . . 2
- 1.3 Objectives . . . 3
- 1.4 Methodology . . . 3
- 1.5 Contribution to the Sustainable Development Goals (SDGs) . . . 4
- 1.6 Structure of the Document . . . 5

2 Introduction to Random-Access Protocols 6
- 2.1 History . . . 6
- 2.2 Basic random Access Protocols . . . 7
  - 2.2.1 Pure ALOHA . . . 7
  - 2.2.2 Slotted ALOHA . . . 9
- 2.3 Collision Resolution Algorithms . . . 10
  - 2.3.1 Capetanakis Collision Resolution Algorithm (CCRA) . . . 11
- 2.4 Free Access and Gated Access Protocols . . . 13
  - 2.4.1 Free Access Protocols . . . 14
  - 2.4.2 Gated Access Protocols . . . 17

3 Age of Information 24

4 Impact of Access Control Mechanisms on Age of Information 31
- 4.1 Gated Access . . . 31
- 4.2 Free Access . . . 35
- 4.3 Pre-emption . . . 38
  - 4.3.1 Pre-emption in Gated Access . . . 39
  - 4.3.2 Pre-emption in Free Access . . . 42

5 Conclusions 50
- 5.1 Future Work . . . 51

List of Acronyms

ACK	Acknowledgement
AoI	Age of Information
CCRA	Capetanakis Collision Resolution Algorithm
CRA	Collision Resolution Algorithms
CRI	Collision Resolution Interval
CRDSA	Collision Resolution Diversity Slotted ALOHA
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access Collision Detection
FDM	Frequency Division Multiplexing
G	Transmission Load
IIoT	Industrial Internet of Things
IoT	Internet of Things
mMTC	Massive Machine-type Communication
PMF	Probability Mass Function
RTC	Real-Time Communication

S Throughput

SDGs Sustainable Development Goals

SIC Successive Interference Cancellation

TDM Time Division Multiplexing

UHF Ultra High Frequency

List of Figures

2.1 Collision occurrences in pure ALOHA. 7
2.2 Collision occurrences in Slotted ALOHA. 9
2.3 Comparison of throughput in pure and Slotted ALOHA 10
2.4 Example of a Collision Resolution Interval for the CCRA. 12
2.5 Tree Diagram for the Collision Interval of Figure 2.4. 12
2.6 Illustration of Free Access protocol with CCRA. 14
2.7 Throughput vs. Arrival Rate for different user counts in Free Access: (a) 100 users, (b) 500 users. 15
2.8 Throughput vs. Arrival Rate for 100 and 500 users in Free Access. 16
2.9 Delay vs. Arrival Rate for different user counts in Free Access: (a) 100 users, (b) 500 users. 16
2.10 Delay vs. Arrival Rate for 100 and 500 users in Free Access. 17
2.11 Illustration of the Gated Access protocol with CCRA. 17
2.12 Throughput vs. Arrival Rate for different user counts in Gated Access: (a) 100 users, (b) 500 users. 18
2.13 Throughput vs. Arrival Rate for 100 and 500 users in Gated Access. 19
2.14 Comparison of Throughput vs. Arrival Rate for 100 users in Gated and Free Access. 19
2.15 Delay vs. Arrival Rate for different user counts in Gated Access: (a) 100 users, (b) 500 users. 20
2.16 Delay vs. Arrival Rate for 100 and 500 users in Gated Access. 20
2.17 Comparison of Delay vs. Arrival Rate for 100 users in Gated and Free Access. . 21
2.18 Probability Mass Function (PMF) of Collision Resolution Interval (CRI) duration for different arrival rates in a 100-user Gated Access system. 21
2.19 PMF of CRI duration for different numbers of users in a 100-user Gated Access system. 22
2.20 Comparison of throughput (a) and average delay (b) in an infinite-user Gated Access system as a function of the arrival rate λ. 23
2.21 Average delay as a function of throughput in an infinite-user Gated Access system. 23

3.1 AoI evolution for a single user in a slotted system. 26
3.2 Normalized average AoI as a function of the aggregated arrival rate λ in a slotted ALOHA system. 27
3.3 Example of communication between a single user and the receiver for the three Slotted ALOHA variants. 28
3.4 Throughput as a function of the aggregated arrival rate λ for different Slotted ALOHA variants. 29
3.5 Normalized AoI as a function of the aggregated arrival rate λ for different Slotted ALOHA variants. 30

4.1 Normalized average AoI for Gated Access using CCRA and Reactive SA as a function of λ. 32
4.2 Standard deviation of average AoI for Gated Access using CCRA and Reactive SA as a function of λ. 32

4.3 Comparison of AoI for Gated Access using CCRA with (a) 100 and (b) 500 users separately. 33
4.4 Comparison of average AoI for systems with 100 and 500 users in Gated Access using CCRA. 33
4.5 Probability mass function of delay values for different arrival rates λ in Gated Access. 34
4.6 Comparison of average AoI for Gated Access, Free Access, and Reactive Slotted ALOHA schemes as a function of λ. 36
4.7 Comparison of standard deviation of AoI for Gated Access, Free Access, and Reactive Slotted ALOHA schemes as a function of λ. 36
4.8 Comparison of AoI for Free Access with (a) 100 and (b) 500 users separately. . . 37
4.9 Comparison of average AoI for systems with 100 and 500 users in Free Access. . 37
4.10 Illustration of the pre-emption mechanism in Gated Access using CCRA. . . . 39
4.11 Comparison of AoI behavior with and without pre-emption in Gated Access for user B. 40
4.12 Comparison of average AoI for Gated Access with and without pre-emption as a function of λ. 41
4.13 Comparison of average AoI for Gated Access, Gated Access with pre-emption, and Reactive Slotted ALOHA as a function of λ. 41
4.14 Standard deviation of AoI as a function of λ, comparing Gated Access, Gated Access with pre-emption, and Reactive Slotted ALOHA. 42
4.15 Illustration of packet transmissions under Free Access with pre-emption using CCRA. 43
4.16 Comparison of AoI behavior with and without pre-emption in Free Access for user D. 44
4.17 Comparison of average AoI for Free Access with and without pre-emption. . . . 45
4.18 Standard deviation of AoI for Free Access and Free Access with pre-emption. . 45
4.19 Comparison of average AoI for systems with 100 and 500 users in Free Access with pre-emption. 46
4.20 Illustration of contention time and delay in Free Access with pre-emption. . . . 47
4.21 Average delay and contention time in Free Access with pre-emption. 47
4.22 Comparison of AoI between Gated Access with pre-emption, Free Access with pre-emption, and Reactive Slotted ALOHA. 48
4.23 Standard deviation of average AoI for Gated Access, Free Access with pre-emption, and Reactive Slotted ALOHA. 48
4.24 Comparison of AoI between Free Access, Free Access with pre-emption, Gated Access, Gated Access with pre-emption, and Reactive Slotted ALOHA. 49
4.25 Standard deviation of average AoI for Free Access, Free Access with pre-emption, Gated Access, Gated Access with pre-emption, and Reactive Slotted ALOHA. 49

1 Introduction

1.1 Introduction

RTC plays a fundamental role in modern networked systems, where timely and accurate data transmission is essential for effective decision-making. Many applications, such as industrial monitoring, smart home automation, and autonomous systems, rely on continuously updated information to track changes in their environment. In these contexts, it is not only important to minimize transmission delays, but also to ensure that the received data remains fresh and relevant for the system's operation.

Consider an IoT-based gas leak detection system in an industrial facility. In such a scenario, sensors continuously monitor gas concentrations and transmit periodic readings to a central system. If a hazardous level is detected, an alarm may be triggered. In the case of a single alarm message, delay is a critical metric, as it determines how quickly an operator or automated response system receives the warning. However, for continuous monitoring, it is also important that the system has access to the most recent sensor data rather than outdated information. The relevance of this aspect depends on the application; in some cases, a steady stream of updated readings provides more useful insights than a single, low-latency packet.

This concept is captured by the metric known as AoI, which quantifies the freshness of the information available at the receiver. Unlike conventional metrics such as latency and throughput, which focus on transmission speed and efficiency, AoI measures how outdated the most recently received update is. While low latency ensures rapid delivery of individual packets, low AoI ensures that the receiver has access to up-to-date information over time. Depending on the application, optimizing AoI may be more relevant than minimizing delay alone.

In large-scale networks, multiple devices must share a common communication channel, leading to potential transmission conflicts. One widely used approach to handling medium access is through random access protocols, which allow devices to transmit data without a precoordinated schedule. These protocols are particularly suited for decentralized systems such as the IoT, where devices generate traffic unpredictably. However, random access inherently leads to collisions, where multiple devices transmit simultaneously, causing transmission failures and requiring retransmissions. These collisions introduce additional delays, affecting both latency and AoI.

Traditional multiple access schemes such as Time Division Multiplexing (TDM) and Frequency Division Multiplexing (FDM) avoid such collisions by allocating dedicated time slots or frequency bands to each device. However, these methods rely on strict synchronization and predefined allocation, making them inefficient in highly dynamic and decentralized environments like the IoT, where traffic patterns are often unpredictable.

Among random access methods, Slotted ALOHA is a widely studied approach, where devices transmit at the beginning of predefined time slots. While this reduces the chance of partial packet collisions, it does not eliminate simultaneous transmissions. When multiple devices attempt to transmit in the same slot, a collision occurs, leading to retransmissions. The frequency of collisions, retransmission policies, and access mechanisms directly impact AoI, as packets may become outdated before they are successfully delivered.

This thesis investigates the impact of random-access mechanisms on AoI, evaluating different approaches to managing medium access. In particular, two key access strategies are analyzed:

- **Uncoordinated access (Free Access):** Devices transmit as soon as they generate a new packet, without any waiting mechanism or contention resolution. This allows for immediate transmissions but leads to frequent collisions, potentially increasing AoI.
- **Collision-managed access (Gated Access):** Devices wait for ongoing collisions to be resolved before attempting transmission. This structured approach reduces contention but introduces additional delays before new packets can be transmitted.

By studying these access strategies, this thesis explores how factors such as system load, contention level, and retransmission policies affect AoI. Additionally, structured collision resolution mechanisms are analyzed to determine their effectiveness in reducing AoI compared to simpler random access schemes.

To quantify these effects, key performance metrics are evaluated, including AoI distributions, standard deviation of AoI, and the probability mass function (PMF) of delay values. Furthermore, the concept of Collision Resolution Intervals is introduced as a means of analyzing the time required to resolve collisions before new transmissions can take place.

The contribution of this thesis lies in its systematic evaluation of different access strategies and their impact on AoI, providing a comparative analysis that extends beyond conventional performance metrics like delay and throughput. By focusing on information freshness in IoT environments, this research offers insights that can help improve medium access control mechanisms for time-sensitive applications.

1.2 Motivation

As a Telecommunications Engineering student, I chose this topic because understanding how information stays fresh in a network is very important for many modern communication systems. Real-time data is critical in areas like mobile networks, the Internet of Things, and wireless communications, which are all part of my field of study.

The AoI is a useful way to measure how up-to-date the data is, which is different from traditional metrics like delay or throughput. By studying random access protocols and their impact on AoI, I can learn how different communication methods affect the quality and reliability of data transmission.

This knowledge is relevant because many current and future telecom systems need to deliver fresh information quickly and efficiently, especially as networks become more crowded and complex. Working on this topic also helped me develop skills in designing and analyzing communication protocols, modeling network behavior, and using simulation tools. I have learned to develop and improve communication systems in practice. This experience will be very relevant for my future career in telecommunications, where real-time communication is essential.

1.3 Objectives

The primary objective of this thesis is to gain a comprehensive design experience in the context of real-time communication systems, particularly those that rely on random-access mechanisms. Through both theoretical and practical exploration, the study aims to engage in the full design cycle: identifying a performance-critical problem, developing analytical models, simulating alternative solutions, and evaluating trade-offs to propose design improvements. Within this broader goal, the following specific objectives are pursued:

1. **Understanding AoI in Communication Networks:**
   Provide a clear and structured introduction to the concept of AoI, explaining its relevance as a performance metric in modern real-time systems such as IoT, cyber-physical systems, and autonomous networks. Emphasis will be placed on contrasting AoI with traditional metrics like latency and throughput.

2. **Evaluating Random Access Protocols:**
   Investigate how different random-access control mechanisms affect the timeliness of information updates. This includes understanding the operational principles of these protocols and how they interact with network traffic and user dynamics.

3. **Analyzing AoI Performance Metrics:**
   Examine key statistical and probabilistic properties that shape AoI performance, such as average AoI, peak AoI, update intervals, and the distribution of packet delivery times. These analyses will help quantify how different network behaviors affect information freshness over time.

4. **Comparing System Behaviour Under Varying Conditions:**
   Analyze how AoI performance scales with system parameters such as the number of users, arrival rates, and transmission probabilities. This objective involves identifying the trade-offs between increased network contention and the freshness of information.

5. **Optimizing Transmission Strategies:**
   Identify and recommend optimal protocol configurations that minimize AoI while maintaining acceptable levels of throughput, reliability, and fairness. This involves exploring different parameter spaces, such as access probability in Free Access and gating intervals in Gated Access protocols.

6. **Developing Practical Insight Through Design-Oriented Analysis:**
   Reinforce the design experience by iteratively modeling, simulating, evaluating, and improving communication protocols. The intention is not only to analyze existing mechanisms but to understand the implications of design decisions and provide informed recommendations for future network architectures.

1.4 Methodology

The methodology for this study is structured around a combination of theoretical analysis, mathematical modelling, and simulation-based evaluation. The objective is to comprehensively analyze the behavior of AoI in various random-access communication environments and assess the efficiency of different protocol designs. The process consists of the following stages:

1. **Literature Review:**
   A comprehensive review of state-of-the-art research on AoI and random-access protocols was carried out. This includes foundational work on ALOHA-based protocols, carrier sense mechanisms, and collision resolution algorithms, as well as recent contributions focusing on AoI optimization.

2. **Mathematical Modelling:**
   Analytical models were developed to characterize the evolution of AoI under various access control schemes, including Free Access and Gated Access. Metrics such as average AoI, peak AoI, and system throughput were formally defined and derived under both finite and infinite population scenarios. These models provide insight into the inherent trade-offs in protocol design.

3. **Simulation Framework:**
   A custom simulation environment was implemented in MATLAB to replicate the behavior of different random-access protocols. The framework allows for flexible configuration of network parameters such as user population, slot structure, and transmission probabilities. The simulations mimic real-world network dynamics to complement and validate the theoretical models.

4. **Performance Evaluation:**
   The simulation outputs were systematically compared with the theoretical predictions. Key performance indicators such as AoI, collision rates, and channel utilization were analyzed to assess the consistency and reliability of each protocol under varying network loads and contention levels.

5. **Comparative Analysis:**
   By evaluating multiple access protocols under consistent test conditions, the strengths and limitations of each scheme were highlighted. This comparative approach supports a design-oriented perspective, focusing on how access strategies influence information freshness and overall network performance.

6. **Conclusion and Recommendations:**
   Based on both the analytical results and simulation outcomes, conclusions were drawn regarding the suitability of different random-access strategies for AoI-sensitive applications. Recommendations are offered for future research, protocol improvements, and real-world deployment scenarios.

1.5 Contribution to the SDGs

This thesis mainly contributes to the following Sustainable Development Goals:

Firstly, it strongly aligns with **SDG 9: Industry, Innovation and Infrastructure** by providing insights for optimizing communication protocols, which are fundamental for building modern, reliable, and resilient telecommunication infrastructures. This supports the development of advanced networks required in many critical applications.

Secondly, it contributes moderately to **SDG 11: Sustainable Cities and Communities** by enabling more efficient real-time communication for smart city applications, improving safety and resource management.

Additionally, this work supports **SDG 4: Quality Education** by advancing knowledge and skills in telecommunications engineering, fostering academic and technical development.

Finally, the research also relates at a lower level to **SDG 7: Affordable and Clean Energy** and **SDG 13: Climate Action** by potentially improving energy efficiency in communication networks through reducing unnecessary transmissions.

1.6 Structure of the Document

This thesis is organized into five chapters, each addressing a key aspect of the study.

- **Chapter 1: Introduction**
  This chapter provides an overview of the research context, the motivation behind the study, the objectives pursued, and the methodology adopted. It sets the foundation for understanding the relevance and scope of the thesis.

- **Chapter 2: Introduction to Random-Access Protocols**
  This chapter presents a review of classical and modern random-access protocols, including Pure ALOHA, Slotted ALOHA, and collision resolution algorithms. It also introduces the concept of Free Access and Gated Access as access control strategies.

- **Chapter 3: Age of Information**
  This chapter introduces the concept of AoI as a performance metric for real-time systems. It explains its significance compared to traditional metrics like delay or throughput, and outlines how it is measured and interpreted.

- **Chapter 4: Impact of Access Control Mechanisms on Age of Information**
  This core chapter analyzes the influence of Free and Gated Access protocols on AoI, under both finite and infinite user scenarios. It includes an analytical comparison and explores the effect of pre-emption mechanisms on system performance.

- **Chapter 5: Conclusions**
  This chapter summarizes the key findings of the study, highlights the main contributions, and suggests directions for future work.

Additional sections include a glossary of acronyms and a list of references used throughout the thesis.

2 Introduction to Random-Access Protocols

Random-access protocols are fundamental to enabling multiple users to share a common communication channel efficiently. Unlike centrally coordinated multiple access schemes, where resources are allocated in advance, random access operates in a decentralized manner, allowing users to transmit data without prior coordination. This approach is particularly useful in environments with sporadic and unpredictable traffic patterns, such as mobile networks, satellite communications, and the IoT, where devices generate data intermittently [1].

The origins of random-access protocols date back to the early 1970s with the introduction of the ALOHA protocol, which allowed devices to send packets whenever necessary, relying on retransmissions upon detecting collisions. Over time, enhancements such as Slotted ALOHA and more advanced collision resolution strategies have improved network efficiency by mitigating contention and optimizing resource utilization. These protocols continue to be relevant today, especially in the context of Massive Machine-type Communication (mMTC), where a large number of devices need to access the channel with minimal overhead.

Recent advancements in random access have introduced techniques that better manage contention and improve system efficiency. Notably, methods exploiting Successive Interference Cancellation (SIC) have significantly enhanced performance by allowing receivers to recover packets from collisions instead of treating them as lost. These advances bridge the gap between traditional random access and more structured approaches, making them increasingly competitive with scheduled access methods. As a result, modern random-access protocols are now being reconsidered for large-scale applications, including IoT networks and future wireless standards [2].

2.1 History

The origins of random-access protocols date back to around 1970, when the University of Hawaii developed the ALOHA system to enable computer communications across the Hawaiian Islands using Ultra High Frequency (UHF) radio waves [3]. In the original ALOHA model, users transmitted data packets without synchronization. If an Acknowledgement (ACK) was not received within a specific timeframe, the user would assume a collision had occurred and retransmit the packet after a randomly determined back-off period.

While innovative, the pure ALOHA approach faced significant limitations due to frequent collisions, particularly as network traffic increased. To mitigate this, Slotted ALOHA was introduced, dividing time into fixed intervals (slots) [4]. Here users could only begin transmissions at the start of a time slot, which minimized the period during which collisions could occur and effectively doubled throughput compared to pure ALOHA. As networks expanded and collisions became a challenge, more advanced protocols emerged to further minimize collisions. Carrier Sense Multiple Access (CSMA) introduced the concept of ”listening before transmitting” where devices checked the channel for activity prior to sending data [5]. This approach was further enhanced with Carrier Sense Multiple Access Collision Detection (CSMA/CD) enabling devices to detect collisions during transmission and immediately halt their activity,

thereby conserving bandwidth [6]. Similarly, Carrier Sense Multiple Access Collision Avoidance (CSMA/CA) adopted a preventive approach by incorporating waiting periods and ACK mechanisms to minimize the likelihood of collisions, making it particularly effective in wireless environments [7].
The evolution of random-access protocols extended beyond terrestrial networks into satellite communications. Techniques like Collision Resolution Diversity Slotted ALOHA (CRDSA) emerged to address challenges such as long propagation delays and high operational costs associated with satellite links. CRDSA improved efficiency by transmitting multiple replicas of each packet, enabling receivers to recover original data even if some copies were lost or collided [8].

In modern applications, particularly in the Industrial Internet of Things (IIoT), random access protocols offer a scalable and flexible solution for managing a large number of devices. Traditional permission-based access methods can be inefficient in scenarios with sporadic traffic and dynamic connectivity. While random access does not guarantee ultra-reliable low-latency communication, it reduces signaling overhead and allows devices to transmit without requiring centralized scheduling, making it well-suited for mMTC scenarios.

2.2 Basic random Access Protocols

2.2.1 Pure ALOHA

Pure ALOHA represents the simplest form of random access. In the pure ALOHA system, users can transmit their data packets at any time, without any synchronization between the devices. The main idea is that each user sends its data packet and waits for an ACK from the receiver. If the ACK is not received, the user will retransmit the data packet after a random back-off period [9].

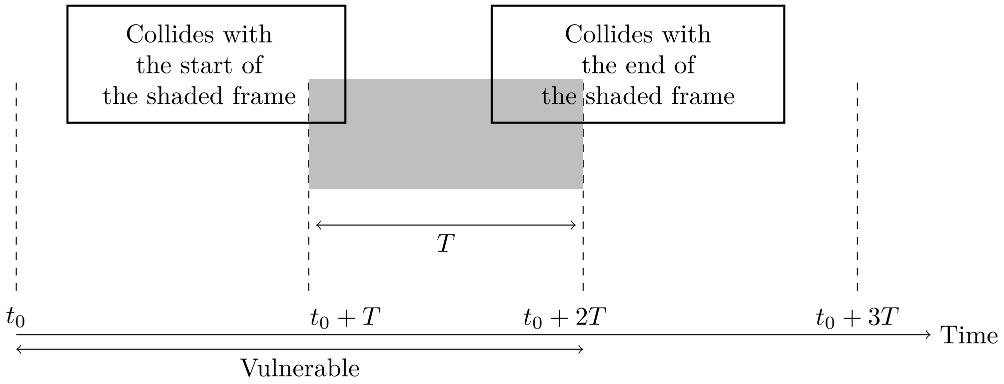


Figure 2.1: Collision occurrences in pure ALOHA.

The process works as follows:

1. A user sends a packet at any arbitrary time.
2. The receiver acknowledges successful reception.
3. If a collision occurs (i.e., multiple users transmit simultaneously and their signals interfere), successful reception is not guaranteed, and affected users retransmit after a random back-off period, following the *collision channel* model [10].

The key limitation of pure ALOHA is that as the number of users increases, the likelihood of packet collisions increases, leading to increased retransmissions and reduced system efficiency. Since users do not coordinate their transmissions, network resources are not optimally utilized, resulting in lower throughput.

Throughput (S), represents the average number of successfully received packets per unit time. It is defined as:

$$S = \frac{\text{Average number of successfully received packets}}{\text{Total time}} \tag{2.1}$$

A successful transmission occurs when exactly one user transmits during the vulnerable period $2T$. If multiple users transmit within this window, a collision occurs, and no packets are successfully received.
The probability of exactly k users transmitting in a given vulnerable period follows a Binomial distribution:

$$P_k = \binom{n}{k} p^k (1-p)^{n-k} \tag{2.2}$$

where p is the probability that a given user transmits, and n is the total number of users. Since a successful transmission requires exactly one user transmitting while all others remain idle, setting $k = 1$ gives:

$$P_{\text{success}} = P(k=1) = np(1-p)^{n-1} \tag{2.3}$$

For large n and small p, with $np = G$ held constant, the Binomial distribution can be treated as a Poisson distribution:

$$P(X=k) = \frac{e^{-G} G^k}{k!} \tag{2.4}$$

The offered Transmission Load (G) represents the average number of transmission attempts per unit time, including both new transmissions and retransmissions. Using this approximation for $k = 1$, we obtain:

$$P_{\text{success}} = P(X=1) = e^{-G} G. \tag{2.5}$$

The probability of zero additional transmissions during this period follows the Poisson distribution:

$$P_{\text{success}} = e^{-\lambda \cdot (2T)}. \tag{2.6}$$

Since we define G as the transmission attempts per unit time, we set $\lambda = G$, leading to:

$$P_{\text{success}} = e^{-2G}. \tag{2.7}$$

S is obtained as the product of the offered load and the probability of a successful transmission:

$$S = Ge^{-2G}. \tag{2.8}$$

To find the maximum throughput, differentiate $S(G)$ and set it to zero:

$$\frac{dS}{dG} = e^{-2G}(1-2G) = 0 \tag{2.9}$$

Solving $1 - 2G = 0$, we find $G = 0.5$, and evaluating at this point:

$$S_{\max} = 0.5 \cdot e^{-1} \approx 0.184. \tag{2.10}$$

This result shows that pure ALOHA achieves a maximum S of 18.4%, meaning that most transmission attempts result in collisions or idle periods.

2.2.2 Slotted ALOHA

Slotted ALOHA was introduced as an improvement over pure ALOHA to reduce collision probability and increase channel efficiency [4]. The key innovation was the introduction of time slots, where transmissions could only begin at the start of a predefined time interval, rather than at any arbitrary moment. This synchronization significantly lowered the chances of packet collisions compared to pure ALOHA.

The working process of Slotted ALOHA can be summarized as follows:

1. Time is divided into discrete slots, each equal to the time required to transmit a single packet.
2. A user wishing to transmit waits for the beginning of the next time slot before sending the packet.
3. If only one user transmits in a slot, the packet is successfully received.
4. If a collision occurs, all involved users assume failure and attempt retransmission after a random back-off period.

Slotted ALOHA improves upon pure ALOHA by restricting transmissions to begin only at the start of fixed time slots as shown in Figure 2.2, reducing the vulnerable period to T instead of $2T$. This decreases the probability of collisions and increases throughput.

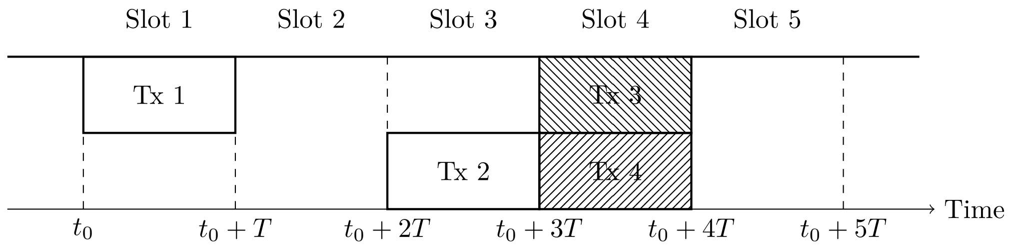


Figure 2.2: Collision occurrences in Slotted ALOHA.

Since the analysis follows the same approach as pure ALOHA, the probability of exactly one user transmitting in a slot follows a Binomial distribution:

$$P_{\text{success}} = np(1-p)^{n-1}. \tag{2.11}$$

Using the Poisson approximation for large n and small p, with $G = np$ held constant, this simplifies to:

$$P_{\text{success}} = Ge^{-G}. \tag{2.12}$$

S is given by:

$$S = Ge^{-G}. \tag{2.13}$$

Maximizing $S(G)$ by differentiation:

$$\frac{dS}{dG} = e^{-G}(1 - G) = 0. \tag{2.14}$$

Solving $1 - G = 0$, we find $G = 1$, and evaluating at this point:

$$S_{\max} = e^{-1} \approx 0.368. \tag{2.15}$$

Thus, Slotted ALOHA achieves a maximum throughput of 36.8%, which is double that of pure ALOHA, due to the reduced vulnerable period. We can see this in Figure 2.3

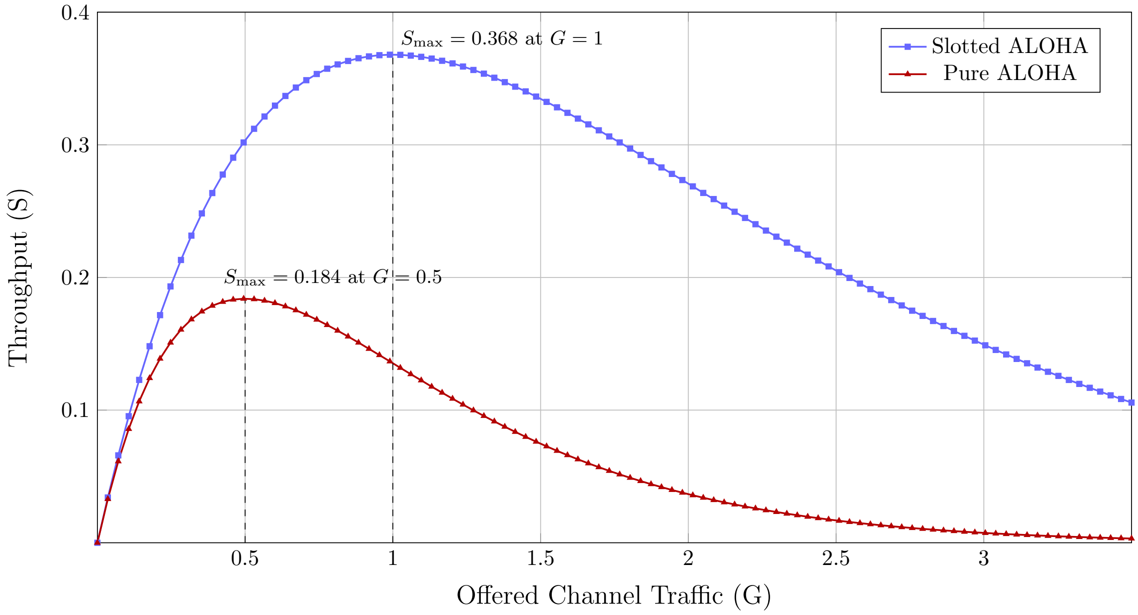


Figure 2.3: Comparison of throughput in pure and Slotted ALOHA

2.3 Collision Resolution Algorithms

In random-access communication networks, Collision Resolution Algorithms (CRA) play a crucial role in managing transmission conflicts that arise when multiple users attempt to send data over a shared channel simultaneously. A well-structured CRA ensures that once a collision occurs, all affected packets are eventually retransmitted and successfully received, avoiding indefinite delays [11].
To analyze how collisions are resolved, we consider an idealized model based on the following assumptions:

1. The channel is time-slotted, and each packet occupies exactly one slot. If multiple packets are transmitted in the same slot, a collision occurs, and the receiver detects it but cannot recover the individual packets.
2. At the end of each slot, the receiver provides feedback to all transmitters indicating whether the slot was:

  - Empty (no transmission occurred),
  - Successful (exactly one packet was received correctly),
  - A collision (two or more packets overlapped, requiring retransmission).

3. Propagation delays are negligible, meaning that the feedback from slot i is immediately available for making transmission decisions in slot $i + 1$. This allows transmitters to react without delay, facilitating efficient resolution of collisions.

For a CRA to be considered effective, it should satisfy the following key criteria: An effective CRA must efficiently resolve collisions to maximize the number of successful transmissions while minimizing retransmission delays, ensuring packets are delivered as quickly as possible. Unlike ALOHA-based systems, which rely on random retransmissions without guaranteeing resolution, structured approaches such as the Capetanakis algorithm provide a deterministic method for handling collisions. This ensures that all packets involved in a collision are eventually retransmitted successfully, while all transmitters remain aware of the resolution process. The introduction of these algorithms has significantly improved the efficiency of random-access protocols, making them more predictable and reliable in various communication scenarios.

2.3.1 Capetanakis Collision Resolution Algorithm (CCRA)

One of the most influential CRA was introduced by John Capetanakis in 1977 [12]. This algorithm, commonly known as the CCRA (or Serial Tree Algorithm), provides an organized approach to resolving collisions based on a binary tree splitting mechanism. CCRA operates as follows:
When a collision occurs, all transmitters involved in the collision flip a fair binary coin.

- Those flipping 0 retransmit in the next slot.
- Those flipping 1 defer their retransmission until the resolution of any further collisions among the 0-flipping group.
- No new packets are introduced into the system until the initial collision is fully resolved.

This process continues recursively, systematically isolating individual packets for successful transmission.

Example: Collision Resolution Using CCRA
Consider an scenario proposed in [11] in which four transmitters (A, B, C, and D) experience a collision in slot 1. The resolution process unfolds as follows:

- **Slot 1:** A collision occurs among A, B, C, and D.
- **Slot 2:** All four transmitters flip a coin. Suppose B and C flip 0, while A and D flip 1. Only B and C attempt retransmission, resulting in a new collision.
- **Slot 3:** B and C flip a coin again. Suppose C flips 0 and B flips 1. Thus, C transmits successfully.
- **Slot 4:** B recognizes the successful transmission of C and retransmits, successfully delivering its packet.
- **Slot 5:** A and D, which were waiting since slot 1, now retransmit.
- **Slot 6:** Suppose both A and D flip 1, resulting in an empty slot.

- **Slot 7:** A and D retransmit again, causing another collision.
- **Slot 8:** Suppose A flips 0 and D flips 1. A transmits successfully
- **Slot 9:** D retransmits and successfully delivers its packet.
- **Slot 10:** The system waits to ensure no remaining delayed packets. Once an empty slot appears, the original collision is resolved.

A,B C,D	B,C	C	B	A,D		A,D	A,D	A	A	
1	2	3	4	5	6	7	8	9	10	11

Figure 2.4: Example of a Collision Resolution Interval for the CCRA.

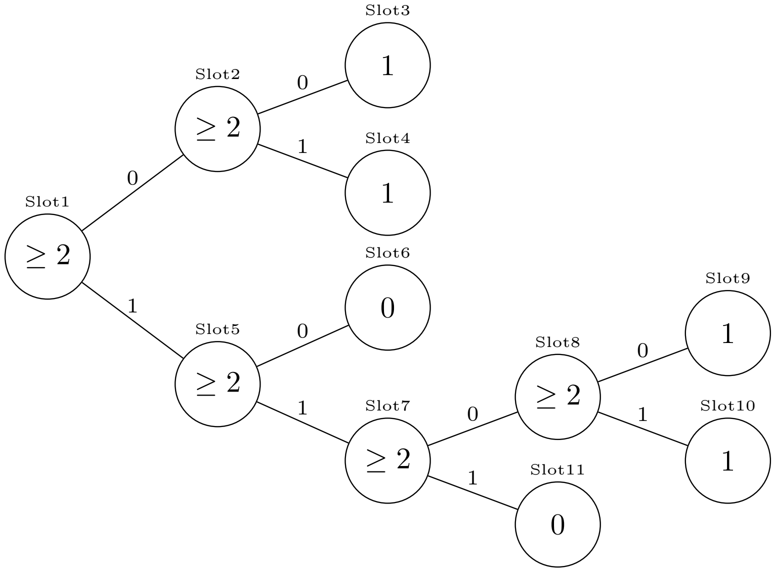


Figure 2.5: Tree Diagram for the Collision Interval of Figure 2.4.

The above example illustrates how the CCRA systematically resolves the collision while maintaining an organized retransmission order.
A binary rooted tree is a structure in which each node either has two branches or none. One fundamental property of such trees is that they contain exactly one more terminal node than intermediate nodes (or two more terminal nodes than intermediate nodes, excluding the root).

This relationship leads to an important characteristic of the CCRA: A collision in a given time slot is considered resolved precisely when the number of subsequent collision-free slots exceeds by two the number of slots that still experience collisions.
For example, referring to Figure 2.4, the collision that occurs in slot 2 is resolved in slot 4, the collision in slot 5 is resolved in slot 11, and the collision in slot 1 is also resolved in slot 11. This pattern illustrates that later collisions tend to resolve quickly.

This fundamental property of CCRA suggests an efficient way to implement the algorithm, as originally proposed by R.G. Gallager [13]:

When a transmitter flips to 1 following a collision in which it was involved, it performs the following steps:

- Initialize a counter to 1.
- Increment the counter by one for every subsequent collision slot.
- Decrement the counter by one for every collision-free slot.
- Retransmit when the counter reaches zero.

Additionally, it is crucial that all transmitters are aware when the original collision (if any) has been fully resolved. This knowledge determines when new packets may be introduced into the system. To achieve this, each transmitter maintains a second counter, and takes the following steps:

- Initialize the second counter to 1 just before the first slot.
- Increment it by one for every collision slot.
- Decrement it by one for every collision-free slot.
- When this second counter reaches zero, the original collision is resolved.

The CRI refers to the time period that begins with the original collision slot and ends when the original collision is resolved. If a slot is collision-free from the start, the CRI consists of just that one slot.
We can analyze the statistics of the CRI length, i.e., the number of slots within a CRI. For example, the CRI depicted in Figure 2.4 (or equivalently, Figure 2.5) consists of 11 slots. Since 4 packets are successfully transmitted during this interval, its throughput is:

$$\frac{4}{11} = 0.364 \text{ [packets/slot]} \tag{2.16}$$

This value is close to the theoretical maximum throughput limit of 0.368 [packets/slot] observed in Slotted ALOHA. The measured throughput suggests that CCRA effectively resolves collisions while maintaining transmission efficiency.

A key feature of CCRA is that it operates using only feedback indicating whether a slot contains a collision or is collision-free. Unlike protocols requiring explicit acknowledgment of successful transmissions (e.g., CSMA/CA), CCRA resolves collisions without needing feedback to distinguish between empty and successful slots. This reduces feedback complexity while maintaining efficient collision resolution.

2.4 Free Access and Gated Access Protocols

In random-access communication systems, the way users are allowed to access the channel significantly affects performance, efficiency, and fairness. The two primary approaches to managing access studied in this thesis are Free Access and Gated Access protocols. These methods define how new transmissions are initiated and how ongoing communications are managed within a network [14].

2.4.1 Free Access Protocols

Systems with sporadic traffic can benefit from Free Access protocols, where transmitters send packets immediately at the beginning of the next available time slot. This method allows quick access to the channel but increases the probability of collisions. Unlike Gated Access, where ongoing transmissions must be resolved before new ones are introduced, Free Access allows new packets to arrive and interfere with ongoing retransmissions, potentially extending the resolution process. These characteristics define the behavior of Free Access protocols:

- A transmitter sends a packet at the beginning of the next available time slot.
- This method allows quick access but increases the probability of collisions.
- Unlike gated access, new packets may arrive and interfere with ongoing retransmissions, potentially extending the resolution process.

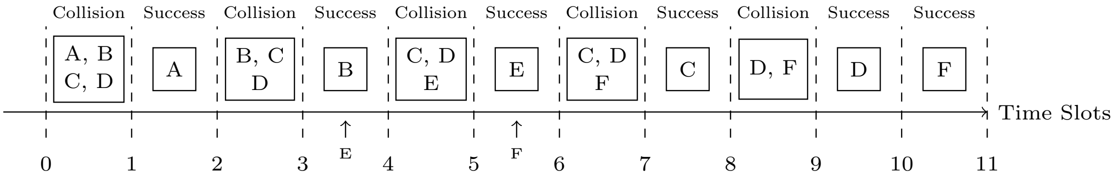


Figure 2.6: Illustration of Free Access protocol with CCRA.

Figure 2.6 illustrates a possible sequence of events in a Free Access protocol using the CCRA. The process follows this timeline:

- **Slot 0: Collision** – Packets A, B, C, and D attempt transmission simultaneously, leading to a collision. Since multiple packets are involved, the algorithm initiates a resolution process by splitting transmissions.
- **Slot 1: Success** – Only transmitter with packet A decides to transmit.
- **Slot 2: Collision** – All the colliding packets, B, C, and D, retransmit resulting in another collision.
- **Slot 3: Success** – Only packet B retransmits. It is now successfully transmitted and removed from contention. Also a new packet E arrives in this slot.
- **Slot 4: Collision** – Packets C, D and E retransmit. Since multiple packets are involved, another collision occurs.
- **Slot 5: Success** – Packet E retransmits successfully and is now removed from contention. A newly generated packet F waits for the next slot to transmit.
- **Slot 6: Collision** – Packets C, D and F retransmit, leading to another collision.
- **Slot 7: Success** – Packet C retransmits successfully.
- **Slot 8: Collision** – Packets D and F retransmit, causing another collision.
- **Slot 9: Success** – Packet D retransmits successfully.

- **Slot 10: Success** – Packet F successfully transmits, marking the final resolution of pending transmissions.

The CCRA efficiently handles the resolution of multiple colliding packets by iteratively splitting transmissions and ensuring that only a single packet is successfully transmitted per slot, while new packets are being added to the ongoing collisions because os the Free Access nature.

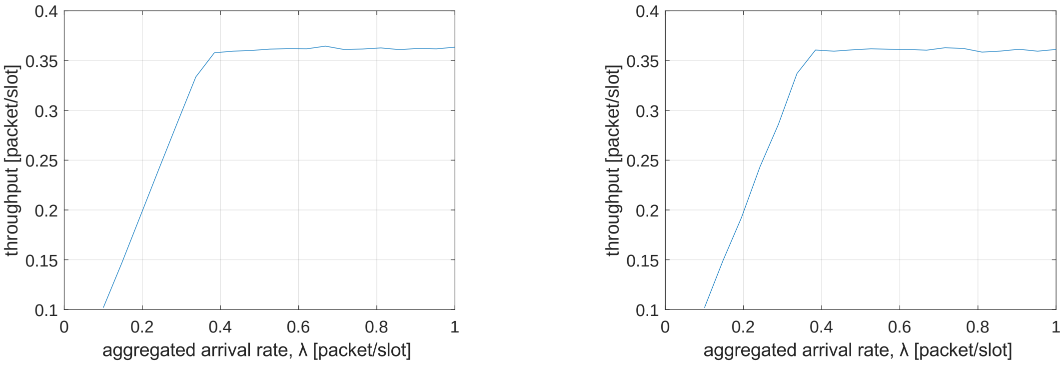
To analyze the performance of Free Access protocols, key metrics have been evaluated based on simulation results for 100 and 500 users that used CCRA to resolve collisions.

Each user independently generates a packet in a given time slot with probability $\frac{\lambda}{n}$, where λ is the total arrival rate and n is the number of users.

$$P_{\text{arrival}} = \lambda/n \tag{2.17}$$

where λ represents the total number of packet arrivals per time slot across all users.
We will first examine the relationship between throughput and arrival rate. Throughput measures the efficiency of successful packet transmissions over time and is given by Equation (2.1).

Figures 2.7a and 2.7b illustrate the throughput performance of Free Access protocols for networks with 100 and 500 users. They show that throughput increases with the arrival rate as more packets are successfully transmitted. However, beyond a certain point, throughput stabilizes at approximately 0.36 packets per slot, indicating that the system has reached its maximum achievable throughput. This saturation occurs because, at high loads, the contention for the channel increases, leading to a higher probability of collisions. As a result, additional transmission attempts beyond this point only lead to more collisions rather than more successful transmissions.



(a) 100 users in Free Access.

(b) 500 users in Free Access.

Figure 2.7: Throughput vs. Arrival Rate for different user counts in Free Access: (a) 100 users, (b) 500 users.

Figure 2.8 confirms that this saturation throughput is independent of the total number of users. Since throughput primarily depends on the aggregated arrival rate rather than the number of transmitters, a system with 100 users and another with 500 users achieve the same throughput when operating under similar traffic conditions. The probability of collisions and successful transmissions remains consistent as long as the overall system load is the same, explaining why both cases converge to the same performance limit.

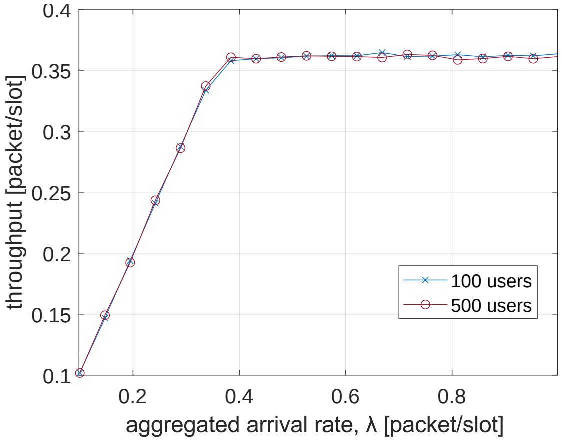
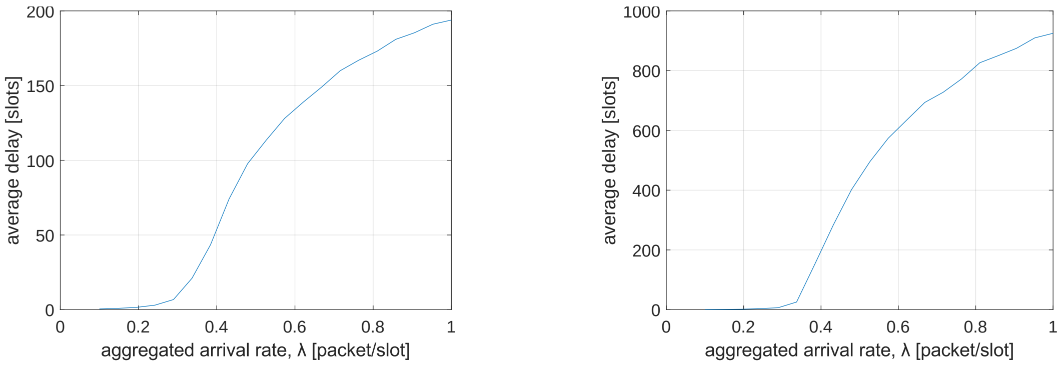


Figure 2.8: Throughput vs. Arrival Rate for 100 and 500 users in Free Access.

Now the relationship between delay and arrival rate is analyzed.
Figure 2.9a shows delay for 100 users, while Figure 2.9b presents results for 500 users. Delay remains low at low arrival rates but rises with increased contention and retransmissions. Unlike throughput, delay keeps increasing with congestion, especially in the 500-user scenario due to more frequent retransmissions.



(a) 100 users in Free Access.

(b) 500 users in Free Access.

Figure 2.9: Delay vs. Arrival Rate for different user counts in Free Access: (a) 100 users, (b) 500 users.

Figure 2.10 compares the delay trends for both network sizes. While both follow a similar pattern, the 500-user network experiences a sharper increase in delay due to higher contention. In Free Access, new packets can collide with ongoing retransmissions, causing some packets to take significantly longer to be successfully transmitted. This results in greater variability in delay as traffic load increases.

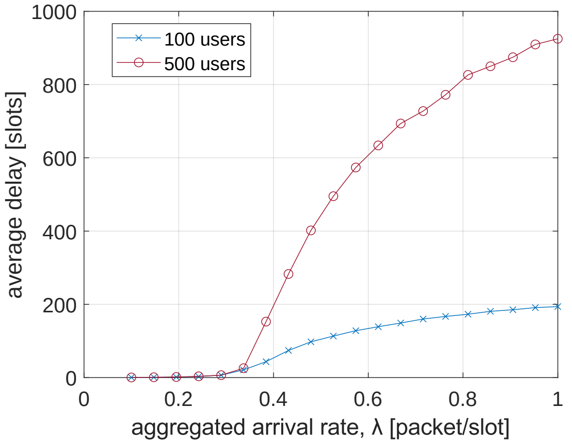


Figure 2.10: Delay vs. Arrival Rate for 100 and 500 users in Free Access.

2.4.2 Gated Access Protocols

In contrast to Free Access, Gated Access protocols regulate packet transmissions by ensuring that new packets are only sent after the resolution of ongoing collisions. This mechanism allows structured access to the channel. The following characteristics define the behavior of Gated Access protocols:

- New packets are only transmitted after all ongoing collisions are resolved, preventing interference.
- Transmitters follow a structured access policy, ensuring an orderly resolution process.
- Unlike Free Access, new arrivals do not interfere with ongoing retransmissions, preventing additional contention during collision resolution.

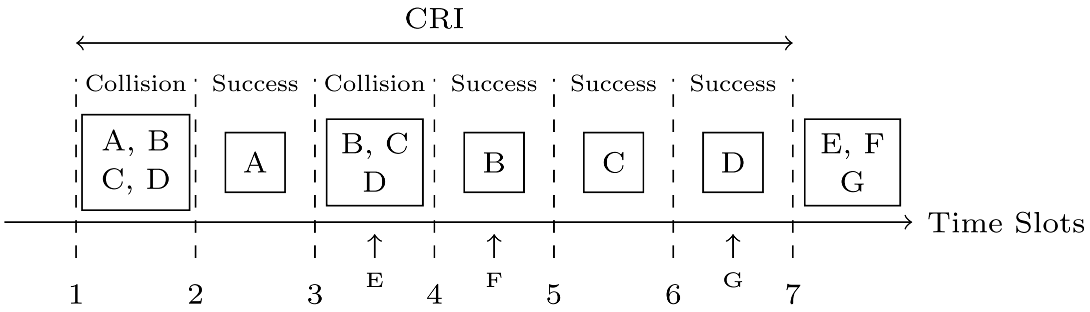


Figure 2.11: Illustration of the Gated Access protocol with CCRA.

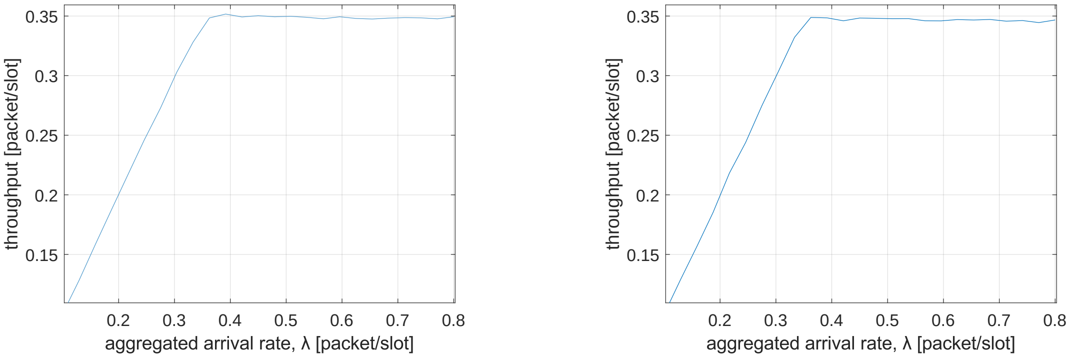
Figure 2.11 illustrates how transmissions occur under gated access using CCRA. The process begins with multiple packets colliding, triggering a structured resolution phase. In this example:

- **Slot 1: Collision** – Packets A, B, C, and D attempt transmission simultaneously, resulting in a collision.

- **Slot 2: Success** - Packet A is successfully transmitted and removed from contention. A new packet E arrives, and waits for the ongoing collision to be resolved.
- **Slot 3: Collision** - Packets B, C, and D attempt retransmission, leading to another collision.
- **Slot 4: Success** - Packet B is successfully transmitted. A new packet F arrives, and waits for the ongoing collision to be resolved.
- **Slot 5: Success** - Packet C is successfully transmitted.
- **Slot 6: Success** - Packet D is successfully transmitted. The CRI is now resolved. A new packet G arrives, and waits for the ongoing collision to be resolved.
- **Slot 7: New CRI** - Newly arrived packets E, F and G start a new CRI

To analyze the performance of gated access protocols, key metrics based on simulation results for 100 and 500 users that used CCRA are evaluated.

First the relationship between throughput and arrival rate is examined.



(a) 100 users in Gated Access.

(b) 500 users in Gated Access.

Figure 2.12: Throughput vs. Arrival Rate for different user counts in Gated Access: (a) 100 users, (b) 500 users.

Figure 2.12a shows the throughput for 100 users, while Figure 2.12b presents results for 500 users. In both cases, throughput initially increases with arrival rate before stabilizing at a maximum value. As in Free Access, there exists a limit to the achievable throughput.

Figure 2.13 compares the trends for both user counts. Notably, throughput stabilizes at similar levels for 100 and 500 users, showing that performance is primarily governed by the aggregated arrival rate rather than the total number of users. This is because, at high loads, the system is continuously processing packets, meaning that adding more users does not change the rate at which transmissions can be successfully completed.

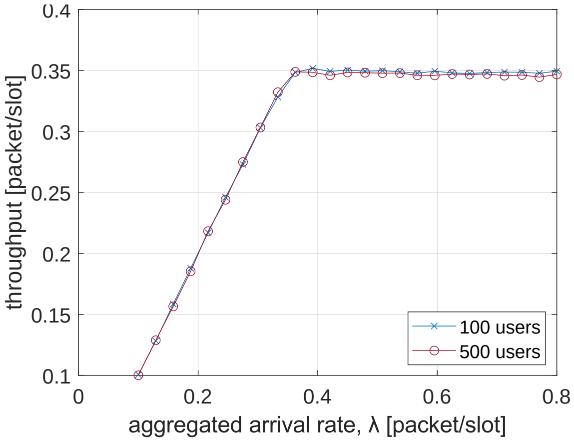


Figure 2.13: Throughput vs. Arrival Rate for 100 and 500 users in Gated Access.

Figure 2.14 compares the throughput performance of Gated and Free Access protocols for 100 users. Free Access achieves a slightly higher peak throughput since nodes can transmit immediately upon packet arrival, making full use of available time slots. However, this also results in higher contention and frequent collisions. In contrast, Gated Access enforces an ordered transmission structure, ensuring that ongoing retransmissions are resolved before new packets attempt transmission. This reduces contention but introduces waiting times, which slightly lower the maximum achievable throughput compared to Free Access. Despite this difference, both protocols exhibit similar behavior at high loads, where throughput stabilizes at its maximum value.

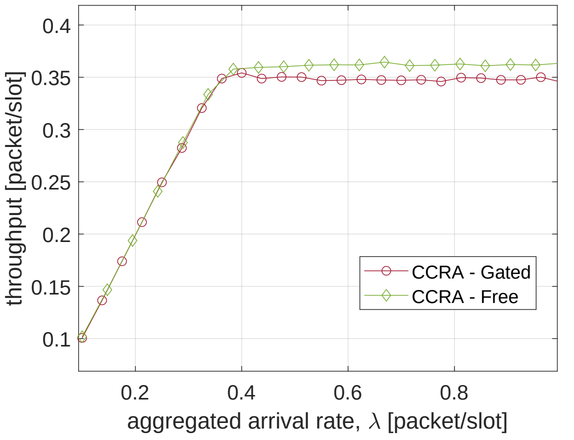
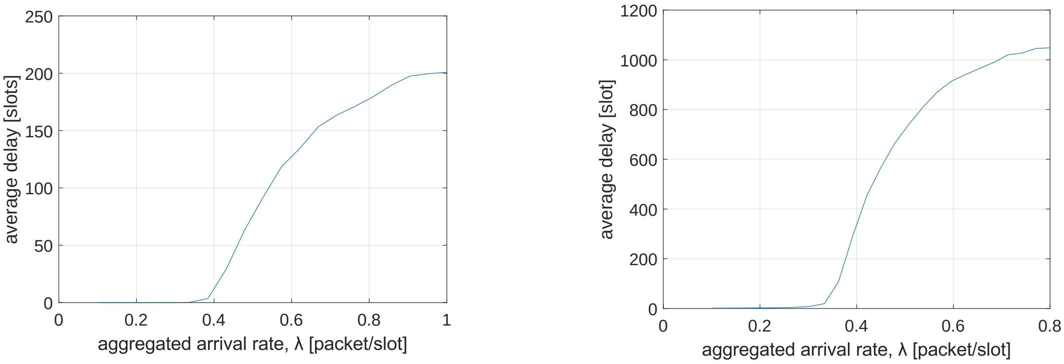


Figure 2.14: Comparison of Throughput vs. Arrival Rate for 100 users in Gated and Free Access.

The delay characteristics of Gated Access are distinct from those observed in Free Access. Figure 2.15a presents the delay trends for 100 users, while Figure 2.15b does so for 500 users. Initially, delay remains low as the system operates efficiently, with minimal waiting times. However, as arrival rate increases, delay grows significantly due to the enforced ordering of transmissions.



(a) 100 users in Gated Access.

(b) 500 users in Gated Access.

Figure 2.15: Delay vs. Arrival Rate for different user counts in Gated Access: (a) 100 users, (b) 500 users.

Figure 2.16 compares both user groups. The 500-user scenario experiences higher delays due to the increased contention and enforced transmission ordering.

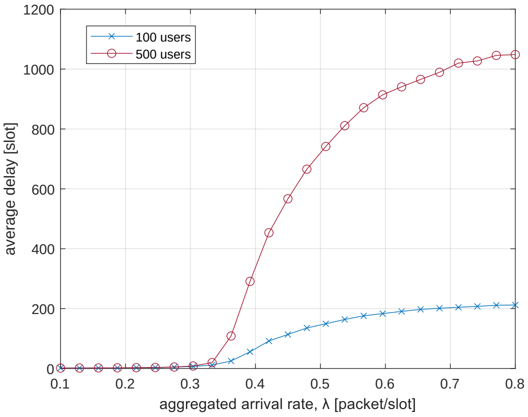


Figure 2.16: Delay vs. Arrival Rate for 100 and 500 users in Gated Access.

As we can see in Figure 2.17 at low arrival rates, both Free and Gated Access exhibit minimal delay since collisions are rare. However, as traffic increases, Free Access experiences a sharp rise in delay due to uncontrolled retransmissions following collisions. In contrast, Gated Access mitigates collisions by enforcing structured transmissions, leading to a more controlled increase in delay. Nevertheless, at high arrival rates, both methods experience significant delays as network saturation results in longer queues and waiting times, limiting system responsiveness.

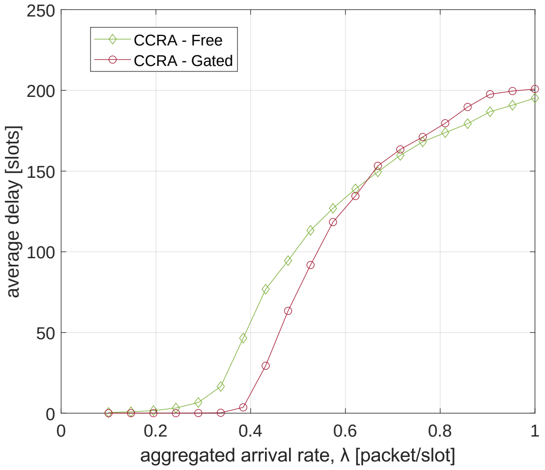


Figure 2.17: Comparison of Delay vs. Arrival Rate for 100 users in Gated and Free Access.

Another important aspect to analyze is the PMF of CRI duration as a function of the arrival rate. As previously discussed, a CRI represents the number of time slots required to resolve a collision. Examining its distribution under different arrival rates provides valuable insights into system efficiency, particularly in understanding how contention resolution scales with increasing traffic load. Figure 2.18 illustrates the probability mass function (PMF) of the CRI duration for different aggregated arrival rates λ in a 100-user Gated Access system. Each color represents a different λ, showing the effect of traffic load on CRI duration. At low λ values, fewer packets are generated per slot, leading to fewer collisions and shorter CRI durations, as there are fewer packets competing for retransmission. This results in a distribution concentrated around smaller CRI values. As λ increases, more packets are generated, meaning that more users contend for access, and the resolution process requires a greater number of slots. Consequently, the CRI duration distribution shifts toward longer intervals, as indicated by the increased spread and higher probabilities of larger CRI values. This trend shows that higher arrival rates increase network contention, extending the time required to resolve collisions before new packets can be transmitted.

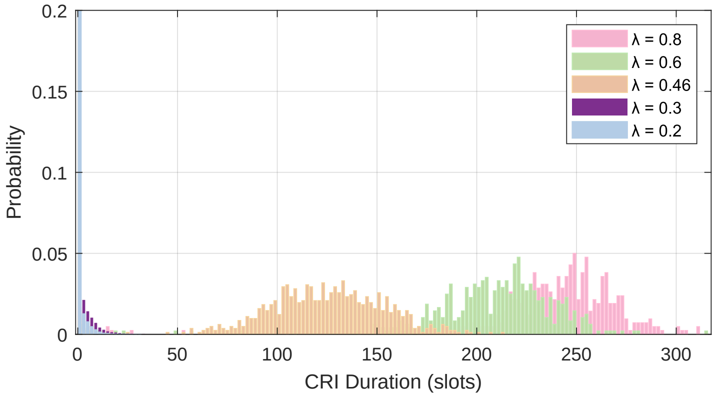


Figure 2.18: PMF of CRI duration for different arrival rates in a 100-user Gated Access system.

Another factor to consider is the PMF of CRI duration as a function of the number of nodes. Figure 2.19 illustrates how the number of competing users impacts the duration of CRI in a

Gated Access system, providing insight into how contention scales with an increasing number of participants.

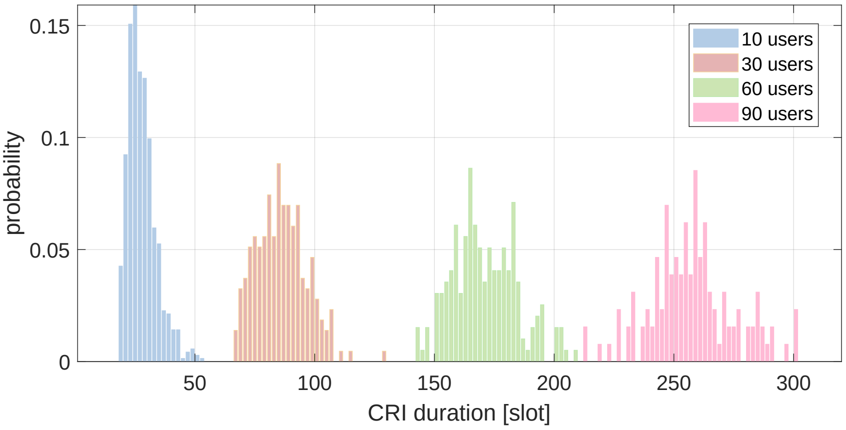
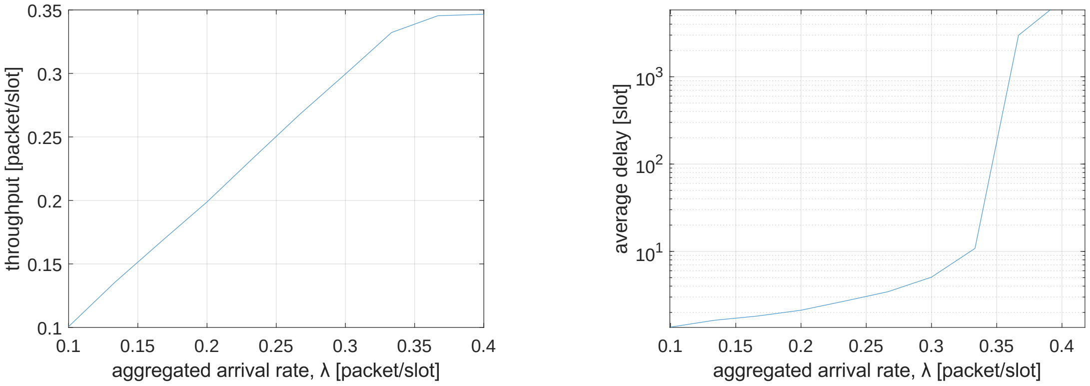


Figure 2.19: PMF of CRI duration for different numbers of users in a 100-user Gated Access system.

When the number of users is small, collisions are less frequent, and when they do occur, they involve fewer packets, allowing for faster resolution. As a result, the PMF is concentrated around shorter CRI durations. However, as the number of users increases, contention for the channel becomes more intense, leading to longer and more complex resolution periods. This results in a broader distribution, where longer CRI durations become more probable. The system requires additional time slots to sequentially resolve larger groups of colliding packets, extending the time before new packets can be successfully transmitted.

In an infinite-user scenario, the behavior of the Gated Access scheme is fundamentally different from the finite-user case. Since there is no limit to the number of potential transmitters, the system continuously experiences new generations of packets. However, due to the structured nature of Gated Access, where new packets must wait for the resolution of previous transmissions before being allowed to contend, a critical bottleneck emerges. As the arrival rate increases, CRI durations become progressively longer, leading to a situation where each CRI exceeds the duration of the previous. This effect results in a growing backlog of packets waiting to be resolved, and beyond a certain threshold, the system enters a congestion state where CRI lengths diverge toward infinity.

Initially, throughput increases with arrival rate, as seen in Figure 2.20a. However, once the system approaches the maximum sustainable throughput around $\lambda \approx 0.35$, resolution times grow exponentially, as depicted in Figure 2.20b. This is a direct consequence of the increasing CRI durations, since all newly generated packets must wait for the completion of an ongoing CRI, a higher arrival rate leads to longer waiting times, extending the time required for each resolution cycle. Eventually, the system cannot continue to process updates efficiently, causing an exponential increase in delay.



(a) Throughput as a function of the arrival rate λ in an infinite-user Gated Access system.

(b) Average delay as a function of the arrival rate λ in an infinite-user Gated Access system.

Figure 2.20: Comparison of throughput (a) and average delay (b) in an infinite-user Gated Access system as a function of the arrival rate λ.

This behavior is further reinforced by the theoretical bounds derived by Massey [11], which confirm the sharp increase in delay at higher arrival rates. As shown in Figure 2.21, the delay-to-throughput relationship follows a steep curve, demonstrating that as the system approaches its throughput limit, the time required to successfully deliver packets escalates uncontrollably.

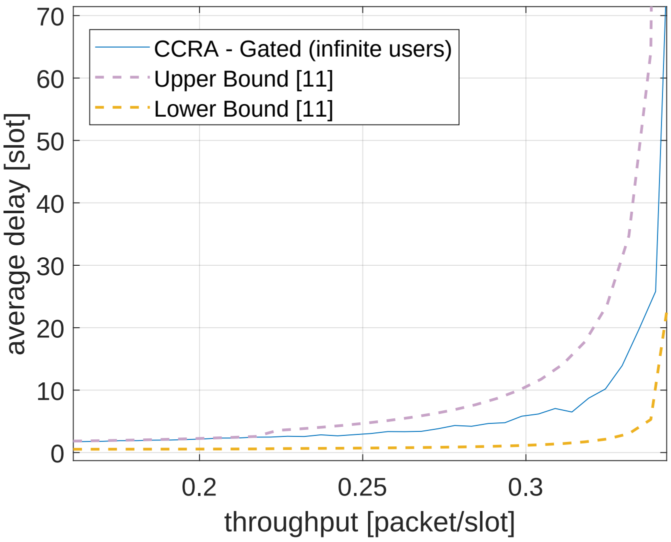


Figure 2.21: Average delay as a function of throughput in an infinite-user Gated Access system.

3 Age of Information

In many modern IoT systems, a large number of devices continuously transmit time-stamped updates to a central gateway. This is particularly relevant in applications such as remote sensing, industrial monitoring, vehicular networks, and environmental tracking. Traditionally, communication performance has been evaluated using metrics like throughput and delay. However, these measures do not fully capture the ability of a system to provide up-to-date information, which is critical in scenarios where real-time status updates are essential.

To address this limitation, the concept of AoI has emerged as a key performance metric, offering insights that complement traditional indicators [15]. While extensive research has been conducted on AoI in point-to-point communications, its behavior in systems employing modern link-layer protocols, particularly in IoT settings, remains less explored. This is especially relevant for random access protocols, which have gained prominence in supporting massive machine-type communications [16].

AoI is a metric that quantifies the freshness of data at the receiver, specifically for a given user. Unlike traditional delay, which measures the time taken for a packet to travel from the sender to the receiver, AoI focuses on how up-to-date the information is at the receiver, considering the time elapsed since the last successfully received packet was generated.

Consider the following example: a sensor produces a reading at time t_0 and transmits it to the server. For a given user (e.g., the sensor), the delay is defined as the time it takes for the reading to reach the server. At time t_1, the packet is successfully received by the server, and the delay for this packet is simply the time difference, $t_1 - t_0$.

Now, if no new updates are received from the sensor, the AoI at time t_1 will also be $t_1 - t_0$, since the last successful update was generated at t_0 and received at t_1. However, as time progresses and no further updates are transmitted, the AoI increases linearly. At time t_2, the AoI would be $t_2 - t_0$, and at time t_3, the AoI would be $t_3 - t_0$, and it will continue to grow until the next successfull reception. This linear increase in AoI reflects how the information at the receiver becomes progressively more outdated. In contrast, delay remains fixed at $t_1 - t_0$, because delay only measures how long it took for a particular packet to arrive at the receiver. So the main difference between these two metrics os that delay is a one-time measurement for each individual packet, while AoI is an ongoing metric that updates over time and reflects the freshness of data continuously.

For each user, the AoI is evaluated independently based on the time of the last successful update received and how long it has been since that update. This continuous evaluation makes AoI a more dynamic and comprehensive metric for assessing the freshness of data, especially in systems where updates occur intermittently, and timely updates are crucial.

At any given time t, AoI is defined as:

$$\delta(t) = t - u(t) \tag{3.1}$$

where t is the current time at the receiver, and $u(t)$ represents the generation time of the most recently received packet. AoI decreases whenever a new update is successfully received and increases linearly in the absence of updates, capturing the growing staleness of information, as shown in Figure 3.1.

To provide a long-term measure of how fresh information remains over a period T, the average AoI can be used:

$$\Delta = \frac{1}{T}\int_0^T \delta(t)\,dt \tag{3.2}$$

where T is the total observed period. This metric is particularly useful in evaluating systems where updates are either periodic or irregular, offering insight into how consistently new information is delivered. However, in some systems, especially those with unpredictable or bursty traffic patterns, other metrics, such as peak AoI, may provide a better understanding of the performance in critical situations. Average AoI may not always be the best choice, as it tends to smooth out the differences in update frequency and packet delivery. For example, in safety-critical applications, the maximum delay between updates can be more important than the average freshness of the information.

Peak AoI quantifies the worst-case AoI right before a new update is received. It is defined as:

$$\delta_{\text{peak}} = t_i - u(t_{i-1}) \tag{3.3}$$

where t_i is the time when the i-th packet is successfully delivered, and $u(t_{i-1})$ is the generation time of the previous successful update. Peak AoI highlights moments when information at the receiver is at its stalest, making it a useful metric for time-sensitive applications.

To further evaluate worst-case performance, the average peak AoI considers the mean of all peak AoI values over a given observation period T. It is calculated as:

$$\Delta_{\text{peak}} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\text{peak},i} \tag{3.4}$$

where N is the total number of successful updates received, and $\delta_{\text{peak},i}$ corresponds to the peak AoI of the i-th update. While average AoI provides a general sense of freshness, average peak AoI focuses on the worst-case staleness over time. This distinction is especially relevant in safety-critical systems, where infrequent but extreme delays can have significant consequences.

Several factors influence AoI in communication systems. A higher update frequency helps maintain fresher information at the receiver, directly lowering the AoI, if all updates are successfully received. However, while a higher update rate could tend to reduce AoI, increased congestion and competition for the shared communication channel often lead to higher collision probabilities, meaning there will be less successful transmissions, causing AoI to increase, as the information at the receiver becomes more outdated.

These factors highlight the importance of AoI as a critical metric in modern communication systems, especially in applications where not only reliability and delay but also data freshness are essential for system performance.

The evolution of AoI over time for a single user follows a characteristic sawtooth pattern, as shown in Figure 3.1, which considers a single user transmitting updates in a time-slotted system. Whenever a new packet is successfully received, AoI drops to the difference between the current time and the generation time of that packet, as stated in Equation 3.1. In the absence of updates, AoI increases linearly over time until another successful update arrives, resetting it once again.

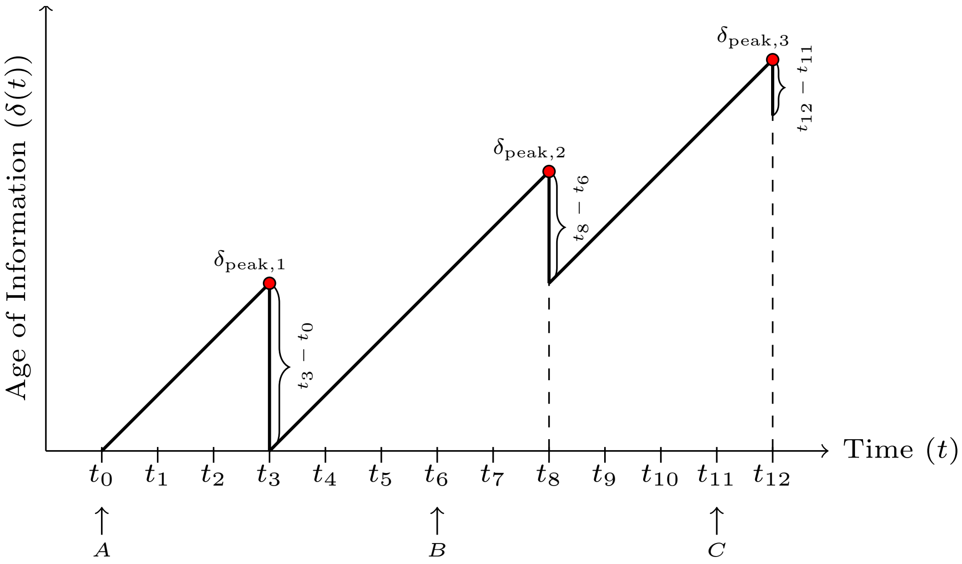


Figure 3.1: AoI evolution for a single user in a slotted system.

In Figure 3.1, at time t_0 a new packet A is generated. This packet is successfully received at time t_3, and at this point, the AoI is reset. The reset value of AoI is the time difference between the generation time t_0 and the reception time t_3. This is referred to as the first observed value of peak AoI, as the AoI increases linearly from t_0 to t_3 and resets at t_3, which is reflected in Equation 3.3. After the first update, the AoI begins to increase again linearly as no new updates are received. At time t_6, a new packet B is generated. This packet is received at t_8, and again, the AoI resets. The reset value of AoI at this point is the time difference between the generation time t_6 and the reception time t_8. This represents the second observed value of peak AoI. Then at time t_{11} packet C is generated. The packet is successfully received at t_{12}, and the AoI is reset again. The reset value of AoI at this point is the time difference between the generation time t_{11} and the reception time t_{12}. This is the third observed value of peak AoI, and it reflects the maximum AoI value reached between the 3 updates.

To gain a broader understanding of how AoI behaves across multiple users in a communication system, we analyze the average AoI as a function of the system's traffic load. For illustration, consider a slotted ALOHA system, where multiple users contend for access to the shared channel. In such systems, the transmission of updates by users can result in packet collisions affecting the freshness of the data received by the central gateway. The packet generation rate, represented by the aggregated arrival rate λ, plays a key role in influencing the freshness of the data. In this study, we focus on the impact of λ on the average AoI, as this provides insights into how frequently updates are delivered and how the system's performance is affected by the rate of packet generation.

Figure 3.2 illustrates the relationship between the normalized average AoI and the aggregated arrival rate λ in a slotted ALOHA system. The normalized average AoI is the average AoI value per λ, normalized by the total number of users sharing the channel, which will be useful for comparison between schemes where different number of users are sharing the channel.

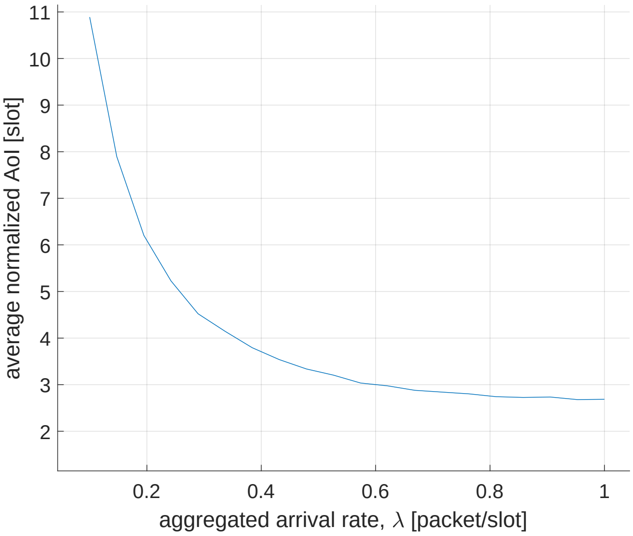


Figure 3.2: Normalized average AoI as a function of the aggregated arrival rate λ in a slotted ALOHA system.

At low λ, updates are rare, and as a result, the system experiences prolonged periods without fresh data. This leads to a higher average AoI. As λ increases, the update frequency increases, and the AoI decreases because updates are received more frequently, keeping the data fresher. However, when λ becomes higher, the system experiences increased contention between users sharing the channel. This results in more packet collisions, which delay successful transmissions and limit the improvement in AoI as λ increases.

Unlike in previous work, such as [16], where AoI is presented in terms of channel load G, in this thesis, results of AoI are presented in terms of λ, as it directly correlates with packet generation rates and allows for a more intuitive understanding of AoI in relation to system traffic. Typically, in slotted ALOHA systems, the relationship between AoI and channel load is expected to exhibit a U-shape, where AoI initially decreases as the update frequency increases, but then increases again due to packet collisions at higher traffic loads. In Figure 3.2, we observe a similar trend, where AoI decreases with increasing λ up to a point, but then differs as it reaches a lower bound, as collisions become more frequent. This behavior reflects the balance between receiving updates more often and dealing with the delays caused by congestion.

To further study the AoI in random access, we analyze three different variants of Slotted ALOHA, where each variant uses a distinct transmission strategy. Figure 3.3 illustrates the behavior of a single user transmitting packets to a receiver for each of the three schemes. In all cases, packet generation is given by λ/n, where λ is the packet arrival rate and n is the total number of users in the system.

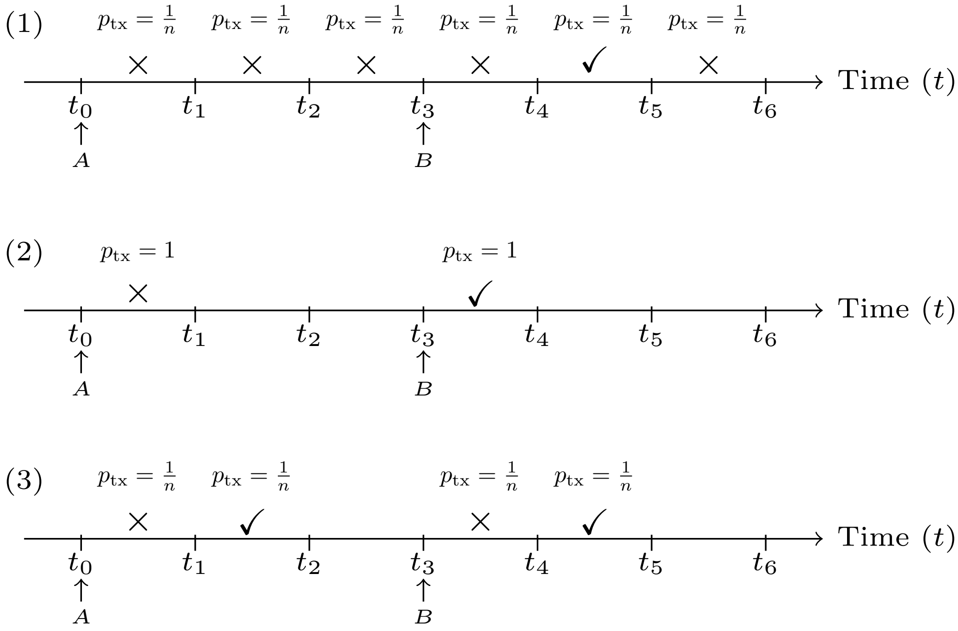


Figure 3.3: Example of communication between a single user and the receiver for the three Slotted ALOHA variants.

In the first variant we will call it SA Max Throughput, (1) in Figure 3.3, the user tries to transmit with probability $p_{tx} = \frac{1}{n}$ at every slot. However, if the packet is not delivered and a new packet is generated, it can overwrite the old packet without any retransmissions. In the example shown in the figure, the user generates Packet A at time t_0 and continuously tries to transmit with $p_{tx} = \frac{1}{n}$ until time t_3, when Packet B is generated. At this point, Packet A is overwritten by Packet B because Packet A was never successfully delivered. This scheme's lack of acknowledgment or retransmission strategy could lead to an increase in AoI, especially when packets are overwritten before they can be successfully transmitted.

In the second variant, we will call it Reactive SA, (2) in Figure 3.3, a user transmits only when it has a newly generated packet, with transmission probability $p_{tx} = 1$. If the transmission fails due to a collision, the packet is lost, and no retransmissions are attempted. This reduces channel congestion but increases packet loss, as unsuccessful packets are never retransmitted. In the example shown, Packet A is generated at t_0. The user transmits it with $p_{tx} = 1$ at the next available slot, but it faces a collision and is never delivered. At t_3, the user generates Packet B and immediately tries to transmit it with $p_{tx} = 1$ successfully. In terms of AoI, if both packets had been successfully transmitted, their AoI would have been minimized, as they would have been received one slot after their generation. Since Packet B was successfully transmitted, it results in the lowest possible AoI, as packets are always received immediately after their generation.

The third variant we will call it SA with Acknowledgment, (3) in Figure 3.3, is similar to the first in that the user attempts to transmit with $p_{tx} = \frac{1}{n}$ at every slot, but once a packet is successfully transmitted, the user stops trying to retransmit it until a new packet is generated. Feedback from the receiver allows the user to stop retransmissions once the packet is successfully delivered. In this case, Packet A is generated at t_0, and the user attempts to transmit it with $p_{\text{tx}} = \frac{1}{n}$. Once Packet A is successfully delivered, it is acknowledged, and the user stops trying to transmit it. At t_3, Packet B is generated, and the user attempts to transmit it in the same way. Packet B is successfully delivered, and its successful transmission is acknowledged, causing the user to stop retransmitting it. We can suposse that scheme (3) would result in a lower AoI compared to scheme (1), as it reduces collision rates by preventing unnecessary retransmissions of already received packets.

To gain deeper insights into the performance of the three Slotted ALOHA variants, we examine the system's throughput S and the average AoI as functions of the aggregated arrival rate λ. These metrics give us a clearer understanding of how each variant performs under varying levels of arrival rate.

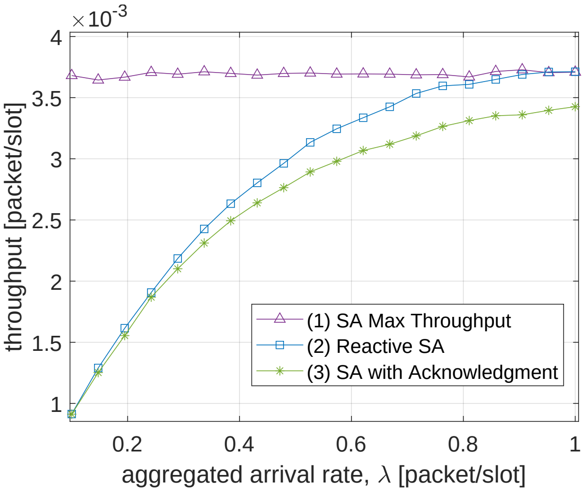


Figure 3.4: Throughput as a function of the aggregated arrival rate λ for different Slotted ALOHA variants.

Figure 3.4 shows the throughput as a function of the aggregated arrival rate λ. As expected, SA with maximum throughput (1) achieves the highest throughput across all values of λ. This is because packets are continuously retransmitted with $p_{\text{tx}} = \frac{1}{n}$, ensuring that even if collisions occur, the same packets are retransmitted multiple times, leading to a much higher number of successful receptions compared to the other schemes. In Reactive SA (2), throughput increases with λ but at a slower rate. This is because packets are transmitted only once when newly generated, avoiding retransmissions of older packets. While this reduces congestion and may lead to higher successful receptions, fewer transmissions overall result in lower throughput. SA with Acknowledgment (3) behaves similarly to Reactive SA, but with the added feedback mechanism. The user stops retransmitting once an acknowledgment is received, which causes more frequent transmissions and thus higher collisions, resulting in fewer successful receptions and lower throughput than SA with maximum throughput.

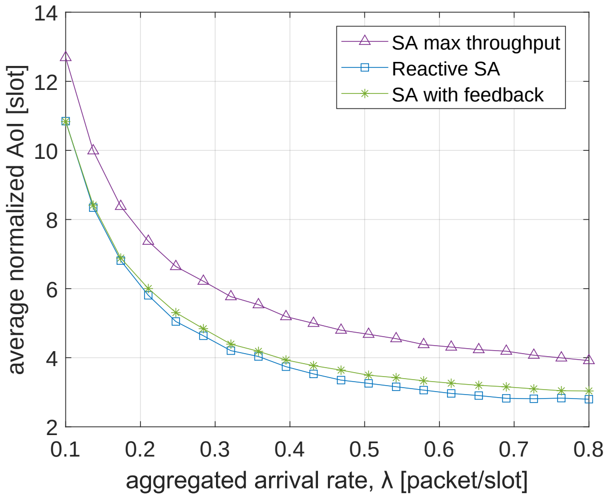


Figure 3.5: Normalized AoI as a function of the aggregated arrival rate λ for different Slotted ALOHA variants.

Figure 3.5 shows the average normalized AoI per slot for the three Slotted ALOHA variants, divided by the number of users. Reactive SA (1) achieves the best AoI because packets, when successfully received, are delivered one slot after their generation, minimizing staleness. SA with Acknowledgment follows, as the reduced transmission attempts lead to fewer collisions and more frequent updates. SA with maximum throughput results in the highest AoI as continuous retransmissions increase congestion and delay successfull updates.

These results highlight the trade-off between throughput and AoI. Depending on the application, it is important to prioritize either throughput or data freshness. In real-time applications, minimizing AoI is often more critical than maximizing throughput, while in other scenarios, higher throughput may be more important.

4 Impact of Access Control Mechanisms on Age of Information

This chapter investigates the impact of different random access schemes on the AoI in communication systems. We start by analyzing the AoI behavior in Gated Access and Free Access schemes, focusing on key performance metrics such as average AoI, standard deviation, and the PMF of delay values.

The chapter first examines Gated Access, exploring how the structured resolution of contention affects AoI under varying arrival rates. It then moves on to Free Access, where packets are transmitted without waiting for contention resolution, highlighting how this access method introduces greater variability in AoI due to more frequent collisions.

Following this, the concept of pre-emption is introduced to both Gated and Free Access schemes to improve the freshness of transmitted updates. Pre-emption allows users to replace older, pending packets with newly generated ones during ongoing contention, ensuring that the most recent information is prioritized. The chapter compares the performance of Gated and Free Access with pre-emption, analyzing their impact on AoI and comparing them to the original schemes without pre-emption.

Finally, a comprehensive comparison is made between all the access schemes, including both versions with and without pre-emption, to determine the best approach for minimizing AoI.

4.1 Gated Access

This first section examines Gated Access, where users must wait for the resolution of ongoing transmissions before new packets can access the system as introduced in Chapter 2. This method imposes a structured transmission order, reducing contention but also introducing waiting times. The results for a finite number of users are analyzed, using the CCRA.

Figure 4.1 presents the relationship between the normalized AoI and the aggregated arrival rate λ for Gated Access using CCRA and Reactive SA schemes with 100 users. At low arrival rates, AoI is high in both schemes due to long intervals between updates, as fewer packets are transmitted throughout both systems. As λ increases, the frequency of successful updates grows, reducing AoI. For values of λ less than approximately 0.45, the Gated Access scheme performs better, achieving a lower AoI compared to Reactive SA. This is because the Gated Access scheme limits contention by allowing only a subset of users to transmit, which leads to fewer collisions and more successful transmissions, thus keeping the AoI lower. However, beyond the threshold of $\lambda \approx 0.45$, Reactive SA starts to outperform Gated Access. In this range, the Gated Access scheme experiences increased contention as more users attempt to transmit, causing longer CRIs. This results in delayed transmissions, and as a consequence, the AoI increases. On the other hand, Reactive SA allows each user to transmit only once per packet generation, reducing unnecessary retransmissions and congestion, which leads to more frequent successful updates and a lower AoI beyond this point.

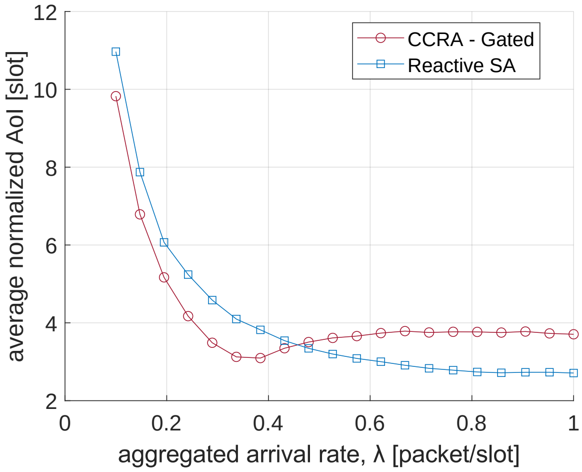


Figure 4.1: Normalized average AoI for Gated Access using CCRA and Reactive SA as a function of λ.

To further analyze the behavior of AoI in Gated Access, Figure 4.2 presents the standard deviation of AoI as a function of the arrival rate λ for both shcemes. A higher standard deviation indicates greater variability in update intervals, which is undesirable in applications requiring consistent and regular update cycles. The results show that Reactive SA exhibits a significantly higher AoI variance compared to Gated Access. In the reactive scheme, packets are transmitted immediately after generation, but if a collision occurs, they are lost with no retransmission. This leads to irregular intervals between successful updates, as some packets do not reach the receiver. In contrast, Gated Access introduces a structured resolution mechanism that ensures that packets are successfully delivered before new transmissions begin, which reduces the variability in update intervals, leading to a lower AoI variance.

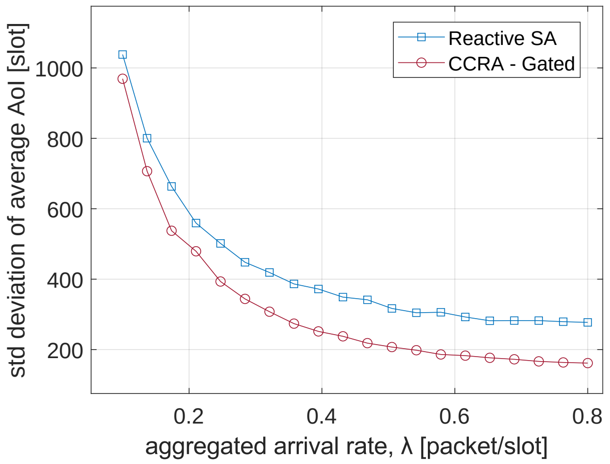
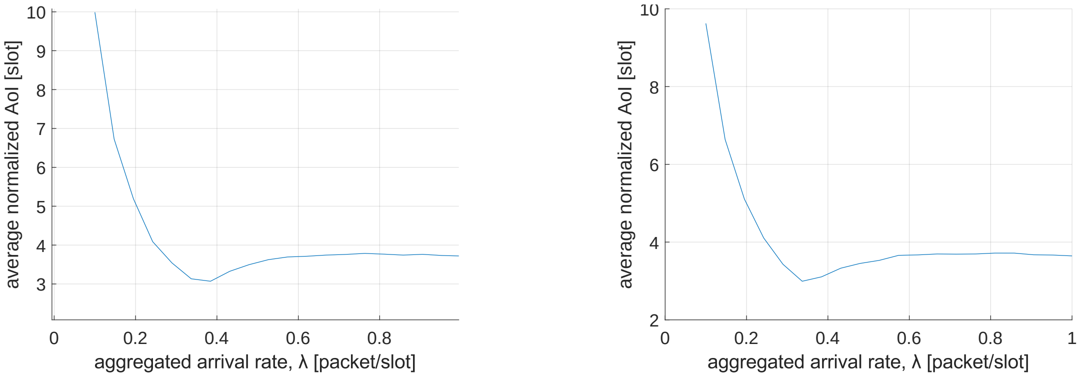


Figure 4.2: Standard deviation of average AoI for Gated Access using CCRA and Reactive SA as a function of λ.

As observed in Chapter 2, the number of users in a system did not significantly impact throughput, and a similar trend is evident in AoI performance. Figure 4.3 and Figure 4.4

demonstrate that systems with 100 and 500 users exhibit nearly identical AoI behavior across all values of λ. This occurs because AoI primarily depends on the aggregated arrival rate rather than the absolute number of users. Since each user generates packets with probability λ/n, the overall system-wide arrival rate remains consistent regardless of the total number of users. Consequently, both small and large systems experience similar levels of contention and resolution dynamics, leading to overlapping AoI curves.



(a) AoI vs. λ for 100 users.

(b) AoI vs. λ for 500 users.

Figure 4.3: Comparison of AoI for Gated Access using CCRA with (a) 100 and (b) 500 users separately.

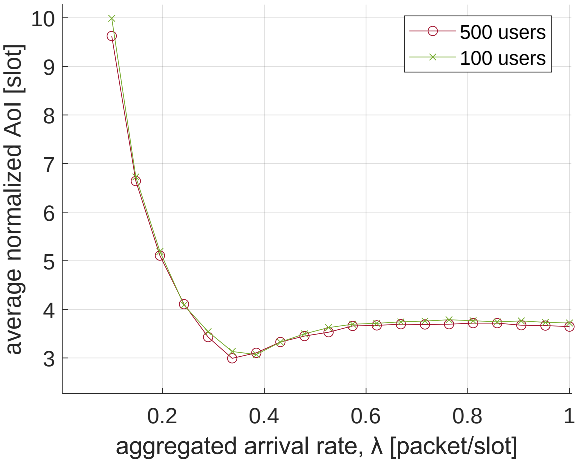


Figure 4.4: Comparison of average AoI for systems with 100 and 500 users in Gated Access using CCRA.

To gain further insight of how AoI differs from delay, Figure 4.5 presents the PMF of delay values for different arrival rates in the Gated Access scheme. The delay is defined as the time between the generation of a packet and its successful reception at the receiver. By analyzing its probability distribution, we can understand the variability in transmission times across different arrival rates.

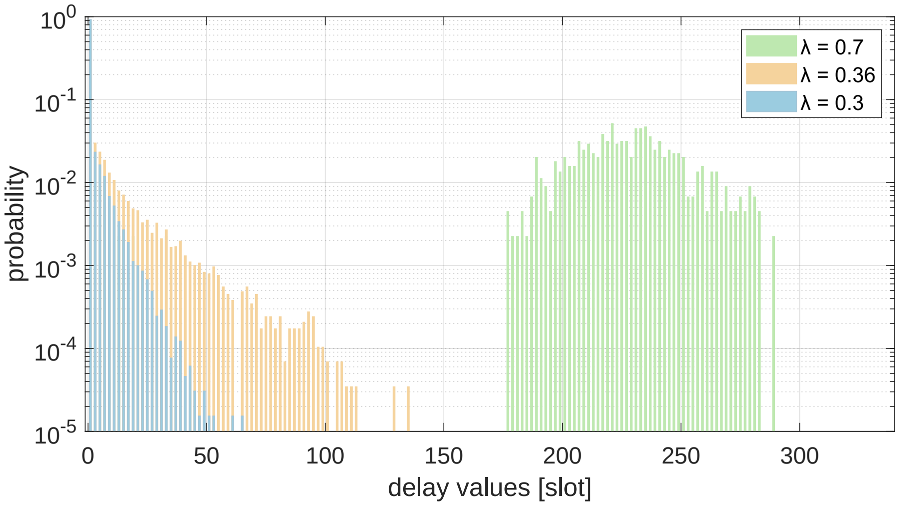


Figure 4.5: Probability mass function of delay values for different arrival rates λ in Gated Access.

Figure 4.5 illustrates how delay distributions change with varying network loads. At low λ ($\lambda = 0.3$), the system experiences shorter delays. In this case, packets are generated less frequently, and CRIs are short. This results in a concentrated probability distribution at low delay values, indicating that most packets are successfully transmitted with minimal waiting time. As λ increases ($\lambda = 0.36$), the PMF shifts toward higher delay values, as expected. More packets are competing for access to the channel, which leads to longer CRIs and delays in packet transmission. At high arrival rates ($\lambda = 0.7$), the delay distribution exhibits a more pronounced shift toward larger delay values. This behavior correlates with longer CRIs as more packets compete for access, which increases the waiting time before packets can be successfully transmitted. Consequently, the probability of longer delays becomes more prominent, resulting in a much broader distribution.

So, which value of λ yields the best AoI? We can observe from Figure 4.1 that the lowest AoI occurs around $\lambda = 0.36$, where the system experiences a balance between update frequency and congestion. This suggests that at this arrival rate, updates occur frequently enough to keep the information fresh, while avoiding excessive congestion and unnecessary retransmissions.
However, looking at the PMF of delay, we see that at $\lambda = 0.3$, the average delay is at its lowest. This is because fewer packets are contending for access, leading to fewer CRIs and faster transmissions. This corresponds to the observation that lower delays do not always translate into the best AoI, as the system struggles to maintain a high frequency of successful updates. This discrepancy between the best AoI and the best average delay can be attributed to the trade-off between packet delivery rate and congestion. While lower delays (such as those observed at $\lambda = 0.3$) provide more timely packet deliveries, they result in fewer updates overall, which increases information staleness over time. On the other hand, $\lambda = 0.36$ strikes a better balance, providing both faster updates and lower congestion, leading to improved AoI performance even if the average delay is slightly higher.

Moving on to the analysis of the Gated Access scheme in an infinite-user scenario, as presented in Chapter 2. We do not have results for AoI in the infinite-user case due to the fundamentally different behavior in such systems. As detailed in Chapter 2 and illustrated in Figures 2.20a and 2.20b, the system experiences ever-growing contention as the arrival rate increases, leading to excessively long CRI durations. Unlike in a finite-user system, where AoI reflects the staleness of information from specific users, in an infinite-user scenario, updates originate from an ever-growing pool of transmitters. As a result, AoI does not provide useful insights into data freshness, since any given user may never successfully update the receiver due to indefinitely long resolution times. Instead, the system behavior is best analyzed through delay and throughput metrics, which describe the aggregate performance of contention resolution rather than the timeliness of individual updates.

For this reason, AoI studies in this thesis will be constrained to finite-user settings, where the metric retains its relevance in measuring the freshness of received data. The results obtained for infinite-user simulations, while useful for validating theoretical bounds and understanding system behavior, do not contribute directly to the assessment of AoI in practical real-time monitoring applications.

4.2 Free Access

This section examines how AoI behaves in Free Access schemes using CCRA, where users transmit without waiting for contention resolution. The following simulations were conducted for 100 users, providing insight into the dynamics of AoI in such systems. In a Free Access scheme, users are allowed to transmit independently of others, which contrasts with Gated Access schemes where users must wait for the resolution of ongoing transmissions before sending new packets, as introduced in Chapter 2. This difference in transmission strategies has significant implications for AoI, as the system must manage the trade-off between congestion and update frequency.

Figure 4.6 shows the average AoI for the three schemes: Gated Access using CCRA, Free Access using CCRA, and Reactive SA. Up until around $\lambda \approx 0.45$, when fewer packets are competing for access, Gated Access and Free Access perform similarly in terms of AoI. In these schemes, the CCRA ensures that all generated packets are successfully delivered, even if collisions occur, as they are retransmitted until successfully transmitted. In contrast, Reactive SA shows a slightly higher AoI because if a packet collides with other packets, it is lost and not retransmitted, meaning that the information is not successfully delivered. This results in higher staleness of the data in Reactive SA compared to Gated and Free Access schemes. Beyond $\lambda \approx 0.45$, Reactive SA outperforms the other two schemes, achieving the best AoI. This happens because Reactive SA's transmission strategy minimizes the number of retransmissions, allowing the system to handle increased packet generation rates more efficiently. In contrast, Gated Access experiences a slight increase in AoI as contention becomes more significant and the system starts facing delays in resolving transmissions. However, the CCRA mechanism in Gated Access ensures that all generated packets are eventually delivered during each CRI, maintaining a low AoI despite the higher congestion. Free Access, however, shows a sharp increase in AoI beyond $\lambda \approx 0.45$. This is because, as the traffic load increases, Free Access suffers from excessive collisions and retransmissions. Older packets continue to stay in the system longer due to multiple retransmission attempts, which leads to very high staleness and an exponential increase in AoI. Therefore, the congestion created by continuous retransmissions results in the highest AoI of the three schemes at higher traffic loads.

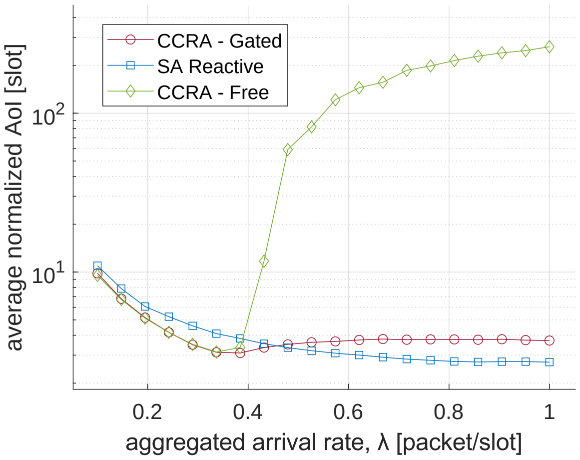


Figure 4.6: Comparison of average AoI for Gated Access, Free Access, and Reactive Slotted ALOHA schemes as a function of λ.

Figure 4.7 illustrates the standard deviation of AoI for all three schemes. Gated Access consistently exhibits the lowest standard deviation, reflecting stable update intervals and transmission times due to its structured resolution process. In contrast, Free Access experiences a sharp increase in AoI variance as λ rises. This is driven by higher collision rates, leading to situations where packets fail to reach the receiver, resulting in highly variable AoI values among users. Since some users packets are successfully transmitted while others are lost, this causes significant variation in the AoI across the system. Reactive Slotted ALOHA maintains a more stable variance compared to Free Access, but still shows more variability than Gated Access due to occasional collisions and delays. While Reactive SA avoids retransmissions, collisions still occur, leading to some packet losses and variability in AoI.

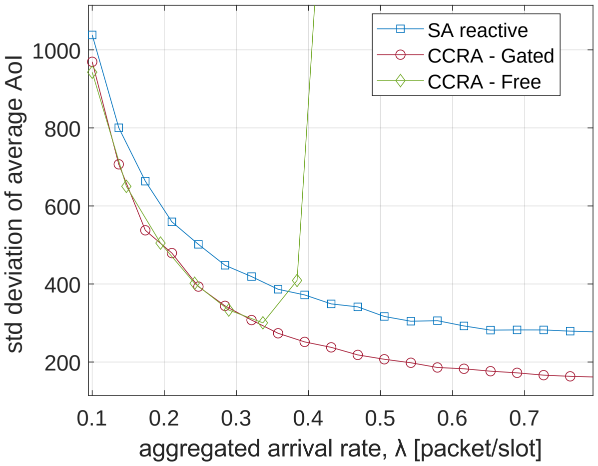
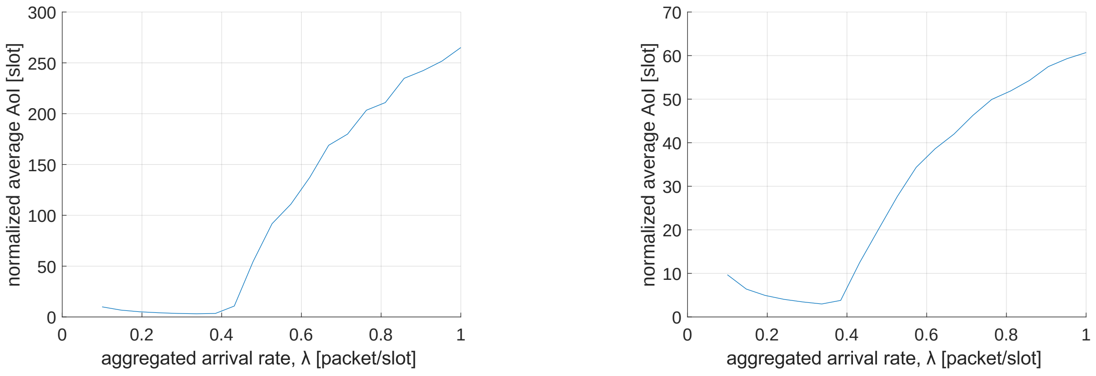


Figure 4.7: Comparison of standard deviation of AoI for Gated Access, Free Access, and Reactive Slotted ALOHA schemes as a function of λ.

These variations can be linked to throughput performance: Free Access achieves a higher peak throughput, as shown in Figure 2.14, because users transmit immediately upon packet arrival, utilizing all available time slots. However, this advantage is offset by frequent collisions, leading to increased variability in AoI. On the other hand, Gated Access, while slightly limiting throughput due to structured transmission and waiting times, achieves a more stable AoI by reducing contention and ensuring that all generated packets are successfully delivered over time.

In summary, Gated Access provides the most stable AoI performance, particularly under high congestion, due to its structured approach that ensures all generated packets are eventually delivered. Free Access, while offering higher throughput, suffers from significant AoI variance, especially at higher traffic loads, due to frequent collisions and retransmissions. Reactive SA strikes a balance, achieving the best AoI beyond $\lambda \approx 0.45$ by minimizing retransmissions, though it shows slightly higher variance compared to Gated Access. The results highlight the trade-offs between throughput, AoI, and AoI variance, emphasizing the need to balance congestion control and efficient packet delivery in random access systems.

If we now want to examine the impact of the number of users on AoI, we can refer to Figures 4.8 and 4.9. The results indicate that the average AoI beyond $\lambda \approx 0.45$, where some packets may never get delivered, is significantly worse for the system with 100 users than the system with 500 users in Free Access. This is in contrast to what we observe in Gated Access, where the number of users does not strongly affect AoI, as long as the system load is the same.



(a) AoI vs. λ for 100 users.

(b) AoI vs. λ for 500 users.

Figure 4.8: Comparison of AoI for Free Access with (a) 100 and (b) 500 users separately.

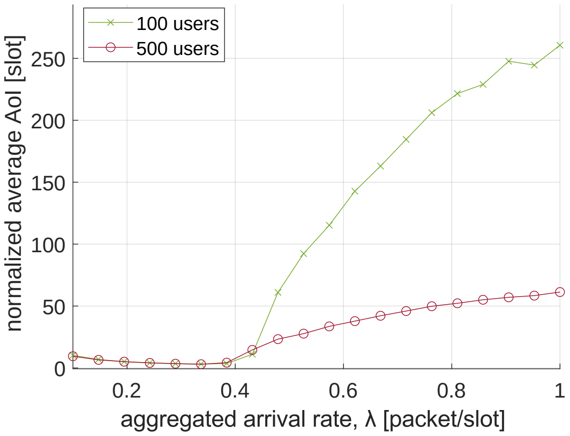


Figure 4.9: Comparison of average AoI for systems with 100 and 500 users in Free Access.

This behavior can be explained by examining the packet generation probability per user and how this affects contention resolution in Free Access.

The probability of packet generation per user is given by:

$$P_{\text{generation}} = \frac{\lambda}{\text{n}} \tag{4.1}$$

where λ is the aggregated arrival rate and n is the total number of users in the system. For example, if $\lambda = 0.8$, the packet generation probability per user would be:

$$P_{\text{generation}}^{(100\text{ users})} = \frac{0.8}{100} = 0.008 \tag{4.2}$$

$$P_{\text{generation}}^{(500\text{ users})} = \frac{0.8}{500} = 0.0016 \tag{4.3}$$

In the case of 100 users, each user attempts to transmit more frequently, leading to a higher probability of collisions between users. This higher collision rate means that unresolved transmissions take longer, which prevents fresh updates from being successfully transmitted and increases the average AoI. In contrast, for 500 users, each user generates packets less frequently, leading to fewer collisions and more efficient contention resolution. This allows fresh updates to be delivered more regularly and reduces the average AoI.

4.3 Pre-emption

In contention-based random access protocols, such as Gated Access and Free Access, users often experience delays due to collisions and retransmissions before successfully delivering their packets. During these delays, updates can become stale, leading to an increased AoI at the receiver. Pre-emption is introduced as a mechanism to prioritize fresher updates by allowing users to replace older, pending packets with newly generated ones before transmission is completed. This technique ensures that when a user finally succeeds in transmitting, the delivered information is as up-to-date as possible.

The core idea behind pre-emption is that an ongoing contention should not prevent new updates from being generated. If a user is already attempting to transmit a packet but generates a newer update while still in contention, pre-emption allows it to discard the outdated packet and transmit the fresher one instead.

This mechanism can significantly improve AoI performance by preventing outdated packets from being delivered, particularly in high-traffic scenarios, where contention resolution might take multiple time slots.
However, pre-emption also introduces trade-offs: by discarding older packets, it increases the risk that some updates are never received. This can be problematic in applications where intermediate updates carry valuable information beyond just the most recent state. However, in this thesis, our focus is solely on the freshness of packets, rather than the specific content they convey.

The following sections explore how pre-emption operates in Gated Access and Free Access, analyzing its impact on contention resolution and AoI performance in both access schemes.

4.3.1 Pre-emption in Gated Access

In conventional Gated Access protocols, users must wait for ongoing CRIs to complete before transmitting newly generated packets. This means that once a user enters contention, it continues attempting to transmit the same packet until it is successfully received. However, in the pre-emption model, this behavior is modified to improve information freshness.
If a user that is already contending in a CRI generates a new update while waiting for its turn to transmit, it discards the previous packet and replaces it with the newly generated one. This ensures that the most recent status information is always prioritized, rather than persisting with potentially outdated data.

The key advantage of this mechanism is that it significantly reduces the risk of delivering stale updates, particularly in high-traffic scenarios where CRIs can last multiple time slots. Without pre-emption, a user might transmit an update that was generated long before the end of the contention process, resulting in an increase in AoI at the receiver. By allowing packet replacement within an ongoing CRI, the system ensures that when a user finally succeeds in transmitting, the delivered information is as fresh as possible.

To illustrate how pre-emption operates in Gated Access, consider the following example:

- **Slot 1: Collision -** users A_1, B_1, C_1, D_1 transmit, resulting in a collision.
- **Slot 2: Collision -** users A_1, B_1 retransmit, causing another collision. During this slot, user B generates a new packet B_2, replacing B_1.
- **Slot 3: Success -** user A_1 transmits successfully and exits contention. A new packet E_1 is generated and will wait for the ongoing CRI to finish.
- **Slot 4: Success -** user B_2 transmits successfully. A new packet F_1 is generated and waits for the next CRI.
- **Slot 5: Collision -** users C_1 and D_1 retransmit, causing another collision. During this slot, user C generates a new packet C_2, replacing C_1.
- **Slot 6: Success -** user D_1 transmits successfully. A new packet G_1 is generated and waits for the next CRI.
- **Slot 7: Success -** user C_2 transmits successfully, ending the CRI.
- **Slot 8: New CRI -** A new CRI starts with users E_1, F_1, G_1 colliding.

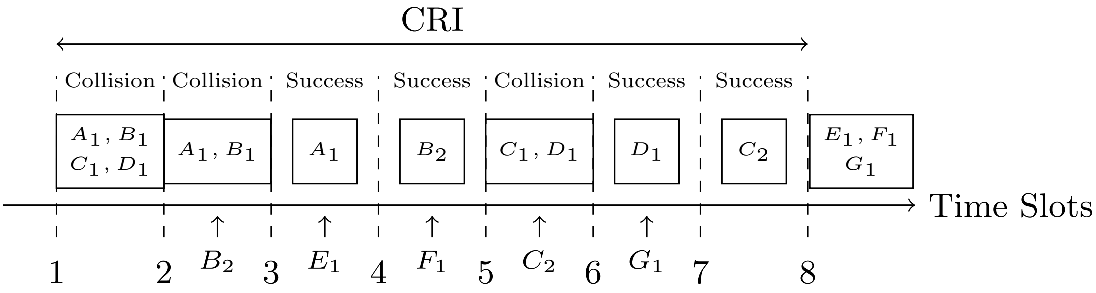


Figure 4.10: Illustration of the pre-emption mechanism in Gated Access using CCRA.

To compare the improvement in AoI between a Gated Access scheme with and without pre-emption, we can refer to the example in Figure 4.10. Focusing only on user B, in the pre-emption scenario, user B generates a new packet B_2 during slot 2, replacing the previous packet B_1. As a result, by slot 4, B_2 is successfully transmitted. The AoI at slot 4 can be calculated using the formula from Equation 3.1:

$$\delta(t_4)^{(\text{user B})} = t_4 - t_2 = 2 \tag{4.4}$$

where t_2 is the generation time of B_2. In contrast, in the scenario without pre-emption, at slot 4, the received packet would be B_1, resulting in a higher AoI calculated as:

$$\delta(t_4)^{(\text{user B})} = t_4 - t_1 = 3 \tag{4.5}$$

where t_1 is the generation time of B_1. Thus, with pre-emption, the AoI is reduced by 1 time slots, resulting in fresher information being delivered to the receiver in the scheme that uses pre-emption, this difference is visible in Figure 4.11.

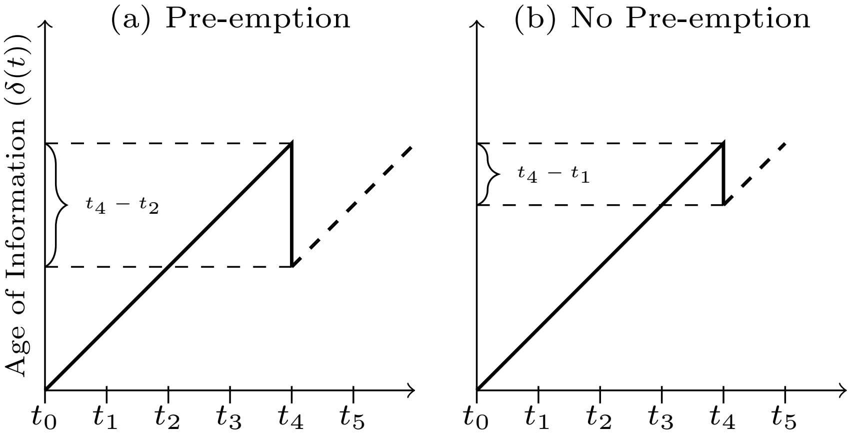


Figure 4.11: Comparison of AoI behavior with and without pre-emption in Gated Access for user B.

To analyze the impact of pre-emption in AoI, Figure 4.12 compares the normalized average AoI of the standard Gated Access scheme with the pre-emption model. At low arrival rates, both schemes exhibit similar AoI performance since packet updates occur infrequently. However, as λ increases, the pre-emption model achieves lower AoI. This improvement is attributed to the fact that users always attempt to transmit their most recent updates, preventing outdated packets from being successfully delivered. The results indicate that pre-emption significantly reduces AoI at high traffic loads, where users frequently generate new updates. Without pre-emption, users persist in retransmitting outdated packets until contention resolution completes, showing that pre-emption ensures that successful transmissions carry fresher information.

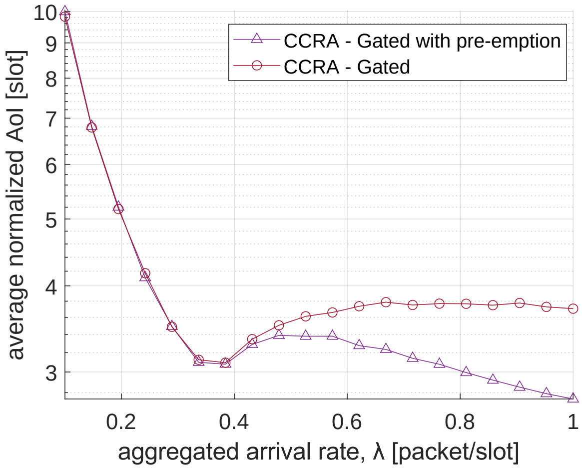


Figure 4.12: Comparison of average AoI for Gated Access with and without pre-emption as a function of λ.

To further contextualize the impact of pre-emption, Figure 4.13 extends the comparison to include Reactive Slotted ALOHA. As observed in previous results, Reactive SA achieves the lowest AoI compared to standard Gated Access at moderate loads due to its immediate transmission policy. However, at higher traffic loads, preemptive Gated Access outperforms standard Gated Access while approaching the performance of Reactive SA.

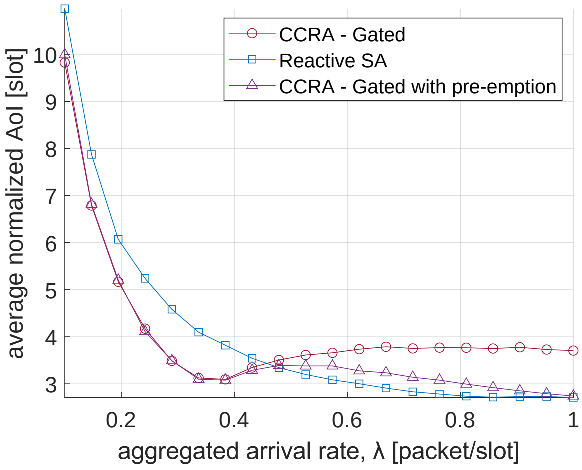


Figure 4.13: Comparison of average AoI for Gated Access, Gated Access with pre-emption, and Reactive Slotted ALOHA as a function of λ.

Figure 4.14 presents the standard deviation of AoI for Gated Access, Gated Access with pre-emption, and Reactive Slotted ALOHA. A lower standard deviation of AoI typically indicates less variability in the intervals between updates. The results from the simulations show that pre-emptive Gated Access exhibits a slightly higher standard deviation than non-pre-emptive Gated Access, especially beyond $\lambda \approx 0.45$. This can be explained by the fact that, under pre-emption, nodes may replace older packets with new ones during the contention resolution process. As the traffic load increases, this replacement mechanism leads to more variability in the time between updates, causing the observed increase in standard deviation. Reactive SA, while achieving the lowest average AoI, shows a higher standard deviation due to its higher packet loss rate, leading to more irregular update intervals.

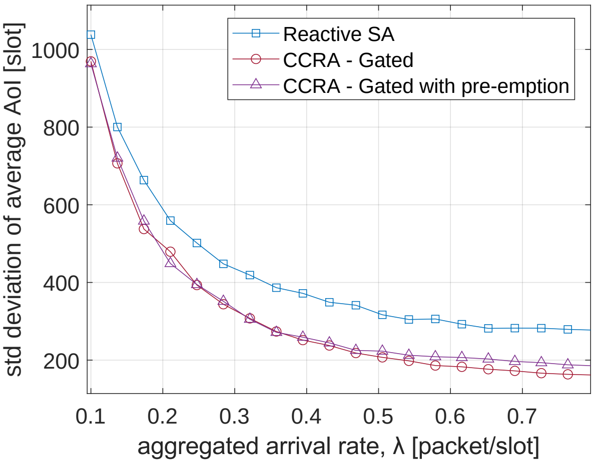


Figure 4.14: Standard deviation of AoI as a function of λ, comparing Gated Access, Gated Access with pre-emption, and Reactive Slotted ALOHA.

The inclusion of pre-emption in Gated Access schemes introduces a significant improvement in data freshness by prioritizing the most recent updates. This enhancement is particularly effective at higher loads, where contention resolution delays can lead to stale packet deliveries. By replacing outdated packets with newly generated ones, pre-emption enables Gated Access to achieve AoI better performance than regular Gated Access schemes, while still benefiting from structured transmission order.

These findings highlight the importance of balancing contention resolution with update prioritization. While structured access mechanisms like Gated Access inherently introduce delays, incorporating pre-emption mitigates these effects.

4.3.2 Pre-emption in Free Access

In the conventional Free Access model, once a user attempts transmission, it continues retransmitting the same packet until it is successfully received or lost due to excessive contention. However, in the pre-emption model, this behavior is modified to improve information freshness.

Under the pre-emption mechanism, if a user that is actively contending in the channel generates a new packet while still trying to transmit a previous one, it discards the older packet and replaces it with the newly generated one. This ensures that the most up-to-date information is always prioritized, rather than persisting with outdated packets that might have been generated several slots earlier. Unlike in Gated Access with pre-emption, where only users who are contending replace their old packets for newly generated ones, Free Access with pre-emption allows any user in the system to update their information.

Figure 4.15 demonstrates the pre-emption mechanism in Free Access, where users replace their currently contending packets with newly generated ones. The sequence of events is described as follows:

- **Slot 1: Collision -** users A, B, C and D try to transmit at the same time, resulting in a collision.

- **Slot 2: Success -** Packet A_1 is successfully transmitted. Meanwhile, user D generates a new packet, replacing D_1 with D_2.
- **Slot 3: Collision -** A collision occurs as packets B_1, C_1, and D_2 attempt transmission simultaneously.
- **Slot 4: Success -** Packet B_1 is successfully retransmitted and is removed from contention. A new packet E_1 arrives and will attempt transmission in the following slot.
- **Slot 5: Collision -** Packets C_1, D_2, and E_1 collide. Since multiple packets are involved, another collision occurs. At the same time, user D generates another new packet D_3, replacing D_2.
- **Slot 6: Success -** Packet E_1 is successfully retransmitted and is removed from contention. A newly generated packet F_1 is introduced into the system.
- **Slot 7: Collision -** Packets C_1, D_3, and F_1 collide, leading to another contention period.
- **Slot 8: Success -** Packet C_1 is transmitted successfully. Meanwhile, a new packet F_2 is generated and replaces F_1.
- **Slot 9: Collision -** Packets D_3 and F_2 collide, extending the contention.
- **Slot 10: Success -** Packet D_3 is successfully transmitted and removed from the contention.
- **Slot 11: Success -** Packet F_2 is successfully transmitted, marking the completion of all pending transmissions.

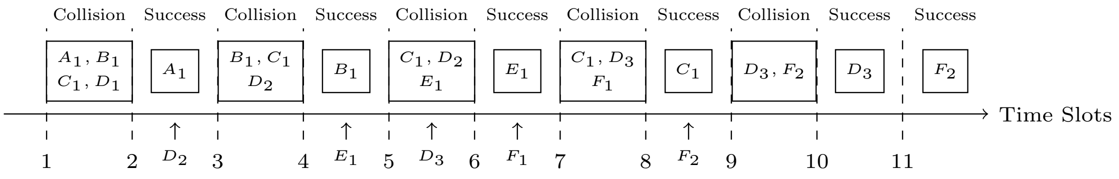


Figure 4.15: Illustration of packet transmissions under Free Access with pre-emption using CCRA.

This example shows how pre-emption allows users to replace outdated packets with newer ones while still contending for transmission. If we want to compare the behaviour of a Free Access scheme with and without pre-emption, we can look at node D from Figure 4.15.

Focusing only on user D, in the pre-emption scenario, user D generates a packet D_1 at slot 1. Then, at slot 2, user D generates a new packet D_2, replacing the previous packet. Finally, at slot 5, user D generates another packet D_3, replacing the previous packet again. As a result, by slot 10, D_3 is successfully transmitted. The AoI at slot 10 can be calculated using the formula from Equation 3.1:

$$\delta(t_{10})^{(\text{user D})} = t_{10} - t_5 = 5 \tag{4.6}$$

where t_5 is the generation time of D_3. In contrast, in the scenario without pre-emption, user D would have had D_1 as the transmitted packet at slot 10, resulting in a higher AoI calculated as:

$$\delta(t_{10})^{(\text{user D})} = t_{10} - t_1 = 9 \tag{4.7}$$

where t_1 is the generation time of D_1. Thus, with pre-emption, the AoI is reduced by 4 time slots, resulting in fresher information being delivered to the receiver in the scheme that uses pre-emption. This difference is visible in Figure 4.16.

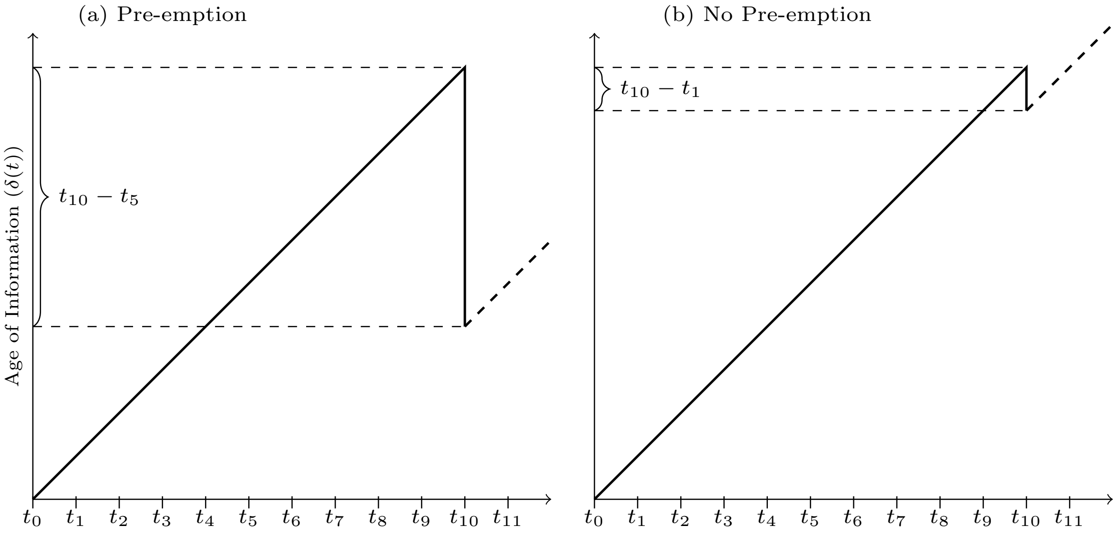


Figure 4.16: Comparison of AoI behavior with and without pre-emption in Free Access for user D.

Figure 4.17 compares the AoI performance in Free Access with and without pre-emption. At low traffic loads, both schemes perform similarly because collisions are infrequent, and most packets are successfully transmitted in their first attempt (same as shown in Figure 4.12 for Gated Access with and without pre-emption at low loads). Since new updates rarely arrive before an older one is transmitted, pre-emption has little effect.

However, as the arrival rate increases, pre-emption helps reduce delays by ensuring that only the freshest updates are transmitted. At very high arrival rates, the benefits of pre-emption diminish as frequent packet replacements increase the likelihood of updates being discarded before successful transmission. However, despite this effect, pre-emption still maintains a lower AoI than the standard Free Access scheme, indicating that its ability to prioritize fresher updates continues to provide an advantage, even under heavy traffic conditions

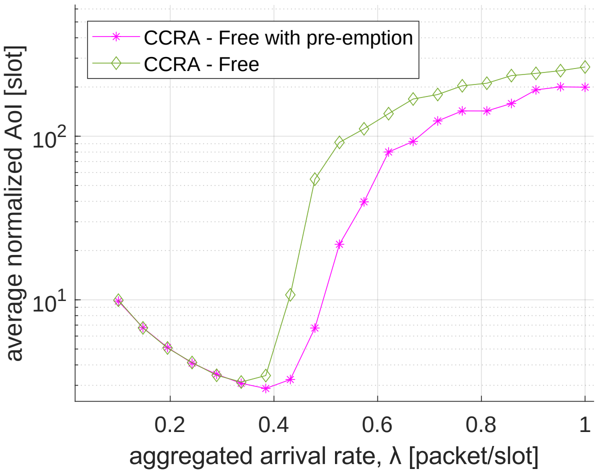


Figure 4.17: Comparison of average AoI for Free Access with and without pre-emption.

Figure 4.18 compares the standard deviation of AoI between Free Access and Free Access with pre-emption. At low traffic loads, both schemes behave similarly. However, as the load increases, Free Access with pre-emption exhibits lower variability in AoI. This is because pre-emption helps resolve the issue of users never transmitting, reducing the wide variation in AoI values.

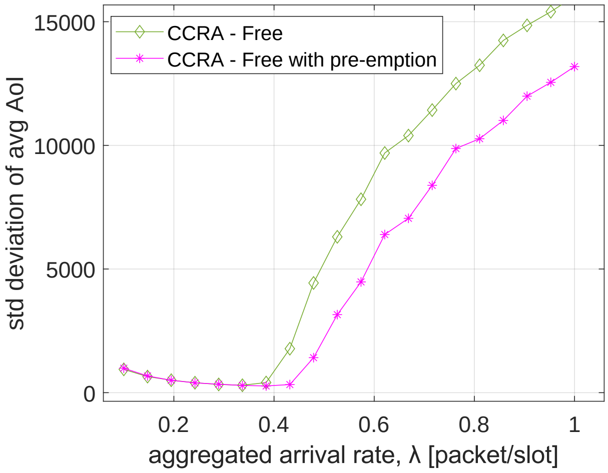


Figure 4.18: Standard deviation of AoI for Free Access and Free Access with pre-emption.

Figure 4.19 illustrates how the number of users affects AoI in Free Access with pre-emption. Similar to the behavior observed in Free Access without pre-emption, the 500-user system achieves better AoI than the 100-user system beyond moderate arrival rates. This occurs because, with fewer users, each user transmits more frequently, leading to persistent retransmissions and increased contention, ultimately delaying fresh updates. In contrast, with 500 users, individual users transmit less often, and the contention resolution process is more effective, allowing fresher updates to be introduced more regularly. At higher loads, pre-emption further amplifies this effect by frequently replacing outdated packets, which benefits a system with many users by maintaining lower AoI. However, at lower loads, pre-emption has minimal impact since new updates are less frequent, resulting in fewer packet overwrites and similar performance between both user scenarios.

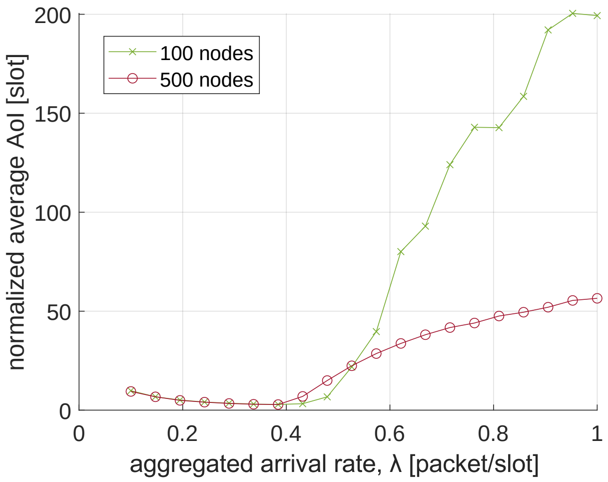


Figure 4.19: Comparison of average AoI for systems with 100 and 500 users in Free Access with pre-emption.

While AoI provides a measure of information freshness, understanding the transmission process requires analyzing additional key performance metrics. In particular, the concepts of delay and contention time are essential to understanding how Free Access with pre-emption affects update transmission dynamics.

Delay is defined as the time elapsed between the generation of a packet and its successful reception at the receiver. It directly reflects how long an individual update takes to reach its destination. However, this does not account for the time a user spends attempting to transmit while facing repeated collisions. To capture this aspect, we introduce the contention time metric. Contention time refers to the duration a user spends in the contention resolution process before either successfully transmitting a packet or replacing it due to pre-emption. Unlike delay, contention time accounts for the period during which a user is actively contending for access but may not necessarily succeed in transmitting the same packet it initially attempted to send.

In Free Access with pre-emption, this distinction is particularly important because users can generate new updates while still contending. When this happens, the older packet is discarded and replaced with the fresher update. As a result, even if a user remains in contention for a long time, the delay of the successfully transmitted packet may be short, since it may have been generated just before the user finally gained access. Oppositely, the overall contention time increases as network load grows, as more users attempt transmission simultaneously, leading to extended collision resolution periods, which can be seen in Figure 4.20.

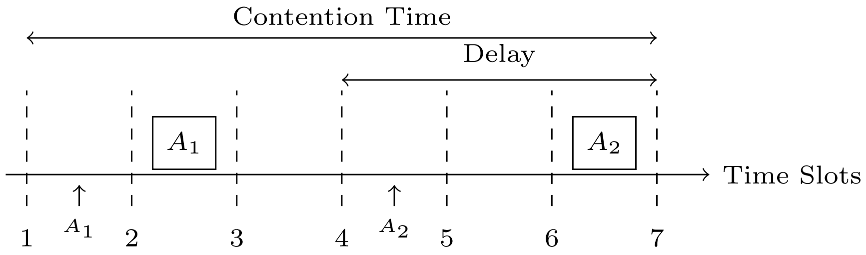
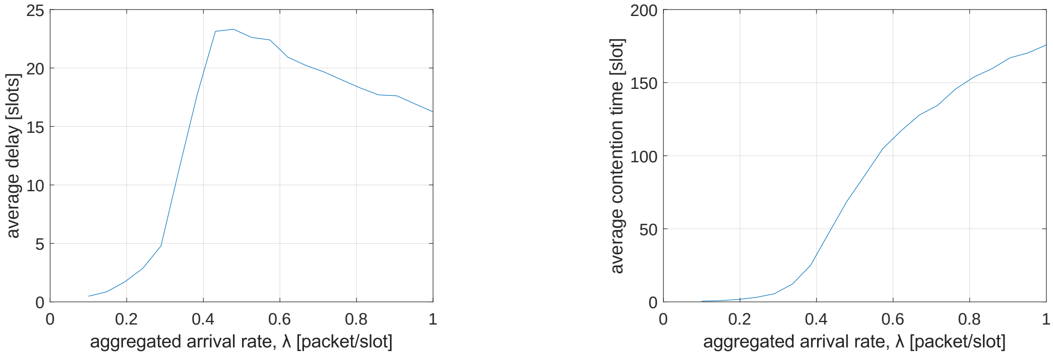


Figure 4.20: Illustration of contention time and delay in Free Access with pre-emption.

To further investigate this behavior, Figure 4.21 presents the delay and the corresponding contention time in Free Access with pre-emption for varying arrival rates. The delay curve in Figure 4.21a initially increases with λ, as higher arrival rates result in increased contention. However, beyond a certain point, pre-emption causes the delay to decrease, as outdated packets are frequently replaced with fresher updates before transmission. In contrast, the contention time curve in Figure 4.21b continues to rise with λ, illustrating that higher arrival rates lead to prolonged contention resolution periods. This confirms that while pre-emption helps prioritize fresh updates, it does not eliminate the time spent in contention.



(a) Average delay as a function of λ.

(b) Average contention time as a function of λ.

Figure 4.21: Average delay and contention time in Free Access with pre-emption.

Figure 4.22 compares the AoI performance of Gated Access with pre-emption, Free Access with pre-emption, and Reactive Slotted ALOHA. Comparing Gated Access with pre-emption and Free Access with pre-emption, the results show that they both behave similarly at low arrival rates, but Gated Access achieves lower AoI than Free Access at moderate to higher arrival rates. This improvement can be attributed to how Gated Access structures its contention resolution process:

In Gated Access, new updates must wait until all ongoing transmissions are resolved before entering the system. This controlled approach reduces unnecessary packet replacement, preventing fresh updates from being discarded before a successful transmission.

In Free Access, new updates can interfere with ongoing contention resolution, frequently replacing older packets before they are successfully transmitted. This continuous pre-emption can be beneficial at moderate loads but becomes counterproductive under high traffic, as updates are overwritten too often, leading to increased packet loss and rising AoI.

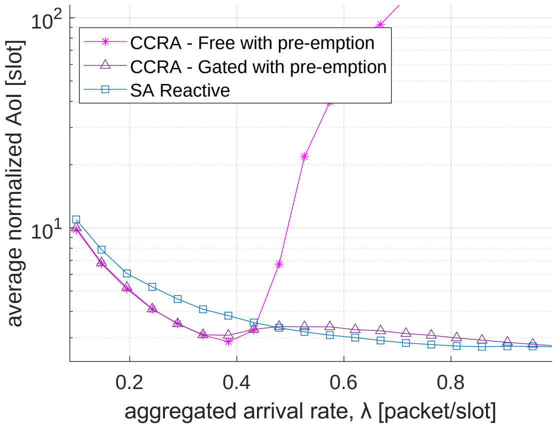


Figure 4.22: Comparison of AoI between Gated Access with pre-emption, Free Access with pre-emption, and Reactive Slotted ALOHA.

Still, at low and moderate loads, both Gated and Free Access with pre-emption show similar performance, as contention is minimal, and updates are successfully transmitted without excessive replacement. While, at high loads, the structured nature of Gated Access with pre-emption allows better control over transmissions, ensuring that updates are eventually delivered rather than repeatedly replaced, leading to a lower overall AoI compared to Free Access with pre-emption.

Figure 4.23 compares the standard deviation of AoI for Gated Access, Free Access with pre-emption, and Reactive Slotted ALOHA. At lower traffic loads, Reactive Slotted ALOHA exhibits the lowest standard deviation, reflecting the stable update intervals that come with its immediate transmission mechanism. In contrast, Free Access with pre-emption shows higher variance in AoI, especially as the arrival rate increases, due to the frequent replacement of older packets with newer ones during contention. Gated Access with pre-emption, on the other hand, maintains a lower standard deviation compared to Free Access with pre-emption, thanks to its structured transmission process that reduces fluctuations in update intervals. At high traffic loads, the difference between Gated Access and Free Access becomes more pronounced, with Gated Access demonstrating more consistent and stable AoI performance.

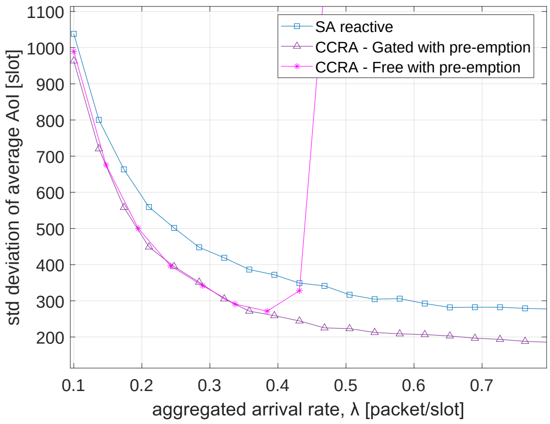


Figure 4.23: Standard deviation of average AoI for Gated Access, Free Access with pre-emption, and Reactive Slotted ALOHA.

Figure 4.24 presents a comprehensive comparison of AoI and Figure 4.25 presents the standard deviation of AoI across all studied access schemes: Free Access, Free Access with pre-emption, Gated Access, Gated Access with pre-emption, and Reactive Slotted ALOHA.

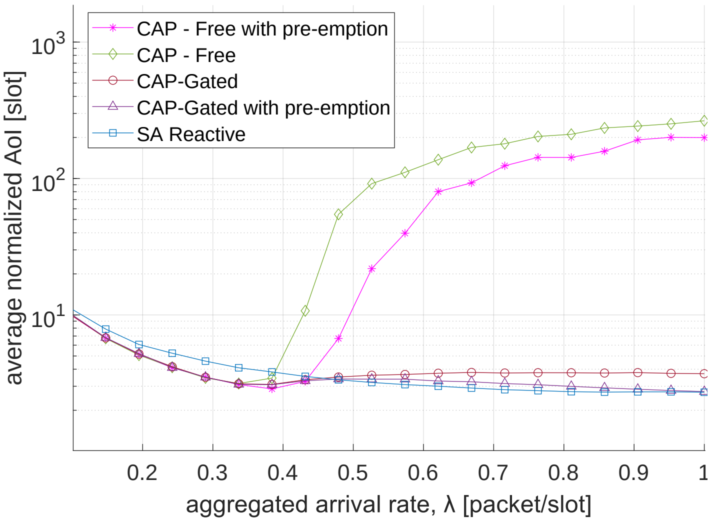


Figure 4.24: Comparison of AoI between Free Access, Free Access with pre-emption, Gated Access, Gated Access with pre-emption, and Reactive Slotted ALOHA.

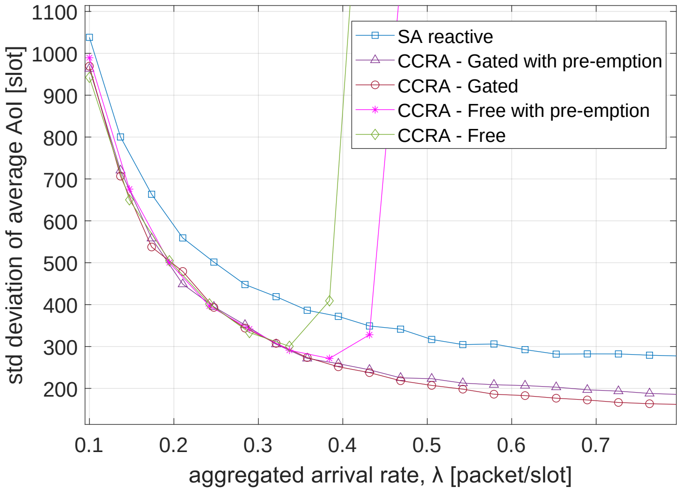


Figure 4.25: Standard deviation of average AoI for Free Access, Free Access with pre-emption, Gated Access, Gated Access with pre-emption, and Reactive Slotted ALOHA.

The results from Figure 4.24 and Figure 4.25 confirm that Gated Access achieves the lowest AoI, benefiting from immediate transmission without retransmissions at higher arrival rates. Among contention-based schemes, Gated Access outperforms Free Access due to its structured resolution process, which prevents excessive packet overwrites. Pre-emption improves AoI at moderate loads by ensuring fresher updates are transmitted, but its effectiveness depends on access control. While Gated Access with pre-emption maintains stable performance even at high loads, Free Access with pre-emption suffers from severe AoI degradation due to uncontrolled packet replacement. The findings highlight that structured access mechanisms and controlled pre-emption lead to better AoI performance.

5 Conclusions

In the previous chapter, the impact of Random Access protocols on AoI has been explored in communication systems, focusing on Gated Access and Free Access schemes. The performance of these schemes has been evaluated under varying packet arrival rates, considering how this factor influences the timeliness and freshness of transmitted data.

The analysis of Gated Access revealed that it effectively reduces AoI if compared to traditional Slotted schemes from low to moderate arrival rates by structuring contention resolution. However, as network load increases, the duration of CRIs grows, leading to higher AoI due to delays in packet delivery. This increase in delay is a consequence of higher contention as more packets compete for access to the shared channel.

In contrast, Free Access was shown to result in significantly worse AoI compared to Gated Access, particularly at higher arrival rates. Due to the absence of a structured transmission order, collisions and retransmissions in Free Access lead to longer delays and higher AoI. At low arrival rates, Free Access may perform similarly to Gated Access, but as the traffic load increases, Free Access becomes much less efficient. The lack of control over transmission order causes frequent retransmissions and increases competition for the channel, which significantly raises the AoI. This highlights the limitations of Free Access in handling higher network loads and maintaining timely updates compared to Gated Access.

The introduction of pre-emption in both access schemes was found to have a significant impact on AoI performance. By allowing nodes to replace outdated packets with newer ones during contention, pre-emption helps prioritize fresh updates, particularly in high-traffic scenarios where retransmissions can delay successful transmissions. This results in a lower AoI, as nodes avoid transmitting outdated information. The pre-emption model reduces delays by ensuring that when transmission occurs, the most recent packet is transmitted. However, pre-emption introduces trade-offs, as it leads to the potential loss of intermediate updates if they are repeatedly overwritten by newer packets. This can be problematic in scenarios where intermediate information is also valuable, but for the purposes of this study, the focus was on data freshness rather than content.

The comparison between Gated Access with and without pre-emption showed that pre-emption improves AoI performance, especially under high traffic loads. By replacing outdated packets during ongoing contention, Gated Access with pre-emption ensures that successful transmissions deliver fresher information. This is in contrast to the standard Gated Access scheme, where nodes may continue retransmitting outdated packets, leading to higher AoI.

Free Access with pre-emption showed improved AoI performance compared to standard Free Access. Pre-emption helped reduce AoI at moderate loads by ensuring that fresher updates were prioritized for transmission. The performance difference between Free Access with and without pre-emption was most noticeable at moderate traffic loads, where pre-emption significantly reduced delays and kept AoI lower. However, at higher loads, the benefits of pre-emption were less pronounced. While pre-emption still provided an advantage over standard

Free Access by prioritizing fresher updates, the increase in traffic led to more frequent collisions and packet replacements, which caused AoI to rise. Despite this, pre-emption continued to outperform standard Free Access, where outdated packets were more likely to be transmitted under heavy traffic conditions. However, Gated Access schemes consistently performed better in terms of AoI, particularly at high loads, due to their more structured and controlled transmission process.

These findings underline the importance of balancing contention resolution with the need for timely updates. Gated Access offers a more controlled update delivery system. Pre-emption improves both Gated and Free Access systems by reducing the risk of outdated updates being transmitted, but it also raises the possibility of intermediate updates being lost. In practical applications, the choice of access mechanism should depend on the importance of data freshness versus the need to ensure that every update is transmitted successfully.

Overall, this thesis has provided a comprehensive analysis of how Random Access protocols, including Gated and Free Access schemes with pre-emption, affect the AoI performance in communication systems.

5.1 Future Work

This thesis has analyzed how random access protocols impact the Age of Information in communication networks. However, there are several ways in which the work could be expanded or improved in the future:

- Study systems with users that have different update rates or priorities, as this is common in practical IoT and sensor networks.
- Consider other packet arrival models besides Poisson, such as bursty traffic or periodic updates, to better reflect real-world behavior.
- Design and evaluate new access schemes that combine features of both Gated and Free Access, allowing the system to switch based on traffic conditions.
- Analyze the impact of energy constraints, especially in battery-powered devices, and how update frequency affects power consumption..
- Investigate whether learning-based approaches, such as reinforcement learning, could help optimize update scheduling and access control over time.

References

[1] Berioli, M., Cocco, G., Liva, G., Munari, A. (2016). Modern random access protocols. Foundations and Trends in Networking, 10 (4), 317–446.

[2] Liva, G., Polyanskiy, Y. (2024). Unsourced multiple access: A coding paradigm for massive random access. Proceedings of the IEEE, 112(9), 1214-1229.

[3] Abramson, N. (1970). The ALOHA system—Another alternative for computer communications. Proceedings of the AFIPS Fall Joint Computer Conference, 37, 281–285.

[4] Roberts, L. G. (1975). ALOHA packet system with and without slots and capture. SIGCOMM Computer Communication Review, 5 (2), 28–42.

[5] Kleinrock, L., Tobagi, F. (1975). Packet switching in radio channels: Part I - Carrier sense multiple-access modes and their throughput-delay characteristics. IEEE Transactions on Communications, 23(12), 1400-1416.

[6] IEEE Standards for Local Area Networks: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. (1985). ANSI/IEEE Std 802.3-1985.

[7] IEEE. (2021). IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks—Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (IEEE Std 802.11-2020).

[8] Casini, E., De Gaudenzi, R., Del Rio Herrero, O. (2007). Contention resolution diversity slotted ALOHA (CRDSA): An enhanced random access scheme for satellite access packet networks. IEEE Transactions on Wireless Communications, 6 (4), 1408-1419.

[9] Abramson, N. (1970). The ALOHA system—Another alternative for computer communications. Proceedings of the AFIPS Fall Joint Computer Conference, 281-285.

[10] Massey, J., Mathys, P. (1985). The collision channel without feedback. IEEE Transactions on Information Theory, 31 (2), 192-204.

[11] Massey, J. L. (1981). Collision-resolution algorithms and random-access communications. In G. Longo (Ed.), Multi-user communication systems (Vol. 265, pp. 221-237). Springer.

[12] Capetanakis, J. I. (1979). Tree algorithms for packet broadcast channels. IEEE Transactions on Information Theory, 25 (5), 505-515.

[13] Gallager, R. G. (1978). Private Communication.

[14] Mathys, P., Flajolet, P. (1985). Q-ary collision resolution algorithms in random-access systems with free or blocked channel access. IEEE Transactions on Information Theory, 31 (2), 217-243.

[15] Yates, R. D., Sun, Y., Brown, D. R. III, Kaul, S. K., Modiano, E., Ulukus, S. (2021). Age of information: An introduction and survey. IEEE Journal on Selected Areas in Communications, 39 (5), 1183–1198.

[16] Munari, A. (2021). Modern random access: An age of information perspective on irregular repetition slotted ALOHA. IEEE Transactions on Communications, 69 (6), 3572-3585.