Information Freshness: How To Achieve It and Its Impact On Low-Latency Autonomous Systems

Biplav Choudhury

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering

Jeffrey H. Reed, Chair
Vijay K. Shah, Co-chair
Wenjing Lou
Luiz A. DaSilva
Mike Mollenhauer

April 29, 2022
Blacksburg, Virginia

Keywords: wireless communications, age of information, deep reinforcement learning, IoT networks, vehicular networks, smart grid

Information Freshness: How To Achieve It and Its Impact On Low-Latency Autonomous Systems

Biplav Choudhury

(ABSTRACT)

In the context of wireless communications, low latency autonomous systems continue to grow in importance. Some applications of autonomous systems where low latency communication is essential are (i) vehicular network's safety performance depends on how recently the vehicles are updated on their neighboring vehicle's locations, (ii) updates from IoT devices need to be aggregated appropriately at the monitoring station before the information gets stale to extract temporal and spatial information from it, and (iii) sensors and controllers in a smart grid need to track the most recent state of the system to tune system parameters dynamically, etc. Each of the above-mentioned applications differs based on the connectivity between the source and the destination. First, vehicular networks involve a broadcast network where each of the vehicles broadcasts its packets to all the other vehicles. Secondly, in the case of UAV-assisted IoT networks, packets generated at multiple IoT devices are transmitted to a final destination via relays. Finally for the smart grid and generally for distributed systems, each source can have varying and unique destinations. Therefore in terms of connectivity, they can be categorized into one-to-all, all-to-one, and variable relationship between the number of sources and destinations. Additionally, some of the other major differences between the applications are the impact of mobility, the importance of a reduced AoI, centralized vs distributed manner of measuring AoI, etc. Thus the wide variety of application requirements makes it challenging to develop scheduling schemes that universally address minimizing the AoI.

All these applications involve generating time-stamped status updates at a source which are then transmitted to their destination over a wireless medium. The timely reception of these updates at the destination decides the operating state of the system. This is because the *fresher* the information at the destination, the better its awareness of the system state for making better control decisions. This freshness of information is not the same as maximizing the throughput or minimizing the delay. While ideally throughput can be maximized

by sending data as fast as possible, this may saturate the receiver resulting in queuing, contention, and other delays. On the other hand, these delays can be minimized by sending updates slowly, but this may cause high inter-arrival times. Therefore, a new metric called the Age of Information (AoI) has been proposed to measure the freshness of information that can account for many facets that influence data availability. In simple terms, AoI is measured at the destination as the time elapsed since the generation time of the most recently received update. Therefore AoI is able to incorporate both the delay and the inter-packet arrival time. This makes it a much better metric to measure end-to-end latency, and hence characterize the performance of such time-sensitive systems. These basic characteristics of AoI are explained in detail in Chapter 1. Overall, the main contribution of this dissertation is developing scheduling and resource allocation schemes targeted at improving the AoI of various autonomous systems having different types of connectivity, namely vehicular networks, UAV-assisted IoT networks, and smart grids, and then characterizing and quantifying the benefits of a reduced AoI from the application perspective.

In the first contribution, we look into minimizing AoI for the case of broadcast networks having one-to-all connectivity between the source and destination devices by considering the case of vehicular networks. While vehicular networks have been studied in terms of AoI minimization, the impact of mobility and the benefit of a reduced AoI from the application perspective has not been investigated. The mobility of the vehicles is realistically modeled using the Simulation of Urban Mobility (SUMO) software to account for overtaking, lane changes, etc. We propose a safety metric that indicates the collision risk of a vehicle and do a simulation-based study on the ns3 simulator to study its relation to AoI. We see that the broadcast rate in a Dedicated Short Range Network (DSRC) that minimizes the system AoI also has the least collision risk, therefore signifying that reducing AoI improves the on-road safety of the vehicles. However, we also show that this relationship is not universally true and the mobility of the vehicles becomes a crucial aspect. Therefore, we propose a new metric called the Trackability-aware AoI (TAoI) which ensures that vehicles with unpredictable mobility broadcast at a faster rate while vehicles that are predicable are broadcasting at a reduced rate. The results obtained show that minimizing TAoI provides much better on-road safety as compared to plain AoI minimizing, which points to the importance of mobility in such applications.

In the second contribution, we focus on networks with all-to-one connectivity where packets

from multiple sources are transmitted to a single destination by taking an example of IoT networks. Here multiple IoT devices measure a physical phenomenon and transmit these measurements to a central base station (BS). However, under certain scenarios, the BS and IoT devices are unable to communicate directly and this necessitates the use of UAVs as relays. This creates a two-hop scenario that has not been studied for AoI minimization in UAV networks. In the first hop, the packets have to be sampled from the IoT devices to the UAV and then updated from the UAVs to the BS in the second hop. Such networks are called UAV-assisted IoT networks. We show that under ideal conditions with a generate-at-will traffic generation model and lossless wireless channels, the Maximal Age Difference (MAD) scheduler is the optimal AoI minimizing scheduler. When the ideal conditions are not applicable and more practical conditions are considered, a reinforcement learning (RL) based scheduler is desirable that can account for packet generation patterns and channel qualities. Therefore we propose to use a Deep-Q-Network (DQN)-based scheduler and it outperforms MAD and all other schedulers under general conditions. However, the DQN-based scheduler suffers from scalability issues in large networks. Therefore, another type of RL algorithm called Proximal Policy Optimization (PPO) is proposed to be used for larger networks. Additionally, the PPO-based scheduler can account for changes in the network conditions which the DQN-based scheduler was not able to do. This ensures the trained model can be deployed in environments that might be different than the trained environment.

In the final contribution, AoI is studied in networks with varying connectivity between the source and destination devices. A typical example of such a distributed network is the smart grid where multiple devices exchange state information to ensure the grid operates in a stable state. To investigate AoI minimization and its impact on the smart grid, a co-simulation platform is designed where the 5G network is modeled in Python and the smart grid is modeled in PSCAD/MATLAB. In the first part of the study, the suitability of 5G in supporting smart grid operations is investigated. Based on the encouraging results that 5G can support a smart grid, we focus on the schedulers at the 5G RAN to minimize the AoI. It is seen that the AoI-based schedulers provide much better stability compared to traditional 5G schedulers like the proportional fairness and round-robin. However, the MAD scheduler which has been shown to be optimal for a variety of scenarios is no longer optimal as it cannot account for the connectivity among the devices. Additionally, distributed networks with heterogeneous sources will, in addition to the varying connectivity, have different sized

packets requiring a different number of resource blocks (RB) to transmit, packet generation patterns, channel conditions, etc. This motivates an RL based approach. Hence we propose a DQN-based scheduler that can take these factors into account and results show that the DQN-based scheduler outperforms all other schedulers in all considered conditions.

Information Freshness: How To Achieve It and Its Impact On Low-Latency Autonomous Systems

Biplav Choudhury

(GENERAL AUDIENCE ABSTRACT)

Age of information (AoI) is an exciting new metric as it is able to characterize the freshness of information, where freshness means how representative the information is of the current system state. Therefore it is being actively investigated for a variety of autonomous systems that rely on having the most up-to-date information on the current state. Some examples are vehicular networks, UAV networks, and smart grids. Vehicular networks need the real-time location of their neighbor vehicles to make maneuver decisions, UAVs have to collect the most recent information from IoT devices for monitoring purposes, and devices in a smart grid need to ensure that they have the most recent information on the desired system state. From a communication point of view, each of these scenarios presents a different type of connectivity between the source and the destination. First, the vehicular network is a broadcast network where each vehicle broadcasts its packets to every other vehicle. Secondly, in the UAV network, multiple devices transmit their packets to a single destination via a relay. Finally, with the smart grid and the generally distributed networks, every source can have different and unique destinations. In these applications, AoI becomes a natural choice to measure the system performance as the fresher the information at the destination, the better its awareness of the system state which allows it to take better control decisions to reach the desired objective.

Therefore in this dissertation, we use mathematical analysis and simulation-based approaches to investigate different scheduling and resource allocation policies to improve the AoI for the above-mentioned scenarios. We also show that the reduced AoI improves the system performance, i.e., better on-road safety for vehicular networks and better stability for smart grid applications. The results obtained in this dissertation show that when designing communication and networking protocols for time-sensitive applications requiring low latency, they have to be optimized to improve AoI. This is in contrast to most modern-day communication protocols that are targeted at improving the throughput or minimizing the delay.

Dedication

To my family.

Acknowledgments

Writing this dissertation fills me with a variety of emotions. However, if I had to single out one of them, it will be gratitude. I have been incredibly blessed and lucky to have had the opportunity and guidance of working with extremely smart and helpful people. Mere words are not sufficient to express my feelings for them, however, I take this as an opportunity to acknowledge those who made a positive impact during the course of my study. First and foremost, I am very thankful to my adviser Dr. Jeffrey Reed for giving me the support and guidance needed at every step, along with the freedom to work on very exciting projects. His guidance on identifying research topics, approaching a problem, professional advice, etc. has been invaluable, and I appreciate him for always being so approachable.

I owe my utmost respect to my co-adviser Dr. Vijay K. Shah for being such a great mentor and guiding me at every step. We worked together very closely and I was able to see his passion for research, and hopefully, I was able to imbibe some of it. I thoroughly enjoyed our discussions and working with you has been one of the highlights of my graduate study. I will always look up to your knowledge, energy, and passion for research, and will revere your support in every aspect during my Ph.D.

I would like to thank my other committee members, Dr. Wenjing Lou, Dr. Mike Mollenhauer, and Dr. Luiz A. DaSilva for providing valuable feedback during my preliminary examinations, which helped me to improve my work by exploring additional areas. I also thank Hilda and Makensi for always being so approachable and helpful with administrative work. The advisers at Graduate School and the Cranwell International Center have been very supportive and helped me navigate the various procedures involved with my internships, jobs, and courses. During my internship at Qualcomm, Anthony was a great mentor and I am extremely proud of the work we accomplished.

I made some great friends during my stay here at Blacksburg. I am thankful to Gaurang with whom I wrote my first paper and learned the fundamentals of paper writing. Thanks for being so patient with me and always helping me out with the variety of doubts that I came to you with. Ali, sharing the lab with you was a fun and learning experience, and I

admire you for how focused you were on your work. Raghu has been a great friend, more as a mentor, and has guided me on different topics, from research to course planning to job search to life in general, and I greatly value his advice. I shared a lab with Shubhajeet and Tengchan, and I looked forward to meeting you guys every day. Seeing both of you prepare for the final stages of your Ph.D. motivated me a lot. I had a great co-author in Avik and our collaboration formed a major part of my work. Our long zoom discussions were extremely fruitful for me. It was great to work with Aidin and I admire how you approach your work with utmost sincerity. Tarun, I marvel at the practical approach and the knowledge you have on almost any topic. During my final days, I learned a lot from Prasenjit and am grateful to have had the opportunity to work with you. I wish you all the best in your future endeavors and am sure you will accomplish great things.

Finally, I would like to thank my family for their continued support and love over the years. Maa, Deuta, and Maina, you have been my constant source of motivation. Whatever little I have achieved, I owe them to you all. Maa and Deuta, I have been extremely lucky to have been blessed with your work ethic and I believe it to be my biggest strength. I have the best sister in the world in Maina and our entertainment video calls were easily the best part of my day. I look forward to the day when you gift me the thing you promised. Swaswati, you have been a very powerful force and I thank you for being a part of my life.

Lastly, I bow with reverence to thank Lord Krishna for responding to all of my silent prayers and granting countless blessings, knowledge, and opportunities. Thank you for everything.

Contents

List of Figures

List of Tables

List of Abbreviations

3GPP Third generation partnership project

AoI Age of Information

BSM Basic Safety Message

C-V2X Cellular Vehicle-to-Everything

CAM Cooperative Awareness Message

CBR Channel Busy Ratio

CPS Cyber-Physical Systems

CR Collision Risk

CSI Channel State Information

CSMA Carrier Sense Multiple Access

DCC Decentralized Congestion Control

DSRC Dedicated Short Range Communication

ETSI European Telecommunication Standards Institute

EvC Eigen Vector Centrality

FCFS First Come First Served

FTM Fine Time Measurement

IoT Internet of Things

LOS Line of Sight

LTE Long Term Evolution

MDP Markov Decision Process

PDR Packet Delivery Ratio

QoS Quality of Service

RB Resource Block

SAE Society of Automotive Engineers

TAoI Trackability-aware Age of Information

TTC Time To Collision

UAV Unmanned Aerial Vehicles

URLLC Ultra Reliable Low Latency Communications

V2V Vehicle-to-Vehicle

Chapter 1

Introduction

Wireless communications have seen a tremendous rate of growth in the past two decades. The coming of 5G and ongoing studies into 6G promises that this growth will only accelerate in the future. This has led to many innovative applications like real-time streaming, remote surgery, sensor monitoring, etc. that rely on high data rates with extremely high reliability. In these applications, the communication technology acts as the backbone where any shortcomings in the communication network will affect the system performance. Therefore, this era, where the performance of any system relies on the fast and reliable transfer of information, depends on a robust and reliable communication network as an enabler.

Network performance has been traditionally measured using metrics like end-to-end delay, throughput, etc. However, when we consider time-sensitive monitoring and control applications, these metrics fail to capture the full picture. For example, delay can be minimized by sending packets at a very low rate. However, this leads to a high packet inter-arrival time. Similarly, high throughput can be obtained by operating the network into the saturation region. However, this congests the network and the packets encounter high contention and queuing delays at the source. This results in outdated packets being received at the receiver. Therefore the network should ideally operate in a zone where the inter-arrival time is less while ensuring packets are not delayed at the source.

Age of Information (AoI) is a recently proposed metric [63] that provides an indication of how fresh the information is from the receiver's point of view. AoI is able to capture both the inter-arrival time and the delay, hence it is useful for measuring the network performance in monitoring and control applications. To put it in simple words, it measures how much the receiver is aware of the sender's current state. The calculation of AoI is explained in Fig. 1.1, and is inspired from [58]. A single source and destination pair is considered where the source transmits packets to the destination over a wireless channel. At the source, the packet generation instants are t_1, t_3 and those packets are received at the destination at t_2, t_4 respectively. The interval between the transmission and reception instant for a particular

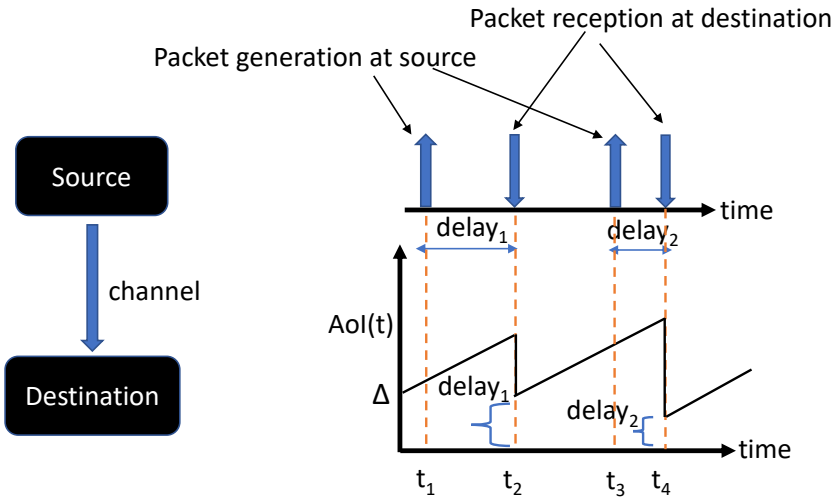


Figure 1.1: Age of Information

packet is that packet's delay. Assuming that the initial AoI is Δ, AoI linearly increases until a packet is received at t_2, at which the AoI reduces to the packet delay. In a similar manner, the AoI increases linearly again until the next packet is received. Finally, the area under the resulting saw-tooth patterned plot is divided by the total observation interval to get the average AoI between the pair of sender and receiver.

Specifically, when we consider time-sensitive applications, a low AoI is desirable and due to this, AoI has been a topic of active research for the past few years [67]. It was first studied in the context of vehicular networks [34, 62]. Other areas of AoI applications include sensor monitoring, [64], data storage [59], caching [124], video monitoring [50] etc. Related AoI metrics like Peak AoI (PAoI), Trackability-aware AoI (TAoI), Age of Incorrect Information (AoII), Value of Information (VoI), etc. have also been proposed that tries to relate the AoI with some other performance metric related to the system. The primary objective in these works is how to effectively utilize the system resources so that the system AoI/PAoI/-TAoI/AoII/VoI is minimized. In order to achieve this, tools from stochastic analysis [80, 81], machine learning [21, 33, 70], game theory [48, 90] etc. have been used.

1.1 Motivation and Contributions

In this dissertation, we investigate different scheduling schemes to help achieve a low AoI across three different autonomous systems that rely on low-latency communications. The applications are primarily categorized in terms of the connectivity between the source and the destination. As shown in Fig. 1.2, each of the three systems have a different nature of connectivity. The first type is the broadcast network where packets from each node is transmitted to every other node, leading to a one-to-many relationship between the source and destination. Vehicular networks are a prime example of this type. The second type of connectivity is many-to-one, where packets from multiple sources arrive at a single destination. We also consider the presence of a relay between the source and the destination, representative of UAV-assisted IoT networks. Here, packets from multiple IoT devices are transmitted to a base station via UAVs that act as relays. Finally, the third type of connectivity is distributed networks where each source is connected to different destination devices. One example is a smart grid where measurements from different parts of the grid are used at various controllers to maintain a stable operating state. In addition to the nature of connectivity, other major differences between the three applications considered in this dissertation are the resources being allocated for scheduling, centralized vs distributed manner of measuring AoI, impact of mobility, the objective in improving AoI, etc. These differences make it challenging to design scheduling schemes that universally address AoI minimization across different types of applications.

1.2 Age of Information: its role and limitations in terms of safer V2V networks

First, in Chapter 2, we look into AoI minimization in broadcast networks. Broadcast networks involve a one-to-all relationship between the source and destinations. A typical example of a broadcast network is a vehicular network. While there are existing works on minimizing AoI in vehicular networks, they have two main shortcomings - (i) they do not account for the vehicle's mobility in the AoI, and (ii) the improvement in system performance from a reduced AoI is not investigated. To address these shortcomings, we do a simulation-based study to investigate the effect of the packet broadcast rate on the system AoI in an

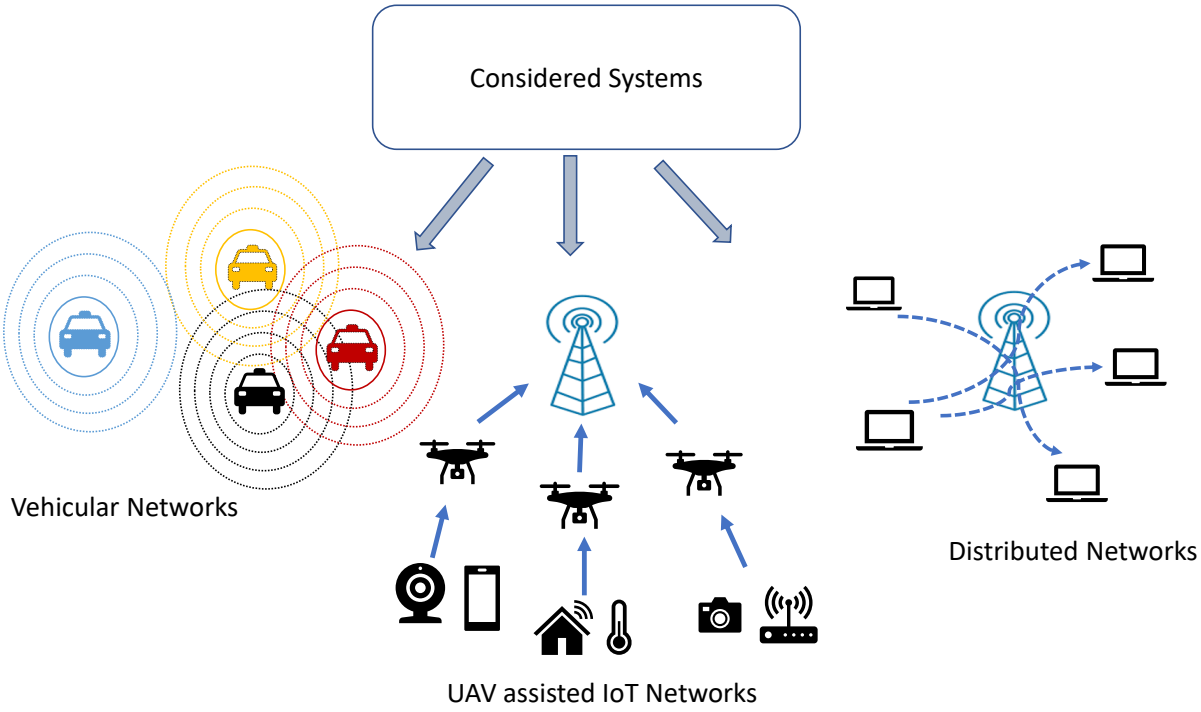


Figure 1.2: Applications considered in this dissertation

802.11p Dedicated Short Range Communication (DSRC) vehicular network using the ns3 simulator. Vehicular networks are examples of broadcast networks where each vehicle tries to communicate its packets to all other vehicles in its range. We propose a new safety metric called the Collision Risk (CR) based on the localization accuracy of the vehicles, that takes into account both the network and mobility characteristics. The results indicate that the broadcast rate that minimizes the system AoI also minimizes the collision risk, thereby maximizing the on-road safety of the network. We also see an interesting relation between the AoI optimal rate and the throughput optimal rate - while the two rates are far apart at lower vehicle densities, they approach each other as the vehicle density increases. The next part of the study builds on the above results while taking more practical aspects of vehicular networks into consideration. We use the Simulation of Urban Mobility (SUMO) toolbox to model the vehicle movements as realistically as possible, where overtaking, lane changes, etc. are considered. We also allow the vehicles to use location estimation algorithms as mandated in standard Society of Automotive Engineers (SAE) J2945/1 to estimate the location of the neighboring vehicles during instants of no packet receptions. This presents an

interesting scenario: vehicles with predictable mobility patterns can be easily tracked with location estimation algorithms. However, for the unpredictable vehicles, they need to constantly broadcast their packets as the location estimation algorithm cannot track them well. Using these insights, we propose a new metric called the Trackability-aware AoI (TAoI) that takes into account the mobility in addition to the AoI. We also propose a new broadcast rate algorithm based on minimizing the TAoI that exceeds the safety performance of the AoI-based broadcast rate algorithm and the standard 10 Hz broadcast rate.

The specific contributions in this chapter are-

- Designed a metric called collision risk that takes into account both the network performance and mobility aspects of the vehicles.
- Demonstrated the relation between the AoI and the on-road safety of the vehicular network based on minimizing the collision risk.
- Proposed a new variant of AoI, called TAoI, that takes both the mobility and the AoI into account. We show that TAoI minimization achieves a better result in the on-road safety compared to plain AoI minimization.

The work presented in this chapter has led to the following publications-

- **Choudhury, B.**, Shah, V. K., Dayal, A., Reed, J. H. (2020, May). Experimental analysis of safety application reliability in V2V networks. In 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring) (pp. 1-5). IEEE.
- **Choudhury, B.**, Shah, V. K., Dayal, A., Reed, J. H. (2021, May). Joint age of information and self risk assessment for safer 802.11 p based V2V networks. In IEEE INFOCOM 2021-IEEE Conference on Computer Communications (pp. 1-10). IEEE.

1.3 Learning-based Age of Information-minimizing scheduling in UAV-relayed IoT Networks

In Chapter 3, we focus on the other type of connectivity seen in wireless networks which is called the all-to-one type. Here multiple source devices transmit their packets to a single

destination. An example is IoT networks where multiple IoT devices measure a particular physical phenomenon and transmit these measurements to a base station (BS). The existing studies on AoI minimization in IoT networks usually consider a single-hop network where the packets are directly transmitted from the IoT devices to the BS. However, there are scenarios in which a base station is not available (remote or disaster-struck regions) due to which UAVs have been deployed to act as relay units. These UAVs collect the packets generated by IoT devices and forward them to the nearest BS. Such networks are called UAV-assisted IoT networks. This presents a unique two-hop scenario where the first hop is between the device and a UAV, and the second hop is between the UAV and the BS. This is different from traditional one-hop scheduling which has been the subject of focus in most of the recent works on AoI in UAV networks. In our scenario, each time step involves two actions: first, sampling of the packets from the IoT devices by the UAVs, and secondly, updating the sampled packets from the UAVs to the BS. We prove, via mathematical analysis and simulations, that under ideal conditions of generate-at-will packet generation at the IoT devices and no packet loss at the channels, a Maximal Age First - Maximal Age Difference (MAF-MAD) scheduler has the best AoI minimization performance. However, under general conditions, a Deep Q Network (DQN) based scheduler is able to outperform the MAD scheduler as it is able to learn the network characteristics like the periodicity of packet generation at the IoT devices and packet loss of the channels. However, DQN has an inherent drawback of scalability where it is not able to converge for large action spaces. Therefore to account for this, we propose another variation of deep reinforcement learning called Proximal Policy Optimization (PPO) that doesn't have this drawback. Additionally, the PPO scheduler is able to adapt to changing network conditions which the DQN scheduler is not able to do.

The specific contributions in this chapter are-

- Considered AoI minimization for UAV-assisted IoT networks with two time-ordered steps, i.e., sampling and updating of packets from UAV devices to a base station via UAVs as relays.

- Proposed MAD scheduler and its optimality in AoI minimization in such two-hop networks is shown via mathematical analysis and simulations.

- Proposed a DQN-based scheduler for general conditions which outperforms MAD and other schedulers.

- Proposed a PPO-based scheduler that scales to larger networks and adapts to changing network conditions better than the DQN-based scheduler.

The work presented in this chapter has led to the following publications-

- **Choudhury, B.**, Shah, V. K., Ferdowsi, A., Reed, J. H., Hou, Y. T. (2021, October). AoI-minimizing Scheduling in UAV-relayed IoT Networks. In 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS) (pp. 117-126). IEEE.

- **B. Choudhury**, P. Karmakar, V.K. Shah, J. H. Reed, "Learning based AoI minimizing scheduling policies for Age of Information minimization in two-hop UAV-assisted IoT networks", submitted to IEEE Transactions on Wireless Communications.

1.4 Design and Analysis of AoI-Optimal 5G Schedulers in Smart Grid

Finally, Chapter 4 deals with AoI minimization for networks with a variable type of connectivity where each source can have a different number of destinations. They are usually seen in distributed networks where information from one device is sent to multiple other devices, and an example is a smart grid. A smart grid consists of multiple devices called distributed energy resources (DERs), and they communicate with one another to ensure the grid operation remains stable. Currently, smart grid connectivity is implemented using optical fiber cables which is expensive and cannot be scaled easily as additional components are added to the grid. Due to this, 5G is being actively investigated as an enabler of the communication among smart grid components. Hence we first design a co-simulation environment where the performance of smart grid applications is analyzed under a 5G network by focusing on the Radio Access Network (RAN) scheduling. The 5G environment is implemented in Python and the smart grid is simulated in PSCAD/MATLAB, and an interface is developed for them to work together. The initial experiments are done to find out if 5G can support time-critical smart grid operations. Based on the encouraging results that 5G is able to support the considered applications reliably, we then propose AoI minimizing schedulers at the RAN (maximal age difference, maximal age first) and compare them to standard 5G schedulers

like round-robin, proportional fairness, etc. The results obtained show that the AoI-based schedulers have a better AoI performance than the standard 5G schedulers due to which the smart grid applications show improved stability, with the maximal age difference (MAD) being the best performing scheduler. Hence as smart grids become one of the important 5G application areas, the results obtained show the need to design better smart grid-centric schedulers that target minimizing AoI. Then for the general case of distributed networks with heterogeneous sources, we show that the MAD scheduler is not the optimal one as it cannot account for factors like a different number of destinations for each source device, varying resource block requirements per device, channel qualities, and traffic generation patterns, etc. Therefore a reinforcement learning-based approach is suitable under such situations and we propose a DQN-based scheduler that outperforms all other considered schedulers under all considered settings.

The specific contributions in this chapter are-

- Designed a co-simulation platform to simulate the effect of communication-specific effects like delays, packet loss, etc on the stability of the smart grid.
- Demonstrated that 5G can reliably support smart grid applications. Additionally, we show that AoI minimizing 5G schedulers provide more stability to the smart grid as compared to traditional 5G schedulers like round-robin and proportional fairness.
- Evaluated the MAD scheduler's AoI minimization in distributed networks and showed that MAD is no longer optimal here due to its inability to take into account the varying connectivity among devices.
- Proposed a DQN-based scheduler to account for the varying connectivity and heterogeneous sources which outperforms MAD and all other schedulers.

The work presented in this chapter has led to the following publications-

- **B. Choudhury***, R. Iyer*, V. K. Shah and A. Mehrizi-Sani, "Power Systems Performance under 5G Radio Access Network in a Co-Simulation Environment," 2021 IEEE 30th International Symposium on Industrial Electronics (ISIE), 2021, pp. 01-06, doi: 10.1109/ISIE45552.2021.9576295.

- **B. Choudhury**, R. Iyer, A. Mehrizi-Sani, V. K. Shah, and J. Reed, "Design and analysis of age-optimal 5G schedulers for distributed control of microgrids", *submitted to IEEE Systems Journal*.
- **B. Choudhury**, P. Karmakar, V. K. Shah, and J. Reed, "Learning based Age of Information Minimization in 5G-enabled Distributed Networks", *to be submitted*.

*Note: * signifies equally contributing authors.*

The following publications are not included in this dissertation:

- Dayal, A., Shah, V. K., **Choudhury, B.**, Marojevic, V., Dietrich, C., Reed, J. H. (2021, June). Adaptive Semi-Persistent Scheduling for Enhanced On-road Safety in Decentralized V2X Networks. In 2021 IFIP Networking Conference (IFIP Networking) (pp. 1-9). IEEE.
- S. Javad, V. K. Shah, Q. Fan, **B. Choudhury**, L. Liu, and J. H. Reed "On the Joint Optimization of Number, Placement and Backhaul Connectivity of Multi-UAV Networks", *submitted to IEEE IoT journal*.
- G. Naik, **B. Choudhury**, J. M. Park, (2019). IEEE 802.11 bd and 5G NR V2X: Evolution of radio access technologies for V2X communications. IEEE access, 7, 70169-70184.

Chapter 2

Age of Information: Its Role and Limitations In Achieving Safer V2V Networks

Vehicle-to-Vehicle (V2V) communications is envisioned as one of the key enablers for Intelligent Transportation Systems (ITS), mainly due to its potential in improving on-road safety in tomorrow's smart cities [4]. The two major standards for V2V communications are the IEEE 802.11 based Dedicated Short Range Communication (DSRC) and LTE-based Cellular-V2X (C-V2X). DSRC uses 802.11p to establish a decentralized network where every vehicle transmits at fixed time intervals, usually 100 ms [88]. C-V2X, on the other hand, was developed in the Release 14 of 3GPP and is a cellular technology with the additional capability of using sidelink interface PC5 for direct V2V communications. Newer versions of both DSRC and C-V2X are being developed, respectively called as 802.11bd and NR-V2X [88].

The three major applications that will be supported by V2V communications are - (i) *Safety Applications* - vehicles periodically broadcast their kinematics parameters (such as vehicle position, speed etc.) in messages called Basic Safety Messages (BSMs) [114], (ii) *Cooperative Message Applications* - these are cooperative traffic efficiency messages controlled by traffic management systems with the objective of improving traffic flow [31], and (iii) *Media Sharing* - vehicles can share infotainment content in a peer-to-peer fashion [51]. In this work, we consider a 802.11p V2V network communicating using DSRC standard where the focus is on safety applications. BSMs carry time-sensitive safety-critical information such as the sender vehicle's location, speed and heading, which upon reception, allows a receiving vehicle to accurately localize the nearby sender vehicles. Thus, BSMs enable several safety applications like cooperative collision warning (CCW) [101], electronic emergency brake light (EEBL) [111] and slow/stopped vehicle alert (SVA) [55] by minimizing the number of collision risky situations. Therefore it is critical that the location information in a BSM is up-to-date and

fresh, as old location information doesn't provide any benefit from a safety point-of-view. To quantify this freshness of information, Age of Information (AoI) has been introduced as a new metric and is defined as the time elapsed since the last BSM was generated [63]. In the context of safety applications and BSMs, lower AoI means fresher location information which makes AoI one of the key indicators to measure the performance of time-sensitive safety applications in V2V networks.

This is because safety beacons facilitate accurate positioning or *localization* of neighboring vehicles. In such V2V networks, it is imperative that the safety beacons are not lost due to packet collisions resulting from congestion, and that the beacons are up-to-date and *fresh*. Lost or outdated safety beacons will negatively impact the reliability of safety applications as it will increase the occurrence of *collision risky* situations; due to the wrong localization of neighboring vehicles or in other words, high *tracking error*. Tracking error is the difference between a vehicle's actual location and perceived location of that vehicle by its neighbors (via safety beacons), and more on this will be explained in detail in the upcoming sections. Safety beacon rate, i.e., the rate at which beacons are broadcast, becomes a critical parameter in order to ensure a safe tracking error. In DSRC standard, it is set to 10 Hz, which may not work well in all V2V network scenarios [88]. Consider the following two instances: (i) high vehicle density scenarios - the V2V network would likely suffer from high channel congestion and undesirable level of packet collisions even when beacon rate is as low as 10 Hz, and (ii) low vehicle density scenarios - the reliability of safety applications can be further enhanced with beacon rates above 10 Hz, as higher beacon rates would improve the freshness of the information and packet collisions do not worsen (because the channels can support higher beacon rate for fewer vehicles).

In recent years there has been a growing body of research on minimizing the AoI in V2V networks (and various other communication networks) [20, 27, 28, 35, 62, 63, 74, 77, 80, 92, 117]. For instance in [63], the authors propose a distributed broadcast rate control algorithm for a DSRC network that minimizes system AoI by iteratively minimizing the locally computed AoI at each vehicle. Refer to Sec. 2.1 for detailed discussion on related works. These prior works propose to minimize AoI and present AoI based rate control algorithms with the consideration that lower AoI will always result in better safety performance in V2V networks.

We perform a comprehensive investigation of the safety applications reliability against vary-

ing safety beacon rates and vehicle densities through experimental analysis based on ns-3 network simulator. We utilize *Collision Risk* as the metric for safety application reliability, which is computed using a kinematics-based model based on tracking error. We observe that there exists an optimal beacon rate for which the collision risk is minimized, and this optimal rate is unique for varying vehicle densities. For instance, in our experiments, the safety-optimal beacon rate for 50 vehicles/km is 25 Hz whereas it is 10 Hz for 200 vehicles/km.

However when taking in practical considerations like tracking ability of the vehicles, we demonstrate that minimizing AoI does not always improve on-road safety performance in V2V networks. This is mainly because AoI solely depends upon the time instants of the BSM generation and reception [81], and does not take into account – (i) *Value of BSM information* - does the reception of a BSM at a certain receiving vehicle actually reduce the collision risk posed by the sender vehicle?, and (ii) *Current knowledge of the receiver vehicle* – what does the receiver already know about the sender vehicle, for example, using tracking capability (as per SAE J2945 [18])? To counter these shortcomings of AoI, we propose a novel metric, termed *Trackability-aware Age of Information* (TAoI), that jointly considers AoI and the self risk assessment of vehicles, measured as the self trackability of the vehicles. In general, *trackability* refers to how accurately the location of a vehicle can be estimated by a neighbor[1] vehicle (based on its most recently received BSM), and better trackability means better on-road safety (as shown in Sec. 2.5). However, since 802.11p V2V networks are decentralized networks, we consider that each vehicle assesses its own trackability behavior (i.e., either trackable or non-trackable) at certain pre-specified time intervals, termed *measurement intervals*, and piggybacks this information in the BSM broadcasts. This allows the receiving vehicles to identify the trackability behavior of sender vehicle. Our extensive simulation experiments based on realistic SUMO traffic traces and ns-3 network simulator show that the proposed TAoI based BSM broadcast rate control protocol significantly outperforms both the baseline 10 Hz broadcast rate (up to 40%) and AoI based rate control algorithm (up to 12%), in terms of improving on-road safety, i.e., collision risk reduction, in all considered V2V scenarios. Furthermore, we also witness that the packet delivery ratio is much better in case of TAoI compared to that of AoI, with similar or lower usage of communication channel resources. *To the best of our knowledge, this is the first work that proposes to improve on-road safety of V2V networks by jointly taking into account – the AoI and self risk assessment of vehicles.*

[1] neighbor vehicle means vehicles within the radio transmission range.

This chapter is organized as follows. Sec. 2.1 discusses the related works in terms of AoI and vehicular networks. Sec. 2.2 describes the tracking error and collision risk whereas Sec. 2.3 presents the two network metrics, i.e., AoI and throughput. Sec. 2.4 details the experimental analysis. Then Sec. 2.5 presents the system model under the SUMO mobility model. Sec. 2.6 explains the limitations of AoI when considered for safety of vehicles. Sec. 2.7 presents an overview of proposed metric TAoI, while Sec. 2.8 discusses the details of TAoI aware rate control algorithm. Sec. 2.9 describes the results, followed by conclusions in Sec. 2.10.

2.1 Related Work

This section discusses the related works on safety performance, AoI and throughput in the context of V2V networks.

In [76], the authors analyzed the effect of beacon rate on network performance of safety messages in V2V networks, and developed a framework to recommend beacon rates based on the message utility maximization. Remote mobility estimators were used in [76] and [91] to control tracking error using adaptive beacon rates. The authors used Channel Busy Ratio (CBR) for controlling beacon rate in [45] to improve safety information dissemination range. [40] and [78] define risk metrics based on the vehicle dynamics and traffic situation, which are then used to select beacon rates. In [105], an adaptive beacon rate algorithm was designed to improve safety in terms of beacon inter-reception time. All these works focus on improving safety performance via controlling beacon rates. On contrary, this chapter investigates the relationship of safety performance with varying vehicle densities and queue sizes and disciplines, in addition to beacon rates. Furthermore, we investigate the interconnection of safety application reliability with two network metrics, i.e., AoI and Throughput.

AoI was introduced in [63] and a distributed algorithm for rate adaptation to reduce AoI was proposed in it. In [27], the authors develop an analytical model that takes into account the Carrier Sense Multiple Access (CSMA) contention to improve the AoI in a V2V network. Llatser et al. [74] simulated a convoy of automated vehicles where AoI was analyzed along

with other parameters for changing convoy-size and beacon rates. A centralized scheduling scheme for beacon broadcasting was designed in [92] to minimize AoI and it was shown to outperform DSRC in terms of fairness and number of effective neighbors. Other papers that deal with AoI can be found in [68]. On the other hand, throughput is a classical metric and it has been well studied for various networks. [49] lists several of the works that analyses throughput in V2V networks. However, none of these works investigate their interconnection with safety performance, which is a key requirement to minimize collision risky situations.

Rate (or broadcast interval) control has been actively investigated as a means of reducing the AoI in vehicular networks. The analytical model formulated by Baiocchi et al. in [27] uses the connectivity graph of the network to demonstrate the relationship of average system AoI with vehicle density and broadcast intervals. Llatser et al. [74] considered a convoy of vehicles and analyzed the change in AoI with changing broadcast intervals and convoy size. A similar case of platooning was considered in [77] where the European Telecommunication Standards Institute's (ETSI) Decentralized Congestion Control (DCC) was analyzed in terms of AoI performance - it was shown that DCC's AoI can be improved by modifying its broadcast intervals to target a specific Channel Busy Ratio (CBR). In [117], Vinel et al. show that 100ms broadcast interval for Cooperative Awareness Messages (CAM), European equivalent of BSM, reduces AoI better compared to mobility based CAM triggering process defined in ETSI EN 302 637-2 [6]. Our work is closest to [63] where a decentralized broadcast rate control algorithm was implemented that improves the system AoI, by iteratively minimizing the locally computed AoI at each vehicle.

In addition to rate control, other approaches that have been used to improve AoI are piggybacking neighbor's information [62], Eigen-vector Centrality (EvC) of the topology [28], optimal contention window [80], machine learning [21] etc. However considering we focus on rate control, these approaches are out-of-scope with respect to our chapter.

The prior works has several limitations – (i) *none of these AoI based rate control protocol works evaluate how AoI improves the on-road safety metric, i.e., collision risk reductions.* It is worth noting that very recently the authors in [35] performed experimental analysis on AoI and safety metrics. However, they consider a simplistic highway model which does not capture the realistic vehicle mobility behaviors. (ii) On similar lines, *most of existing work consider over-simplified vehicle mobility models*, with no lane changing or acceleration/de-

celeration behaviors, and finally (iii) *prior work adopt a risk-neutral approach to AoI* - where each vehicle is given the same importance in reducing their AoI. In this work, we address these limitations by proposing a novel TAoI metric and subsequently a TAoI based broadcast rate control protocol for improving on-road safety of V2V networks.

2.2 Tracking Error and Collision Risk

The calculations for the tracking error and the resultant collision risk are explained below:

2.2.1 Tracking Error

Consider a basic transmitter vehicle and a receiver vehicle that sends and receives safety beacons, respectively. Let the most recent beacon from sender u to receiver v carry the information that u's location is $x_{uv}^{t_1'}$, which is true at the packet generation time t_1'. At time t_1 when the beacon is received, the position of u has changed to $x_u^{t_1}$. The tracking error at v while tracking u at time t_1 with a reception is [91]:

$$\delta pos_{uv}^{t_1} = |x_u^{t_1} - x_{uv}^{t_1'}|, \quad \text{where} \quad t_1' < t_1 \tag{2.1}$$

We consider x-coordinate to calculate the tracking error as the vehicles move in x-direction only. See Sec.2.4.1 for details. The error in Eqn.2.1 occurs because (i) the beacon takes non-zero time to be delivered to the receiver after being generated, and (ii) the sender would likely move a certain distance during this time period. For large time delays, the beacon's information is relatively outdated. Packet losses will further deteriorate this error. If there were beacon receptions at instants t_1 and t_2 whose generation times are t_1' and t_2' respectively, the errors at times t_1 and t_2 are calculated using Eqn.2.1. At times $\widetilde{t}$ with no beacon receptions, where $t_1 < \widetilde{t} < t_2$, we compute the tracking error at vehicle v at time $\widetilde{t}$, as follows:

$$\delta pos_{uv}^{\widetilde{t}} = |x_u^{\widetilde{t}} - x_{uv}^{t_1'}| \tag{2.2}$$

where $x_u^{\widetilde{t}}$ is the location of vehicle u at time instant $\widetilde{t}$, which can be computed as the product of vehicle u's speed and time difference $|\widetilde{t} - t_1|$. $x_{uv}^{t_1'}$ is the location of u as per the last beacon received at vehicle v from vehicle u at t_1. Therefore the tracking error is calculated using

Eqn.2.1 at instants of beacon reception and Eqn.2.2 at instants of no reception.

In the observation interval T, the average tracking error for a certain vehicle pair $u-v$, is $\delta pos_{uv} = \frac{1}{T}\int_T \delta pos^t_{uv}$, where t represents any instant in the observation interval T.

2.2.2 Collision Risk

Time To Collision (TTC) for a pair of vehicles is defined as the time taken for the distance between the pair of vehicles to become 0, which denotes a possible collision between them. We relate the collision risk to the error in the TTC, which results from the tracking error. Every vehicle continuously monitors the TTC with respect to its neighbors as it is a key indicator of safety application reliability [40]. High TTCs mean that the neighboring vehicles are far apart and there is no immediate threat of collision and vice versa.

At any time t, the receiver v calculates TTC to its neighbor u based on the beacon sent by u as:

$$TTC^t_{uv,calc} = |\frac{1}{s_{uv}}(x^t_v - x^{t'}_{uv})| \tag{2.3}$$

x^t_v is v's actual position at time t and $x^{t'}_{uv}$ is u's location as per the last beacon received at v from u with generation time t'. s_{uv} is the relative velocity between them. But, the actual TTC at time t is:

$$TTC^t_{uv,act} = |\frac{1}{s_{uv}}(x^t_v - x^t_u)| \tag{2.4}$$

as at t, sender u is at x^t_u and not $x^{t'}_{uv}$. So the tracking error affects the TTC calculation. This TTC error at any time t can be calculated as the absolute value of the difference between Eqn.2.3 and Eqn.2.4, and is equal to $\delta TTC^t_{uv} = \frac{\delta pos^t_{uv}}{s_{uv}}$. In the observation interval T, the average TTC error for u-v is:

$$\delta TTC_{uv} = \frac{1}{T}\int_T \delta TTC^t_{uv} = \frac{\delta pos_{uv}}{s_{uv}} \tag{2.5}$$

This δTTC^t_{uv} can lead to risky situations and its significance in collision warning was studied in [102]. E.g. suppose v has overestimated/underestimated the TTC to u and is about to take a route/maneuver decision based on this erroneous value of TTC. Such a situation can

be safely averted by manual intervention if the time taken by the driver to react and apply the brakes to make the vehicle stop is less than δTTC^{t}_{uv}. But if this error exceeds the driver's controllability, it presents a risky scenario. We classify the scenario to be risky if:

$$\delta TTC_{uv} > t_{react} + t_{brake} \tag{2.6}$$

We use a Gaussian distribution to model the speeds of the vehicles with a mean of 25 m/s and a variance of 3 m/s [42]. t_{react} is taken to be 1s [40] and is the time taken by the driver to respond to the situation and apply the brakes. We take the common deceleration of all the vehicles as 4.6 m/s^2 [40], and this leads to a value of $t_{brake} = 5.43$ s for the vehicles to come to a complete stop. Eqn. 2.6 then becomes: $\delta TTC_{uv} > 6.43$ s for a scenario to be risky.

For every simulation performed, we count the proportion of the vehicle pairs whose δTTC_{uv} exceeded 6.43 s during the interval they communicated. This is taken as the measure of the overall safety performance of the network. The logic is that if δTTC_{uv} was under threshold for a particular pair, the receiver's routing/maneuver decisions would have considered the correct location of the sender most of the times. So it is unlikely that the receiver's actions will cause any collisions. It should be kept in mind that the TTC error arises solely due to the tracking error, which in turn arises due to the delay and packet loss in the communication. Thus the safety performance is being measured from a purely communications point of view, and no other phenomenon is considered.

2.3 Age of Information and Throughput

In this section, we briefly discuss two network metrics, i.e., Age of Information (AoI), and Throughput. We then study their interconnection with safety performance, i.e., collision risk, under various V2V scenarios.

2.3.1 Age of Information (AoI)

AoI at the receiving vehicles is calculated as the time elapsed since the received beacon was generated at the sender until the beacon's reception time. It is equal to the end-to-end delays at the instants of reception, with linear increase between receptions. While delay is

associated with the packets and is undefined when there is no reception, AoI can be thought of as a continuous variable that is defined for the entire observation interval. We define the observation interval T as the interval between the reception of the first and the last beacons between a sender and a receiver. The average AoI is calculated as the total area under the AoI plot normalized by the observation interval [63], i.e. $\triangle_{uv} = \frac{1}{T}\int_T \triangle^t_{uv}$, where $\triangle^t_{uv}$ represents the AoI of vehicle u's information at vehicle v. It is calculated as $\triangle^t_{uv} = t - g(t)$ at instant t, with $g(t)$ being the generation time of the most recent packet received by v. The system AoI with N vehicles is calculated across $N(N-1)$ unique pairs of sender and receiver as:

$$\triangle = \frac{1}{N(N-1)} \sum_{u} \sum_{v \neq u} \triangle_{uv} \tag{2.7}$$

More details on AoI will be provided in Sec. 2.6. As our work is focused on safety applications, we don't consider LCFS queues of sizes more than 1, as having a bigger LCFS queue provides no practical utility in terms of time-sensitive information. This is because an LCFS queue will send out the most recently generated packet first making the older packets wait, and these older packets do not add any value to the receiver vehicle after the newer packet has been received. They are differentiated as *informative* and *non-informative* packets [112]. However for the sake of completeness, we show the variation of AoI in a specific case with LCFS queues in Fig. 2.1. It can be seen that in LCFS, queue sizes do not make a big difference in AoI. In the rest of our analysis, we won't be showing the LCFS results as LCFS of the considered queue sizes (1,5,10,100 packets) perform very close to a FCFS queue of size 1 as per Fig.2.1[2]. Therefore, the AoI performance of an LCFS queue can be roughly assumed to have a performance similar to that of an FCFS for queue size 1. The reception of non-informative packets do not lead to any reduction in AoI [112]. This can also be understood in terms of tracking error: once the recent location of a vehicle is known via a recent beacon, the location information obtained from the older beacons that are transmitted after the newer beacon in an LCFS queue do not add any value. Tracking error for any LCFS queue then becomes similar to the tracking error of an FCFS queue of size 1. As collision risk is computed directly from the tracking error, the above property of LCFS queue applies to collision risk in LCFS queues too.

[2]a small but negligible AoI difference between queue sizes of 1 and 100 can be noticed for 200 veh/km.

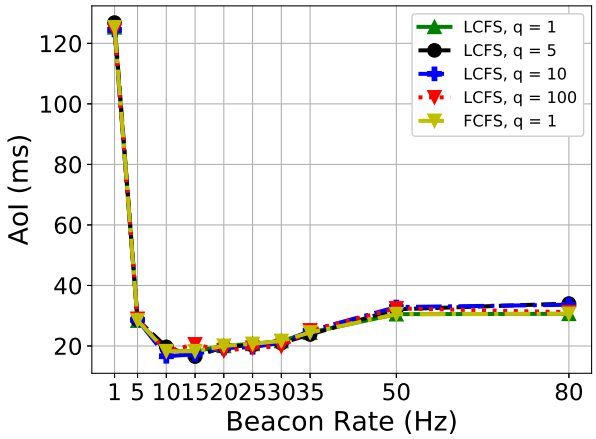


Figure 2.1: Age in an LCFS queue

2.3.2 Throughput

Throughput is a measure of the rate of data transfer, and it is calculated as the total data transferred in the entire simulation. Taking τ as the observation interval and K as total packets transferred in that duration, total throughput $= \frac{K}{\tau}$ packets per second. Average throughput per vehicle can be obtained by normalizing the total throughput with the number of vehicles. Throughput is not affected by LCFS/FCFS, and the performance of an LCFS queue would be exactly same as than of a FCFS queue of the same size.

2.4 Experimental Analysis

This section describes the simulation setup, followed by performance analysis of safety application reliability, i.e., collision risk and two network metrics, i.e., AoI and Throughput, against varying beacon rates and vehicle densities.

2.4.1 Simulation Setup

We utilize Network Simulator 3 (ns-3) for our experimental study. We consider that the vehicles are placed uniformly along a 3-lane highway of length 1 Km. The vehicles move along the highway at constant speeds along the x-axis without any lateral movement across the lanes, until the simulation ends. Unless otherwise stated, the beacon/transmission rates were varied from 1 Hz to 80 Hz, and a small random number is added to this rate to ensure that all the vehicles do not attempt transmission at the same time [63]. The packet sizes are set to 320 bytes The queue sizes are varied between 1, 5, 10 and 100 packets, and the queue

discipline is FCFS. Recall LCFS is equivalent to FCFS with size 1, irrespective of queue sizes for LCFS. A list of the important parameters used in the simulation along with their values are given in Table 2.1.

Table 2.1: Parameters used to set up the simulation

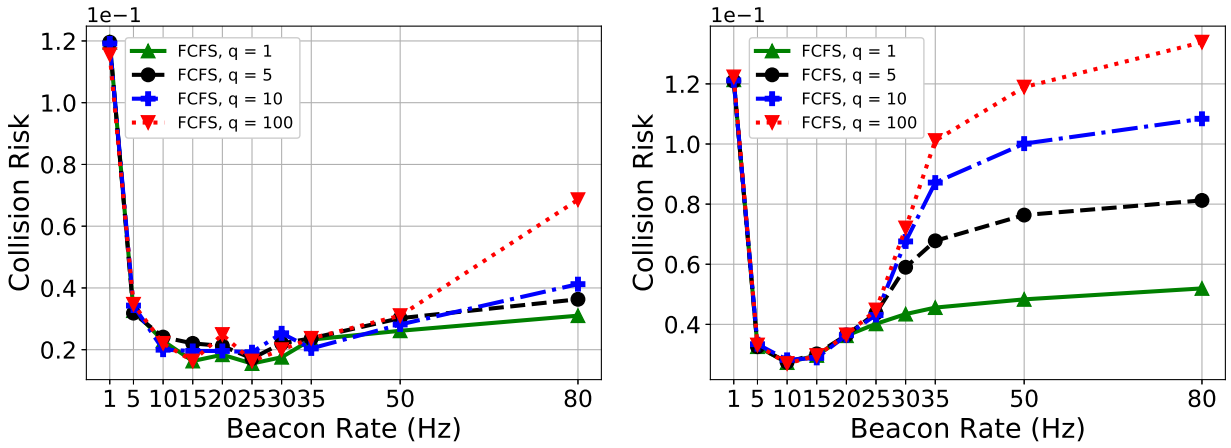
Parameters	**Value**
Vehicle Density	50 veh/km - 200 veh/km
Number of Lanes	3
Lane Width	4 m
Packet Size	320 bytes
Data Rate	6 Mbps
Loss Model	Log Distance Propagation Loss
Path Loss exponent	$\gamma = 3$
Channel Frequency	5.9 GHz
Channel Bandwidth	10 MHz
Vehicle Speeds	N(25 m/s, 3 m/s)
t_{brake}	5.43 s
t_{react}	1 s
Antennas/Spatial Streams	1/1

2.4.2 Experimental Results

Next, we discuss the comparative analysis of collision risk, AoI, and throughput against varying rates and densities.

Collision Risk Analysis: Fig. 2.2 depicts how collision risk varies with varying beacon rates, under considered vehicle densities of 50 vehicles/Km and 200 vehicles/Km. Collision risk is minimal at 25 Hz for 50 veh/km whereas 10 Hz for 200 veh/km, which means that there is a unique beacon rate for a certain vehicle density that maximizes the safety application reliability. Furthermore, note that there exists a convex relationship between the collision risk and beacon rates, which hints that gradient-descent algorithms may be utilized to analytically determine the optimal safety-optimal beacon rate (which will be a part of our future work). From this analysis, it is clear that enforcing a fixed beacon rate of 10 Hz (by DSRC standard) may not be optimal in all V2V scenarios. Regarding the queuing sizes, collision risk remains unaffected for lower beacon rates, however, it increases gradually after 35 Hz for lower vehicle density, and after 25 Hz for higher vehicle density. Longer queues perform poorly compared to that of shorter queues, and interestingly, queue size 1 performs the best. These observations can mainly be credited to its correlation with AoI (next paragraph).

AoI Analysis: Fig.2.3 depicts the relationship between AoI and safety beacon rates under



(a) 50 veh/Km (b) 200 veh/Km

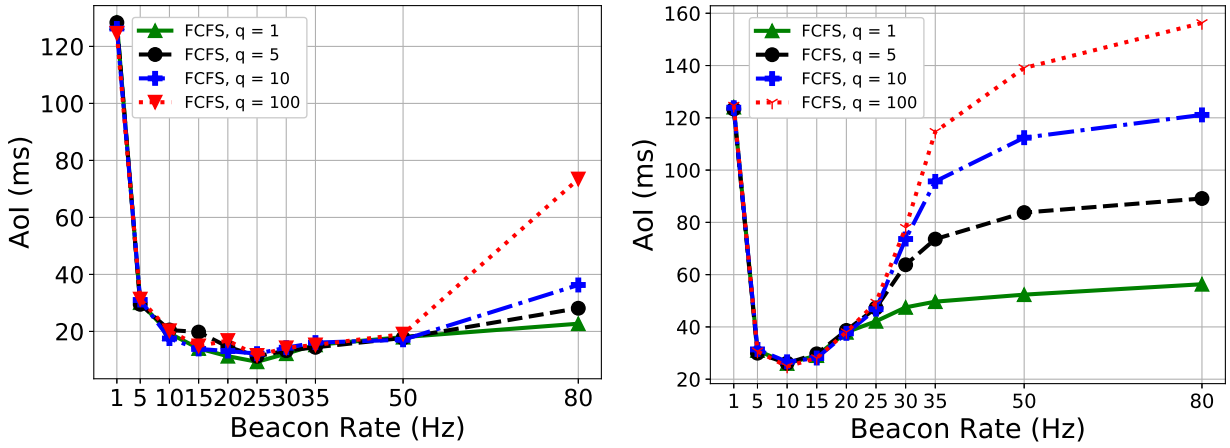
Figure 2.2: Collision risk vs Beacon rates

varying vehicle densities. Similar to collision risk, it is interesting to see that AoI is also a convex function of beacon rate. More interestingly, the AoI-optimal beacon rate is the same as the safety-optimal beacon rate, for both the considered vehicle densities. This is an important result as it establishes that AoI and safety application reliability are highly correlated, and safety application's beacon rate algorithm can reliably undertake AoI minimization as the sole objective, rather than safety application reliability[3].

Similar to collision risk, AoI also remains unaffected for lower beacon rates, and after optimal beacon rate, it increases gradually for both the density scenarios. Smaller queues have lower AoI and this can be largely attributed to lower queuing delays. Fig. 2.4 shows that delays are lower for shorter queues, which in turn results in lower AoI and vice-versa. At very high beacon rates, the network becomes saturated, which results in a saturation in AoI, as also pointed out in the literature [27].

An important observation is that at lower beacon rates, delay is less (Fig.2.4) as the channel resources are available whereas the collision risk is very high (Fig. 2.2). This shows that lower delay doesn't guarantee higher safety performance. However similar to collision risk, AoI is higher for both lower and higher beacon rates. AoI is higher for lower beacon rates due to the high inter-arrival times while at higher beacon rates, AoI is higher mainly due to higher queuing delays.

[3]Computing safety application reliability, i.e., collision risk, in real-time is very hard in V2V networks compared to that of AoI, a network metric.



(a) 50 veh/Km (b) 200 veh/Km

Figure 2.3: Age of Information vs Beacon rates

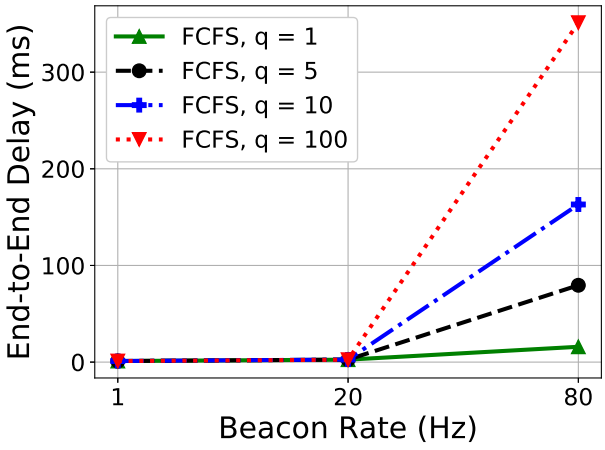


Figure 2.4: End-to-end-delay

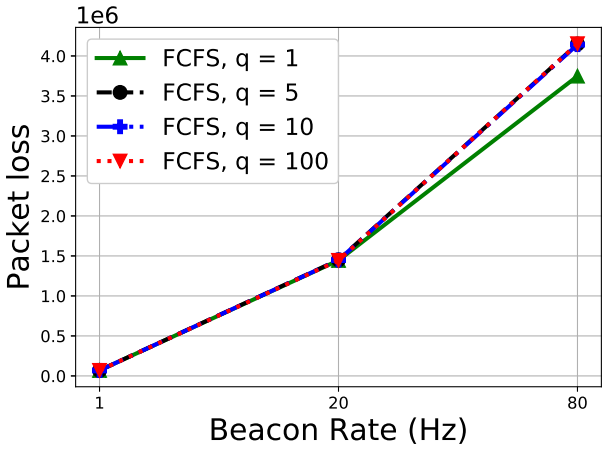
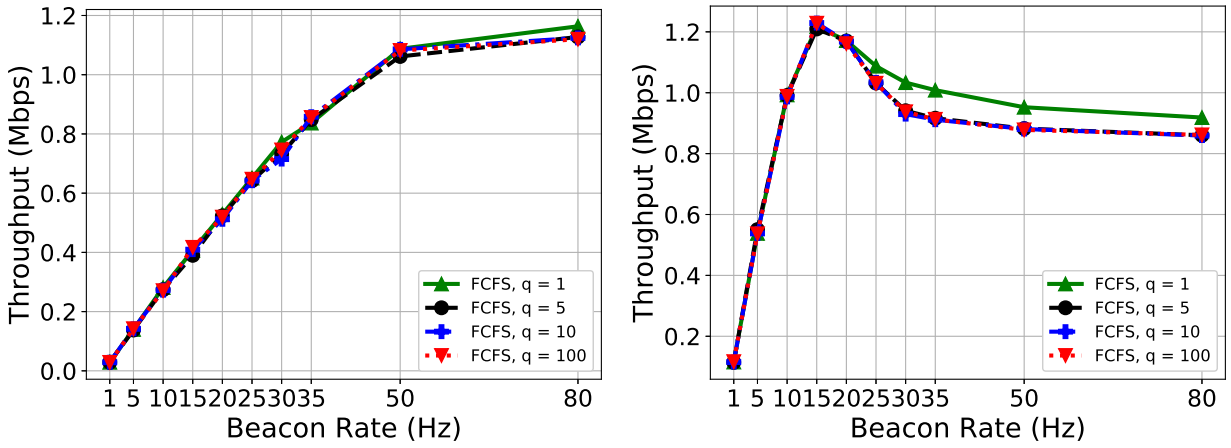


Figure 2.5: Packets lost in collisions



(a) 50 veh/Km (b) 200 veh/km

Figure 2.6: Throughput vs Beacon rates

Throughput Analysis: Fig. 2.6 shows the gradual increase in throughput until it peaks, and then it saturates to a lower value. For 200 veh/Km, it peaks at 15 Hz and starts saturating from 35 Hz; while it doesn't saturate until 80 Hz for 50 veh/Km.

Notice that for lower vehicle density, the throughput-optimal beacon rate (80 Hz) is very far from safety-optimal rate (25 Hz), however, for higher vehicle density, throughput-optimal beacon rate (15 Hz) is closer to the safety-optimal rate (10 Hz), and performs quite well in terms of collision risk. Similar to collision risk and AoI, a queue size of 1 is throughput-optimal, and this can be attributed to the fact that packet loss due to packet collisions is less for smaller queues, compared to that of larger queues [108] (See Fig. 2.5).

In this section, we performed a comprehensive experimental analysis of safety application reliability (in terms of collision risks) against beacon rates and vehicle densities for V2V networks, and concluded that there exists a unique beacon rate for which the safety application reliability is maximized, and it is unique for each vehicle density. Furthermore, our experiments showed that there exists a strong interconnection between collision risk and Age of Information (AoI) notwithstanding vehicle densities, and other queuing parameters. Whereas, the throughput-optimal rate approaches safety-optimal beacon rate only under higher vehicle densities. Queue size of 1 is both AoI and throughput optimal in all

considered V2V scenarios.

In the next part of the analysis, we will allow for the vehicles to track the location of the neighbor vehicles as mandated by SAE-J2945/1. We also use Simulation of Urban MObility (SUMO) to model the vehicles' mobility as realistically as possible. The new system model is explained below.

2.5 System Model

Our system model consists of $\mathcal{N}$ vehicles moving in an m lane rectangular road as shown in Fig. 2.7. Each vehicle $u \in \mathcal{N}$ moves with a certain velocity $s_u \in (0, s_{max}]$. The movement of each vehicle is implemented using the Krauss' mobility model [69], which regulates its acceleration (and deceleration); and the lane changes are integrated as per [44]. Unlike existing literature, this ensures a realistic representation of real-world driver and vehicle mobility behaviors.

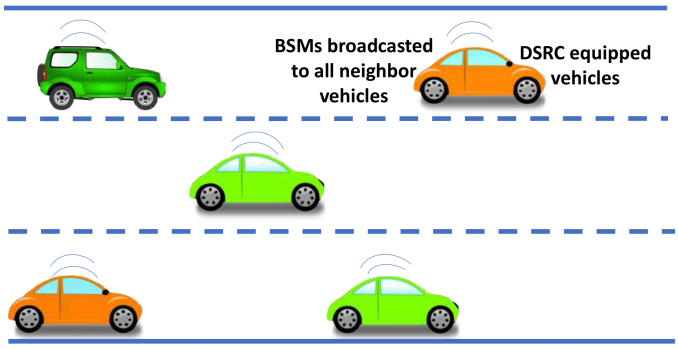


Figure 2.7: System model: A simple 802.11p based V2V network

Each vehicle uses 5.9 GHz DSRC wireless technology that has been designed to support vehicular safety applications (such as CCW, EEBL and SVA) in V2V networks [5]. DSRC uses existing IEEE 802.11p standard as its PHY and MAC layers; mainly because of the widespread availability of IEEE 802.11 chip-sets as well as its performance and cost savings. We assume each vehicle is equipped with a single 10 MHz DSRC radio, and the transmission power is fixed. Vehicular safety applications rely heavily on periodic broadcast of BSMs

which contain the vehicle's position, heading, and other information about the transmitting vehicle. Without loss of generality, consider that each vehicle u broadcasts its BSM at periodic time intervals of Δ (usually 100 ms) which can take any random value in $[\Delta_{min}, \Delta_{max}]$. It means that there is at least Δ inter-reception delay between two consecutive BSMs from a vehicle u received at a neighboring vehicle v. Moreover, since u broadcasts its BSM as per Carrier Sense Multiple Access (CSMA) scheme followed by 802.11p [115], it leads to additional communication delay, (i.e., queuing delay, transmission delay, and propagation delay), denoted by x, at receiving vehicle v. Because of both Δ and x, the information in u's BSM is relatively *outdated* (or stale) by the time it is received by the neighboring vehicle v. *The lack of fresh or up-to-date information at a receiver may lead to wrong positioning of sender vehicle, i.e. v will not be able to localize u correctly and presents a potential on-road safety hazard.*

To address this, a promising solution is to enable a receiving vehicle v to track (or *estimate*) u's current location during these inter BSM reception times (in time instants of no BSM receptions). As per SAE J2945 [23], a linear extrapolation based on the last known velocity, position and heading as per the most recent BSM received can be used to estimate the sender vehicle's current location. However, note that a linear extrapolation will not always work well if the vehicle being tracked doesn't have a fixed linear mobility behavior. This will give rise to a non-negligible error at a vehicle v while estimating the location of neighboring vehicle u, which we call *Tracking error* (TE). Lower TE value means that v is able to track u (or estimate u's location) relatively well, and vice versa. This tracking error across all the vehicles is then used to calculate the on-road safety of V2V networks, measured as *Collision risk*. Note that the tracking error, collision risk and other terms will be re-introduced in the below sections as their calculation will change compared to Sec. 2.2 where we didn't consider tracking ability as per SAE J2945/1 at the vehicles.

Tracking Error (TE), $\tau_{e,uv}$

As mentioned earlier, $\tau_{e,uv}$ is defined as the difference between the ground truth location of the sender vehicle u and the sender's location as estimated by its neighboring vehicle v. Let the most recent BSM received at v from u was generated at time t'. Then at any time $t > t'$, v can estimate u's current location as $(\hat{x}_{uv}(t), \hat{y}_{uv}(t))$ where

$$\begin{aligned} \hat{x}_{uv}(t) &= x_u(t') + s_u(t')\cos(\theta_u(t'))\delta t \\ \hat{y}_{uv}(t) &= y_u(t') + s_u(t')\sin(\theta_u(t'))\delta t \end{aligned} \tag{2.8}$$

where $\delta t = (t - t')$, and $(x_u(t'), y_u(t')), s_u(t')$ and $\theta_u(t')$ respectively denote u's location, speed, and heading information contained in the last BSM received from u. Then the TE $\tau_{e,uv}$ that v has in tracking u at time t is calculated as -

$$\tau_{e,uv}(t) = \sqrt{(x_u(t) - \hat{x}_{uv}(t))^2 + (y_u(t) - \hat{y}_{uv}(t))^2} \tag{2.9}$$

where $(x_u(t), y_u(t))$ is the ground truth location of u at t. Note vehicle u may not see the same TE in tracking vehicle v, i.e., $\tau_{e,vu}(t) \neq \tau_{e,uv}(t)$, because v's BSM may have been generated at different time, say t'' where $t'' \neq t'$, and also v's location, speed and heading may be different to that of u.

The average TE that v has in tracking u in a certain time window T can be computed as

$$\tau_{e,uv} = \frac{1}{T}\int_T \tau_{e,uv}(t) \tag{2.10}$$

Significance. The concept of TE is used to compute the Collision Risk as described later in this Sec. 2.5.

Self Tracking Error (Self-TE), $\tau_{p,u}$

Self-TE $\tau_{p,u}$ is the difference between the actual location of a certain vehicle u, and the self-estimated location of the same vehicle. It is periodically calculated after a certain time interval and this pre-specified time interval at which $\tau_{p,u}$ is (re) computed is referred to as *measurement interval* (t_{MI}). We utilize the same linear extrapolation approach (used in TE) that uses the position, speed and heading at $(t - t_{MI})$ to estimate the u's current location as $(\bar{x}_u(t), \bar{y}_u(t))$ at time t where -

$$\begin{aligned} \bar{x}_u(t) &= x_u(t - t_{MI}) + s_u(t - t_{MI})\cos(\theta_u(t - t_{MI}))t_{MI} \\ \bar{y}_u(t) &= y_u(t - t_{MI}) + s_u(t - t_{MI})\sin(\theta_u(t - t_{MI}))t_{MI} \end{aligned} \tag{2.11}$$

The self-TE of u at time t, denoted by $\tau_{p,u}(t)$, is -

$$\tau_{p,u}(t) = \sqrt{(x_u(t) - \bar{x}_u(t))^2 + (y_u(t) - \bar{y}_u(t))^2} \tag{2.12}$$

Significance. This concept of self TE is the basis for identifying the trackability behavior (i.e., either trackable or non-trackable) of a certain vehicle at each measurement interval t_{MI}. This is the first step of our proposed TAoI rate control protocol. Refer to Sec. 2.7 for more details.

Collision Risk

Here we describe how we compute on-road safety of V2V networks, measured as *Collision Risk* and is inspired from the risk model developed in [35].

In V2V networks, Time To Collision (TTC) is an important safety metric due to which each vehicle continuously monitors the TTC to the neighboring vehicles [26, 39, 40, 113]. TTC for a pair of vehicles is defined as the time needed for the distance between them to become zero, which denotes a potential collision between them. Mathematically, TTC can be computed as the ratio of distance between u and v and their relative velocity. See Fig. 2.8. High TTCs mean that there is no immediate threat of collision between u and v given the current distance and relative velocity between them and vice-versa.

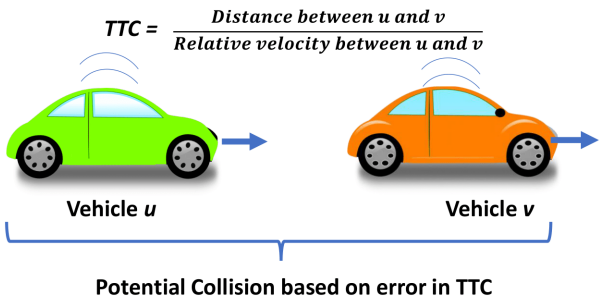


Figure 2.8: TTC Calculation

Now given the TE ($\tau_{e,uv}$) that a v has in tracking u, the error in TTC, denoted by δTTC_{uv}, between the pair $u - v$ can be calculated as [35]–

$$\delta TTC_{uv}(t) = \frac{\tau_{e,uv}(t)}{s_{uv}(t)} \tag{2.13}$$

where $s_{uv}(t)$ is the relative velocity between u and v at t. The significance of δTTC for collision avoidance has been studied in [102]. Specifically, a certain on-road situation can be considered collision risky (or dangerous) whenever the δTTC for any pair of vehicles exceeds the threshold value of δTTC_{thresh}, where δTTC_{thresh} is the time needed for the driver to react to the possible collision and bring the vehicle to a stop. Thus, δTTC_{thresh} is computed as -

$$\delta TTC_{thresh} = t_{react} + t_{brake} \tag{2.14}$$

where t_{react} is 1s and refers to the time taken by the driver to respond to the situation and apply the brakes [35]. t_{brake} is the time taken by the vehicle to come to a complete stop after the brakes have been applied. Taking the deceleration for a vehicle as $a = 4.6\text{m/s}^2$ [35] makes $t_{brake} = s/a$ where s is the velocity of the vehicle. A key difference from the study done in [35] is that in this work, s is not fixed for the entire simulation duration and changes with time for each vehicle.

Given δTTC_{uv} and δTTC_{thresh}, the collision risk for each vehicle pair u - v at any time t can be computed as -

$$CR_{uv} = \begin{cases} 1 & \delta TTC_{uv}(t) > \delta TTC_{thresh} \\ 0 & otherwise \end{cases} \tag{2.15}$$

Using Eq. 2.15, we count the number of instances between each pair of vehicles in which δTTC exceeded δTTC_{thresh}, as the measure of overall on-road safety of the V2V network. Note that δTTC is directly proportional to TE, τ_e, between each pair of vehicles in the network (See Eq. 2.13). Thus, *it is critical the τ_e between every pair of vehicles is minimized in order to enhance on-road safety performance of V2V networks.*

2.6 Age of Information and Safety Performance

In this section, we briefly describe Age of Information (AoI) and then discuss the limitation of AoI based rate control approach on the safety performance of V2V networks.

2.6.1 Age of Information (AoI)

First introduced in [63], Age of Information (AoI) is used to quantify the freshness of information at a destination node (i.e. a receiving vehicle) about some measurements. generated by the source node (i.e. a sender vehicle). More formally, AoI in V2V networks can be defined as the time elapsed since the last successfully received BSM at the receiving vehicle was generated at the sender vehicle. In order to better understand the AoI, let us look at Fig. 2.9 which depicts a realization of AoI, denoted by $AoI_{uv}(t)$, at the receiving vehicle v as a function of time when a sender vehicle u transmit BSMs using a first come first serve (FCFS) discipline. Let t' denote the generation time of most recent BSM at sender vehicle u, then AoI at receiving vehicle v at current time t is calculated as

$$AoI_{uv}(t) = t - t' \tag{2.16}$$

Note that $AoI_{uv}(t)$ is a zigzag-like function with a slope of 1 between the BSM inter-reception intervals and is reset to the end-to-end delay in each time instance when u's new BSM is successfully received at v.

Using Eq. 2.16, the average AoI_{uv} between vehicle v and vehicle u can be calculated as the total area under the $AoI_{uv}(t)$ plot normalized by the observation interval T [63], i.e.

$$AoI_{uv} = \frac{1}{T}\int_{T} AoI_{uv}(t) \tag{2.17}$$

Consider that v has $\mathcal{N}_v \subseteq \mathcal{N}$ neighboring vehicles, i.e., $u \in \mathcal{N}_v$ where $\mathcal{N}$ is the total number of vehicles. Then the average AoI at a receiving vehicle v can be computed as follows.

$$AoI_v = \frac{1}{|\mathcal{N}_v|}\sum_{u\in\mathcal{N}_v} AoI_{uv} \tag{2.18}$$

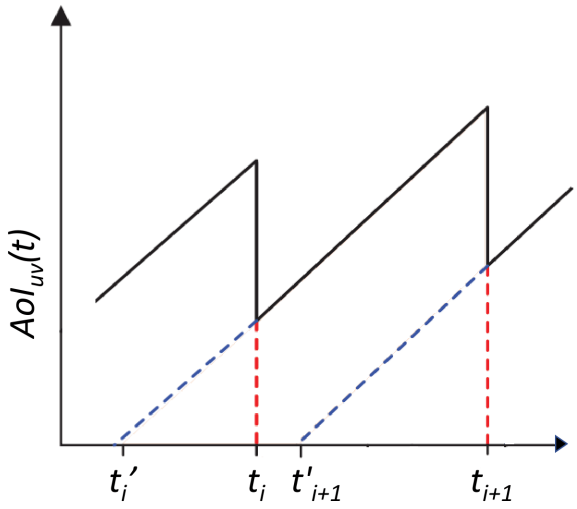


Figure 2.9: Evolution of AoI_{uv} at v based on the BSMs received from u. The generation and reception times of the i^{th} BSM are denoted by t'_i and t_i, which makes the overall delay for that BSM $= (t_i - t'_i)$. The $(i+1)^{th}$ BSM is generated at $t'_{(i+1)}$, and therefore the BSM broadcast interval is given by $\Delta = (t'_{i+1} - t'_i)$. There is no BSM reception between t_i and $t_{(i+1)}$, and the AoI grows linearly in this interval $(t_{i+1} - t_i)$ known as inter BSM reception interval.

Finally, the system AoI with $\mathcal{N}$ vehicles can be calculated across $\mathcal{N}(\mathcal{N}-1)$ unique pairs of sender and receiver as:

$$AoI = \frac{1}{\mathcal{N}(\mathcal{N}-1)} \sum_{v \in \mathcal{N}} \sum_{u \in \mathcal{N}_v} AoI_{uv} \tag{2.19}$$

From the above discussion, it is evident that AoI is less when the receiver vehicle receives BSMs frequently from the sender vehicle. As the reception of frequent BSMs means the receiver has knowledge of the most recent location of the sender, a less AoI means less TE. Therefore minimizing the system AoI in Eq. 2.19 should improve the safety of all the vehicles in the network. However, when practical constraints like limited communication resources, i.e., channel capacity and different mobility behaviors of vehicles are considered, a lower system AoI may not always result in improved safety performance as we shall see in Sec. 2.6.2.

2.6.2 Limitations of AoI in defining Safety Performance

Let us consider a simple V2V scenario with two vehicles u and v. For ease of understanding, we make the following assumptions for our considered V2V scenario -

- Channel capacity is limited to one BSM transmission at any time - it means at any given time t, only one vehicle can broadcast its BSM.
- There is no additional communication delay involved ($x = 0$) - it means the inter-BSM reception intervals between a receiver v and sender u is equal to Δ, where Δ is the BSM broadcast interval.
- The back-off time is taken as Δ. If u broadcasts at time $t = \Delta$, and another sender vehicle v senses the channel busy, then v will attempt its next transmission at 2Δ. Thus, the instants of transmission will be $t_1 = \Delta$, $t_2 = 2\Delta$, and so on, where either u or v gets to transmit.

Under these assumptions, AoI can be computed as -

$$AoI(t+\Delta) = \begin{cases} 0, & \text{if a BSM is received} \\ AoI(t) + \Delta, & \text{otherwise} \end{cases} \tag{2.20}$$

Lets consider that the mobility behavior of u and v is given by the following equations -

$$\begin{aligned} y_u(t) &= 2t, \quad \dot{y}_u(t) = 2 \\ y_v(t) &= t^2, \quad \dot{y}_v(t) = 2t \end{aligned} \tag{2.21}$$

where $y(t)$ and $\dot{y}(t)$ are the position and velocity at time t respectively. Both u and v will estimate each other's current locations as $\hat{y}_{vu}(t)$ and $\hat{y}_{uv}(t)$, respectively using Eq. 2.8. Then the TE that u has in tracking vehicle v ($\tau_{e,vu}$) and v has in tracking vehicle u ($\tau_{e,uv}$) at time t is given by Eq. 2.9.

Let us now see how AoI and an alternative rate control strategies treat AoI and TE in this simple V2V scenario.

Under *AoI* based rate control strategy:

Both u and v should be allowed an equal access to the channel resources to broadcast their BSMs so that the system AoI is minimized. However, given the restriction of at most 1 transmission at any time, it is intuitive that the optimal way of minimizing AoI would be to allow u and v to broadcast their BSM at alternate intervals. E.g. if u transmits at time instance t_1, then v transmits at t_2 followed by u at t_3, and v at t_4, and so on. Lets consider $\Delta t = 1$ for the ease of presentation.

The resulting AoI and TE is shown in Table 2.2. Time instants in red are u's broadcast instants (and rest are v's).

From Eq. 2.17, $AoI_{uv} = \frac{3}{6} = 0.5$. Similarly, $AoI_{vu} = 0.5$. Then from Eq. 2.19, system AoI is

$$AoI = \frac{AoI_{uv} + AoI_{vu}}{2} = 0.5$$

For TE, Eq. 2.10 gives $\tau_{e,uv} = \frac{2}{6} \approx 0.33$, $\tau_{e,vu} = \frac{15}{6} = 2.5$.

Under an alternative rate control strategy:

Where u transmits at t_1 and then v gets to transmit in the remaining time. The resulting AoI and TE are shown in Table 2.3. The average AoI and TE at v are as follows -

$AoI_{uv} = \frac{15}{6} = 2.5$ and $AoI_{vu} =$
$\frac{1}{6} \approx 0.17 \Rightarrow AoI = 1.334$ Similarly, $\tau_{e,uv} = \frac{2}{6} \approx 0.33$ and $\tau_{e,vu} = \frac{9}{6} = 1.5$.

Regarding safety performance, the difference in TE in the two strategies occur for $\tau_{e,vu}$ at t_4 and t_6. The calculation of $\tau_{e,vu}(t_4)$ is explained here - In the AoI scheme, u receives a BSM from v at t_2 when $y_v(t_2) = 4$, $\dot{y}_v(t_2) = 4$. At t_3, $\delta t = t_3 - t_2 = 1$ and u estimates v's location as per Eq. 2.8 as

$$\hat{y}_{vu}(t_3) = y_v(t_2) + \dot{y}_v(t_2) \times 1 = 8$$

TE at u is calculated as $\tau_{e,vu}(t_3) = y_v(t_3) - \hat{y}_{vu}(t_3) = 9 - 8 = 1$. As v doesn't transmit at t_3, u again estimates v's location at t_4 from BSM it last received at t_2 using Eq. 2.8 as -

$$\hat{y}_{vu}(t_4) = y_v(t_2) + \dot{y}_v(t_2) \times 2 = 12$$

where $\delta t = t_4 - t_2 = 2$. Then TE $\tau_{e,vu}(t_4) = y_v(t_4) - \hat{y}_{vu}(t_4) = 16 - 12 = 4$.

The $\tau_{e,vu}(t_4)$ in the alternate strategy is less because v gets to transmit at t_3 unlike in the

AoI strategy. Hence u estimates v's location at t_4 with $\delta t = t_4 - t_3 = 1$ as -

$$\hat{y}_{vu}(t_4) = y_v(t_3) + \dot{y}_v(t_3) \times 1 = 9 + 6 \times 1 = 15$$

and the new TE is $\tau_{e,vu}(t_4) = y_v(t_4) - \hat{y}_{vu}(t_4) = 16 - 15 = 1$. In this way, $\tau_{e,vu}$ reduces from 4 to 1, thereby improving safety, even though system AoI has degraded ($1.334 > 0.5$) when the alternative rate control strategy is used.

Table 2.2: AoI rate control strategy

Time	t_1	t_2	t_3	t_4	t_5	t_6
$AoI_{uv}(t)$	0	1	0	1	0	1
$y_u(t)$	2	4	6	8	10	12
$\hat{y}_{uv}(t)$	0	4	6	8	10	12
$\tau_{e,uv}(t)$	2	0	0	0	0	0
$AoI_{vu}(t)$	1	0	1	0	1	0
$y_v(t)$	1	4	9	16	25	36
$\hat{y}_{vu}(t)$	0	0	8	12	24	32
$\tau_{e,vu}(t)$	1	4	1	**4**	1	**4**

Table 2.3: Alternative rate control strategy

Time	t_1	t_2	t_3	t_4	t_5	t_6
$AoI_{uv}(t)$	0	1	2	3	4	5
$y_u(t)$	2	4	6	8	10	12
$\hat{y}_{uv}(t)$	0	4	6	8	10	12
$\tau_{e,uv}(t)$	2	0	0	0	0	0
$AoI_{vu}(t)$	1	0	0	0	0	0
$y_v(t)$	1	4	9	16	25	36
$\hat{y}_{vu}(t)$	0	0	8	15	24	35
$\tau_{e,vu}(t)$	1	4	1	**1**	1	**1**

Based on these discussions, we can make the following key observations – even though u's AoI at v (AoI_{uv}) is lower in AoI based rate control strategy compared to that of the alternative one, v incurs the same TE ($\tau_{e,uv}$) in tracking u in both the strategies. It means minimizing AoI_{uv} in the AoI based rate control strategy did not help in improving u's safety. On the other hand, AoI_{vu} is lower in alternative rate control strategy and it leads to a significant reduction in TE, $\tau_{e,vu}$, which will promisingly improve safety of v.

This is because, as shown in Eq. 2.21, u does not change its mobility behavior (i.e., speed and heading), or in other words, u *is trackable*. It means, after v receives the first BSM from u at

t_1, v accurately estimates u's current location at times $t_2, \ldots t_6$ and thus, leading to $\tau_{e,uv} = 0$ for the remaining time, even in the absence of fresh BSM receptions. This fact is being harnessed by the alternative rate control strategy where after allotting a single transmission for v, all the remaining times $(t_2, \ldots t_6)$ are allotted to v. As v's mobility behavior keeps changing with time (i.e., *v is non-trackable*), location estimation doesn't perform well in tracking v due to which frequent BSM receptions are required at u from v to enable better tracking of v so that $\tau_{e,vu}$ is reduced.

Because the non-trackable vehicles have high TE and as a result are likely to have a higher value of collision risk (Sec. 2.5), we call these *risky* vehicles. Similarly, the trackable vehicles are called *non-risky* vehicles.

From the above V2V scenario (and key observation), it is clear that minimizing AoI does not always improve the safety of V2V networks. It becomes critical to take into account the trackability of vehicles. With that, the following deductions can be made about the relation between AoI and safety in the context of vehicular networks -

- AoI minimization is not important for non-risky vehicles that can be tracked well by its neighbors.
- For risky vehicles, AoI minimization will improve the TE, and thus, on-road safety of the network.

2.7 Trackability aware AoI (TAoI) metric

To address the shortcomings of AoI, we propose a novel *Trackability aware AoI* (TAoI) metric that accounts for both AoI and trackability (i.e., risky or non-risky) of neighboring vehicles for improved safety performance of V2V networks. To get the trackability information about the neighboring vehicles, a certain vehicle v can compute the TE $\tau_{e,uv}$ for each neighbor u at instants of BSM receptions using Eq. 2.8 and Eq. 2.9. However, this TE approach has two major limitations – (i) *a vehicle learns the value of TE only at BSM reception instants*, where the inter-BSM reception time intervals may be very large (100's of milliseconds), and (ii) *additional communication overheads in V2V networks*, because the receiver vehicle which computes TE, has to send back the feedback (with TE information) to every neighboring vehicle.

A better approach to measure trackability (or riskiness) is to do *self risk assessment* by utilizing self TE $\tau_{p,v}$ as vehicles having high self TE are likely to have high TE. It is computed locally at each vehicle, and does not suffer from the aforementioned limitations. Each vehicle identifies itself as a *non-risky* or *risky* vehicle based on its self-TE value, and shares this information with neighboring vehicles by piggybacking it along with its BSM. Based on this information, every vehicle is aware of which of the neighborhood vehicles are risky.

For a sender-receiver pair u-v, calculation of the instantaneous and average TAoI is similar to the calculation of AoI. At any time t, $TAoI_{uv}$ at v with respect to u is given by

$$TAoI_{uv}(t) = AoI_{uv}(t) \times \mathbf{I}(\tau_{p,v} \geq \tau_{p,th}) \tag{2.22}$$

where $AoI_{uv}(t)$ is AoI at v with respect to u and is computed using Eq. 2.16. $\mathbf{I}(\tau_{p,v} \geq \tau_{p,th})$ is an indicator function that equals 1 only when the self TE $\tau_{p,v}$ exceeds a fixed threshold $\tau_{p,th}$ and is 0 otherwise. Therefore, TAoI takes on the value of AoI when $\tau_{p,v}$ is greater than $\tau_{p,th}$, otherwise it is 0.

Similarly, the average TAoI for the pair u-v over the observation interval T is calculated as:

$$TAoI_{uv} = \frac{1}{T} \int_T TAoI_{uv}(t) \tag{2.23}$$

Then the average TAoI at v becomes -

$$TAoI_v = \frac{1}{|\mathcal{N}_v^r|} \sum_{u \in \mathcal{N}_v^r} TAoI_{uv} \tag{2.24}$$

where $\mathcal{N}_v^r \subseteq \mathcal{N}_v$ is the set of risky neighboring vehicles of v and can be computed as follows -

$$\mathcal{N}_v^r = \{u | \mathbf{I}(\tau_{p,u} \geq \tau_{p,th}), u \in \mathcal{N}_v\} \tag{2.25}$$

Finally the system TAoI across all the vehicle pairs is -

$$TAoI = \frac{1}{\mathcal{N}(\mathcal{N}-1)} \sum_{v \in \mathcal{N}} \sum_{u \in \mathcal{N}_v^r} TAoI_{uv} \tag{2.26}$$

In the example presented in Sec. 2.6.2, we briefly discuss how TAoI based rate control strategy will always reduce the TE of each neighboring vehicle (whether risky or not), and thus, is a promising metric for improved on-road safety.

- The value of $TAoI_{vu}(t)$ at vehicle u is 0 because u identifies itself a non-risky vehicle, i.e., $\mathbf{I}(\tau_{p,u} \geq \tau_{p,th}) = 0$ for $\tau_{p,th} > 0$. Therefore it does not need to transmit BSMs frequently (or at lower broadcast interval Δ_u) to reduce its $TAoI_{vu}$. This doesn't harm u's safety because once v receives u's BSM, then v learns u's non-riskiness behavior; and the obtained u's information (i.e., location, speed, and heading) contained in BSM will allow v to track it accurately. Here, u only needs to broadcast fresh BSM only once unless it changes its mobility behavior.
- At vehicle v, $TAoI_{uv}(t) = AoI_{uv}(t)$ because v identifies itself as a risky vehicle, i.e., $\mathbf{I}(\tau_{p,v} \geq \tau_{p,th}) = 1$. Here, v must broadcast its BSM as frequently as possible to minimize its $TAoI_{uv}$ (or AoI_{uv}). This will reduce $\tau_{e,uv}$ and ensures v's safety. Note that since u does not need to transmit unless its mobility behavior is changed, v can utilize the additional channel resources to transmit more frequently and thus, further reducing $TAoI_{uv}$ (or AoI_{uv}), and improving the overall on-road safety of the network.

To summarize, TAoI metric harnesses the capability of tracking or estimation (using linear extrapolation as per SAE J2945) and enables a non-risky vehicle to broadcast its BSM with much lower priority, i.e, at higher inter broadcast intervals Δ, and a risky vehicle with higher priority, i.e., at much lower Δ. This will greatly reduce the TE corresponding to risky vehicles without compromising the TE of the non-risky vehicles, and thus improve the overall safety of V2V networks.

Problem Overview. The problem of designing an optimal rate control strategy that minimizes system TAoI can be formulated as an Integer Linear Programming (ILP) problem.

At each time $t \in T$, let $C(t)$ denote the total channel capacity in V2V networks and $\mathcal{N}_v^r(t)$ denote the list of risky vehicles in the neighborhood of any vehicle v. Note, $\mathcal{N}_v^r(t)$ does not include v's non-risky neighbors, for which $\mathbf{I}(\tau_{p,u} \geq \tau_{p,th}) = 0$ where $u \in \mathcal{N}_v$. Finally, let $r_v(t)$ denote the rate at which v's broadcasts BSMs, and it can be calculated as $r_v(t) = \frac{1}{\Delta_v(t)}$. Here $\Delta_v(t)$ is the inter broadcast interval of v at time t. Since $\Delta \in [\Delta_{min}, \Delta_{max}]$, $r_v \in [r_{min}, r_{max}]$.

Objective function. The objective is to minimize the overall system TAoI across all vehicles over the entire time period.

$$\min_{r_v(t)} \sum_{t \in T} \frac{1}{\mathcal{N}(\mathcal{N}-1)} \sum_{v \in \mathcal{N}} \sum_{u \in \mathcal{N}_v^r} TAoI_{uv}(t) \tag{2.27}$$

$$\text{subject to} \sum_{u \in (\mathcal{N}_v \cup v)} r_u(t) < C(t), \forall v \in \mathcal{N}, t \in T \tag{2.28}$$

$$r_{min} \leq r_v(t) \leq r_{max}, \forall v \in \mathcal{N}, t \in T \tag{2.29}$$

Constraints. Eq. 2.28 constrains that the sum total of broadcast rate of all vehicles in the vicinity must be less than the total channel capacity. Eq. 2.29 restricts the broadcast rate between the minimum and maximum allowable rates.

Intuitively, the above ILP formulation will provide a centralized optimal solution to the TAoI minimization problem. However, it is highly impractical in our considered 802.11p based V2V networks because of *lack of global information.* The exact calculation of system TAoI at any given time t, given in Eq. 2.27, requires knowledge of TE $\tau_{e,uv}$ for all vehicle pairs in the network. However, each vehicle v can only estimate TE $\tau_{e,uv}$ for its neighboring vehicles based on the BSMs it receives from those vehicles. Furthermore, it assumes the *knowledge of vehicle's mobility and channel information at future time instants*, which is largely impossible to obtain in any time-varying network, including V2V networks.

Thus, we propose a novel TAoI based rate control algorithm that attempts to minimize system TAoI of V2V networks in a decentralized manner, as shown in Sec. 2.8.

2.8 TAoI-based Rate Control Algorithm

This section presents the decentralized TAoI rate control algorithm, which allows each vehicle to determine its broadcast time interval Δ (where $\Delta \in [\Delta_{min}, \Delta_{max}]$) such that the locally measured average TAoI is minimized. Our algorithm is run at each vehicle at every measurement interval t_{MI}. However, note that t_{MI} does not need to be synchronized across all the vehicles in the network.

The algorithm operates in two steps – first, each vehicle identifies itself as a risky or non-risky vehicle by computing its self TE, and shares this information with the neighboring vehicles. Second, based on this knowledge of the risky and non-risky vehicles in their neighborhood,

each vehicle adapts its broadcast interval Δ so as to minimize the locally measured average TAoI. The details of the algorithm are as follows:

Step 1: Vehicle Self-Risk Assessment. As depicted in Algorithm 1, each vehicle v utilizes Eq. 2.12 and calculates its self-TE $\tau_{p,v}$, based on its actual and self-estimated location every t_{MI} (See line 1). Depending upon on the self-TE value, v identifies itself as a non-risky vehicle (i.e., $\mathbf{I}(\tau_{p,v} \geq \tau_{p,th}) = 0$) or a risky vehicle (i.e., $\mathbf{I}(\tau_{p,v} \geq \tau_{p,th}) = 1$), and sets a *self riskiness* flag respectively as 0 or 1. See lines (2 - 7). Following this, v piggybacks this information in all its BSMs broadcasts in line 8 so that the neighboring vehicles are aware of v's self riskiness behavior. As the vehicle's riskiness (risky or non-risky) behavior may change over time, as also previously discussed, the entire algorithm (including this step) is (re) run at the beginning of every t_{MI}.

Algorithm 1 Self Assessment of riskiness at node v

Input: At vehicle v under consideration at time t - actual location $(x_v(t), y_v(t))$, self-estimated location $(\bar{x}_v(t), \bar{y}_v(t))$

Output: Set the flag for riskiness as 1 or 0.

1: Calculate the Self Tracking Error $\tau_{p,v}$ using $(x_v(t), y_v(t))$ and $(\bar{x}_v(t), \bar{y}_v(t))$ based on Eq .2.12.
2: **if** $\mathbf{I}(\tau_{p,v} \geq \tau_{p,th}) == 1$ **then**
3: Mark v as a risky vehicle
4: Set *riskiness flag* $= 1$
5: **else**
6: Mark v as a non-risky vehicle
7: Set *riskiness flag* $= 0$
8: Piggyback the *riskiness flag* in all the BSMs broadcast between current t_{MI} and next t_{MI}.

Step 2: Broadcast Rate Adaptation. In this second step, our algorithm iteratively minimizes the locally measured average $TAoI_v$ at each vehicle v by adapting the broadcast interval, i.e., Δ_v at each t_{MI} through one of the following three possible actions[4] – (i) **DECR** - decrease broadcast interval, (ii) **INCR** - increase broadcast interval, and (iii) **SAME** - maintain broadcast interval. The pseudocode for this step is presented in Algorithm 2.

As shown in lines 2-3, each vehicle v first checks for channel congestion, and the congestion criteria is adopted from Kaul et al. [63]. If the channel is congested, v will unequivocally increase its broadcast interval Δ_v to avoid congesting it further. Then lines 4-5 check if v is non-risky, for which Δ_v is unchanged. Otherwise v is a risky vehicle (line 6), and the special case where v has no risky neighbors is handled in lines 7-8 : v will reduce its Δ_v as a risky

[4]Note that the concept of utilizing these actions for rate adaption in inspired by the AoI rate control algorithm proposed in [63].

Algorithm 2 TAoI-based Rate Control at vehicle v

Input: Average of the broadcast intervals of all neighbor vehicles Δ_{avg}, $TAoI_v$ of the previous t_{MI} denoted by $TAoI'_v$, action taken in the previous t_{MI} Ω, broadcast interval change factor β, previous broadcast interval Δ'_v.

Output: New broadcast interval Δ_v for node v

Calculate AoI_v and $TAoI_v$ for the current t_{MI} using Eq. 2.18 and Eq. 2.24 respectively.
if $AoI_v > 2\Delta_{avg}$ **then** ▷ if channel is congested
    $\boldsymbol{\alpha_v}$ = INCR
if $\mathbf{I}(\tau_{p,v} \geq \tau_{th}) == 0$ **then** ▷ If v is non-risky
    $\boldsymbol{\alpha_v}$ = SAME
else if $\mathbf{I}(\tau_{p,v} \geq \tau_{th}) == 1$ **then** ▷ If v is risky
    if $|\mathcal{N}_v^r| == 0$ **then** ▷ no risky neighbors
        $\boldsymbol{\alpha_v}$ = DECR
    else if $TAoI_v < TAoI'_v$ **then** ▷ if $TAoI_v$ has improved
        new action $\boldsymbol{\alpha_v} = \boldsymbol{\Omega}$
    else if $TAoI_v > TAoI'_v$ **then** ▷ if $TAoI_v$ has degraded
        new action $\boldsymbol{\alpha_v} = \boldsymbol{\Omega}^c$
    else if $TAoI_v == TAoI'_v$ **then** ▷ no change in $TAoI_v$
        new action $\boldsymbol{\alpha_v}$ = SAME
if $\boldsymbol{\alpha_v}$ == INCR **then**
    $\Delta_v = \beta\Delta'_v$
else if $\boldsymbol{\alpha_v}$ == DECR **then**
    $\Delta_v = \frac{\Delta'_v}{\beta}$
else if $\boldsymbol{\alpha_v}$ == SAME **then**
    $\Delta_v = \Delta'_v$
return Δ_v

vehicle should transmit its BSMs frequently. The lines 9 - 12 forms the core of the algorithm where $TAoI_v$ for the current and previous t_{MI} are compared. If $TAoI_v$ has improved based on the previous action, the previous action is repeated. Otherwise, the complimentary of the previous action is selected, where INCR and DECR are compliments of each other. If there is no change in $TAoI_v$, action chosen is SAME as shown in lines 13 - 14. Finally, lines 15-20 are used to calculate the new broadcast interval based on the action selected in the above steps. The factor by which the intervals change if the action was INCR or DECR is given by β. The new broadcast interval returned is maintained until the end of current t_{MI}.

Note there are three controlling parameters in the TAoI based rate control algorithm, as discussed below.

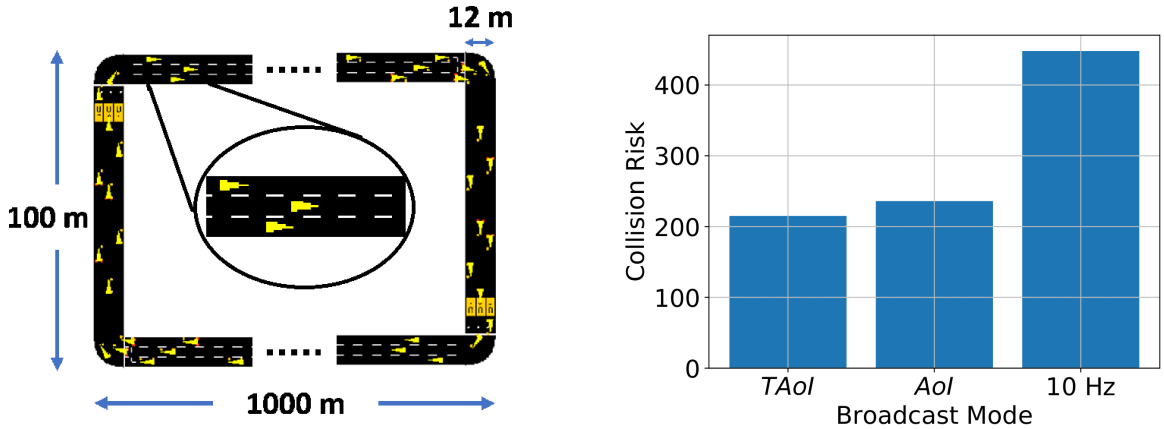
- *Interval change factor (β).* β is used to calculate the new broadcast interval from the previous broadcast interval. When intervals are increased (action is **INCR**), the new interval is obtained by multiplying the old interval with β and vice-versa. We set β=1.1 as per [63].

- *Measurement Interval (t_{MI}).* t_{MI} is set at 1s. It can be higher or lower depending on how frequently the network changes. Even though t_{MI}=1s is a low value for evaluating TAoI in a network as a vehicle may not have received enough BSMs to get a correct estimate of its local TAoI, smaller values of t_{MI} helps the network to minimize AoI (and hence TAoI) faster [63].

- *Self TE threshold $\tau_{p,th}$.* As per the recommendations made in the SAE-J2945 [18], we set $\tau_{p,th}$ = 0.5m. However for practical considerations, this can be adjusted to reflect how accurate the location information needs to be, or in other words, what is the threshold for being classified as a risky vehicle. For $\tau_{p,th} = 0$, $TAoI = AoI$.

The calculation of the optimal β and t_{MI} are beyond the scope of this study and will investigated as a part of our future work.

Time Complexity. Intuitively enough, Algorithm 1 has a running time complexity of $O(1)$. The time complexity of the Algorithm 2 is $O(\mathcal{N})$. This is because the algorithm computes TAoI at vehicle v for each of the neighboring vehicles, which may be the total number of vehicles in the worst case scenario. Thus, the total time complexity of TAoI based rate control algorithm is $O(\mathcal{N}+1) = O(\mathcal{N})$.



(a) Road Layout with a zoomed in picture showing cars moving in 3 lanes (b) Collision Risk for different broadcast modes

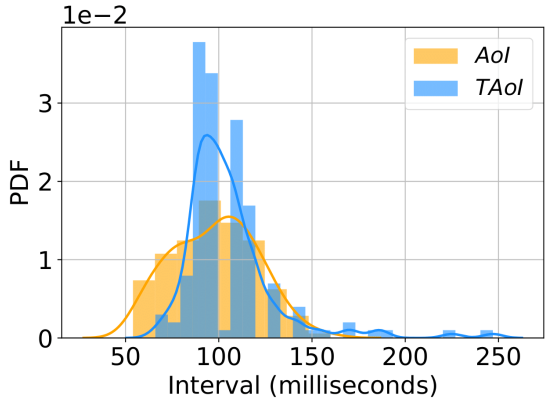
Figure 2.10: Scenario for 150 vehicles

Note the V2V network is a time-varying network where the neighborhood observed by each vehicle keeps changing over time. Because of this, the algorithm will keep changing its Δ_v for each vehicle v based on the new value of local $TAoI_v$. This means that there may not be a broadcast interval that is safety-optimal at all times. Hence we do not pursue convergence analysis.

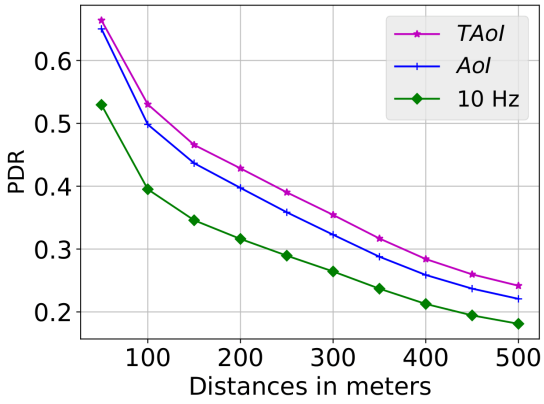
2.9 Performance Evaluation

This section evaluates the TAoI based rate protocol against the baseline AoI based rate protocol [63] and standard 10 Hz.

Simulation Setting The 802.11p V2V network is simulated using network simulator-3 (ns-3) [15]; and SUMO traffic simulation [16] package is used to model the vehicle's mobility as it provides realistic mobility traces for V2V networks. A small part of the road simulated in SUMO with the vehicles in it is shown in Fig. 2.10a where the yellow triangles represent the vehicles. Unless otherwise stated, all the experiments are performed with 150 vehicles moving in a 3-lane rectangular road of 1000m $\times$ 100m. The total simulation time is 100 seconds. Table 2.4 lists other important parameters.



(a) Density plot showing distribution of broadcast intervals



(b) Variation of PDR with distance

Figure 2.11: Rates and PDR for 150 vehicles

Performance metrics. For the comparative analysis of proposed TAoI rate control algorithm against the baseline AoI rate control protocol and standard 10 Hz, we consider the following two key performance metrics.

- **Collision Risk** - It measures the overall on-road safety of V2V networks, and is defined as the number of instances between each pair of vehicles in which δTTC exceeded δTTC_{thres}. Revisit Sec. 2.5 for details.
- **Packet Delivery Ratio (PDR)** - It is the probability that all vehicles within the range of transmitting vehicle, successfully receives the transmitted BSM. It is defined as $PDR_v = \frac{PR_v}{PD_v}$ where PD_v is number of BSMs sent by vehicle v and PR_v is the number of BSMs sent by v received by neighboring vehicles $u \in \mathcal{N}_v$.

Experimental Results Before we present the results, we show the broadcast intervals selected by four different vehicles. As seen in Fig. 2.12a, the first 2 vehicles are risky as their self-TE are (0.51m, 0.92m) greater than $\tau_{p,th}$. So TAoI prioritizes them by assigning them lower intervals compared to AoI protocol as seen in 2.12b. The opposite case can be seen for vehicles 3 and 4 which are non-risky, due to which the proposed TAoI protocol allocates it higher intervals compared to AoI rate protocol.

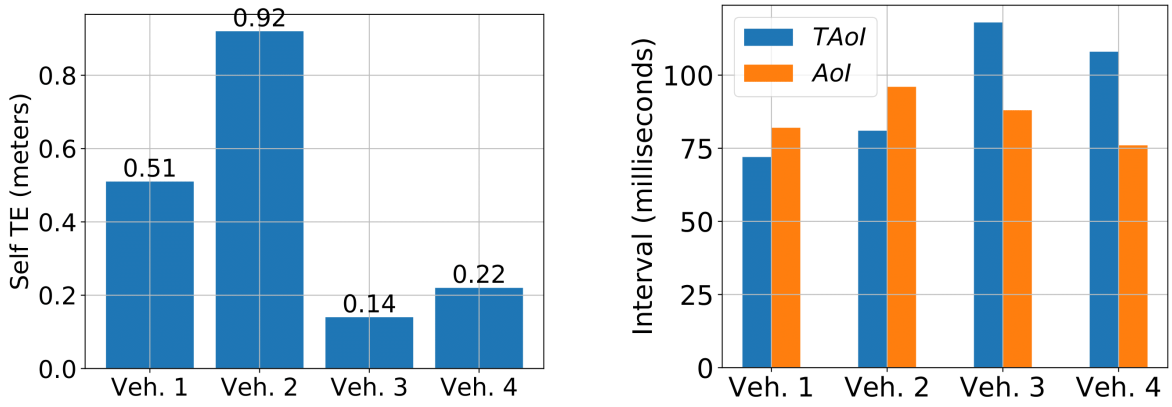
The collision risk, distribution of the broadcast intervals and PDR for 150 vehicles are shown in Fig. 2.10b, 2.11a and 2.11b. From Fig. 2.10b, it can be seen that both AoI and

Table 2.4: Simulation Parameters

Parameters	**Value**
Number of vehicles ($\mathcal{N}$)	100 - 250
Number of Lanes (m)	3
Lane Width	4 m
BSM size	1000 bytes
Transmission Power	20 dBm
Data Rate	6 Mbps
Loss Model	Log Distance Propagation Loss [107]
Path Loss exponent	$\gamma = 3$
Fading	Nakagami-m [107]
Channel Frequency	5.9 GHz
Channel Bandwidth	10 MHz
Max velocity (s_{max})	25 m/s
Antennas/Spatial Streams	1/1
$\tau_{p,th}$	0.5 m
t_{MI}	1 s
β	1.1

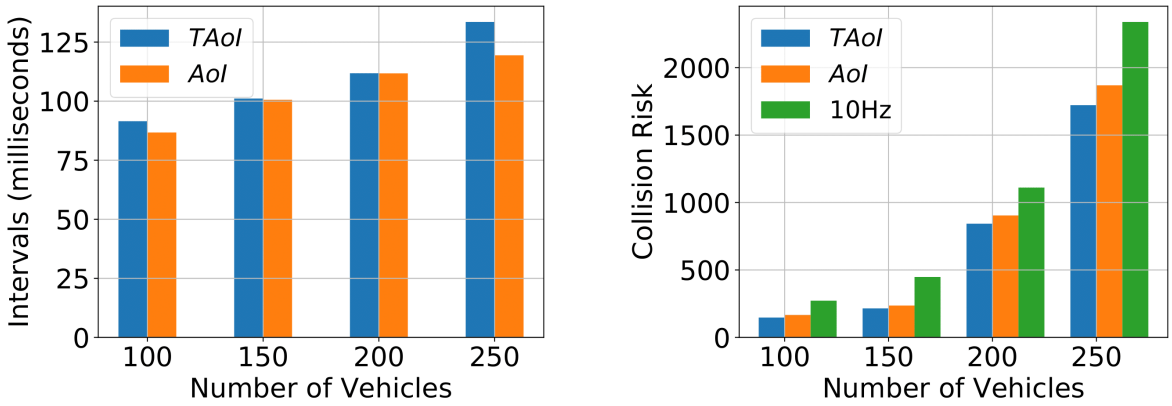
TAoI improves safety compared to the default 10Hz broadcasting, which points to an strong relation between safety performance and improvement in AoI. Regarding the comparison between AoI and TAoI, the improvement in safety as as result of TAoI prioritizing the risky vehicles can be seen in Fig. 2.10b. This shows that non-risky vehicles do not pose much of a threat in V2V networks. To show the different broadcast intervals for different vehicles, the density plot of the distribution of the broadcast intervals is then shown in Fig. 2.11a - it can be seen that both the protocols have a majority of their vehicles broadcasting around similar broadcast intervals. Due to the presence of a spread control factor[5] in [63], the intervals for AoI protocol form a roughly symmetric distribution around the average interval. But with TAoI protocol, as each vehicle can independently select any interval without any spread control, the intervals are spread out with a few vehicles transmitting at very high intervals. Additionally the intervals are higher in our algorithm as vehicles increase their interval as soon as congestion is detected, which is not done in the case of [63]. As a result of the higher intervals in TAoI, channel congestion reduces and this results in a better PDR for TAoI, as shown in Fig. 2.11b. Hence the results show the benefit of a TAoI rate protocol as compared to plain AoI.

[5]As system AoI was shown to be minimized only when each vehicle broadcasts at the same interval in [63], spread control is a necessary operation in system AoI minimization presented there.



(a) Self TE as measured by 4 vehicles (b) Broadcast interval of the 4 vehicles

Figure 2.12: Difference between rates selected by predictable and unpredictable vehicles



(a) Broadcast Interval at different densities (b) Collision Risk at different densities

Figure 2.13: Scalability of the rate protocols

Scalability Analysis To see how the collision risk of the algorithm scales with the number of vehicles, results for 100, 150, 200 and 250 vehicles are shown. Their average broadcast intervals along with the collision risk are shown in Fig. 2.13a and Fig. 2.13b respectively. For 100 vehicles, both the algorithms have an average interval of lower than 100ms, with average interval around 100ms for 150 vehicles and an average interval of more than 100ms for both 200 and 250 vehicles. This is consistent with the observation in [35] where it has been shown that lower densities have lower average intervals that minimize the AoI and vice versa. It is seen that the same relationship between the interval and density holds for TAoI.

It is important to note from Fig. 2.13a and 2.13b that TAoI consistently improves collision

risk across different vehicle densities, while keeping the average broadcast intervals higher or equal to the intervals in AoI rate protocol. The improvement in collision risk ranges around 8%-12% and 24%-40% compared to the AoI rate protocol and default 10Hz broadcasting respectively. This result comprehensively proves that TAoI is a much better metric for controlling the broadcast interval - it utilizes the channel resources much efficiently as compared to AoI rate protocol which is evident from the better safety performance while maintaining the broadcast intervals at same or higher levels compared to AoI rate protocol.

2.10 Conclusion

In this chapter, we showed that AoI based rate control protocol does not always improve on-road safety in 802.11p V2V networks. To address this, we proposed a novel metric, termed, *Trackability-aware* Age of Information, termed TAoI, that jointly takes into account the AoI and self risk assessment of vehicles, measured as self tracking error. Following this, we propose a decentralized TAoI based rate protocol for V2V networks that attempts to minimize locally measured TAoI at each vehicle, which in turn, improve the on-road safety of the V2V network. Our extensive experiments based on realistic SUMO traffic traces on top of ns-3 simulator demonstrate that TAoI rate control protocol greatly improves on-road safety of V2V networks compared to that of AoI rate control and standard 10 Hz rate, in all considered V2V scenarios.

Chapter 3

Learning-based Age of Information-minimizing Scheduling in UAV-relayed IoT Networks

3.1 Introduction

Unmanned Aerial Vehicles (UAVs), as fixed aerial base stations, are being increasingly used to collect time-sensitive information (or status updates) from Internet of Things (IoT) devices and forward them to the nearest Terrestrial Base Station (TBS). We refer to such communication networks as *UAV-relayed IoT networks*. UAV-relayed IoT networks are a promising solution in several scenarios – (i) extending wireless services to IoT devices in remote smart agricultural farms where there is no direct wireless coverage of a TBS [85], (ii) offloading data processing task of a UAV to the TBS equipped with an edge server [75] etc. An example is the smart city plans in Las Vegas where multiple UAVs are being deployed as "advanced contextual and situational awareness modules" to monitor and assess the data captured by IoT devices [104].

A critical requirement in UAV-relayed IoT networks (and general IoT networks) is that the information received at the TBS from the IoT devices is timely or *fresh* [38]. To measure the level of information freshness, the metric called "Age of Information" (AoI) was conceived [61]. It has since attracted significant interest from the research community and is under active investigation. See a survey on AoI [122] and an online bibliography [17]. AoI is defined as the time elapsed since the most recent information was generated. AoI is fundamentally different from traditional metrics such as, latency, at lower network layers (transport, network or link layer), which only focuses on the elapsed time for delivering information between two end nodes in a network.

There has been active research on designing schedulers to minimize AoI in UAV-assisted IoT networks [52], general IoT networks [118], vehicular networks, and other time-sensitive control applications [36]. However, existing research on AoI scheduling (including UAV-assisted IoT networks) has been largely limited to single-hop wireless networks, which is not applicable to our UAV-relayed IoT network scenario. This is because our considered UAV-relayed IoT network involves two hops – (i) [hop 1] - IoT device to UAV, and (ii) [hop 2] - UAV to the TBS. Scheduling for such two-hop network differ from single-hop networks as it involves two time-ordered steps – (i) *sampling* of IoT devices by the UAVs, followed by the (ii) *updating* of these sampled information packets (from UAV) to the TBS. Therefore, designing scheduling policies for such two-hop UAV-relayed IoT networks becomes a non-trivial problem and is the focus of this work. Further, existing AoI research make a number of assumptions – (i) *non-lossy channel conditions* - In reality, wireless channels are lossy and fluctuates rapidly which must be taken into account in designing schedulers (ii) *generate-at-will* traffic generation model at IoT devices – IoT devices usually follow periodic traffic generation model, where an IoT device generates sampling updates at certain fixed periodic time intervals, which may be unknown apriori [25].

In this chapter, we investigate designing optimal scheduling policies for two-hop UAV-relayed IoT networks under real-world conditions. Specifically, we consider the following: (i) we consider lossy channel conditions for both hops i.e., (IoT devices to UAV) and (UAV to the TBS), (ii) we allow each IoT device to generate packets at periodic traffic generation models, which may be unknown to the scheduler.

The main contributions of this work are the following.

- This chapter investigates designing AoI scheduling policies for two-hop UAV-relayed IoT networks, which involves two time-ordered steps – (i) *sampling* of information packets from IoT devices to the UAV relay, and (ii) *updating* of sampled information packets (corresponding to each IoT device) from UAV relay to the TBS.
- We first present a Maximal AoI First - Maximal AoI Difference (MAF-MAD) scheduling policy. Specifically, MAF-MAD scheduler prioritizes sampling of an IoT device to the UAV relay which has highest AoI at the UAV relay, and similarly, updates the sampled information packet (at UAV relay) of the device with highest AoI difference between the UAV and the TBS. We show that MAF-MAD is the optimal scheduler under non-lossy channel conditions and generate-at-will traffic generation models at

the IoT devices.

- Next, we propose an AoI minimizing scheduling policy, based on Deep Q Network (DQN), which we call, DQN based scheduler. We show that DQN based scheduler significantly outperforms MAF-MAD scheduler when the channels are lossy and IoT devices follow periodic traffic generation models, esp. in case of small networks (with 10's IoT devices). This is because DQN is able to learn the characteristics of the network, i.e., channel fluctuations and traffic generation models. However, it does not scale well with network size mainly due to large action space (as we shall see later in the chapter).

- Our simulation results show that both MAF-MAD and DQN based schedulers outperform all considered baseline schedulers, namely, Maximal AoI First (MAF), Round Robin (RR) and Random schedulers employed at both hops under all considered simulation scenarios.

The chapter is arranged as follows: Sec. 3.2 discusses the related works and Sec. 3.3 provides the system model for our work. In Sec. 3.4, we explain the different schedulers. The results for the schedulers with the DQN-based scheduler is shown in Sec.3.5. The disadvantages of the DQN-based scheduler is discussed in Sec.3.6 and the results with the PPO-based scheduler is in Sec. 3.7. The chapter is concluded in Sec. 3.8.

3.2 Related Work

As the availability and ease of deployment of UAVs continue to improve, its role in IoT networks under the context of information freshness has seen great interest in recent years. In [19], the authors study a scenario where a single battery constrained UAV monitors multiple ground nodes. A DL based approach was used to develop a joint trajectory and scheduling design that minimizes AoI at the UAV. The work in [22] considers a similar setting where a UAV has to visit multiple sensor nodes under battery constraints. Similarly, a framework for finding the trajectory that minimizes the average AoI and maximum AoI is presented in [73]. A cellular Internet of UAVs is considered in [53] where a distributed trajectory selection framework is proposed to minimize AoI. Note all these works (along with

[54, 123, 126, 127]) focus mainly on determining the optimal trajectory of the UAVs with the scheduling of packets being only over one hop from IoT devices to the UAV.

Since our UAV-relayed IoT network employ UAV as relays between IoT devices and the BS, we also discuss the related works on AoI for relay networks. Authors in [32] consider a scenario with a single node transmitting to multiple nodes via relays. The source waits for an acknowledgment before transmitting the next sample, and the best waiting time is obtained in closed form. Reference [24] proposes optimal AoI minimizing online and offline policies where a single energy constrained source transmits its information to a single receiver via an energy constrained relay. In [86], a discrete-time stochastic hybrid system analysis is done, first under one relay and a direct link, and then with two relays and no direct link between a single transmitter and a single receiver. Our work is closest to [106] where a relay is used to sample and forward the information of multiple IoT devices to multiple destinations. However, it considers a generate-at-will traffic model for traffic generation at the IoT devices with only a single UAV acting as the relay.

3.3 System Model and Problem Formulation

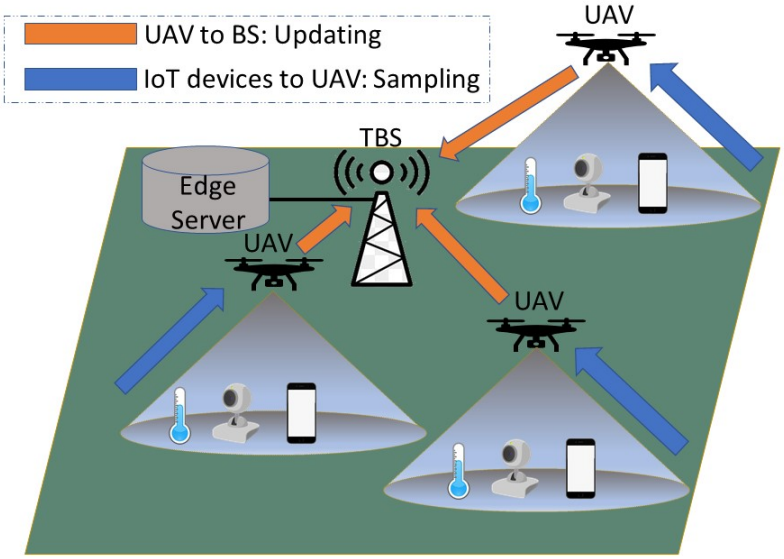


Figure 3.1: UAV-relayed IoT networks system model

3.3.1 Network Model

As shown in Figure 3.1, we consider a UAV-relayed IoT network with a terrestrial base station (TBS), M IoT devices and N UAVs. The TBS is placed at a (random) fixed 2D location in a certain geographical region. The TBS is equipped with edge server, where the collected time-sensitive information data are processed. Similarly, M IoT devices are distributed randomly in the considered region. Since IoT devices are usually energy-constrained and have limited wireless range (most IoT devices have range less than 100 meters [10]), the IoT devices are outside the wireless coverage of the nearest TBS, such as, in case of rural agricultural smart farms, where the IoT devices are far away from the nearest TBS. Therefore N UAVs are deployed as fixed aerial base stations (ABS) in the geographical region to provide end-to-end wireless coverage between the nearest TBS and all M IoT devices. Each UAV $n \in N$ is deployed in the region of interest such that

(i) UAV n lies within the wireless coverage of the TBS and, provides *backhaul links* between itself and TBS, and

(ii) UAV n provides wireless coverage as *access links* to m_n subset of IoT devices, where $m_n \subseteq M$.

Even though a certain IoT device may fall within the wireless coverage of two or more UAVs, we assume that each IoT device is associated to a unique UAV based on some association policy (e.g., max power association policy [99]), and thus, the subset of IoT devices, m_n, is unique to each UAV, and $M = \cup_{n \in N} m_n$. We assume that there is neither intra-cell interference (i.e., UAV and it's associated IoT devices) nor inter-cell interference (i.e., UAV to UAV). This is realistically possible by employing non-overlapping orthogonal channels for both access links [98] and backhaul links [72].

Let K denote the number of non-overlapping orthogonal channels for communication between the TBS and all N UAVs. Similarly, let L_n denote the total number of channels available between a certain UAV n and it's associated IoT devices. For easiness of the presentation, we consider a fixed $L = L_n, (\forall n \in N)$ channels for access links between a certain UAV and it's associated IoT devices. However, our model can be easily extended to unique L_n channels for access links between a UAV and its associated IoT devices.

Next we discuss different traffic generation models at each IoT device and channel conditions considered in our model. Assume T is the total observation duration and it is divided into

equal-length time slots denoted by t.

Traffic generation at the IoT devices. We consider two different types of traffic generation models at each IoT device m – (i) *generate-at-will*: m generates new packet at each time slot t [19], and (ii) *periodic packet generation*: Here, m generates information packets at a fixed time interval denoted by p_m. It means, m generates a new packet at time slots $p_m t, 2p_m t, \ldots, \lfloor \frac{T}{p_m} \rfloor t$. Note that though p_m is a fixed value for a certain IoT device m, it may be unique to each IoT device, i.e., $p_m \neq p_{m'}$, where $m, m' \in M$. The scheduler doesn't have any prior information on the periodicity of traffic generation at a certain IoT device.

Channel Conditions: Because of the fluctuations in the wireless channel, some packets in both the two hops are lost in transmission. It is referred to as link outage and occurs whenever the transmission rate exceeds the channel capacity [116]. Similar to [106], we capture the link outage by considering that the link between device m and the associated UAV n_m has a non-zero probability of dropping the transmitted packets, which we refer to as, *Sample Loss Probability* and is denoted by $l_{s,m} \in (0, 1)$. Similarly, $l_{u,m} \in (0, 1)$ denotes the *Update Loss Probability*, which refers to the packet loss probability between UAV n_m and the TBS.

Scheduling: At each slot t, our considered UAV-relayed IoT networks involves two simultaneous scheduling steps – (i) *sampling* of IoT devices by the UAVs and (ii) *updating* of sampled information packets from a certain UAV to the TBS. The set of IoT devices sampled by UAV n is denoted by $S_n(t)$ and total devices sampled by all the UAVs is $\mathcal{S}(t) = \{\mathcal{S}_1(t), \mathcal{S}_2(t), \ldots, \mathcal{S}_N(t)\}$. Similarly, updated devices are denoted as $\mathcal{U}(t)$. When a device m is sampled, it transmits its most recent packet to the serving UAV n_m and it will replace any of its older packet at n_m [71]. Then when it is updated, the packet is transmitted from UAV n_m to the TBS. At any slot, each channel can support at most 1 packet.

3.3.2 Age of Information (AoI) at the UAV and TBS

We employ Age of Information (AoI) to measure the freshness of information. In particular, AoI is defined as *time elapsed since the generation time of the most-recent packet received (by the UAV/TBS in our case).* The AoI of IoT device m at UAV n_m is denoted as

$$AoI_m^{UAV}(t) = t - \tau_{s,m} \tag{3.1}$$

where t is the current slot and $\tau_{s,m}$ is the generation time of the *most recently sampled* packet of device m that was received successfully at UAV n_m. Similarly, the AoI of device m at the TBS is

$$AoI_m^{TBS}(t) = t - \tau_{u,m} \tag{3.2}$$

where $\tau_{u,m}$ is the generation time of the *most recently updated* packet of the mth IoT device that was received successfully at the TBS. Based on Eqn. (3.1) and (3.2), AoI increases linearly in time slots of no reception and decreases at reception instants.

The evolution of AoI with time at the UAVs and TBS is described next. At t, if device m was selected for sampling, i.e., $m \in \mathcal{S}(t)$, its AoI at the UAV changes as

$$AoI_m^{UAV}(t+1) = \begin{cases} t + 1 - \tau_{s,m} & \text{with prob. } 1 - l_{s,m} \\ AoI_m^{UAV}(t) + 1 & \text{with prob. } l_{s,m} \end{cases} \tag{3.3}$$

Else if the device m was not sampled, i.e., $m \notin \mathcal{S}(t)$, its AoI at the UAV changes as

$$AoI_m^{UAV}(t+1) = AoI_m^{UAV}(t) + 1 \tag{3.4}$$

Similarly AoI for IoT device m at the TBS when selected for update such that $m \in \mathcal{U}(t)$ and when not selected for update such that $m \notin \mathcal{U}(t)$ are shown in Eqn. (3.5) and (3.6).

$$AoI_m^{TBS}(t+1) = \begin{cases} AoI_m^{UAV}(t) + 1 & \text{with prob. } 1 - l_{u,m} \\ AoI_m^{TBS}(t) + 1 & \text{with prob. } l_{u,m} \end{cases} \tag{3.5}$$

$$AoI_m^{TBS}(t+1) = AoI_m^{TBS}(t) + 1 \tag{3.6}$$

Thus a device's AoI increases at the UAVs and the TBS if-

- it was sampled/updated but the packet was lost in transmission,
- it was not sampled/updated.

As the information is relayed through the UAVs, AoI_m^{UAV} directly impacts AoI_m^{TBS}. Note that the communication between a device and the TBS has a delay of a single slot, due to which information sampled by an UAV at a certain slot cannot be updated to the TBS in the same slot [106].

Finally, the average AoI of all IoT devices at the UAVs and the TBS during the observation interval T is calculated as Eqn. (3.7) and (3.8) respectively

$$AoI^{UAV}(T) = \frac{1}{M}\sum_{t=1}^{T}\sum_{m=1}^{M} AoI_m^{UAV}(t) \tag{3.7}$$

$$AoI^{TBS}(T) = \frac{1}{M}\sum_{t=1}^{T}\sum_{m=1}^{M} AoI_m^{TBS}(t) \tag{3.8}$$

Problem formulation: Our objective to design a scheduler that ensures minimum AoI corresponding to all IoT devices at the TBS. Given $S_n(t)$ and $\mathcal{U}(t)$ respectively denote the set of the IoT devices sampled by each UAV $n \in N$ and updated to the TBS at a certain time slot $t \in T$, we formulate the AoI-aware scheduling problem as the minimization of the average AoI of all IoT devices at the TBS (Eqn. (3.8)) subject to the limited channel constraints, as follows.

$$\begin{aligned} &\min AoI^{TBS}(T) \\ \text{s.t.}\quad &|\mathcal{S}_n(t)| \leq L,\ \forall n \in N \text{ and } t = 1, 2...T && (3.9)\\ &|\mathcal{U}(t)| \leq K,\ t = 1, 2...T && (3.10) \end{aligned}$$

where constraint (3.9) and (3.10) refers to the limited number of channels for the sampling and updating respectively. Table 3.1 summarizes all the notations used in the chapter.

Table 3.1: Notations

Meaning	**Symbol**
Number of IoT devices	M
Number of UAVs	N
IoT devices associated to UAV n	m_n
UAV providing coverage to device m	n_m
Packet generation periodicity of device m	p_m
AoI of device m at UAV	AoI_m^{UAV}
AoI of device m at TBS	AoI_m^{TBS}
Packet loss between device m and UAV n_m	$l_{s,m}$
Packet loss between device m and TBS	$l_{u,m}$
Devices sampled at t by UAV n	$\mathcal{S}_n(t)$
Devices sampled at t=$\{\mathcal{S}_1(t), \mathcal{S}_2(t), ..\mathcal{S}_N(t)\}$	$\mathcal{S}(t)$
Devices updated at t	$\mathcal{U}(t)$
Channels at each UAV	L
Channels at the TBS	K

3.4 Scheduling Policies

We first propose a simple low-complexity centralized , *Maximal AoI First-Maximal AoI Difference* (MAF-MAD) scheduling algorithm to solve the above problem. MAF-MAD runs at the TBS and its underlying idea is to sample users with maximum AoI at the serving UAV, and update IoT devices with maximum AoI difference between the UAV and the TBS. We show that MAF-MAD is an optimal scheduler for UAV-relayed IoT networks under ideal conditions where (i) the channels are non-lossy and (ii) all the IoT devices follow generate-at-will traffic generation models. However, under general conditions (i.e., lossy channels and periodic traffic generation models), the scheduling decisions need to incorporate packet generation instants and channel non-idealities, which is not possible in case of MAF-MAD scheduler. Thus, we propose a centralized learning algorithm based on Deep Q networks (DQN), which we call *DQN-based scheduler*. DQN based scheduler also runs at the TBS and learns the unique network characteristics, and thus, promises to perform well for UAV-relayed IoT networks under realistic conditions.

3.4.1 MAF-MAD Scheduler

The algorithmic details of Maximal AoI First - Maximal AoI Difference (MAF-MAD) scheduler are presented in Algorithm 3. At any time $t \in T$, the inputs to the this algorithm are

AoI of each IoT device $m \in M$ at its serving UAV $AoI_m^{UAV}(t)$ and at the TBS $AoI_m^{TBS}(t)$. For the sampling step in line 3, each UAV n samples L IoT devices with the maximal AoI, $AoI_m^{UAV}(t)$, first (MAF) from the m_n IoT devices associated with it. Note that ideally we should consider AoI difference between the UAV and the IoT device, but this is not feasible as the scheduler doesn't know the traffic generation instants of the IoT devices. However when devices generate packets using generate-at-will traffic model, any device which is selected to be sampled will generate a new packet at that slot due to which $AoI_m^{UAV}(t)$ also becomes the AoI difference between the UAV and the IoT device.

Then for the updating step, devices with *maximal AoI difference* (MAD) between the UAV and TBS are prioritized. This AoI difference is calculated as

$$AoI_m^{diff}(t) = AoI_m^{TBS}(t) - AoI_m^{UAV}(t) \tag{3.11}$$

where $AoI_m^{TBS}(t) \geq AoI_m^{UAV}(t)$. The TBS updates K users out of the total M IoT devices with the highest $AoI_m^{diff}(t)$, as shown in line 5.

Algorithm 3 MAF-MAD scheduler

Input: $AoI_m^{UAV}(t)$ and $AoI_m^{TBS}(t)$ for $m \in M$.
Output: $\mathcal{S}(t)$,$\mathcal{U}(t)$.

1: $\mathcal{S}(t) = \phi$
2: **for** UAV $n = 1, 2, ..N$ **do**:
3: $\quad \mathcal{S}_n(t) = \arg\max_{S \subseteq m_n, |\mathcal{S}_n(t)| \leq L} \{AoI_m^{UAV}(t)\}_{m=1}^{m_n}$
4: $\quad \mathcal{S}(t) = \mathcal{S}(t) \cup \mathcal{S}_n(t)$
5: $\mathcal{U}(t) = \arg\max_{U \subseteq M, |\mathcal{U}(t)| \leq K} \{AoI_m^{diff}(t)\}_{m=1}^{M}$

Theorem 3.1. *Under ideal conditions, i.e., no lossy channels and generate-at-will sampling at IoT devices, MAF-MAD is the optimal scheduling policy for minimizing average AoI at the TBS for status updates generated at each IoT device.*

Proof. As also presented in [106], to minimize the AoI at a destination ($AoI^{TBS}(T)$ in our case), at each slot $t = 1, 2, \ldots, T$ the optimal scheduler needs to

- sample devices whose packet reception at the relay (UAV in our case) will lead to the maximum reduction of average AoI at the relay,

- update devices whose packet reception at the TBS will lead to the maximum reduction of average AoI at TBS.

We first show that sampling as per MAF-MAD offers the maximum reduction in average AoI at all the UAVs.

AoI reduction at the UAVs: Consider a UAV n providing coverage to $|m_n|$ (out of M) IoT devices. At slot t, the average AoI of these IoT devices at UAV n is

$$AoI_n^{UAV}(t) = \frac{1}{M}\sum_{i=1}^{m_n} AoI_i^{UAV}(t) \tag{3.12}$$

and from the result cited above, the optimal scheduler will sample those devices which will lead to maximum reduction of average AoI at n from $AoI_n^{UAV}(t)$ to $AoI_n^{UAV}(t+1)$.

For simplicity, assume L=K=1. Therefore the scheduler needs to select L=1 device to sample. Let device $j \in m_n$ have the largest AoI at n such that

$AoI_j^{UAV}(t) = \max(AoI_1^{UAV}(t), AoI_2^{UAV}(t), \ldots AoI_{m_n}^{UAV}(t))$. From Algorithm 3 step 3, at t the MAF-MAD scheduler will sample the jth device. Then the reduction in the average AoI at n is

$$\begin{aligned}\Delta AoI_n^{UAV}(t+1) &= \frac{1}{M}(AoI_n^{UAV}(t) - AoI_n^{UAV}(t+1)) \\ &= \frac{1}{M}\sum_{i=1}^{m_n}(AoI_i^{UAV}(t) - AoI_i^{UAV}(t+1))\end{aligned} \tag{3.13}$$

For the $|m_n| - 1$ unsampled devices, their $AoI^{UAV}(t+1)$ increases by 1 (Eqn. (3.4)). $AoI^{UAV}(t+1)$ for j becomes $t + 1 - \tau_{s,m}$ as per Eqn. (3.3) under no packet loss. Additionally for generate-at-will, j will generate a packet at t, i.e., $\tau_{s,m} = t$, as it was sampled at t. Therefore Eqn. (3.13) becomes

$$\begin{aligned}\Delta AoI_n^{UAV}(t+1) &= \frac{1}{M}\Big(\sum_{i=1,i\neq j}^{|m_n|}(AoI_i^{UAV}(t) - AoI_i^{UAV}(t+1)) + (AoI_j^{UAV}(t) - AoI_j^{UAV}(t+1))\Big)\\ &= \frac{1}{M}(-(|m_n|-1) + AoI_j^{UAV}(t) - (t+1-t))\\ &= \frac{1}{M}(AoI_j^{UAV}(t) - |m_n|) \end{aligned} \tag{3.14}$$

Now for any other scheduler that selects device j' to sample where $j' \neq j$ and hence $AoI_j^{UAV}(t) > AoI_{j'}^{UAV}(t)$, the reduction in AoI at n will be given by

$$\Delta^{'} AoI_n^{UAV}(t+1) = \frac{1}{M}(AoI_{j'}^{UAV}(t) - |m_n|) \tag{3.15}$$

Now comparing Eqn. (3.14) and (3.15), $\Delta AoI_n^{UAV}(t+1) > \Delta^{'} AoI_n^{UAV}(t+1)$. Because all the UAVs $n \in N$ operate independently, the same result holds for other UAVs too. Hence the MAF-MAD scheduler offers the largest AoI reduction at all the UAVs in the sampling step.

AoI reduction at the TBS: In the updating step, the scheduler needs to select K=1 device to update. From Algorithm 3 step 5, at t the MAF-MAD scheduler will update the kth device such that $AoI_k^{diff}(t) = \max(AoI_1^{diff}(t), AoI_2^{diff}(t), \ldots, AoI_M^{diff}(t))$. After the update, the reduction in average AoI at the TBS is

$$\begin{aligned}\Delta AoI^{TBS}(t+1) &= \frac{1}{M}(AoI^{TBS}(t) - AoI^{TBS}(t+1))\\ &= \frac{1}{M}(\sum_{i=1}^{M}(AoI_i^{BS}(t) - AoI_i^{TBS}(t+1))) \end{aligned} \tag{3.16}$$

For $M-1$ devices that were not updated, their $AoI^{TBS}(t+1)$ increases by 1 (Eqn. (3.6)). $AoI^{TBS}(t+1)$ for the updated device k will change as per Eqn. (3.5) when there is no packet loss. Therefore Eqn. (3.16) becomes

$$\begin{aligned}\Delta AoI^{TBS}(t+1) &= \frac{1}{M}\Big(\sum_{i=1,i\neq k}^{M}(AoI_i^{TBS}(t) - AoI_i^{BS}(t+1)) + (AoI_k^{TBS}(t) - AoI_k^{TBS}(t+1))\Big) \\ &= \frac{1}{M}(-(M-1) + AoI_k^{TBS}(t) - AoI_k^{BS}(t+1)) \\ &= \frac{1}{M}(-M+1+AoI_k^{TBS}(t) - AoI_k^{UAV}(t) - 1) \\ &= \frac{1}{M}(AoI_k^{diff}(t) - M) \end{aligned} \tag{3.17}$$

Now for any other scheduler that selects device k' to sample where $k' \neq k$ and hence $AoI_k^{diff}(t) > AoI_{k'}^{diff}(t)$, the reduction in average AoI at the TBS will be given by

$$\Delta^{'} AoI^{TBS}(t+1) = \frac{1}{M}(AoI_{k'}^{diff}(t) - M) \tag{3.18}$$

Comparing Eqn. (3.17) and (3.18), we see $\Delta AoI^{TBS}(t+1) > \Delta^{'} AoI^{TBS}(t+1)$. Hence MAF-MAD also leads to the maximum reduction in average AoI at the TBS. Therefore when there is no packet loss at any of the sampling and updating steps, and the devices generate traffic based on generate-at-will policy, MAF-MAD is the optimal scheduler for minimizing $AoI^{TBS}(T)$.

□

3.4.2 Deep Q Network (DQN) based scheduler

The requirement of taking the channel non-idealities and traffic generation patterns motivates the use of a reinforcement learning (RL) based scheduler. In RL, an agent is able to take an informed decision on which actions to take after interacting with the environment as the interaction allows it to learn about the environment. A typical RL framework has an agent (scheduler in our case) that takes an *action* (devices to sample and update) after observing the environment *state*, following which it gets a *reward* which is like a feedback on the quality of the action taken. We use a Q-learning algorithm [95] to design a scheduler with the objective of AoI minimization.

In Q-learning, a state-action value function $Q^{\pi}(x(t), a(t))$ is defined as the net reward when the system performs action $a(t)$ while starting from state $x(t)$ and then follows policy π. Q-learning tries to calculate the Q-value function by learning a policy which will maximize the cumulative reward. This is done using the Bellman update rule shown in Eqn. (3.19) below:

$$Q_{t+1}(x(t), a(t)) = Q_t(x(t), a(t)) + \beta[r(t) + \gamma \max_a Q_t(x(t+1), a) - Q_t(x(t), a(t))] \quad (3.19)$$

where $r(t)$ is the reward at slot t. β and γ are the learning rate and the discount factor respectively. Discount factor signifies how important the future rewards are and is set to a value between 0 and 1. As our problem has a terminal state when $t = T$, it is an episodic task and hence γ is set to 1 [19].

The agent approximates the Q-function using Eqn. (3.19) and then takes the action that maximizes the reward. As the agent will not have the correct estimate of the Q-values corresponding to many state-action pairs as a consequence of not visiting them, it needs to explore in addition to exploiting the known values of state-action pairs. This is known as the exploration-exploitation trade-off. An epsilon (ϵ) greedy approach where the agent takes a random action with the objective of exploring the environment with probability ϵ and acts greedily with probability $1 - \epsilon$ is generally used to ensure the agent doesn't get stuck with sub-optimal actions.

While Eqn. (3.19) can be theoretically used for all scenarios, its iterative nature makes it infeasible to be used for large state spaces as it will suffer from very slow convergence in addition to needing a large memory [19]. Moreover this approach cannot be used when there are unobserved states [95] and visiting every state-action pair is not practical. This is where the approximating power of neural networks are put to use as they can extract the important features from the available data points and summarize them in smaller dimensions. Neural networks approximates the Q-value function as $Q(x, a|\theta)$ where θ are its weights, and this approach for Q-learning is known as Deep Q-Networks (DQN) and is based on a Markov Decision Process (MDP) formulation of the problem [84]. The goal is to find the optimal weights θ so that the neural network can approximate the optimal Q-value function as close as possible. However the use of a single neural net may cause instability and therefore two neural networks with the same architecture are used: the *current neural network* with

weights θ and a *target neural network* with weights θ^-. While the current network acts as a function approximator and its weights are updated iteratively after each slot, the target network computes the target Q-value function and its weights are fixed for a while and updated every O slots [84].

At each time step, the *experience* of the agent is stored in *experience replay* which is also called replay memory and has a size R. The experience at time t is denoted as $exp(t) = (x(t), a(t), r(t), x(t+1))$ and provides a summary of the agent's experience at that time. The replay memory is then used to randomly sample a mini-batch of stored experiences $x(t'), a(t'), r(t'), x(t'+1)$ which will be used to train the neural network. The primary benefit of using random samples from replay memory to learn instead of consecutive samples as the agent encounters them is that consecutive experiences have high correlation which may lead to inefficient learning [82]. Therefore at each episode, a mini-batch of B past experiences are fed as an input to the DQN. The loss function is given by

$$L(\theta) = [r(t') + \gamma \max_{a(t')} Q(x(t+1), a(t')|\theta^-) - Q(x(t), a(t)|\theta)]^2 \tag{3.20}$$

where $Q(x(t+1), a(t')|\theta^-)$ is evaluated by the target network and $Q(x(t), a(t)|\theta)$ is evaluated using the current network. The weights are then updated as shown in Eqn. (3.21).

$$\theta = \theta + \alpha(y(t') - Q(x(t'), a(t')|\theta))\nabla_\theta Q(x(t'), a(t')|\theta) \tag{3.21}$$

where $y(t') = r(t') + \max_a Q(x(t'+1, a|\theta^-))$ and ∇_θ denotes the gradient with respect to θ. The state, action and reward in the context of our work is explained next.

State Space

State is the scenario encountered by the scheduler. We define our state as

$$x(t) = (t, \{AoI_m^{UAV}(t)\}_{m=1}^M, \{AoI_m^{TBS}(t)\}_{m=1}^M) \tag{3.22}$$

where t is the ongoing slot. At t=1, $AoI_m^{UAV}(t)$ and $AoI_m^{TBS}(t)$ are initialized to 1. The set

of state space is denoted by $\mathcal{X}$.

Action Space

As described in Sec. 3.3, scheduling involves two simultaneous steps of sampling updating devices at each slot. Therefore the action at each slot t is the same as we do in the DQN-based scheduler–

$$a(t) = (\mathcal{S}(t), \mathcal{U}(t)) \tag{3.23}$$

where $\mathcal{S}(t)$=$\{S_1(t), S_2(t), ..S_N(t)\}$ is the set of all devices sampled by all the UAVs at slot t such that $|S_n(t)| \leq L \; \forall \; n \in N$. Similarly, $\mathcal{U}(t)$ is the set of devices selected to update their packets to the TBS where $|\mathcal{U}(t)| \leq K$. The set of action space is denoted by $\mathcal{A}$.

Reward

As the objective is improving the information freshness at the TBS, the reward is given by the negative of the average AoI of all the devices at the TBS at slot t

$$r(x(t), a(t)) = r(t) = -\sum_{m=1}^{M} AoI_m^{TBS}(t) \tag{3.24}$$

Based on action $a(t)$, the environment transitions from state $x(t)$ to a new state $x(t+1)$ according to the state transition probabilities described in Eqn. (3.3), (3.4), (3.5), (3.6) while resulting in a reward $r(t)$. Thus it is a finite horizon MDP with finite state and action spaces, which makes it suitable for DQN.

The DQN based scheduler is shown in Fig. 3.2 and explained in Algorithm 4. The hyper-parameters and the network are initialized in lines 1 - 2. For each step, a random action is taken with probability ϵ or the action with the highest Q-value is selected with probability 1-ϵ in lines 6 - 8. The selected action is performed and environment transitions to the new state in lines 9-10. Line 11 stores the resulting experience in the replay memory. Then a mini-batch of samples is used to train the network in lines 12-17, after which the weights

of the current network are updated in lines 18. Line 19 updates the target network weight every O slots.

Convergence: Convergence of neural networks are hard to be analytically investigated and is strongly dependent on the set of hyper-parameters used [57]. Selection of hyper-parameters is a challenging task and therefore a reasonable set of hyper-parameters is found by trying out different values. More details on the hyper-parameters used in our case are presented in Sec. 3.5. Similar to [19, 53, 106, 127], we limit our investigation into the convergence to simulations, where the neural network converges under the hyper-parameters used and the results presented for the DQN scheduler are the values obtained after it's convergence.

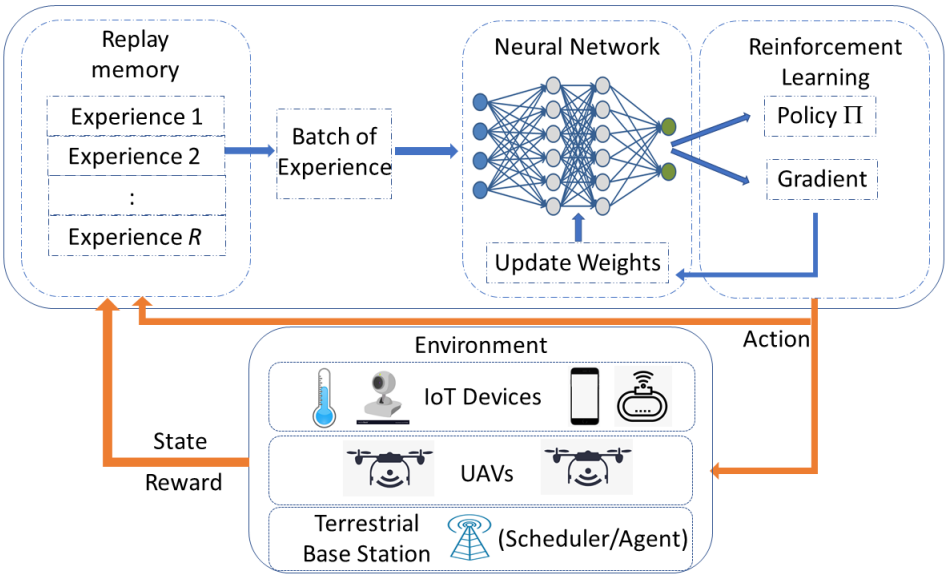


Figure 3.2: Architecture of the Deep Q Network used

3.5 Results

In this section, the results are presented in terms of average AoI at the TBS achieved for MAF-MAD and DQN schedulers, and compare them to three baseline schedulers (See sub-section V.A). We first focus on smaller IoT network setting (with 10's IoT devices) and investigate the efficacy of proposed algorithms against the baseline ones. Furthermore, we also investigate the effect of various control parameters, such as the number of channels, IoT devices and UAVs. Finally, we discuss the comparative analysis of all the schedulers in the case of larger IoT network setting as well.

Algorithm 4 DQN scheduler

Initialize the replay memory R, probability ϵ, current network weights θ, target network weights θ^-. Set episode = 1.
Initialize the current network $Q(s, a|\theta)$ and the target network $Q(s, a|\theta^-)$.
for episode=1:E **do**
    initialize the environment by setting $t = 1$.
    for t=1:T **do**
        Select an action a:
            select a random action $a \in \mathcal{A}$ with probability ϵ,
            otherwise select $a = \arg\max_a Q(x(t), a|\boldsymbol{\theta})$.
        Perform action a.
        Observe the reward $r(t)$ and the new state $x(t+1)$.
        Store experience $\{x(t), a(t), r(t), x(t+1)\}$ in replay memory.
        Sample a random mini-batch B of transitions $x(t'), a(t'), r(t'), x(t'+1)$ from the replay memory.
        Calculate target value:
        if $x(t'+1)$ is a terminal state **then**
            $y(t') = r(t')$,
        else
            $y(t') = r(t') + \max_a Q(x(t'+1, a|\theta^-))$.
        Update θ using the weight update rule as per Eqn. (3.21).
        Update target network by setting $\theta^- = \theta$ every O steps.
        Episode ends if $x(t+1)$ is the terminal state.

3.5.1 Baseline schedulers

- **Maximal AoI First (MAF) scheduler** – devices with highest AoI at both the UAV and TBS are selected for sampling and updating respectively [109]. For sampling, each UAV selects L devices with highest $AoI_m^{UAV}(t)$ out of the m_n devices associated to it. Similarly for updating, K devices out of the total M with highest $AoI_m^{TBS}(t)$ get to update their packets.

- **Round Robin (RR) Scheduler** – under RR, the available channels are assigned in equal and circular fashion among the devices which ensures fairness among the devices [87]. In our version of RR, all the devices are given equal weights unlike weighted RR where the weights play a role in the fairness with higher weights demanding to higher allocation.

- **Random scheduler** – This a baseline scheduler where a random set of devices are

selected to be sampled and updated under the pre-specified channel constraints.

3.5.2 Simulation setting

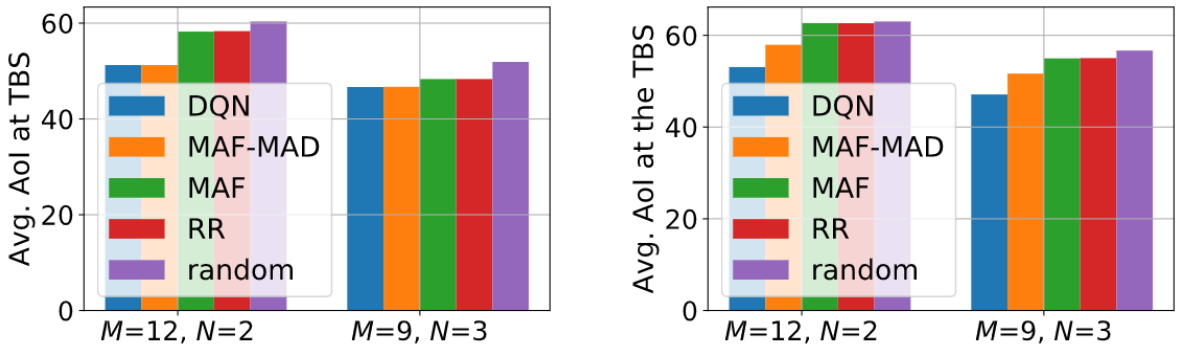
The IoT devices and UAVs are placed randomly [79] in an area of $l = b$ =1000m and observation interval T=10 slots. Simulations for the DQN scheduler are performed using the tf-agents library [9]. The fully connected neural net has two hidden layers and the architecture is $|\mathcal{X}| \times H_1 \times H_2 \times |\mathcal{A}|$. The DQN is implemented on an NVIDIA DGX station with an Intel Xeon E5-2698 v4 CPU and an NVIDIA Tesla V100 GPU (32 GB memory). Data communication between CPU and GPU takes place through a PCIe 3.0 X16 slot. DQN hyper-parameters are shown in in Table 3.2. The results for the other schedulers are averaged over 10^4 runs.

Table 3.2: DQN Parameters

Neurons in hidden layer 1 H_1	1024
Neurons in hidden layer 2 H_1	1024
Relay memory size R	50,000
Mini batch size B	16
Learning rate β	0.001
Learning rate decay rate	0.95
Learning rate decay step	10,000
Discount factor γ	1
Activation function	ReLU
Update steps O	10
Optimizer	Adam
Total episodes E	10^6

3.5.3 Smaller IoT networks

These IoT networks contain 5-10 IoT devices. Such networks are used for monitoring indoors/outdoors particulate matter pollutants where a single type of sensor is able to monitor particulates of various sizes [97]. We use $L = K = 1$ and simulate 2 scenarios: first with $M = 12$ IoT devices equally assigned to $N = 2$ UAVs, and then $M = 9$ IoT devices equally assigned to $N = 3$ UAVs.



(a) ideal conditions (b) general conditions

Figure 3.3: $AoI^{TBS}(T)$ for smaller IoT networks

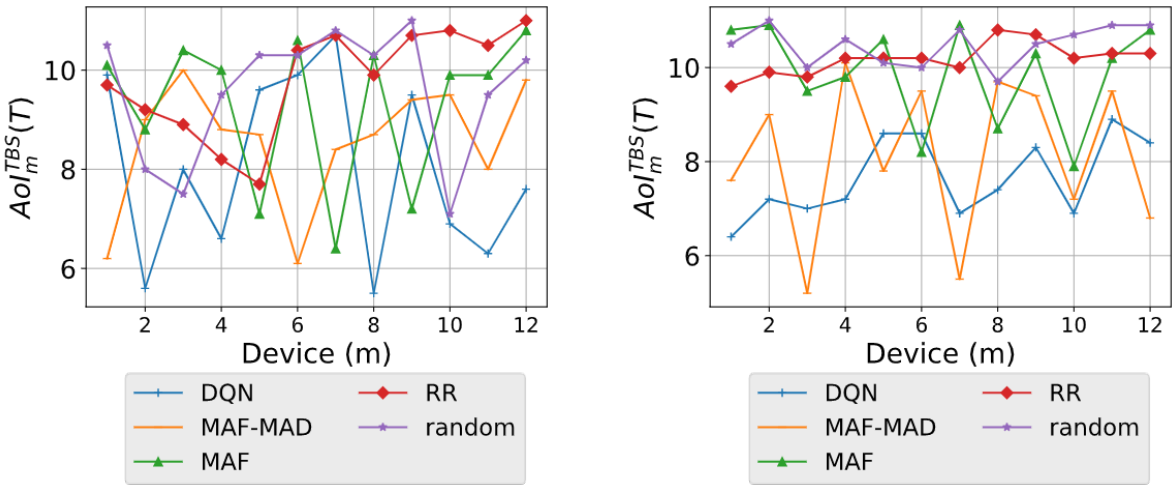
Under Ideal Conditions

Here the devices generate traffic using a generate-at-will model with no packet loss in any of the channels. The results shown in Fig. 3.3a confirm that MAF-MAD is the optimal scheduling policy and outperforms all other scheduling policies. As DQN is able to learn the the optimal scheduling, it's performance also converges to MAF-MAD's performance. MAF and RR have similar performance while random performs the worst. With MAF-MAD and DQN performing similarly, their performance exceeds MAF, RR and random by 5%-10%, 5%-10% and 11%-17% respectively.

Under general conditions

The results are shown in Fig. 3.3b and we see that DQN outperforms MAF-MAD and all other schedulers under these conditions, i.e., generate at will packet generation model and non-lossy channels. Therefore, MAF-MAD scheduler is no longer optimal under these conditions. MAF-MAD is followed by MAF, RR and random. DQN's performance exceeds MAF-MAD, MAF, RR and random by 9%-10%, 16%-18%, 17%-18% and 18%-20% respectively.

To understand why DQN has the best performance, we show the individual AoI of the devices at TBS at $t = T$ for lossy channel with generate-at-will traffic generation in Fig. 3.4a and non-lossy channels with periodic traffic generation in Fig. 3.4b. $N = 2$ UAVs are providing coverage to $M = 12$ IoT devices with the IoT devices equally distributed between the UAVs. For Fig. 3.4a, while the results could be shown for any channel condition, we pre-assign the



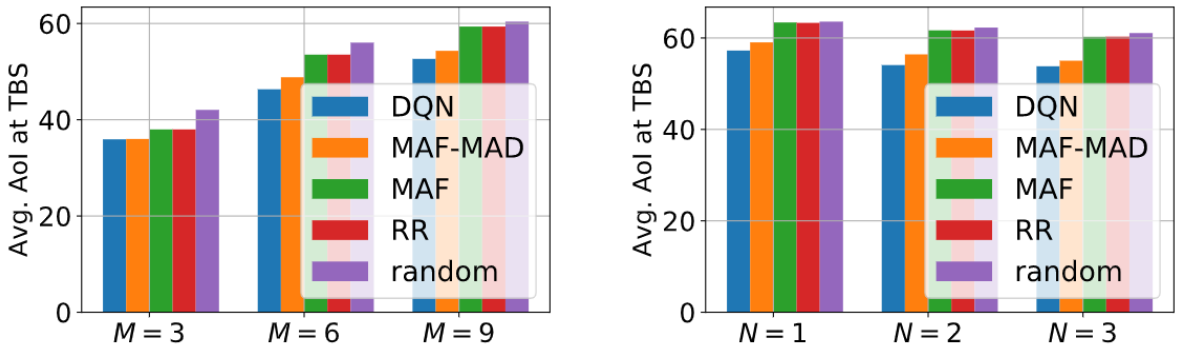
(a) packet loss (b) periodic traffic generation

Figure 3.4: AoI of individual IoT devices

packet loss probabilities $l_{s,m}$ and $l_{u,m}$ for ease of reproducibility [71]. Under lossy channel conditions, focusing on channels that improve packet receptions will minimize average AoI and the DQN based scheduler will learn this. The average AoI per device for DQN, MAF-MAD, MAF, RR and random were - **8.00**, 8.55, 9.29, 9.34, 9.58 respectively and as expected, DQN has the best performance. Similarly in Fig. 3.4b, p_m is selected randomly between 2,3,4 and pre-assigned for the devices. The average AoI for DQN, MAF-MAD, MAF, RR and random were - **7.65**, 8.10, 9.88, 10.18, 10.47 respectively, and it can be seen that the DQN based scheduler has lowest AoI for most of the devices. This means DQN is able to learn the periods of packet generation at the devices and sample them accordingly, which the other schedulers are unable to do.

We now investigate the effect of different parameters on average AoI at the TBS under general conditions.

Variable M (IoT devices) For L=2, K=1, we deploy $N = 3$ UAVs and vary the total number of IoT devices M. The IoT devices are assigned equally among the 3 UAVs. The results are shown in Fig. 3.5a - as M increases, average AoI at the TBS increases too. This is because while M increases, number of channels (L, K) remain same due to which each device now has lesser chances of getting sampled and updated, resulting in increased average AoI at the TBS.



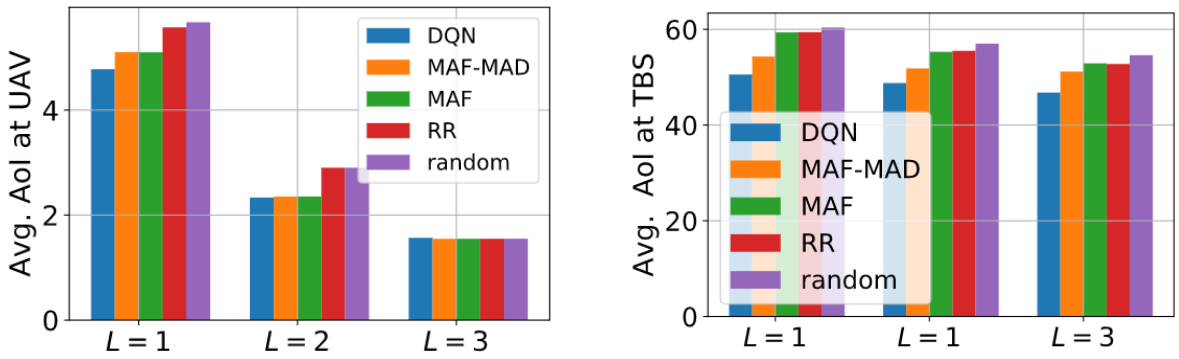
(a) variable number of users (M) (b) variable number of UAVs (n)

Figure 3.5: $AoI^{TBS}(T)$ for varying users and UAVs

Variable N (UAVs) With L=2, K=1, we place $M = 10$ IoT devices and vary the number of UAVs deployed. The IoT devices are assigned equally among the UAVs with the remaining IoT devices assigned to the last UAV, if any. The results are shown in Fig. 3.5b - for constant M, K and L, average AoI at the TBS remains mostly same. This is because even though N is increased, K remains the same for all. Hence capacity of information transfer from the UAVs to the TBS is the same which results in similar average AoI at the TBS.

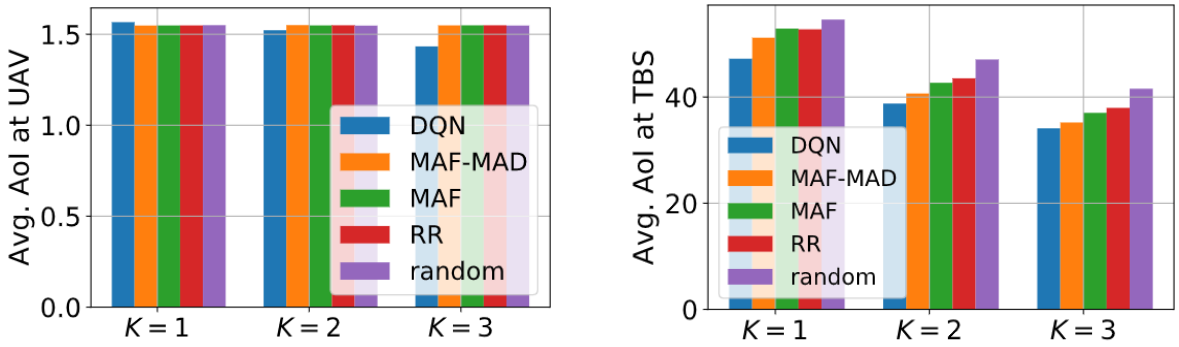
Varying number of channels (L and K) First for fixed K=1, M=9 IoT devices are allocated equally between N=3 UAVs. L is varied and the results for average AoI at the TBS are shown in Fig. 3.6b. As L increases, we don't see any noticeable improvement in average AoI at the TBS. A higher L will improve the $AoI^{UAV}(T)$ (see Fig. 3.6a), but this benefit is not passed on to the TBS as seen from Fig. 3.6b. This indicates that increasing L doesn't help much in improving average AoI at the TBS. The average AoI at the UAV for L=3 in Fig. 3.6a is the same for all the schedulers as $L = m_n = 3$, $\forall n \in N$, due to which all the devices get equally sampled.

Then for the same setting as above, we vary K keeping other parameters constant. L is set to 3 so that each UAV can sample all 3 devices under it at each slot. Therefore the $AoI^{UAV}(T)$ is the same for all the four schedulers as seen from Fig. 3.7a. Then average AoI at the TBS is shown in Fig. 3.7b. Here we see an improvement as K is increased due to which more information can be transferred from the UAVs to the TBS at each slot resulting in lower average AoI at the TBS. This shows that the channels between the UAVs and the TBS is more important and critical bottleneck compared to the channels between the IoT



(a) $AoI^{UAV}(T)$ (b) $AoI^{TBS}(T)$

Figure 3.6: Variable channels at the UAVs (L)



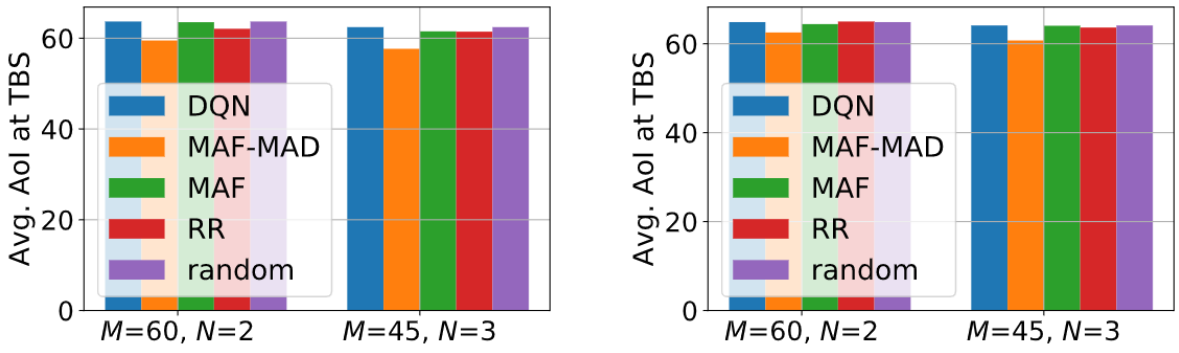
(a) $AoI^{UAV}(T)$ (b) $AoI^{TBS}(T)$

Figure 3.7: Variable channels at the TBS (K)

devices and the UAVs in improving average AoI at the TBS.

3.5.4 Larger IoT networks

Now we consider networks requiring around 50 IoT devices and these types of networks are deployed where low number of devices are not able to capture enough information. E.g. in agricultural fields, multiple sensors for capturing moisture, temperature levels etc are needed [43].



(a) $AoI^{UAV}(T)$ (b) $AoI^{TBS}(T)$

Figure 3.8: AoI(T) fro larger networks

Experimental Results

We set $L = K = 2$ and simulate two scenarios: first we place $M = 60$ IoT devices and assign then equally among $N = 2$ UAVs, and secondly $M = 45$ devices are assigned equally among $N = 3$ UAVs. The results under ideal and general scenarios are shown in Fig. 3.8a and Fig. 3.8b respectively.

This result helps us draw key challenges and a future road-map for further research. From Fig. 3.8a and Fig. 3.8b, it is evident that DQN does badly in terms of AoI minimization for larger networks with MAF-MAD having the best performance under both conditions. This is because, as the network size increases, the action space also increases and DQN *doesn't* provide satisfactory performance when the action space is very large [125]. Due to this, works involving DQN for scheduling in IoT networks to improve AoI have been limited to smaller networks [19, 53, 54, 106, 127]. An alternative approach for larger scenarios is distributed multi-agent learning [121] where each UAV and the TBS acts as independent agents, and this will be explored in a future version of this work.

3.6 Motivation for a Proximal Policy (PPO) Based Scheduler

While the MAF-MAD scheduler provides the best AoI minimization under ideal conditions, however as shown in our preliminary work [37] and also in Sec. 3.7, it doesn't do well under

general conditions with lossy channels and periodic packet generation at the IoT devices, both of which are unknown apriori at the scheduler. This motivates for a reinforcement learning (RL) based approach where the scheduler can learn these characteristics. In the previous section, we showed that while a DQN based scheduler works well for networks having small number of actions, it doesn't scale with network size. In this work, we propose an actor critic based approach, PPO, that is suitable for such scenarios. Details on the PPO scheduler and comparison between the PPO and DQN schedulers will be discussed in the upcoming sections.

PPO Preliminaries

Before presenting the proposed method, we give a brief introduction to deep RL and PPO in this section.

Episode, Return, and Value function In RL, mathematical entities commonly known as agents, learn to make optimal decisions by interacting in an unknown environment and exploiting the received feedback. Usually, Markov Decision Process (MDP)[1] is used to simulate such environment that requires decision making in situations where outcome is partly random and partly under the control of the decision maker or agent. MDP is formally defined as the tuple $\{\mathcal{X},\mathcal{A},\mathcal{T},p(x_0),\gamma\}$. Therefore, the environment consists of a transition function $\mathcal{T}:\mathcal{X}\times\mathcal{A}\to p(\mathcal{X})$ and a reward function $\mathcal{R}:\mathcal{X}\times\mathcal{A}\times\mathcal{X}\to\mathbb{R}$. At time slot t, the agent observe some state $x_t\in\mathcal{X}$ and picks an action $a_t\in\mathcal{A}$ with policy $\pi(a_t|x_t):\mathcal{X}\to p(\mathcal{A})$. As a result, the environment transitions to a next state $x_{t+1}\sim\mathcal{T}(\cdot|x_t,a_t)$ and returns a scalar reward $r_t=\mathcal{R}(x_t,a_t,x_{t+1})$. The first state x_0 is sampled from the initial state distribution $p(x_0)$. Finally, $\gamma\in[0,1]$ is the discount factor.

The agent interacts with the environment until it reaches the terminal state or reaches the time limit T[2], completing an episode $\tau=\{x_0,a_0,r_0,\ldots,x_t,a_t,r_t,\ldots,x_{T-1},a_{T-1},r_{T-1},x_T\}$. Thus quality of state x_t can be realized as the cumulative sum of rewards from x_t to the end of the episode τ, which is known as the return G_t, see Eqn. (3.25). However, due to the stochastic nature of the environment, there can be many possible next states x_{t+1}, resulting multiple return G_t values from state x_t for different episodes. Therefore, quality of state x_t is defined

[1]Markov Decision Process is a discrete-time stochastic control process.
[2]Note that, T is the observation interval and usually very small than total time duration $\mathbb{T}$

by the value function $V(x_t)$, as the expected return from state x_t, see Eqn. (3.26).

$$G_t = \sum_{\hat{t}=t}^{T} \gamma^{\hat{t}-t} r_{\hat{t}} \tag{3.25}$$

$$V(x_t) = \mathbb{E}\left[G_t \mid x_t\right] \tag{3.26}$$

Actor-Critic methods The primary objective of the agent is to behave optimally with policy $\pi(a_t|x_t)$ so that it can obtain a high cumulative reward $R_\tau = r_0 + r_1 + r_2 + \cdots + r_{T-1}$ in episode τ. Lets say $p(\tau)$ is the probability of sampling episode τ with current policy $\pi(a_t|x_t)$. In stochastic policy gradient methods, we maximize the cost function $J(\pi) = \sum_{\tau=1}^{\infty} p(\tau) R_\tau$ with respect to the policy $\pi(a_t|x_t)$ to increase the likelihood of sampling the most rewarding episode under the policy.

In practice, the state x_t can be multi-dimensional, making it infeasible for traditional table based RL approaches [119]. This provides a clear path for the application of universal function approximators (i.e, neural networks) to model policy $\pi(a_t|x_t)$ with parameter θ. Similarly, value function $V(x_t)$ is modeled with parameters ψ. The parameterized policy $\pi(a_t|x_t;\theta)$ and the value function $V(x_t;\psi)$ is commonly known as the **actor** and the **critic** network respectively.

While the actor network interacts with the environment and collect experiences $\{x_t, a_t, r_t, x_{t+1}\}$, the advantage is computed as shown in Eqn. (3.27). Advantage A_t represents the incentive obtained by taking action a_t at state x_t from the prospective of the critic network. Subsequently, the actor and the critic network is jointly optimized by back-propagating the gradients from Eqn. (3.28) and Eqn. (3.29).

$$A_t = r_t + \gamma V(x_{t+1};\psi) - V(x_t;\psi) \tag{3.27}$$

$$\nabla_\theta J(\theta) = \mathbb{E}\left[\nabla_\theta \log \pi(a_t|x_t;\theta) A_t\right] \tag{3.28}$$

$$\nabla_\psi \mathcal{L}(\psi) = \mathbb{E}\left[(G_t - V(x_t;\psi)) \nabla_\psi V(x_t;\psi)\right] \tag{3.29}$$

Proximal Policy Optimization The actor-critic algorithms that rely on the policy gradient formulation suffer from divergent behavior unless we impose a mechanism to constrain the change of the actor's policy. The primary reason is the on-policy data collection for optimizing the actor and the critic network. This issue is addressed in [100] by introducing

trust-region policy updates. Since then, actor-critic algorithms have excelled at tackling real-world problems due to their increased efficiency and stability with respect to Q-Learning. Proximal Policy Optimization (PPO) is the recent addition to such class of algorithms and therefore inherits the aforementioned properties. Moreover, PPO improves the efficiency further and achieves trust-region policy updates by simply clipping the probability ratio $\zeta_t \in [1-\epsilon, 1+\epsilon]$ between the current $\pi(a_t|x_t;\theta)$ and the old $\pi(a_t|x_t;\theta_{old})$ actor policies. PPO optimizes the actor network with the gradients shown in Eqn. (3.30) and also leverages entropy $\mathcal{H}(\theta)$ based regularization to encourage exploration at early stages of learning. The **combined gradient** in Eqn. (3.31) optimizes the *actor* and the *critic* networks simultaneously, where c_1, c_2 are constants.

$$\nabla_\theta J(\theta) = \mathbb{E}\left[\nabla_\theta Min(\zeta_t A_t, [\zeta_t]_{1-\epsilon}^{1+\epsilon} A_t)\right] \tag{3.30}$$

$$\text{where,} \quad \zeta_t = \frac{\pi(a_t|x_t;\theta)}{\pi(a_t|x_t;\theta_{old})}$$

$$\nabla_{\theta\cup\psi}\mathcal{L}(\theta\cup\psi) = -\nabla_\theta J(\theta) + c_1\nabla_\psi\mathcal{L}(\psi) - c_2\nabla_\theta\mathcal{H}(\theta) \tag{3.31}$$

MDP formulation

In this section, we formulate the packet scheduling to minimize AoI at UAV-relayed IoT networks as an MDP for the PPO scheduler. The state, action and reward in the context of our work is explained next.

State Space State is the scenario encountered by the scheduler. For the PPO-based scheduler, we define our state as

$$x_t = \{\hat{t}, \{AoI_m^{UAV}(\hat{t})\}_{m=1}^{M}, \{AoI_m^{TBS}(\hat{t})\}_{m=1}^{M}\}_{\hat{t}=t}^{t-z+1} \tag{3.32}$$

where t is the ongoing time slot and z is the stacking size. The stacking size is discussed in Sec. 3.7.1. At t=1, $AoI_m^{UAV}(t)$ and $AoI_m^{TBS}(t)$ are initialized to 1. The state space is denoted by $\mathcal{X}$.

Action Space As described in Sec. 3.3, scheduling involves two simultaneous steps of sampling and updating devices at each time slot. Therefore the action at each time slot t is

$$a_t = (\mathcal{S}(t), \mathcal{U}(t)) \tag{3.33}$$

where $\mathcal{S}(t)$=$\{S_1(t), S_2(t), ..S_N(t)\}$ is the set of all devices sampled by all the UAVs at time slot t such that $|S_n(t)| \leq L \; \forall \; n \in N$. Similarly, $\mathcal{U}(t)$ is the set of devices selected to update their packets to the TBS where $|\mathcal{U}(t)| \leq K$. The action space is denoted by $\mathcal{A}$, where $|\mathcal{A}| = (\prod_i {}^{m_i}C_L) \times {}^{M}C_K$. Note that the Eqns. (3.23) and (3.33) are identical as the actions in both the DQN and PPO-based schedulers are same.

Reward As the objective is improving the information freshness at the TBS, the reward is given by the negative of the average AoI of all the devices at the TBS at time slot $t+1$

$$r_t = \mathcal{R}(x_t, a_t, x_{t+1}) = -\sum_{m=1}^{M} AoI_m^{TBS}(t+1) \tag{3.34}$$

Note that Eqns. (3.24) and (3.34) are similar as both the MDP formulations are based on minimizing the AoI at the TBS. Based on action a_t, the environment transitions from state x_t to a new state x_{t+1} according to the state transition probabilities described in Eqn. (3.3), (3.4), (3.5), (3.6) while resulting in a reward r_t. Thus it is a finite horizon MDP with finite state and action spaces, which makes it suitable for RL approaches.

Voting mechanism

Here, we propose the voting mechanism, which efficiently captures the action space $\mathcal{A}$ for sampling and updating packets at UAV $n \in N$ and TBS respectively. The advantage of using voting to sample L packets from m_n devices at UAV $n \in N$ is following- at time slot t, we compute votes $v_i^t \; \forall \; i \in m_n$ devices and sample packets from $S_n(t)$ devices with top L votes as shown in Eqn. (3.35). Hence, we can capture ${}^{m_n}C_L$ possible sampling actions with m_n votes. Similarly, we can update K packets to the TBS from M devices with Eqn. (3.36). The benefit of this approach will be clearer if we consider all N UAVs and the TBS, since we only have to compute $(\sum_{n=1}^{N} m_n) + M = 2M$ votes to represent an action space with cardinality $|\mathcal{A}| = (\prod_i {}^{m_i}C_L) \times {}^{M}C_K$.

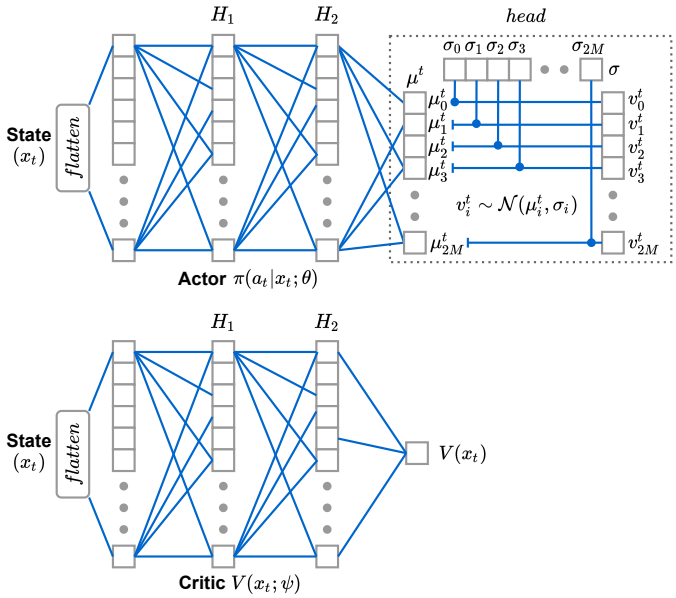


Figure 3.9: Actor and Critic network architectures

$$\mathcal{S}_n(t) = \underset{\mathcal{S}' \subseteq m_n, |\mathcal{S}'| \leq L}{\arg\max} \{v_i^t\}_{i=1}^{m_n} \; \forall \; n \in N \tag{3.35}$$

$$\mathcal{U}(t) = \underset{\mathcal{U}' \subseteq M, |\mathcal{U}'| \leq K}{\arg\max} \{v_i^t\}_{i=1}^{M} \tag{3.36}$$

The PPO-based scheduler leverages this voting mechanism, as shown in Figure 3.9. At time slot t, the actor network $\pi(a_t|x_t;\theta)$ outputs $2M$ normal distributions $\{\mathcal{N}(\mu_i^t, \sigma_i)\}_{i=1}^{2M}$ with mean $\mu^t = \{\mu_i^t\}_{i=1}^{2M}$ and standard deviation[3] $\sigma = \{\sigma_i\}_{i=1}^{2M}$ to compute the votes as $\{v_i^t : v_i^t \sim \mathcal{N}(\mu_i^t, \sigma_i)\}_{i=1}^{2M}$. Thereafter, the votes are converted to $\mathcal{S}_n(t) \; \forall \; n \in N$ and $\mathcal{U}(t)$ as per Eqns. (3.35) and (3.36), which are combined to obtain the final action a_t as given in Eqn. (3.23).

Algorithmic Description

As shown in Algorithm 5, in the beginning, the PPO-based scheduler randomly initializes the parameters θ and ψ of the actor $\pi(.)$ and critic $V(.)$ networks, respectively (Line 1). It

[3]Note that, PPO based scheduler explores the action space by increasing entropy $\mathcal{H}(\theta) = \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2}$, see Eqn. (3.31).

Algorithm 5 PPO-based scheduler

1: Initialize actor $\pi(a_t|x_t;\theta)$, critic $V(x_t;\psi)$
2: Initialize actor's old parameters, $\theta_{old} \leftarrow \theta$
3: **for** iteration $\leftarrow 1, 2 \ldots \lceil \frac{\mathbb{T}}{\mathcal{E}\times T} \rceil$ **do**
4: Replay Buffer $\mathcal{B} \leftarrow \{\}$
5: **for** episode $\leftarrow 1, 2 \ldots \mathcal{E}$ **do**
6: **for** time slot t $\leftarrow 0, 1, 2, \ldots T-1$ **do**
7: Observe current state x_t
8: Execute action $a_t \sim \pi(\cdot|x_t;\theta_{old})$
9: Observe next state x_{t+1} and, reward r_t
10: Compute Advantage A_t using Eqn. (3.27)
11: $\mathcal{B} \leftarrow \mathcal{B} \cup \{x_t, a_t, r_t, x_{t+1}, A_t\}$
12: **for** *epoch* $\leftarrow 1, 2, \ldots \eta$ **do**
13: Sample minibatch $b_m \sim \mathcal{B}$ of size $|b_m| \leq E \times T$
14: Compute gradients $\nabla_{\theta\cup\psi}\mathcal{L}(\theta \cup \psi)$ using Eqn. (3.31)
15: Optimize parameters θ and ψ with the gradients:
16: $\theta \leftarrow \theta - \alpha\nabla_\theta\mathcal{L}(\theta \cup \psi)$
17: $\psi \leftarrow \psi - \alpha\nabla_\psi\mathcal{L}(\theta \cup \psi)$
18: Update actor's old parameters, $\theta_{old} \leftarrow \theta$

also initializes the actor's old parameters θ_{old} with θ (Line 2).

Next, the replay buffer $\mathcal{B}$ is (re)initialized to store new experiences (Line 4). Following it, the actor interacts with the environment for $\mathcal{E}$ episodes, where each episode is of length T. At time slot t of an episode, the actor and the critic network observes the state x_t. Note that, the actor network utilizes its old parameters θ_{old} to compute action a_t. Next, action a_t is executed in the environment, resulting transition to next state x_{t+1} and the PPO based scheduler receives reward r_t(Line 8-9). The critic network is utilized to compute the value of the state $V(x_t;\psi)$ to estimate the advantage A_t of taking action a_t on state x_t using Eqn. (3.27) (Line 10). At this point, the PPO-based scheduler obtains a new experience $\{x_t, a_t, r_t, x_{t+1}, A_t\}$, which is stored in the replay buffer $\mathcal{B}$ (Line 11). At the end of $\mathcal{E}$ episodes, the replay buffer contains $\mathcal{E} \times T$ entries, which trains the actor and critic networks as discussed next (Lines 12 - 18).

A minibatch b_m is sampled from the replay buffer $\mathcal{B}$, where $|b_m| \leq \mathcal{E} \times T$. Subsequently, the gradient $\nabla_{\theta\cup\psi}\mathcal{L}(\theta \cup \psi)$ is computed on minibatch b_m using Eqn. (3.31). Next, the gradient *(i)* $\nabla_\theta\mathcal{L}(\theta \cup \psi)$ *with respect to actor parameters* θ *(ii)* $\nabla_\psi\mathcal{L}(\theta \cup \psi)$ *with respect to critic parameters* ψ is utilized for optimizing (gradient descent) θ and ψ, where α is the learning

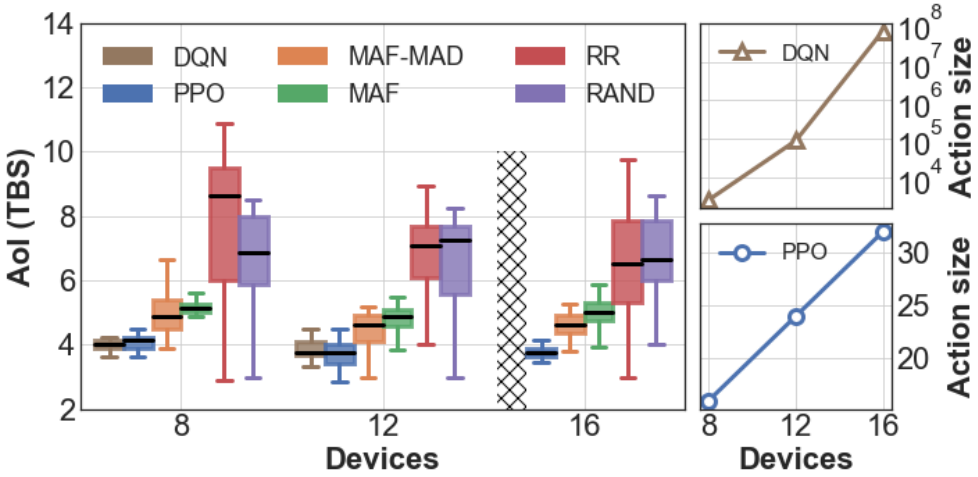


Figure 3.10: DQN vs PPO

rate. These gradient updates improves the actor and critic parameters and the PPO based scheduler performs multiple such updates over η epochs on the replay buffer $\mathcal{B}$. At the end of η epochs (Line 17), the actor network's old parameters θ_{old} is updated with the new actor network parameters θ (Line 18), which will be utilized for interacting with the environment in the next iteration (See line 3). The PPO based scheduler runs for $\lceil \frac{\mathbb{T}}{\mathcal{E} \times T} \rceil$ number of such iterations (Line 3-18) that improves the actor network parameters θ over old parameters θ_{old}, where $\mathbb{T}$ denotes the total time duration considered for packet scheduling.

Note that the convergence of neural networks are hard to be analytically investigated and is strongly dependent on the set of hyper-parameters used [57]. Selection of hyper-parameters is a challenging task and therefore a reasonable set of hyper-parameters is found by trying out different values. More details on the hyper-parameters used in our case are presented in Sec. 3.7. Similar to [19, 53, 106, 127], we limit our investigation into the convergence to simulations, where the neural network converges under the hyper-parameters used and the results presented for the DQN and PPO scheduler are the values obtained after their convergence.

3.6.1 PPO-based scheduler vs DQN-based scheduler

Here, the advantages of the PPO-based scheduler over the DQN-based scheduler proposed in our preliminary work [37] will be highlighted.

Smaller Action Size In our preliminary work [37], we showed that DQN suffers from performance issues in networks with larger action spaces. The number of actions for the DQN-based scheduler is given by $(\prod_i {}^{m_i}C_L) \times {}^{M}C_K$, which means the action size increases rapidly as the network size increases and DQN *doesn't* provide satisfactory performance when the action size is very large [125]. Due to this, works involving DQN for scheduling in IoT networks to improve AoI have been limited to smaller networks [19, 53, 54, 106, 127]. This can be seen in Fig. 3.10 where we consider 3 different UAV-relayed IoT networks with $M = 8, 12$, and 16 IoT devices. Note that the number of UAV channels is $L = \frac{m_n}{2}$ and BS channels is $\frac{M}{2}$ as it causes the maximum number of actions for the DQN-based scheduler.

As shown in Fig. 3.10, the DQN-based scheduler sees a factorial increase in the action size, where the action size of PPO-based scheduler increases linearly as $2M$ with the number of IoT devices M. While both DQN and PPO perform relatively well for the first two networks with $M = 8$ and 12 devices, the DQN-based approach is not feasible for $M = 16$ as it's action size is of the order 10^8 (memory required is very large) whereas the PPO's action size is 32. This shows several folds improvement of PPO-based scheduler over DQN-based scheduler in reducing the action size, and thus, much better scalability of the PPO-based scheduler compared to the DQN-based scheduler.

Scheduler's Generalization ability In certain situations, the training and the deployment network conditions could be different. E.g., devices might change the periodicities or the channel qualities can change. Under such situations, it is desirable if a trained scheduler is able to adapt to the new environment faster than training a new scheduler from scratch. Because PPO is an on-policy algorithm, it is able to adapt. Whereas DQN is an off-policy algorithm which will always need a new scheduler to be trained every-time there is a change in the network conditions.

Consider a UAV-relayed IoT network E_1 with the parameters given in Table.3.3. Let us take two different networks, E_2 and E_3 both of which are similar to E_1 with the only difference mentioned in the following-

- In network E_2, the channel conditions for 10% IoT devices and UAVs have changed with respect to E_1,
- In network E_3 the periodicity for 10% of IoT devices have changed with respect to E_1.

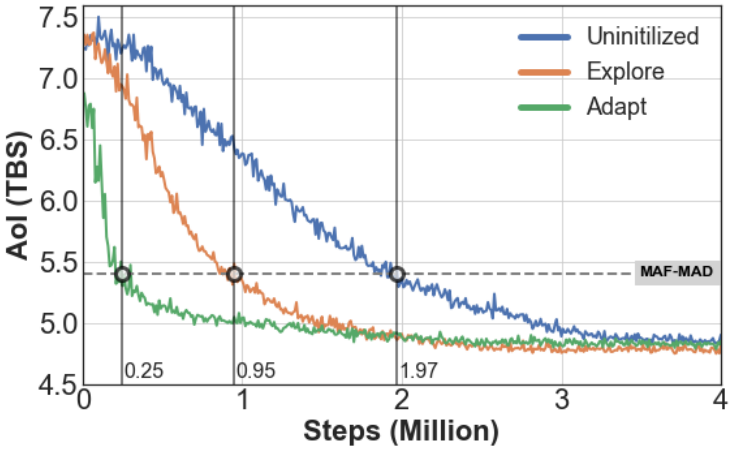


Figure 3.11: Network E_2 : Adapting to changing channel conditions

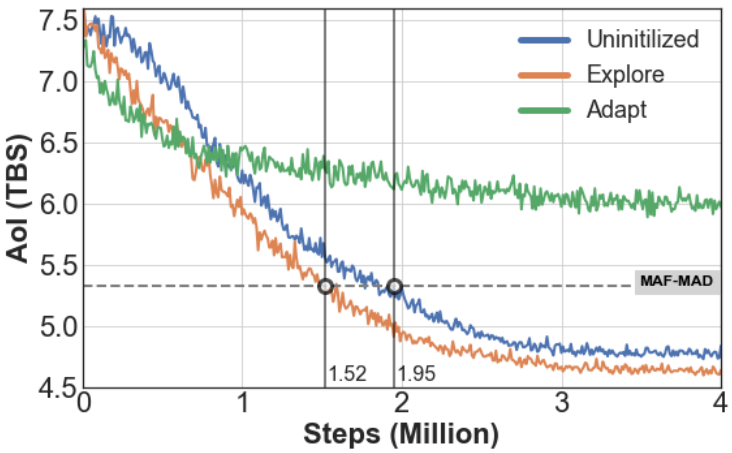


Figure 3.12: Network E_3 : Adapting to changing periodicity

In both E_2 and E_3, a PPO-based scheduler will be trained from scratch to get the baseline performance and this will be termed as *uninitialized*. To showcase the ability of the PPO-based scheduler to generalize to a changed network environment, a PPO-based scheduler trained in E_1 will be deployed in E_2 and E_3 using 2 different approaches –

(i) *explore* meaning that scheduler trained for E_1 will be reused except for the actor head which will be re-trained,

(ii) *adapt* meaning that scheduler trained for E_1 will be entirely reused including the actor head.

Note that, the actor head is comprised of the output layer of the actor network as shown in Fig. 3.9.

Fig. 3.11 shows the results for network E_2. Both *explore* and *adapt* schedulers converge earlier than the *uninitialized*, with the *adapt* scheduler converging exceptionally fast. Therefore additional exploration to retrain the actor head doesn't lead to any benefit. However Fig. 3.12 shows the results for network E_3 where the *adapt* scheduler is unable to perform well and doesn't even outperform MAF-MAD scheduler. Therefore re-training the actor head becomes important. The performance of the *uninitialized* and *adapt* schedulers are similar, signifying that the PPO-based scheduler generalizes well for changed channel conditions as compared to changed device periodicity.

3.7 Results

This section discusses the performance evaluation of the proposed PPO-based scheduler against the baseline schedulers in terms of minimizing AoI at the TBS. The IoT devices and UAVs are placed randomly [79] in a simulation area of $l = b = 1000$ m and the observation interval $T = 20$ slots. Unless otherwise stated, we consider a general network setting with 30 IoT devices which are uniformly distributed among the 3 UAVs. The default parameters for the simulation study are listed in Table 3.3. As discussed in previous section, because of the large memory requirements, it is not feasible for DQN-based scheduler to deal with general simulation setting (with > 12 IoT devices), we do not showcase the results for DQN-based scheduler.

The PPO-based scheduler (and DQN-based scheduler results presented in the previous sec-

tion 3.6.1) are implemented on an NVIDIA DGX station with an Intel Xeon E5-2698 v4 CPU and an NVIDIA Tesla V100 GPU (32 GB memory). Data communication between CPU and GPU takes place through a PCIe 3.0 X16 slot. The hyper-parameters for DQN are kept the same as in [37], while the hyper-parameters for the PPO implementation are shown in Table 3.4 respectively.

Table 3.3: Default network parameters

Number of IoT devices, M	30
Number of UAVs, N	3
IoT devices per UAV, m_n	$(10, 10, 10)$
Channels at each UAV, L	4
Channels at the TBS, K	10

Table 3.4: PPO Hyper-parameters

Neurons in hidden layer 1 H_1	256
Neurons in hidden layer 2 H_1	256
Relay memory size $\lvert\mathcal{B}\rvert$	2048
Minibatch size $\lvert b_m\rvert$	512
Learning rate α	2.5×10^{-4}
Discount factor γ	0.99
Activation function	ReLU
Clip range ϵ	0.2
Optimizer	Adam
Epochs η	10
Total episodes $\mathcal{E}$	5×10^5

3.7.1 AoI Stacking

As shown in in Eqn. (3.32), the state space consists of the AoI of the devices stacked for previous z slots. A higher stack size provides more historical information at the expense of larger memory requirements. Here, Fig. 3.13 shows the AoI at the TBS achieved for different stack sizes. The convergence is initially faster with $z = 16$, but it takes more time to reach the optimal value. Whereas $z = 1$ converges very slowly. As, with higher stack size and increased information in the state, the parameters of the PPO-based scheduler increases as well. Hence it takes less time to reach a considerable performance but struggles to fine tune itself against the environment. $z = 4$ provides a good trade-off between the convergence

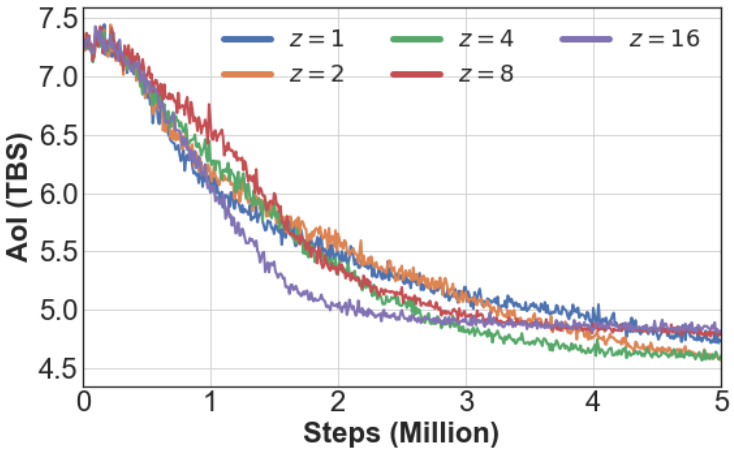


Figure 3.13: Effect of different stack sizes (z)

speed and the optimal performance. Thus the remaining experiments will utilize $z = 4$ slots[4] for the state space.

3.7.2 Ideal vs general environment

Ideal environments are characterized by lossless channels and generate-at-will traffic generation policy at the devices. The results shown in Fig. 3.14 confirm the analytical results derived in Theorem 3.1 that MAF-MAD is the optimal scheduling policy and outperforms all other scheduling policies under such ideal environments. As PPO is also able to learn the the optimal scheduling, it's performance also converges to MAF-MAD's performance. Overall, the PPO and MAF-MAD outperforms MAF, RR, random by 22.3%, 42.1%, 38.3% respectively. However when a general environment is considered where the ideal assumptions do not hold, MAF-MAD is no longer the optimal scheduler and PPO does better. It outperforms MAF-MAD, MAF, RR, random by 10.6%, 25.1%, 30.9%, 37.9% respectively. While the results could be shown for any channel condition and periodicities, they are pre-assigned for ease of reproducibility [37, 71, 106].

Note that the range of the AoI achieved by the PPO-based scheduler is much less than the other schedulers. The PPO-based scheduler considers the channel conditions while allocating resources. Therefore the devices with bad channels are scheduled more frequently so that their AoI do not suffer, at the expense of a higher AoI of the devices with better channels.

[4]Note that in our preliminary work [37], we used a stack size of $z = 1$ for the DQN-based scheduler.

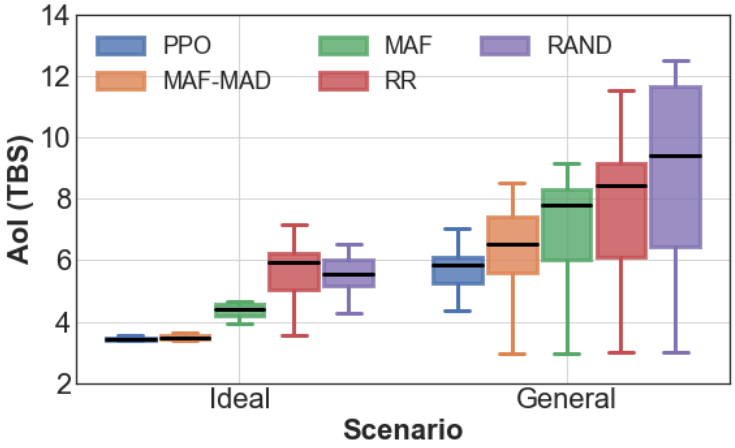


Figure 3.14: Ideal vs general environment

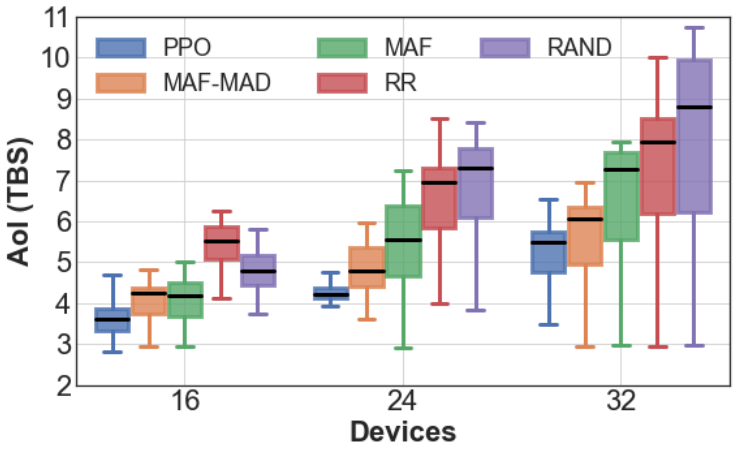


Figure 3.15: AoI for varying number of IoT devices

However for the other schedulers, the devices with good channels see a very low AoI and and devices with bad channels will have high AoI, resulting in a higher range. This is because unlike our PPO-based scheduler, the counterpart schedulers do not have the ability to learn the various channel conditions/fluctuations.

For the rest of the experiments below, general conditions with lossy channels and periodic packet generation at the IoT devices will be considered.

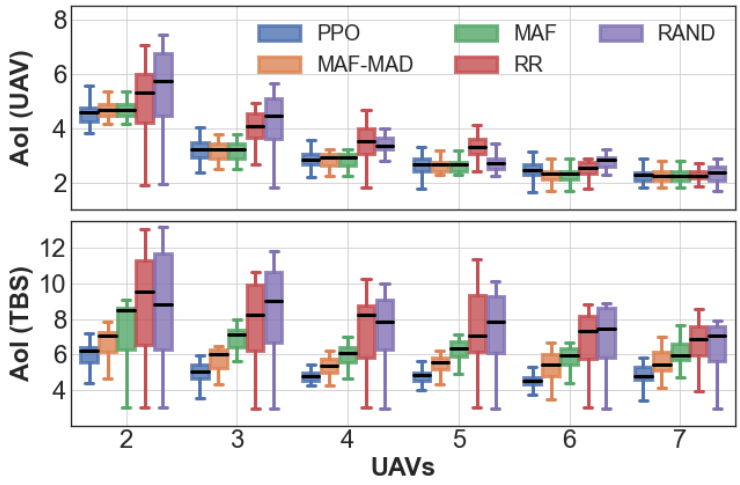


Figure 3.16: AoI for varying number of UAVs

3.7.3 Effectiveness of PPO-based scheduler

In this section, we analyze the effectiveness of the PPO-based scheduler in minimizing the AoI at the TBS with changing network parameters such as the number of IoT devices, UAVs, UAV channels and TBS channels.

Varying number of IoT devices (M) As shown in Fig. 3.15, the PPO-based scheduler outperforms the baselines in all considered scenarios by a significant improvement of AoI at the TBS. However, an upward trend in the AoI can be observed as we increase the number of IoT devices in the network. The primary reason is the constant sampling and updating capacity of the network. Therefore, the increase of IoT devices results in less frequent sampling and updating of packets from a particular IoT device. Thus, we observe an increased average AoI at the TBS for all schedulers.

Varying number of UAVs (N) The PPO-based scheduler shows superiority in minimizing AoI at TBS with changing the number of UAVs in the network, see Fig. 3.16. As shown in the figure, the AoI at UAV is improved with more UAVs, which directly improves the AoI at the TBS as per Eqn (3.5). However, after a point, all schedulers show equivalent AoI at the UAV as most IoT devices can be sampled at once, and there is little scope of improvement due to scheduling. Despite similar AoI at the UAV, the PPO-based scheduler outperforms the baselines and improves the AoI further as it employs better scheduling decisions at the

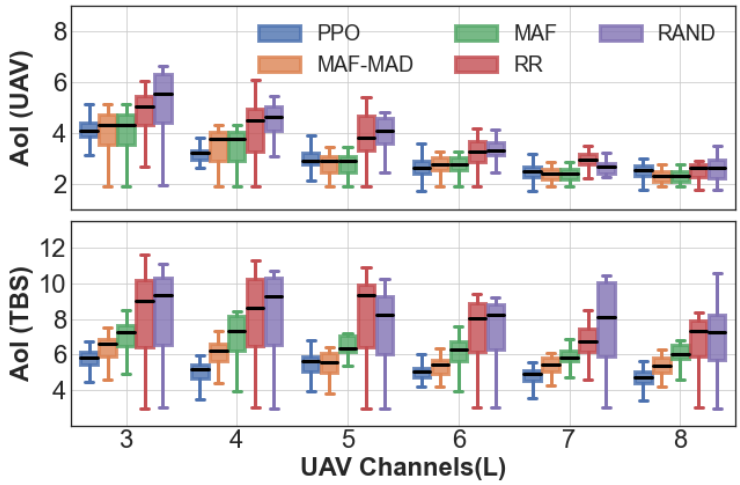


Figure 3.17: AoI for varying UAV Channels (L)

TBS.

Varying number of UAV channels (L) The increasing number of UAV channels has a similar effect to increasing the number of UAVs, as in both ways, the sampling capacity of the network is improved. Hence, the AoI at the UAV improves up to a point when most of the IoT devices can be sampled at once and then becomes equivalent for all schedulers, see Fig. 3.17. Nevertheless, the PPO-based scheduler outperforms the baselines in minimizing the AoI at the TBS in all considered scenarios. Note that the PPO-based scheduler achieves a 61.8% improvement in AoI at the UAV with increased UAV channels from 3 to 8. However, all this gain is not transferable to the TBS, and it only sees a 23.7% improvement in the AoI due to the bottleneck at UAV to TBS communication.

Varying number of TBS channels (K) On the contrary, increasing the number of TBS channels does not change the sampling capacity of the network. Thus, the AoI at the UAV remains unchanged for a particular scheduler with the increase of 10 to 20 in TBS channels, see Fig. 3.18. Instead, the network's updating capacity is improved, allowing more packets to be transferred from the UAVs to the TBS. Therefore, the TBS sees a 31.0% improvement in AoI for PPO-based scheduling and outperforms the baselines. Thus, the channels between the UAVs and the TBS (K) are more critical bottlenecks than the channels between IoT devices and the UAVs (L) in minimizing AoI at the TBS.

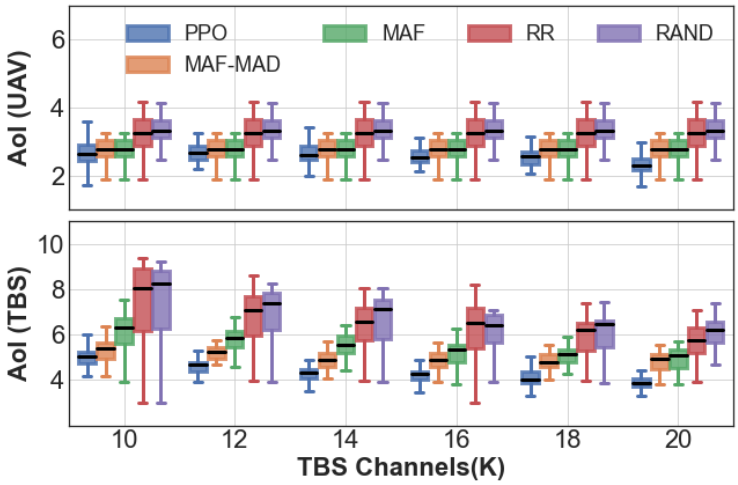


Figure 3.18: AoI for varying TBS Channels (K)

3.8 Conclusion

We propose an MAF-MAD scheduler and prove its optimality in improving AoI in a UAV-relayed IoT network with two hops, where the UAVs act as static ABS that relay packets between the IoT devices and the TBS under no packet loss in any of the channels and generate-at-will traffic generation model at the IoT devices. However when packet loss in the channels and periodicity in packet generation by the IoT devices are considered, MAF-MAD scheduler is no longer optimal. Therefore we propose a PPO-based scheduler which performs better than the MAF-MAD scheduler under such realistic general conditions. We also show that the channels between the UAVs and the TBS play a much important role in improving AoI at the TBS compared to channels between the IoT devices and UAV, which points to the importance of backhaul links compared to access links for AoI minimization. In our future work, we will consider mobility and optimal UAV placement under UAV-relayed IoT networks with the objective of improving information freshness at the TBS.

Chapter 4

Design and Analysis of Age-Optimal 5G Schedulers in Smart Grids and General Distributed Networks

4.1 Introduction

5G is the most recent wireless communication standard that improves rates and reliability over older generations such as 4G/LTE. Prominent applications of 5G communications include remote mining, vehicular communications, healthcare, and the military [8]. All these applications rely on the extremely high data rates and reliability supported by 5G. In general, 5G applications are categorized into three main types based on their performance requirements: (i) *enhanced mobile broadband (eMBB)*: eMBB involves providing 5G connectivity to traditional cellphones which are expected to be its most commonly used application, (ii) *ultra-reliable low latency communications (URLLC)*: URLLC is designed for critical applications where readability is extremely important, e.g., driving or robotic surgery, and (iii) *massive machine-type communications (mMTC)*: mMTC is designed for massive internet of things (IoT) devices where battery performance is more important than data rates. The different characteristics of these applications are described in Fig. 4.1 [14].

5G also has applications in the control of the modern power system, often referred to as the smart grid. Smart grid leverages communication techniques to collect and distribute information about the system measurements to various grid components and the control center. For example, real-time feedback on electricity consumption can be used to dynamically adjust energy generation with the objective of improving energy efficiency [46]. Traditionally, communication in a power grid is designed using optical fibers which can be expensive as the grid becomes larger due to the large-scale connectivity needed between all the components

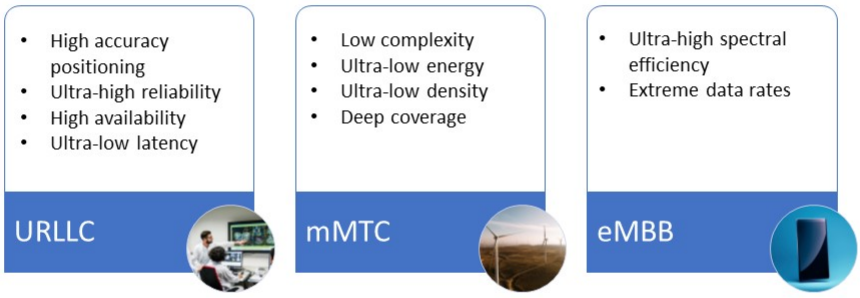


Figure 4.1: 5G application types

involved [11]. Therefore, there is a timely opportunity for a use case for providing connectivity to power grid components through 5G. Third-generation partnership project (3GPP), which is the standards organization in charge of developing protocols for mobile telecommunications, recently formed a study group to standardize the end-to-end architecture of 5G networks specifically built for aiding smart grid operations and is expected to be formalized by Release 18[1].

There has been a recent focus on investigating the benefits that connectivity can bring to smart grids. The authors in [47] compare the response of a fault management system 4G-LTE and 5G systems. The results indicate that the machine-to-machine connectivity abilities of 5G provide significant performance improvement compared with 4G. Reference [103] reviews several applications that 5G can enable in smart grids and analyzes the associated challenges. In [120], a network servicing both eMBB and URLLC applications are considered, and its ability in handling line protection through URLLC is analyzed. The results show that provided the network coverage is above a certain threshold, differential line protection can be supported by the network. However, these research studies consider only a black-box view of the 5G network without considering the implications of its various subsystems on the overall performance. This chapter tries to address this gap by assessing the smart grid's performance under a 5G RAN. The contributions of the first part of this work are:

- A 5G radio access network (RAN) simulator is implemented in Python and is interfaced with general-purpose power systems simulators such as PSCAD and MATLAB to create a co-simulation environment. Inside this co-simulation, the power grid com-

ponents communicate with each other through 5G links under the control of a 5G base station, or gNodeB. This allows us to simulate the effect of RAN resource allocation on the smart grid. The RAN simulator is robust in terms of both interfacing with the different software environments and the number of devices that communicate within the power system.

- The ability of 5G to support smart grid applications, namely the power park and coordinated set point modulation, is investigated using this co-simulation environment.
- The performance of the smart grid under a 5G network is compared with an ideal scenario where the power grid components can communicate instantaneously which provides insight into the stability of the smart grid under 5G.

The rest of the chapter is organized as follows: Section 4.2 describes the overall design of the co-simulator and the interactions between the power and communication systems. Section 4.3 to Section. 4.5 investigate the performance of two smart grid applications under 5G. The distributed control in smart grid and the impact of reduced AoI in it is analyzed in Sections 4.6-4.9. Then the differences in general case of distributed AoI minimization as compared to the other applications considered in this dissertation is explained in Section. 4.10. Finally, the performance of different schedulers under this distributed setting is presented in Section.4.11.

4.2 Co-simulator Design

The objective of developing the co-simulation environment is to investigate the performance of a smart grid in the presence of a 5G RAN. The power system applications are implemented in PSCAD and MATLAB, and the communication environment that includes the RAN scheduler, channels and the gNodeB is implemented through a Python script. The Python script models the 5G communication between the smart grid distributed energy resources (DER). Consider the scenario shown in Fig. 4.2. The DERs generate packets containing their state information, i.e., their local measurements, which need to be transmitted to the other DERs depending on the application. This transmission can be done when the DERs are allocated resources by the gNodeB, where these resources are the means to support the communication links. In 5G, these resources are termed resource blocks (RB) and RBs are

allocated based on the number of packets a DER wants to transmit, which is known as that DER's buffer status report (BSR). The operation of the co-simulation is explained below:

- Each DER transmit its BSR status to the Python-based gNodeB at an interval of τ. This allows gNodeB to get the buffer status, i.e., packets pending to be sent from each DER and this information is used in scheduling.
- Based on BSR, gNodeB allocates RBs to DERs using a pre-defined resource allocation policy π at each transmission time interval (TTI). Based on the allocated RBs, DERs in MATLAB and PSCAD are allowed to transmit their information, i.e., pass their values to the Python simulator. This information exchange between PSCAD and Python script is done through writing the values to a text file [89].
- At each TTI, time is synchronized between the communication and power systems by allowing both of them to simultaneously step through the same duration of time.

The general design of the co-simulator is shown in Fig. 4.3. More details are discussed in Section 4.3.

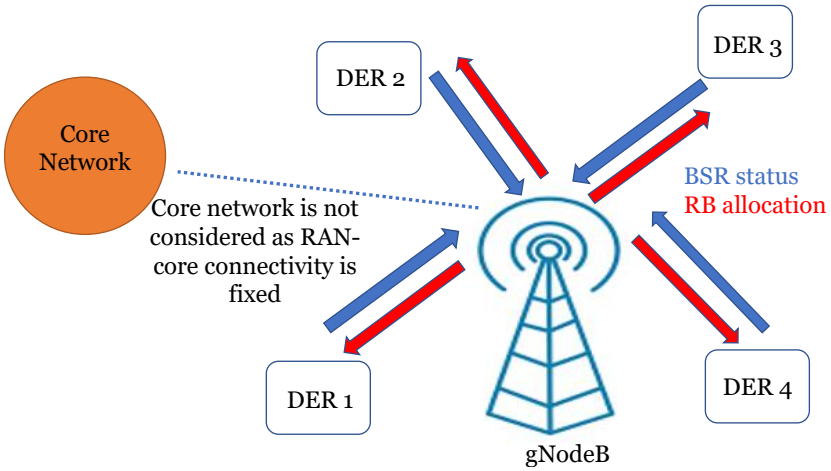


Figure 4.2: Schematic diagram of a 5G Radio Access Network

4.3 Communication System Design

For the RB allocation, a round-robin policy π is used in our case where equal number of RBs are granted to each DER in a circular fashion, i.e., RBs are allocated to each DER in the

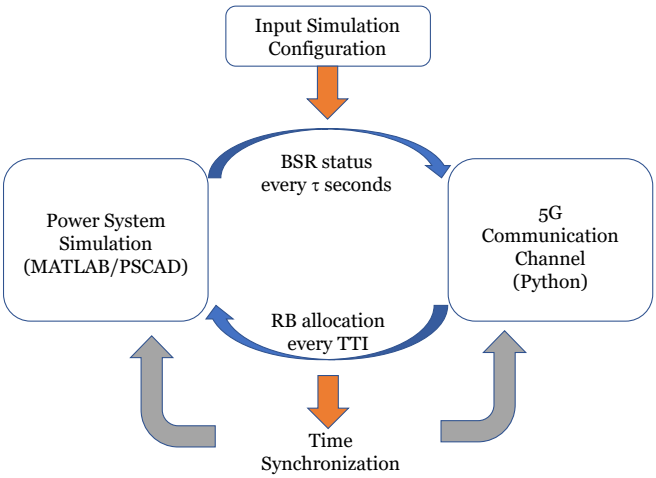


Figure 4.3: Design of the proposed co-simulator.

same sequence. Due to the changing channel conditions, each DER transmits channel state information reference signal (CSI-RS) to gNodeB which is an indicator of channel quality. This indicator is called channel quality index (CQI) and provides information about the modulation order that can be used, where modulation order is the number of symbols that can be transmitted. Higher CQI allows the use of higher modulation order that translates to higher throughput. The channel is modeled as a random time-varying channel which results in varying modulation orders over the allocated RBs. The total number of RBs are M and based on the RB allocation, the throughput achieved in Mbps in each of the communication link is calculated as per 3GPP TS 38.306 [3] and shown in Eqn. (4.1).

$$Throughput = 10^{-6} \sum_{j=1}^{J} (v_{\text{Layers}}^{(j)} Q_m^{(j)} f^{(j)} R_{\max} \frac{12 N_{PRB}^{BW(j),\mu}}{T_s^{\mu}} (1 - OH^j)) \tag{4.1}$$

J denotes the number of carriers aggregated when carrier aggregation is applicable. 5G allows a maximum of 16 carriers to be aggregated. v_{Layers}^{j} denotes the maximum number of layers in the jth component carriers (CC), and is usually the number of streams. CC is restricted by the number of antennas used. $Q_m^{(j)}$ refers to the modulation order which is dependent on the channel quality as decided by the CQI. $f^{(j)}$ is the scaling factor used to scale throughput for various CC combinations. $R_{\max}$ is the maximum allowed coding rate and is set at $\frac{948}{1024}$ as per [13]. $N_{\text{PRB}}^{BW(j),\mu}$ is the number of RBs allocated to a single DER during the scheduling. μ defines the 5G numerology selected, and this numerology decides

the symbol time, T_s^μ, calculated as $T_s^\mu = \frac{10^{-3}}{14.2^\mu}$. Finally, the overhead $OH(j)$ for carrier j is decided by the frequency band used (FR1 or FR2). The communication between the gNodeB and DERs occur in the the FR1 band [41] (410 MHz to 7.125 GHz) and the allocated channel has a bandwidth of B. The throughput achieved as per Eqn. (4.1) decides the time instants when the packets are received. Table 4.1 all the key communication parameters used.

Note that the BSR and CQI reports are assumed to be sent out of band, i.e, without the need of resources for transmission and are also assumed to be received without any errors at the gNodeB [12]. Therefore the gNodeB is always aware of the BSR and CQI at each TTI. The 5G RAN is usually connected to a core network (CN). The CN is a fiber cable-based network due to which its delay and reliability are fixed and do not vary [120]. Therefore, the CN is not implemented and the focus in on the RB allocation at the RAN so that the network performance is investigated from the perspective of RB allocation.

Table 4.1: Parameters used to set up the simulation

Parameter	**Value**
Aggregated carriers J	2
Modulation order $Q_m^{(j)}$	2, 4, 6, 8
Maximum layers for jth carrier $v_{\text{Layers}}^{(j)}$	2
Scaling factor $f^{(j)}$	0.8
Numerology μ	2
Number of RBs M	3
RBs allocated per DER $N_{\text{PRB}}^{\text{BW(j)}}$	1
Scheduling policy π	Round robin
BSR periodicity τ	1 ms
Transmission time interval TTI	1 ms
Carrier frequency f_D	2.63 GHz
Bandwidth B	5 MHz
Packet size L	150 Bytes [7]

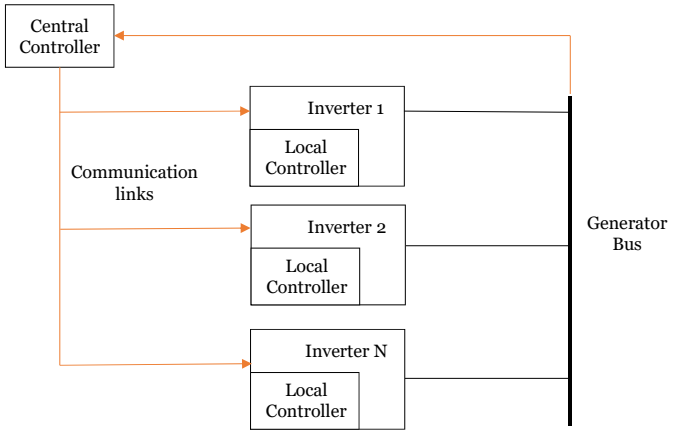


Figure 4.4: Power park system model.

4.4 Use Case I: Power Park

4.4.1 System Description

This case involves distributed control of N inverters connected to a common bus [96], as shown in Fig. 4.4. The idea of frequency partitioning is employed to distribute the control tasks among the central controller (remote) and each inverter's distributed controller, because it is helpful in minimizing the communication bandwidth required between the remote and the local controllers. The partition works by allowing the low frequency part of the control signals to be provided by the central controller and the local controllers provide the high frequency part. The central controller's reference signal is represented as $y^{\star}$. As the system desired disturbance rejection at high frequencies, the reference signals for the local controller is set at zero. The test system is designed with three inverters and each of them has an inner current control loop embedded within an outer voltage control loop. At each inverter, the 5G network is used to transmit the control signals between the central and remote controllers. Reference [96] explain this system in more detail.

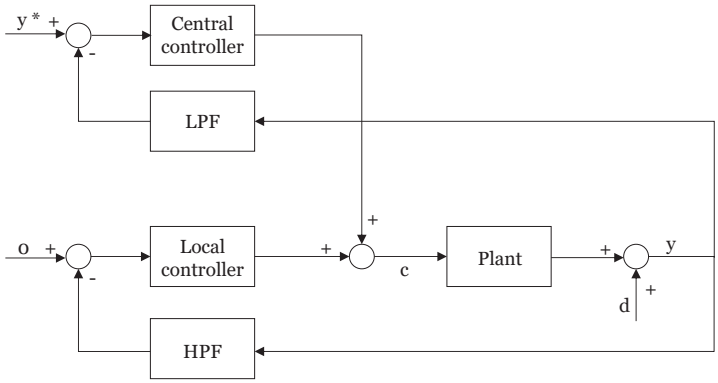


Figure 4.5: Frequency partitioning for inverter control.

4.4.2 Performance Evaluation

Set point change in voltage at the generator bus

The results will evaluate the common generator bus's voltage regulation in the dq-frame of reference. V_{oq} is set to zero and a step change is applied in V_{od} from 0.2 to 0.3 at $t = 0.07$ s. The system response is shown in Fig. 4.6. $V_{\text{od-ideal}}$ denotes the system simulation response under ideal conditions (5G channel is not used). $V_{\text{od-5G}}$ denotes the system response with 5G communication channel. Table 4.3 shows the overshoot and settling time of the responses. The settling times for both the cases are around 0.01 s. In the case of 5G, the overshoot changes from 1% to 10.3%.

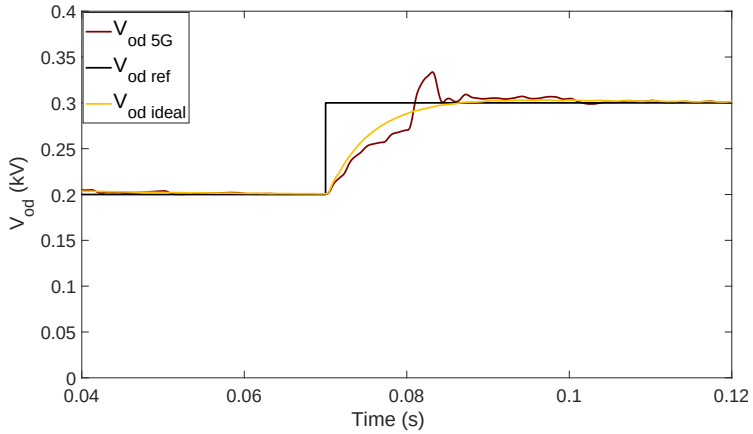


Figure 4.6: Comparison of the system response (d-component of the voltage at the generator bus) to a step change in set-point with and without the 5G network.

Table 4.2: Overshoot and Settling time for set point change in voltage at the generator bus (Power park case)

Case	Overshoot (%)	Settling Time (ms)
Ideal	1	11.1
5G	10.3	14.19

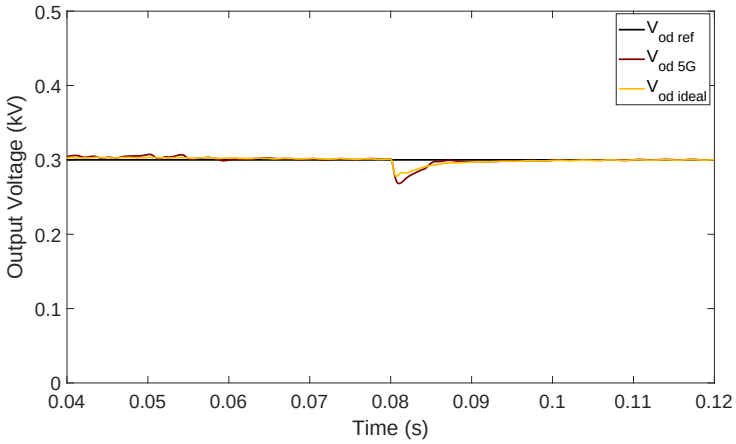


Figure 4.7: Load switching of a three phase balanced RL load. Output voltage in dq frame (d-axis component).

Load switching

Here the performance will be analyzed for a load switching application. A three-phase balanced RL load is switched on and is connected to the system at $t = 0.08$ s, and the resulting transient response is shown in Fig. 4.7. The results show very similar transient performance for both cases. The transient behavior lasts for 5 ms in both the cases and the system does not lose its stability in either of them. When 5G is used,the drop in voltage is around 0.01 V more.

Single-phase to ground fault at the generator bus

A single-phase A-G fault is simulated at $t = 0.08$ s for 0.03 s at the generator bus and the results are in Fig. 4.8. It should be seen that the application doesn't lose its stability under 5G and is able to continue to track the reference value after the fault is cleared. The transient responses with 5G and ideal communication last for 0.065 s and 0.028 s respectively, after the fault is cleared.

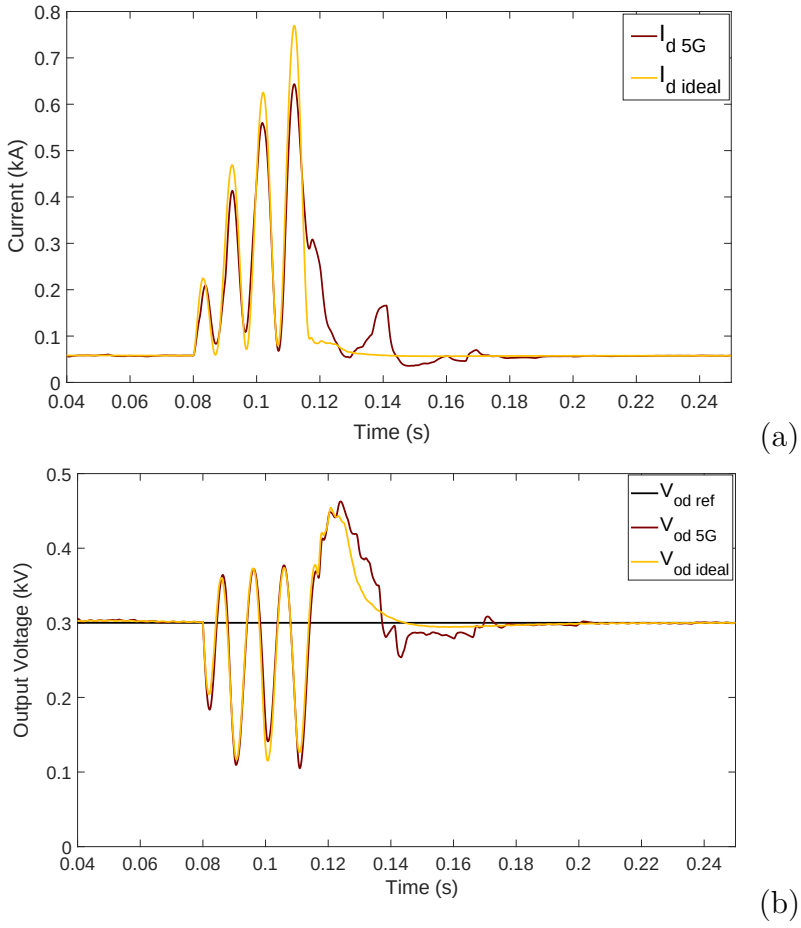


Figure 4.8: Single-phase fault to ground (a) d-component of the output current of one of the inverters in dq-frame. (b) d-component of the output voltage at the generator bus in dq-frame.

Three-phase to ground fault at the generator bus

A three-phase ABC-G fault is simulated at $t = 0.08$ s for 0.03 s at the generator bus. Fig. 4.9 shows the results. The system with 5G network does not lose its stability. It is able to continue tracking the reference value of the direct axis voltage at the generator bus after the three-phase to ground fault is cleared. The transient responses with 5G and ideal communication last for 0.13 s and 0.1 s respectively, after the fault is cleared.

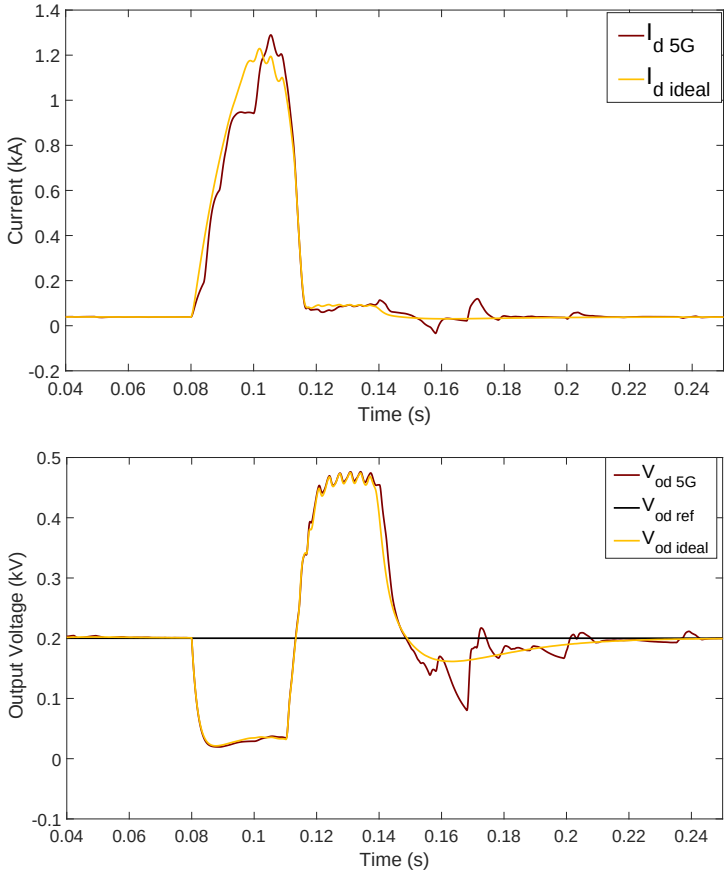


Figure 4.9: Three-phase fault to the ground (a) d-component of the output current of one of the inverters in dq-frame. (b) d-component of the output voltage at the generator bus in dq-frame.

4.5 Use Case II: Coordinated Set Point Modulation of DERs

4.5.1 System Description

In setpoint modulation, the system set-point x_{sp} is modulated depending on the system response $x(t)$ [83]. For a single DER, this is given as

$$x'_{sp} = x_{sp} + me(t), \tag{4.2}$$

where x'_{sp} is the modulated set point, m is the design parameter, and $e(t)$ is the tracking error measured as the deviation of of the system response from the set point.

$$e(t) = x_{sp} - x(t). \tag{4.3}$$

For better dynamic response, the tracking error $e(t)$ is replaced by the the predictive dynamic behaviour of the tracking error. That is,

$$x'_{sp} = x_{sp} + m\hat{e}_{pred}(t). \tag{4.4}$$

Another level of control can be added with the help of communication between different DERs and this is known as coordinated set point modulation [110]. Consider N devices connected at the point of common coupling (PCC) that are capable of exchanging information over a communication channel. The devices that are connected through a communication link exchange information about their respective predictive tracking errors. The existing equation for a single DER can be modified to represent the system response for the ith DER as

$$x'_{i_{sp}} = x_{i_{sp}} + m_i\hat{e}_{i_{pred}}(t) \tag{4.5}$$

The secondary level of control $u(t)$ can then be defined as:

$$u(t) = m_i \sum_{j=1, j\neq i}^{N} a_{ij}\hat{e}_{j_{pred}}(t) \tag{4.6}$$

where a_{ij} denotes if there is a communication link between DER i and DER j. Hence the

control equation can be represented by

$$x'_{i_{sp}} = x_{i_{sp}} + m_i \hat{e}_{i_{pred}}(t) + m_i \sum_{j=1, j\neq i}^{N} a_{ij} \hat{e}_{j_{pred}}(t) \tag{4.7}$$

4.5.2 Performance Evaluation

Fig. 4.10 shows the three devices (DERs) that interact with each other using coordinated set-point modulation.

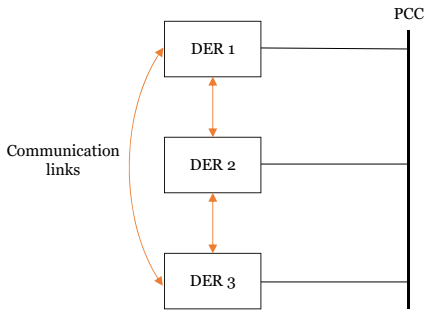


Figure 4.10: System model of coordinated set point modulation of DERs.

Staggered set point change

A staggered set point change is applied to the individual devices. They are as follows - step change in reference point from 0 to 1 at $t = 0.5$ s for DER 1, at $t = 1$ s for DER 2 and at $t = 1.5$ s for DER 3. These set points can represent any electrical quantity that needs control (voltage, real power or reactive power). Fig. 4.11 shows the combined system response at the PCC, under an ideal communication network and a 5G network respectively. Table 4.3 shows the overshoot and the settling time of the system response under the two conditions. While the settling time degrades by 172 ms and the overshoot almost doubles as compared to the ideal communication, the DERs are still able to track the set point changes with 5G.

Simultaneous set point change

A simultaneous set point change is applied to all the devices such that the step change in reference point happens from 0 to 1 at $t = 0.5$ s. This is then brought back to 0 for all the

devices at $t = 2$ s. The overall response at the PCC, under ideal simulation conditions and in the presence of a 5G network, is shown in Fig. 4.12 and Table 4.3 provides the overshoot and the settling time. Even though the settling time increases by 84 ms and the overshoot is higher by 11.4% under the 5G network, the set points are tracked successfully without loss in stability.

Communication failure in one of the devices

Such a scenario is seen when one of the devices is incapable of communicating its state to the other two devices. The response is shown in Fig. 4.13. Compared to ideal communication, although there is an increase in overshoot and settling time, the system is able to track the set points owing to the control algorithm used. This can be seen from the first two terms in Eqn. (4.7). Hence the results obtained in this study confirms that 5G can support the stability of smart grid operations for Power Park and Set Point Modulation, where the stability is measured as overshoot and settling time. This offers an attractive alternative to provide communication in smart grids as the traditional manner of providing connectivity in smart grid using fiber optic cables is expensive and doesn't scale well.

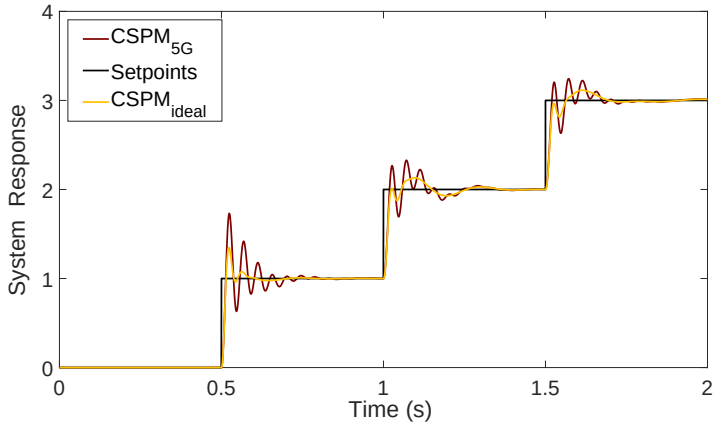


Figure 4.11: Comparison of system response for coordinated set point modulation (staggered set point change) with and without 5G network.

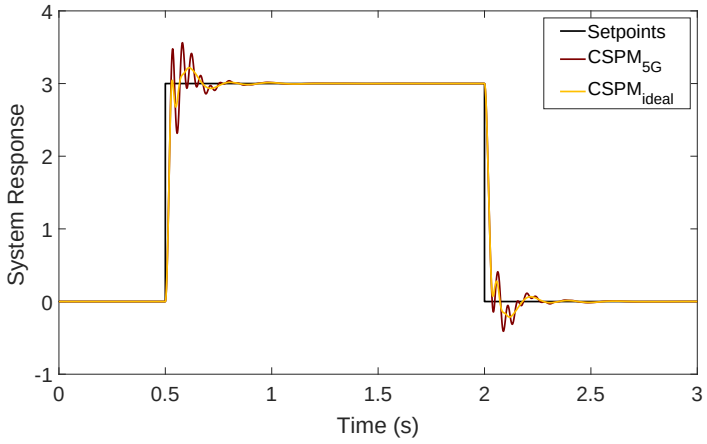


Figure 4.12: Comparison of system response for coordinated set point modulation (simultaneous set point change) with and without the 5G network.

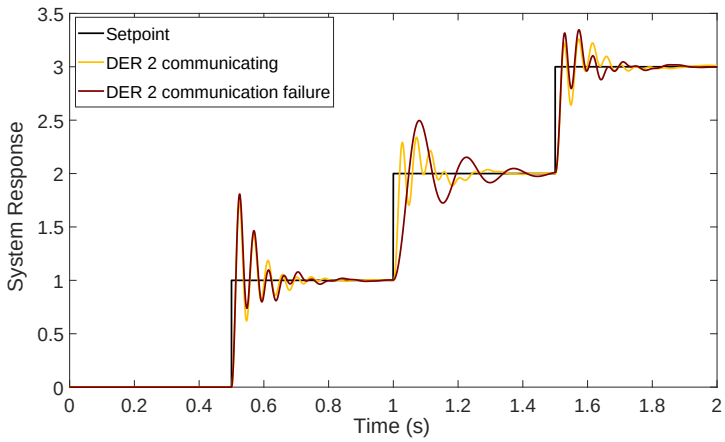


Figure 4.13: Comparison of system response for coordinated set point modulation (staggered set point change) under ideal communication conditions and when communication for device 2 fails.

Table 4.3: System Performance for set-point changes (Coordinated set-point modulation case).

Setpoint change method	**Case**	**Overshoot (%)**	**Settling Time (ms)**
Staggered	Ideal	34	78
	5G	73.1	250
Simultaneous	Ideal	7.2	153
	5G	18.6	237

To conclude, the effects of delays induced by 5G scheduling on the performance of smart grids have been investigated. Results from the two cases considered show that the grid is able to track the set point changes and handle transients in the presence of a 5G communication network. There is a slight degradation in the system performance as compared to ideal simulation values without the presence of a 5G network, but the system still maintains stability. Once the capability of 5G in supporting smart grid operations is verified, we then focus on identifying how to further improve the smart grid performance by improving the network performance in terms of improved AoI.

4.6 Distributed Control Methodology

The distributed control among the smart grid and the impact of communication on it will be evaluated based on the coordinated set point modulation technique explained in Sec. 4.5, and more details about this concept can be found in [110].

4.7 Communication Network Description

Communication among the DERs is enabled by a 5G network which provides wireless coverage to the micorgrid. The communication links between the DERs are represented in an adjacency matrix A, where $a_{ij} = 1$ means DER i sends packets containing its local measurements to DER j, and $a_{ij} = 0$ means that DERs i and j do not communicate. The packet

generated at DER i has to be first uploaded to the gNodeB. Then, in the downloading step, this packet is transferred from the BS to DER j. An example of an adjacency matrix is shown in Table 4.4, and the corresponding network is shown in Fig. 4.14. The set of DERs that generate packets to be uploaded to the BS are the source DERs and denoted as G. The DERs receiving these packets are the destination DERs and denoted as R. Note that the same DER may be in both R and G which means it is both a source and a destination.

4.7.1 5G Communication Network Model

Frequency division duplex (FDD) configuration is used which means both the downlink and uplink transmission happens independently at the same time over different frequencies. Under FDD, the allocated frequency is split into two parts. One part is used for the uplink communications and the other part is used for the downlink communications. The BS is in charge of scheduling to decide which DERs are allowed to upload and download packets. The uplinks are used to support the uploading of the packets generated at the G DERs to the BS. Then the downlinks are used in downloading these packets from the BS to the R DERs.

In 5G, time domain is equally slotted into transmission time intervals (TTI). Similarly in the frequency domain, bandwidth, i.e., range of radio wave frequencies allocated, is equally partitioned into orthogonal subcarriers. These subcarriers together carry the packets. Scheduling for transmission is performed for each TTI. A group of 12 subcarriers over a TTI is called a resource block (RB). The BS maintains two RB pools corresponding to the uplinks and downlinks [2], and at each TTI, these RBs have to be allocated to the DERs:

- B_u RBs for uploading packets from DERs in G to the BS. The maximum number of uplinks possible is $|G|$, where $||$ denotes cardinality. Therefore, the scheduler needs to assign B_u RBs among the $|G|$ possible uplinks.

- B_d RBs for downloading packets from the BS to the DERs in R. However, a particular DER $i \in R$ may be receiving packets from $\bar{i}$ other DERs where $\bar{i} \geq 1$. For example, a DER may need voltage measurements from 2 different DERs as input. Therefore the set of all downlink paths from the BS is given by $\bar{R}$ where $|\bar{R}| = \sum_{i=1}^{R} \bar{i}$ and $|\bar{R}| \geq |R|$. Therefore the scheduler needs to assign B_d RBs to the $|\bar{R}|$ possible downlinks.

Table 4.4: Example adjacency matrix representing communication links between DERs

	DER 1	DER 2	DER 3
DER 1	0	1	0
DER 2	1	0	0
DER 3	1	1	0

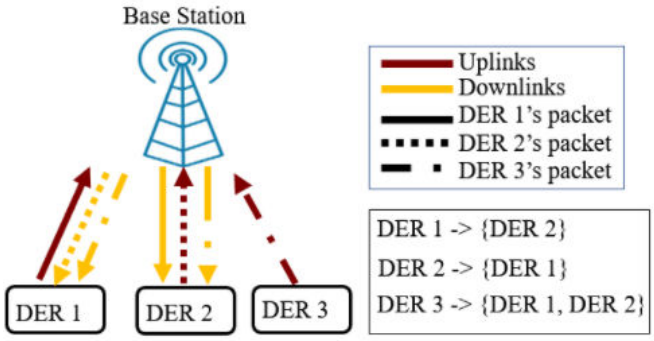


Figure 4.14: Network resulting from adjacency matrix in Table 4.4.

In the example shown in Table 4.4, $N = 3$, $G = 3$, $R = 2$, $\bar{R} = 4$, and the resulting scenario is shown in Fig. 4.14. All the three DERs upload their packets to the BS, after which packets from DER 1 are downloaded by DER 2, packets from DER 2 are downloaded by DER 1, and packets from DER 3 are downloaded by both DERs 1 and 2.

4.7.2 Channel Conditions

The channel is modeled as a time varying Rayleigh fading channel with path loss exponent γ. The quality of a wireless channel is measured using channel state information (CSI). Employing the minimum mean square channel estimation error technique [56], the perfect channel state information (CSI) between the sender and the receiver is calculated as

$$h^i_{\tau,\mathrm{f}} = \hat{h}^i_{\tau,\mathrm{f}} + e^i_{\tau,\mathrm{f}} \tag{4.8}$$

where i denotes the DER that is allocated the RB (τ, f), i.e., RB corresponding to TTI τ and frequency f. $\hat{h}^i_{\tau,\mathrm{f}} \sim \mathcal{CN}(0, 1-\sigma_e^2)$ and $e^i_{\tau,\mathrm{f}} \sim \mathcal{CN}(0, \sigma_e^2)$ are the estimated CSI and error in estimating CSI [65] where $\mathcal{CN}$ represent a complex normal distribution. Assuming perfect knowledge of the CSI, the actual signal-to-noise ratio (SNR) is calculated as

$$\mathrm{SNR}^{i}_{\tau,\mathrm{f}} = \frac{|h^{i}_{\tau,\mathrm{f}}|^2 p^{i}_{\tau,\mathrm{f}} d^{-\gamma}}{\sigma_o^2}, \tag{4.9}$$

where $p^{i}_{\tau,\mathrm{f}}$ is the transmission power, d is the distance, and σ_o is the noise power. A transmission is successful if

$$\mathrm{SNR}^{i}_{\tau,\mathrm{f}} \geq \mathrm{SNR}_{\mathrm{th}}, \tag{4.10}$$

where $\mathrm{SNR}_{\mathrm{th}}$ is the minimum SNR constraint per RB [65, 66]. $\mathrm{SNR}_{\mathrm{th}}$ depends on the service type, and modulation and coding scheme (MCS). smart gridss are served under ultra reliable low latency communications (URLLC) category. For URLLC service and assuming modulation and coding scheme (MCS) = 4, $\mathrm{SNR}_{\mathrm{th}}$ is fixed at 7 dB as this guarantees block error probability of 10^{-5} [65, 66]. $\mathrm{SNR}^{u}_{\mathrm{t,f}} < \mathrm{SNR}_{\mathrm{th}}$ causes a transmission failure. In such cases, the same packet is retransmitted immediately in the next TTI [93].

4.7.3 Traffic Model

Traffic generation at the DERs is modeled as generate-at-will [19], where the DERs generate a fresh packet whenever they are selected by the scheduler to upload their packets to the BS. This is practically justified because we consider the 5G network under numerology $\mu = 0$. This numerology decides the waveform configuration, and different numerologies lead to different parameters like subcarrier spacing/symbol time and cyclic prefix. Using numerology $\mu = 0$ leads to TTI = 1 ms, whereas the DERs generate new packets at a much faster rate (order of microseconds). Therefore, the transmitted packet is always the most recently generated packet and this corresponds to the generate-at-will packet generation model.

4.7.4 Resource Block Allocation Process

The RB allocation process is motivated from [93]. A packet generated by DER i needs B_i RBs to be successfully transmitted. So, it takes B_i RBs from B_u to fully upload DER i's packet to the BS, and another B_i RBs from B_d to fully download DER i's packet to DER j. Therefore, some transmissions take more than 1 TTI if the previous transmissions did not complete as a result of insufficient number of RBs being allocated. At TTI τ, the number of

RBs needed to complete uploading DER i's current packet is denoted by $B_i^{\text{need}}(\tau)$, and the scheduler allocates $B_i^{\text{alloc}}(\tau)$ RBs, where $B_i^{\text{need}}(\tau) \geq B_i^{\text{alloc}}(\tau)$. In case the packet is new, and the scheduler decides to transmit the whole packet in a single TTI, $B_i^{\text{alloc}}(\tau) = B_i^{\text{need}}(\tau) = B_i$. Similarly in the downloading scenario, the number of RBs needed to finish downloading DER i's current packet to DER j in TTI τ is $B_{i,j}^{\text{need}}(\tau)$, while the RBs allocated is $B_{i,j}^{\text{alloc}}(\tau)$.

Note that when $B_i^{\text{need}}(\tau) > B_i^{\text{alloc}}(\tau)$ or $B_{i,j}^{\text{need}}(\tau) > B_{i,j}^{\text{alloc}}(\tau)$, the transmission is incomplete. The number of RBs needed in the next TTI is $B_i^{\text{need}}(\tau + 1) = B_i^{\text{need}}(\tau) - B_i^{\text{alloc}}(\tau)$ and $B_{i,j}^{\text{need}}(\tau + 1) = B_{i,j}^{\text{need}}(\tau) - B_{i,j}^{\text{alloc}}(\tau)$ for the uplink and downlink. The packet is not fully uploaded until $B_i^{\text{need}}(\tau) = 0$ and not fully downloaded until $B_{i,j}^{\text{need}}(\tau) = 0$, after which the next packet is considered for uploading/downloading. The RB allocation process explained above is constrained by the total number of available RBs in B_u and B_d.

4.7.5 Age of Information (AoI)

AoI is a new metric that is increasingly used to quantify the freshness of information at the receiver, which is measured as the time elapsed since the receiver received a packet. Mathematically, it is equal to the delay experienced by the packet at instants of packet reception and increases linearly during the inter-arrival time. At TTI τ, the AoI for DER i at the BS is denoted by $\text{AoI}_i(\tau)$, and AoI of DER i at DER j is denoted as $\text{AoI}_{i,j}(\tau)$, where $i \in G$ and $j \in R$. The general expression for AoI_i is

$$\text{AoI}_i(\tau) = \tau - g_{u,i}, \tag{4.11}$$

where τ is the ongoing TTI, and $g_{u,i}$ is the generation time of the most recently uploaded packet of DER i that is received successfully at the BS. Similarly, $\text{AoI}_{i,j}(\tau)$ is calculated as

$$\text{AoI}_{i,j}(\tau) = \tau - g_{d,i}, \tag{4.12}$$

where τ is the ongoing TTI, and $g_{d,i}$ is the generation time of the most recently downloaded packet of DER i that is received successfully at DER j. For DERs that do not communicate, i.e., $a_{i,j} = 0$ as per the adjacency matrix, $\text{AoI}_{i,j}(\tau) = 0$. The time average AoI of the source DERs G at the BS is

$$\mathrm{AoI}^{\mathrm{BS}}(T) = \sum_{\tau=1}^{T} \sum_{i=1}^{G} \mathrm{AoI}_i^{BS}(\tau). \tag{4.13}$$

The freshness of information is measured as the sum AoI of all the source DERs at the destination DERs:

$$\mathrm{AoI}(T) = \sum_{\tau=1}^{T} \sum_{i=1,j=1}^{i=G,j=R} \mathrm{AoI}_{i,j}(\tau) = \sum_{\tau=1}^{T} \sum_{r=1}^{\bar{R}} \mathrm{AoI}_k(\tau). \tag{4.14}$$

To improve the information freshness at the destination DERs, $\mathrm{AoI}(T)$ is minimized with the constraint of the number of available RBs:

$$\begin{aligned} &\min \ \mathrm{AoI}(T) \\ &\text{such that} \sum_{i\in G} B_i^{\mathrm{alloc}}(\tau) \leq B_u, \quad \tau = 1, 2, \ldots, T \\ &\qquad \sum_{i\in G, j\in R} B_{i,j}^{\mathrm{alloc}}(\tau) \leq B_d, \quad \tau = 1, 2, \ldots, T \end{aligned} \tag{4.15}$$

where the constraints restrict the number of RB allocations to the total RBs available. At each TTI τ, the scheduler has to choose the following:

- Number of RBs allocated to each DER in the UL. This is represented as $B_U^{\mathrm{alloc}}(\tau) = \cup\, B_i^{\mathrm{alloc}}(\tau)\ \forall i \in G$.

- Number of RBs allocated to each DER in the DL. This is represented as $B_D^{\mathrm{alloc}}(\tau) = \cup\, B_{i,j}^{\mathrm{alloc}}(\tau)\ \forall i \in G$ and $j \in R$.

4.8 Studied Schedulers

In this section, different schedulers will be evaluated based on their performance in terms of AoI minimization and stability of the smart grid. The schedulers are categorized into two broad classes of AoI-based schedulers and non-AoI-based schedulers.

4.8.1 AoI-Based Schedulers

Maximal Age Difference (MAD) Scheduler

The MAD scheduler prioritizes DERs with highest AoI at the BS for uploading, and the highest AoI difference between the BS and destination for downloading. It is explained in Algorithm 6. The algorithm takes the current AoI at the BS and destinations for all the DERs as inputs. During uploading (line 3), the DERs are ordered in decreasing order of their AoI at the BS. Based on the ordering, RBs from the B_u pool are allocated in lines 4–7. Then in the downlink, lines 8–12 orders the DERs with the highest AoI difference. They are then allocated RBs from the B_d pool. The AoI difference used in the downloading is given as

$$\text{AoI}_{i,j}^{\text{diff}}(\tau) = \text{AoI}_{i,j}(\tau) - \text{AoI}_{i}^{\text{BS}}(\tau). \tag{4.16}$$

Algorithm 6 MAD scheduling at TTI τ

Input: $\text{AoI}_i(\tau)$ and $\text{AoI}_{i,j}(\tau)\ \forall\ i \in G$ and $j \in R, B_u, B_d,\ B_i^{\text{need}}(\tau)\ \forall\ i \in G,\ B_{i,j}^{\text{need}}(\tau)\ \forall\ i \in G$ and $j \in R$.
Output: $B_U^{\text{alloc}}(\tau), B_D^{\text{alloc}}(\tau)$.

1: $B_u^{\text{avail}}(\tau) = B_u, B_d^{\text{avail}}(\tau) = B_d$
2: **for** $l = 1, 2, ..$ **do**
3: **for** $i \in G$ with lth highest $\text{AoI}_i(\tau)$ **do**
4: **if** $B_u^{\text{avail}}(\tau) > 0$ **then**
5: $B_i^{\text{alloc}}(\tau) = \min(B_u^{\text{avail}}(\tau), B_i^{\text{need}}(\tau))$
6: $B_u^{\text{avail}}(\tau) = B_u^{\text{avail}}(\tau) - B_i^{\text{alloc}}(\tau)$
7: $B_U^{\text{alloc}}(\tau) = B_U^{\text{alloc}}(\tau) \cup B_i^{\text{alloc}}(\tau)$
8: **for** (i, j) $i \in G$, $j \in R$ with lth highest $\text{AoI}_{i,j}^{\text{diff}}(\tau)$ **do**
9: **if** $B_d^{\text{avail}}(\tau) > 0$ **then**
10: $B_{i,j}^{\text{alloc}}(\tau) = \min(B_u^{\text{avail}}(\tau), B_i^{\text{need}}(\tau))$
11: $B_d^{\text{avail}}(\tau) = B_d^{\text{avail}}(\tau) - B_{i,j}^{\text{alloc}}(\tau)$
12: $B_D^{\text{alloc}}(\tau) = B_D^{\text{alloc}}(\tau) \cup B_{i,j}^{\text{alloc}}(\tau)$

Maximum Age First (MAF) Scheduler

Here, the input to the algorithm is the current AoI at the BS and destinations for all the DERs. For scheduling, the DERs with the highest AoI at both the BS and the destinations

are prioritized. See Algorithm 7. As per line 3, the DERs are ordered in decreasing manner based on their AoI at the BS for the uploading step after which RBs from the B_u pool are allocated as shown in lines 4–7. Then in the downlink, RBs are allocated from the B_d pool to the DERs with the highest AoI at the destination as per lines 8–12.

Algorithm 7 MAF scheduling at TTI τ

Input: $\text{AoI}_i(\tau)$ and $\text{AoI}_{i,j}(\tau)\ \forall\ i \in G$ and $j \in R, B_u, B_d,\ B_i^{\text{need}}(\tau)\ \forall\ i \in G,\ B_{i,j}^{\text{need}}(\tau)\ \forall\ i \in G$ and $j \in R$.
Output: $B_U^{\text{alloc}}(\tau), B_D^{\text{alloc}}(\tau)$.

1: $B_u^{\text{avail}}(\tau) = B_u, B_d^{\text{avail}}(\tau) = B_d$
2: **for** $l = 1, 2, ..$ **do**
3: **for** $i \in G$ with lth highest $\text{AoI}_i(\tau)$ **do**
4: **if** $B_u^{\text{avail}}(\tau) > 0$ **then**
5: $B_i^{\text{alloc}}(\tau) = \min(B_u^{\text{avail}}(\tau), B_i^{\text{need}}(\tau))$
6: $B_u^{\text{avail}}(\tau) = B_u^{\text{avail}}(\tau) - B_i^{\text{alloc}}(\tau)$
7: $B_U^{\text{alloc}}(\tau) = B_U^{\text{alloc}}(\tau) \cup B_i^{\text{alloc}}(\tau)$
8: **for** (i, j) $i \in G$, $j \in R$ with lth highest $\text{AoI}_{i,j}(\tau)$ **do**
9: **if** $B_d^{\text{avail}}(\tau) > 0$ **then**
10: $B_{i,j}^{\text{alloc}}(\tau) = \min(B_u^{\text{avail}}(\tau), B_i^{\text{need}}(\tau))$
11: $B_d^{\text{avail}}(\tau) = B_d^{\text{avail}}(\tau) - B_{i,j}^{\text{alloc}}(\tau)$
12: $B_D^{\text{alloc}}(\tau) = B_D^{\text{alloc}}(\tau) \cup B_{i,j}^{\text{alloc}}(\tau)$

4.8.2 Non-AoI–Based Schedulers

Under non-AoI based schedulers, traditional schedulers which are generally used for 5G are considered. They are generally defined for a single hop. However, our scenario involves two hops where the first hop is between the source DERs to the BS, and second hop is between the BS and destination DERs. Each hop is treated independently, and the scheduler works independently across both the hops.

Proportional Fairness (PF) Scheduler

As wireless networks are characterized by time-varying channel conditions, the PF algorithm is designed to take advantage of multi-user diversity, while maintaining comparable long-term throughput for all users [60]. Let $R_i(\tau)$ denote the instantaneous data rate that DER i can achieve at TTI τ, and $T_i(\tau)$ be the average throughput for DER i up to TTI τ in the uplink.

The PF scheduler selects the DERs with the highest relative channel quality according to the metric calculated as $R_i(\tau)/T_i(\tau)$ for uploading. Similarly, RB allocation to the DERs in the downlink is decided by $R_{i,j}(\tau)/T_{i,j}(\tau)$. $R_{i,j}(\tau)$ denotes the instantaneous data rate possible in the downlink channel between DER i and DER j in TTI τ, and $T_{i,j}(\tau)$ denotes the average throughput in that downlink. This is explained in Algorithm 8. Once the RBs have been allocated, the average throughput is updated as

$$T_i(\tau+1) = \begin{cases} (1-\beta)T_i(\tau) + \beta R_i(\tau), B_i^{\text{alloc}}(\tau) = B_i^{\text{need}}(\tau) \\ (1-\beta)T_i(\tau), B_i^{\text{alloc}}(\tau) \neq B_i^{\text{need}}(\tau) \end{cases} \tag{4.17}$$

and

$$T_{i,j}(\tau+1) = \begin{cases} (1-\beta)T_{i,j}(\tau) + \beta R_{i,j}(\tau), B_{i,j}^{\text{alloc}}(\tau) = B_{i,j}^{\text{need}}(\tau) \\ (1-\beta)T_{i,j}(\tau), B_{i,j}^{\text{alloc}}(\tau) \neq B_{i,j}^{\text{need}}(\tau) \end{cases} \tag{4.18}$$

Algorithm 8 PF scheduling at TTI τ

Input: $R_i(\tau)$, $T_i(\tau), R_{i,j}(\tau)$ and $T_{i,j}(\tau)$ $\forall$ $i \in G$ and $j \in R, B_u, B_d$, $B_i^{\text{need}}(\tau)$ $\forall$ $i \in G$, $B_{i,j}^{\text{need}}(\tau)$ $\forall$ $i \in G$ and $j \in R$.
Output: $B_U^{\text{alloc}}(\tau), B_D^{\text{alloc}}(\tau)$.

$B_u^{\text{avail}}(\tau) = B_u, B_d^{\text{avail}}(\tau) = B_d$
for $l = 1, 2, ..$ **do**
  for $i \in G$ with lth highest $R_i(\tau)/T_i(\tau)$ **do**
    if $B_u^{\text{avail}}(\tau) > 0$ **then**
      $B_i^{\text{alloc}}(\tau) = \min(B_u^{\text{avail}}(\tau), B_i^{\text{need}}(\tau))$
      $B_u^{\text{avail}}(\tau) = B_u^{\text{avail}}(\tau) - B_i^{\text{alloc}}(\tau)$
      $B_U^{\text{alloc}}(\tau) = B_U^{\text{alloc}}(\tau) \cup B_i^{\text{alloc}}(\tau)$
  for (i, j) $i \in G$, $j \in R$ with lth highest $R_{\text{i,j}}(\tau)/T_{\text{i,j}}(\tau)$ **do**
    if $B_d^{\text{avail}}(\tau) > 0$ **then**
      $B_{i,j}^{\text{alloc}}(\tau) = \min(B_u^{\text{avail}}(\tau), B_i^{\text{need}}(\tau))$
      $B_d^{\text{avail}}(\tau) = B_d^{\text{avail}}(\tau) - B_{i,j}^{\text{alloc}}(\tau)$
      $B_D^{\text{alloc}}(\tau) = B_D^{\text{alloc}}(\tau) \cup B_{i,j}^{\text{alloc}}(\tau)$

Round Robin (RR) Scheduler

Under RR, the available RBs are assigned in equal and circular fashion among the DERs [87]. It does not take channel condition into account while allocating RBs.

Finally, a random scheduler where a random set of DERs is selected to be uploaded and

downloaded along with an ideal scheduler that communicates without any delays are used to compare the performances.

4.9 Performance Evaluation

The final performance between the schedulers in terms of the AoI and smart grid stability will be analyzed in this section. A rectangular region of $l \times b$ is considered, where the BS is located at $(l/2, b/2)$ and the DERs are located randomly in that rectangular region. To model the path loss realistically, the path loss exponent γ is set as 3.76 per third generation partnership project (3GPP) urban path loss model [65]. We consider a 5G numerology of $\mu = 0$. The two test systems are designed so that system I has 3 DERs while the system II has 28 DERs. The number of RBs in the uplink and downlink for test systems I and II are $B_u = 2, B_d = 3$ and $B_u = 10, B_d = 20$ respectively. The size of the packets from different DERs vary such that the number of RBs needed to completely transmit a single packet for DER i, i.e., B_i, vary from 1 to 3. These parameters are listed in Table 4.5. The channels, traffic generation and RB allocation are modeled as described in Section 4.3.

The two systems employ distributed control of DERs and the analysis is carried out in the dq-frame of reference [30]. The set point changes are done by setting V_{oq} to zero V_{od} is changed. The communication between the DERs take place as explained in the previous sections. We propose using a metric called aggregate tracking error (ATE) to capture the system performance. ATE is calculated as

$$\text{ATE} = \sum_{t=\text{start of disturbance}}^{\text{steady state}} |V_{od}(t) - V_{sp}(t)|. \tag{4.19}$$

To get the relative performance of a scheduler as compared to when ideal communication is used, ATE of a considered scheduler is normalized with respect to the ATE of an ideal scheduler. This is referred to as the normalized aggregate tracking error (NATE) and is measured as

$$\text{NATE} = \text{ATE}_{\text{scheduler}}/\text{ATE}_{\text{ideal}}. \tag{4.20}$$

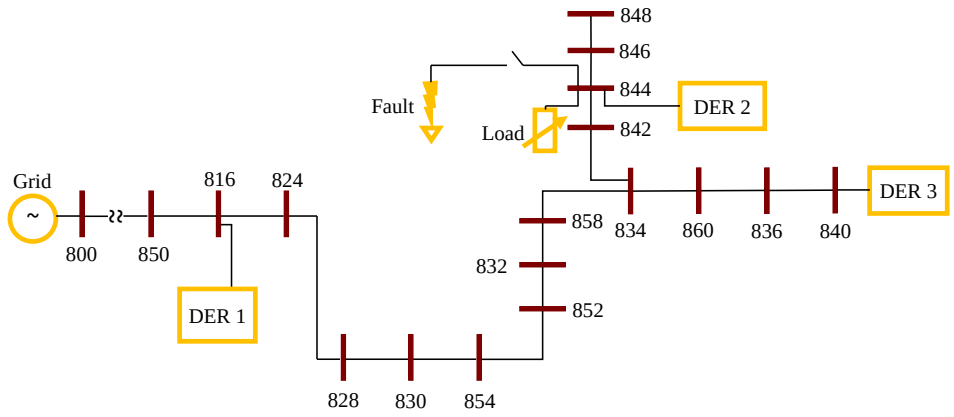


Figure 4.15: IEEE 34-bus test feeder with 3 DERs.

4.9.1 Test System 1

Test system I is the IEEE 34-bus distribution system, see Fig. 4.15. The system has 3 DERs in voltage control mode and is made to operate with a system voltage of 24.9 kV in the grid connected mode. Each of the DERs are implemented as a voltage-sourced converter (VSC) unit behind an LCL filter ($L = 1.8$ mH and $C = 100$ uF). The connectivity between the DERs is as per the adjacency matrix A -

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}. \tag{4.21}$$

Set Point Change in Output Voltage of a DER

At $t = 0.1$ s, the d-component of the output voltage set point of DER 2 is changed from 1.00 pu to 1.10 pu, while V_{oq} is set to zero. The system response under the different schedulers is shown in Fig. 4.16 and the NATE values are shown in Table 4.6. It is seen that the AoI based schedulers significantly outperform the non-AoI schedulers. Compared to the baseline of the ideal scheduler, the NATE values for the MAD and MAF schedulers only increases by 21% and 28%; whereas they increase by 53% and 80% for round robin and proportional fairness schedulers. Random scheduler performs the worst as its NATE degrades by 116% compared to the ideal scheduler.

Table 4.5: Communication Parameters used in simulation

Parameter	**Value**
Deployment region $l \times b$	1000m $\times$ 1000m
Transmission power $p^u_{\tau,\mathrm{f}}$	40 dBm [66]
Noise power σ_o	10^{-10} W
Path loss exponent γ	3.76 [65]
Channel	Rayleigh fading channel
CSI error variance σ_e^2	0.01 [65]
Traffic model	Generate at will
5G numerology μ	0
Number of DERs N	3, 28
Number of RBs in uplink B_u	2, 10
Number of RBs in downlink B_d	3, 20
RBs needed by ith DER B_i^{need}	[1,3] $\forall i \in G$
BS location	$(l/2, b/2)$

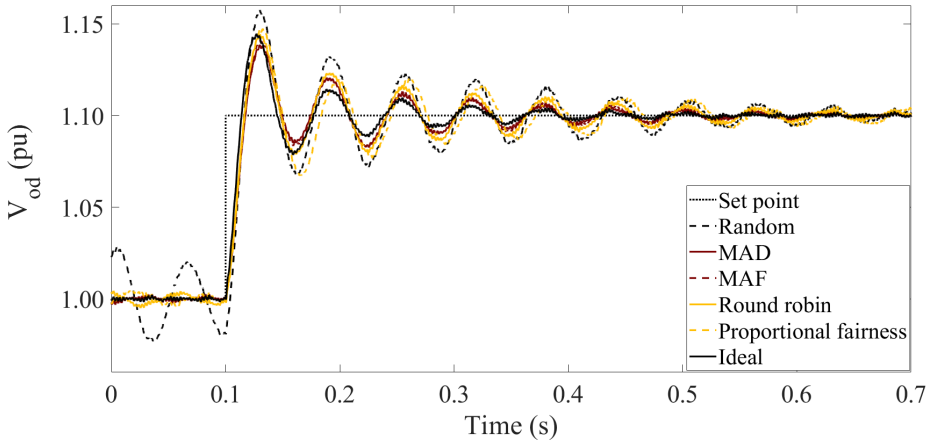


Figure 4.16: IEEE 34-bus system response of DER 2 to a step change in voltage set point from 1.00 pu to 1.10 pu at $t = 0.1$ s under different schedulers.

Three-Phase to Ground Fault

At $t = 0.05$ s, a three-phase ABC-G fault is simulated for a duration of 0.03 s at bus 844. The system response under different schedulers is shown in Fig. 4.17. It can be seen that AoI-aware schedulers have lower NATE values compared with non-AoI aware schedulers. Compared to the baseline of the ideal scheduler, the NATE for the MAD scheduler only increases by 19%; whereas they increase by 29% for round robin. Random scheduler performs

Table 4.6: Normalized Aggregate tracking errors (NATE) for different schedulers (IEEE 34-bus distribution system)

	Normalized aggregate tracking error for different disturbances		
Scheduler	**Voltage set point change**	**Three-phase fault**	**Change in active load**
Ideal	1	1	1
MAD	1.21	1.19	1.22
MAF	1.28	1.26	1.26
Round robin	1.53	1.29	1.37
Proportional fairness	1.80	1.44	1.43
Random	2.16	1.89	2.01

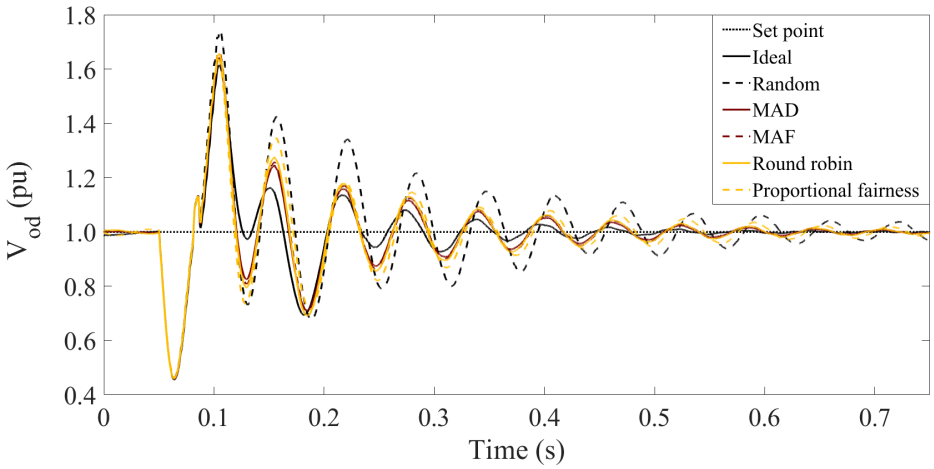


Figure 4.17: IEEE 34-bus system response to a ABC-G fault applied at bus 844 at $t = 0.05$ s for 0.03 s under different schedulers.

the worst as its NATE degrades by 89% compared to the ideal scheduler. These values are summarized in Table 4.6.

Change in Real Power of the Load

The real power of the load connected to bus 844 is changed from 1 MW to 20 MW at $t = 0.1$ s. The system response under different schedulers is shown in Fig. 4.18. Table 4.6 shows that AoI-based schedulers perform much better. Compared to MAD scheduler, the NATE for round robin, proportional fairness, and random degrades by 12.29%, 17.2%, and 64.7%.

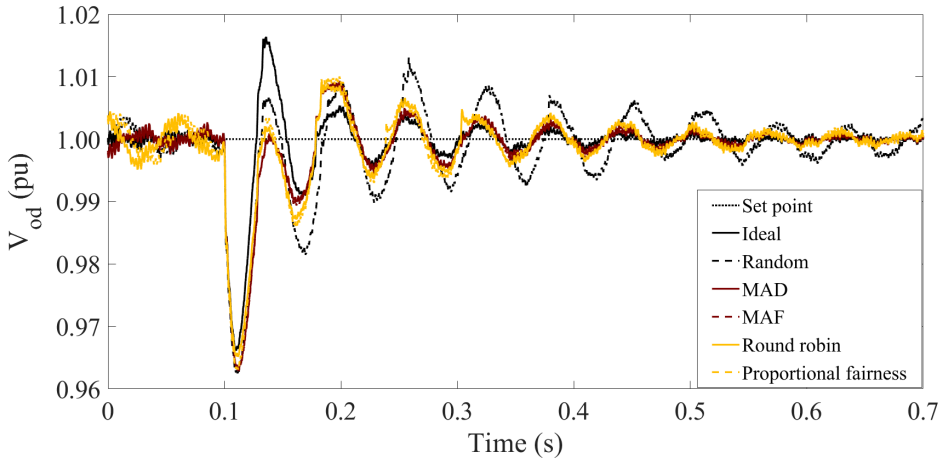


Figure 4.18: IEEE 34-bus system response to a change in real power of the load at bus 844 from 1 MW to 20 MW at $t = 0.1$ s under different schedulers.

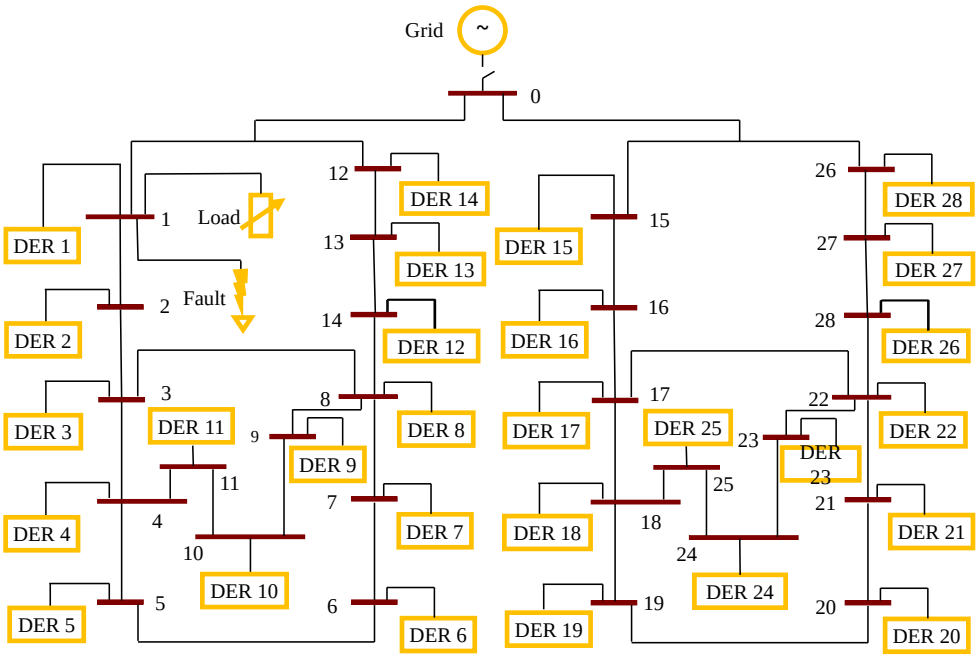


Figure 4.19: CIGRE distribution system with 28 DERs. Loads are not shown.

4.9.2 Test System 2

Test system II is the CIGRE distribution system with 28 DERs, see Fig. 4.19. It operates in the islanded mode with a system base voltage of 0.48 kV. The analysis is carried out in the dq-frame of reference. The DERs (rated 6 MVA) are modeled as a VSC unit behind an LCL

Table 4.7: Normalized aggregate tracking errors (NATE) for different schedulers (CIGRE distribution system)

	Normalized aggregate tracking error for different disturbances		
Scheduler	**Voltage set point change**	**Three-phase fault**	**Change in active load**
Ideal	1	1	1
MAD	1.015	1.188	1.015
MAF	1.016	1.193	1.031
Round robin	1.023	1.220	1.038
Proportional fairness	1.029	1.226	1.043
Random	1.085	1.230	1.068

filter ($L = 7.33$ uH and $C = 1.38$ mF). In this case, the communication among the DERs are:

- DER 1 communicates with DER 2 and DER 1.
- DER 14 communicates with DER 13 and DER 1.
- DER 15 communicates with DER 16 and DER 28.
- DER 28 communicates with DER 27 and DER 15.

The disturbances along with the performance of the schedulers in handling them are described below:

Set Point Change in Output Voltage of a DER

The d-component of the output voltage set point of DER 1 is changed from 1.00 pu to 1.05 pu at $t = 0.1$ s, while V_{oq} is set to zero. The response under different schedulers is shown in Fig. 4.20 and Table 4.7 lists the NATE values. As expected, the AoI based schedulers have better performance. While the MAD and MAF schedulers only see a 1.5% and 1.6% increase in NATE as compared to the ideal scheduler, the round robin and proportional fairness see a 2.3% and 2.9% increase in NATE. Similar to the scenarios described above, random scheduler performs the worst, an increase of NATE of 8.5%, as compared to the ideal scheduler.

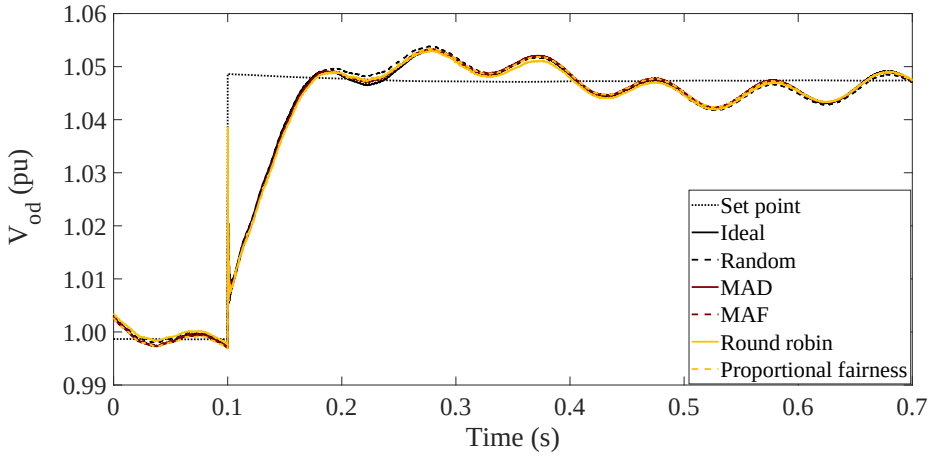


Figure 4.20: CIGRE system response of DER 1 to a step change in voltage set point from 1.00 pu to 1.05 pu at $t = 0.1$ s under different schedulers.

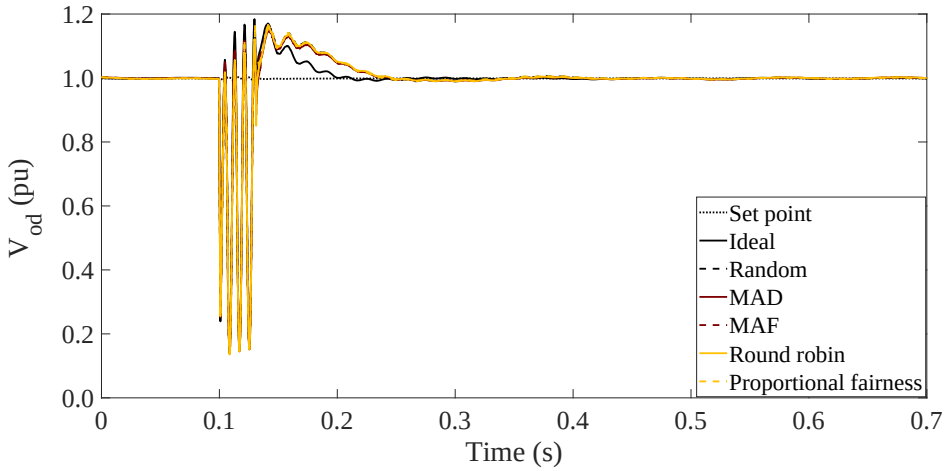


Figure 4.21: CIGRE system response to a ABC-G fault applied at bus 1 at $t = 0.1$ s for 0.03 s under different schedulers.

Three-Phase to Ground Fault

At $t = 0.1$ s, a three-phase ABC-G fault is simulated for 0.03 s at bus 1. Fig. 4.21 shows the AoI schedulers provide much better stability in terms of lower NATE as compared to non-AoI based schedulers. While the MAD scheduler only sees a 19% increase in NATE as compared to the ideal scheduler, the round robin and random scheduler see a 22% and 23% increase in NATE.

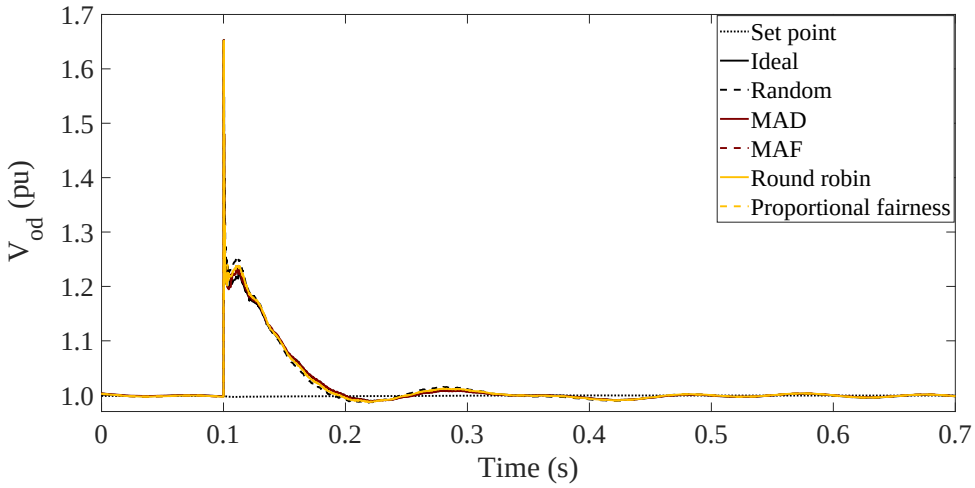


Figure 4.22: CIGRE system response to a change in real power of the load at bus 1 from 4.5 MW to 0.5 MW at $t = 0.1$ s under different schedulers.

Table 4.8: Age of Information (AoI(T)) at the destination DERs for the test cases

Schedulers	**IEEE 34-bus (ms)**	**CIGRE (ms)**
MAD	23992.01	215932.00
MAF	33402.91	232698.74
Round Robin	36696.97	326647.82
Proportional Fairness	47939.47	433371.46
Random	59873.52	514520.33

Change in Real Power of the Load

In this case, the real power of the load connected to bus 1 is changed from 4.5 MW to 0.5 MW at $t = 0.1$ s. The response under different schedulers is shown in Fig. 4.22 and Table 4.7 lists the NATE values for the different schedulers. It can be seen that MAD and MAF schedulers have lower NATE compared with other schedulers. RR and PF have a 2.26% and 2.76% higher NATE compared with MAD. Random scheduler has a 5.22% higher NATE compared with MAD.

Finally, the overall AoI(T) for the IEEE 34-bus system and CIGRE systems are shown in Table 4.8. The AoI based schedulers base their scheduling decisions on the AoI values which ensures the DERs are updated on the most recent state of the system. As a result, the AoI-based schedulers have a better performance in ensuring the stability of the system. Similarly to UAV-assisted internet of things (IoT) networks [37], MAD outperforms MAF. This is because MAD leads to the maximum reduction in AoI at both the BS and the destination

DERs [37]. Among the two non-AoI schedulers, RR has better AoI minimization performance compared to PF. It should be noted that RR also has been shown to have better performance in terms of delay and throughput [87, 94].

4.10 Distributed AoI minimization

It is important to note that the minimizing of AoI in the smart grid scenario presented above is different from traditional approaches of minimizing AoI of packets generated at source devices at a central monitoring station [37] or when each source has a unique destination [106]. This is because in the smart grid scenario involves the minimizing of AoI of packets generated at source devices in a distributed manner at the destination devices. For the general case of a distributed network with heterogeneous sources, each source can have unique properties like variable number destinations, different RB requirements, channel conditions, traffic generation pattern etc. Due to this, the following aspects have to be considered in the scheduling process -

- The uploading step should factor in the number of devices downloading the information from a particular device. Consider two different source devices, $d1$, $d2$, where their information is downloaded by M and N devices respectively where $M > N$. As the AoI of $d1$ will have a greater impact on the final AoI (see Eqn. (4.14)), $d1$ should be prioritized in the uploading step assuming other things to be equal.

- The downloading step should prioritize the devices who were recently uploaded. Currently the MAD scheduler studies in this chapter considers the upload and download as separate steps. This is not desirable as those are sequential operations. Therefore the download step should consider devices who uploaded recently as they will lead to a large improvement in AoI at the destination devices.

- As the packets generated from different sources can have different sizes, those packets will have different RB requirements. Hence this becomes another critical aspect as the number of available RBs is a bottleneck and has to be used judiciously.

- As mentioned in Chapter 3, the packet generation pattern at the devices and channel conditions also need to be accounted in the scheduling decisions.

Table 4.9: Adjacency matrix representing communication links

	Device 1	Device 2	Device 3
Device 1	0	1	1
Device 2	0	0	1
Device 3	1	1	0

To highlight the above points, we devise a heuristic based scheduler that has the same criteria as the MAD scheduler in scheduling devices for uploading, but prioritizes recently uploaded devices in the downloading step. As it considers an ordering between the upload and download steps, it is termed as ordered MAD (oMAD) scheduler. A simple 3 device network where the number of destinations per source is not same and other conditions like RB requirements (1 RB needed per packet), channel conditions (lossless), packet generation periodicity (1 packet per slot) are kept the same across all devices. The observation interval $T = 5$ slots and the total available RBs in the uplink $B_u = 2$ and downlink B_d=3 respectively. The adjacency matrix is shown in Table 4.9:

The results obtained shows that oMAD achieves an AoI of 31.58 compared to 33.64 of MAD, and hence oMAD does better than MAD in terms of minimizing the AoI of the source devices at the destination devices for this particular network configuration. As MAD is proven to be optimal under the scenario where information is collected at a single monitoring station [37] or when each source has a unique destination device [106], and we see oMAD outperforms MAD here, this proves MAD is no longer optimal for distributed networks.

This strongly motivates designing of better scheduling algorithms specifically designed for AoI minimization in distributed networks that take into account the network connectivity and other factors mentioned above. Reinforcement learning based schemes provide a great way to incorporate these factors in its decision process and therefore we propose using a Deep Q Network (DQN) based scheduler. The basics of DQN are explained in detail in Chapter 3, see Sec. 3.4.2 for details. The Markov Decision Process (MDP) framework needed to set up the DQN for the current network is explained next.

State Space The state is defined to be the scenario experienced by the agent, i.e. the scheduler. We define the state at any slot t as

$$x(t) = \{t, \{AoI_i^{BS}(t)\}_{i=1}^{G}, \{AoI_r^{dest}(t)\}_{r=1}^{\bar{R}}\} \tag{4.22}$$

At t=1, $AoI_i^{BS}(t)$ and $AoI_r^{dest}(t)$ are initialized to 1. The set of state space is denoted by $\mathcal{X}$.

Action Space In terms of scheduling, each step involves allocation of RBs to the uplinks and downlinks. Therefore the action is represented as

$$a(t) = (\mathcal{U}(t), \mathcal{D}(t)) \tag{4.23}$$

where $\mathcal{U}(t)$ and $\mathcal{D}(t)$ denote the RB allocation to the uplinks and downlinks respectively.

Reward

As the objective is improving the information freshness of the packets at the final destinations (see Eqn. (4.14)), the reward is given by the negative of the average AoI of all the devices at their destinations at slot t

$$r(x(t), a(t)) = r(t) = \frac{AoI(T)}{T|\bar{R}|} \tag{4.24}$$

Note that compared to Eqn. (4.14) where we consider the sum AoI as the final performance metric, Eqn. (4.24) involves a normalizing term over the sum AoI. This is because the sum AoI has a large spread/variance which could cause the DQN's weights to change quickly and can lead to an unstable learning process [29].

The DQN based scheduler is shown in Fig. 4.23 and explained in Algorithm 9. The hyper-parameters and the network are initialized in lines 1 - 2. For each step, a random action is taken with probability ϵ or the action with the highest Q-value is selected with probability 1-ϵ in lines 6 - 8. The selected action is performed and environment transitions to the new state in lines 9-10. Line 11 stores the resulting experience in the replay memory. Then a mini-batch of samples is used to train the network in lines 12-17, after which the weights of the current network are updated in lines 18. Line 19 updates the target network weight every O slots. Note that the working principle behind the DQN algorithm is explained in

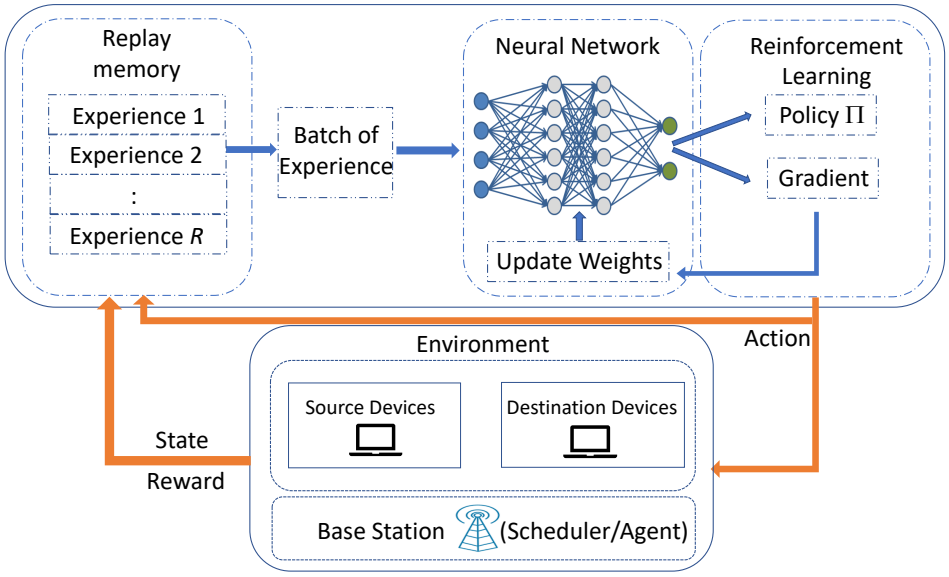


Figure 4.23: DQN network architecture.

detail in Chapter 3 and hence is not repeated again here.

4.11 Results

To evaluate the performance of the various schedulers, two different conditions are considered - (a) ideal conditions where the channels are lossless, the source devices follow a generate-at-will traffic generation model and the packets from each source require 1 RB , and (b) non-ideal conditions where the the wireless channels are lossy, the source devices generate packets at certain fixed periodicities and the packets from different sources require different number of RBs. For the non-ideal conditions, the communication parameters is set as per Sec. 4.7.2. The final results for the ideal and non ideal conditions are shown in Fig. 4.24 and 4.25 respectively. The DQN scheduler outperforms all other schedulers in achieving the least AoI which proves that a learning approach is desirable in such scenarios. It can be seen that for both the ideal and non ideal conditions, the DQN scheduler outperforms oMAD, MAD and all other schedulers. It should be noted that the percentage improvement that DQN scheduler offers over MAD is greater under the non ideal conditions (15%)as compared to the ideal conditions (5%). This is because the only important feature for the ideal condition is the connectivity among different devices, whereas additional factors like RB requirements, channels qualities, traffic generation etc. provides the learning algorithm more features to

Algorithm 9 DQN scheduler

Initialize the replay memory R, probability ϵ, current network weights θ, target network weights θ^-. Set episode = 1.
Initialize the current network $Q(s, a|\theta)$ and the target network $Q(s, a|\theta^-)$.
for episode=1:E **do**
    initialize the environment by setting $t = 1$.
    for t=1:T **do**
        Select an action a:
            select a random action $a \in \mathcal{A}$ with probability ϵ,
            otherwise select $a = \arg\max_a Q(x(t), a|\boldsymbol{\theta})$.
        Perform action a.
        Observe the reward $r(t)$ and the new state $x(t+1)$.
        Store experience $\{x(t), a(t), r(t), x(t+1)\}$ in replay memory.
        Sample a random mini-batch B of transitions $x(t'), a(t'), r(t'), x(t'+1)$ from the replay memory.
        Calculate target value:
        if $x(t'+1)$ is a terminal state **then**
            $y(t') = r(t')$,
        else
            $y(t') = r(t') + \max_a Q(x(t'+1, a|\theta^-))$.
        Update θ using the weight update rule.
        Update target network by setting $\theta^- = \theta$ every O steps.
        Episode ends if $x(t+1)$ is the terminal state.

learn from which the other schedulers are unable to account for.

4.12 Conclusion

In this chapter, we develop a co-simulation environment to evaluate the performance of a smart grid under a 5G Radio Access Network (RAN). The results show that the smart grid remains stable under 5G. Then we focus on the scheduling part of the RAN and compare the Age of Information (AoI) minimization among different schedulers. We see that the minimization of the AoI also correlates to improved stability, which points to the need for AoI minimization based 5G protocols in smart grid applications. Then, we highlight the difference in distributed scenario like the smart grid as compared to scenarios where the AoI is measured at a central monitoring station. Due to this, the Maximal Age Difference (MAD) scheduler, which is proven to be optimal for AoI minimization under many different scenarios,

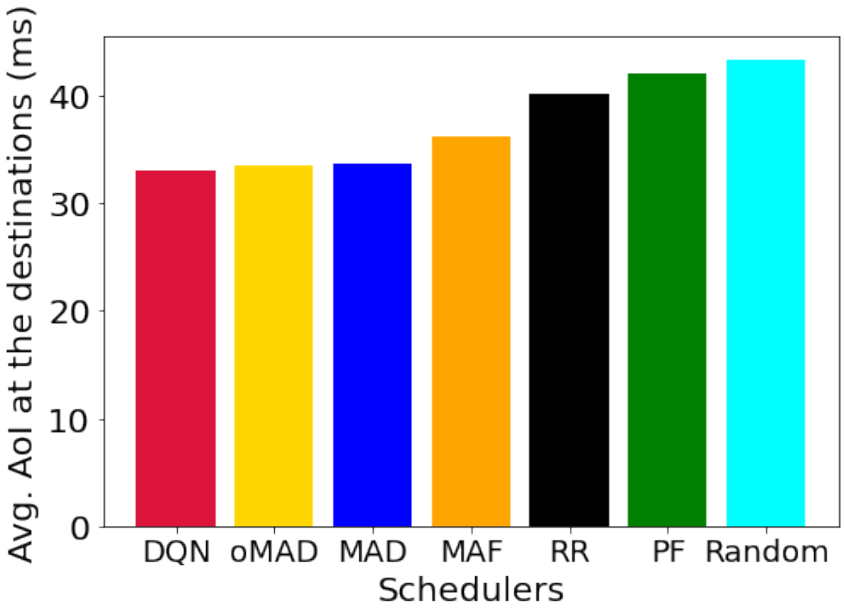


Figure 4.24: AoI obtained under ideal conditions.

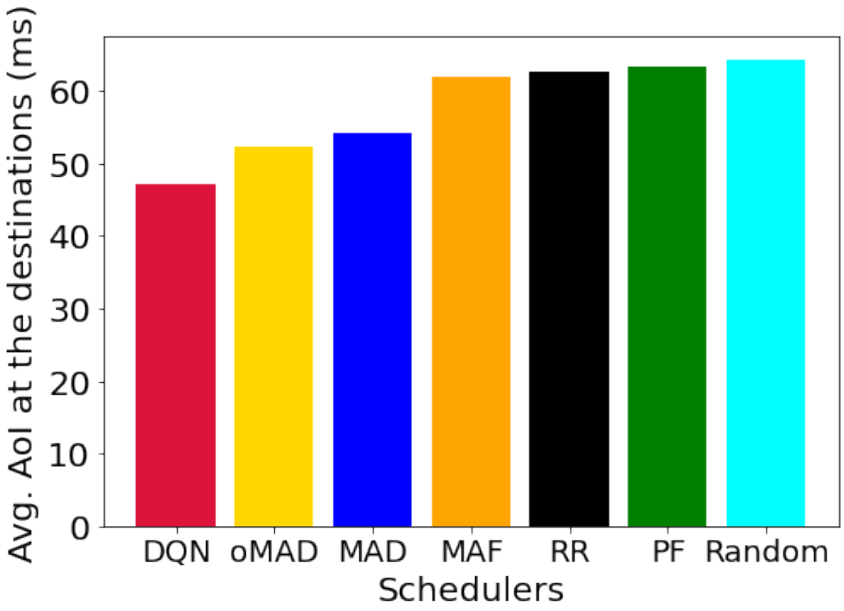


Figure 4.25: AoI obtained under non-ideal conditions.

is no longer optimal in this case. This distributed network setting of smart grid is commonly encountered is a variety of other settings like remote surgery, multi-robot coordination etc. However, it has not been investigated in the current AoI literature. We propose a Deep Q Network based scheduler that outperforms the MAD and all other schedulers in all considered scenarios. This work will be extended to consider modulation and coding scheme (MCS) selection in addition to RB allocation to design a full fledged 5G scheduler for distributed networks.

Chapter 5

Conclusions and Future Work

In this dissertation, we have investigated various means of minimizing the Age of Information (AoI) for three time-sensitive autonomous systems, namely vehicular networks, UAV-assisted IoT networks, and smart grid. AoI is a natural choice to measure these systems' performance from a communication perspective. This is because the fresher the information at the receiver, the better is its awareness of the system state which allows it to make better control decisions. The applications investigated in this dissertation are categorized by the type of connectivity between the source and destination nodes. The first type is the one-to-all connectivity. In Chapter 2, we show that for vehicular networks involving broadcast mode of communication, the AoI and the safety performance are related, but under certain conditions this relationship no longer holds. Therefore, mobility needs to be accounted for in addition to the AoI when designing a rate-control algorithm to improve safety. We consider a binary classification of the vehicles in terms of their current mobility pattern, i.e. risky and non-risky. This can be extended to a multi-class classification that takes into account the road conditions, past and current mobility patterns, etc. This extension can help formulate a unified risk metric that captures the likelihood of a vehicle colliding with another. Additionally, currently we do not consider the vehicles reacting to the information received. Ideally, upon receiving any information that indicates the current trajectory could lead to a collision, the vehicle should change velocity/lanes to avoid the collision. Therefore, future work should incorporate this reactive behavior at the vehicles.

AoI minimization in networks with multiple sources and a single destination, or all-to-one networks, is considered in Chapter 3. The application investigated is a UAV-assisted IoT network. Using mathematical analysis and simulations, we show that a Maximal Age Difference (MAD) scheduler is the optimal scheduler for minimizing AoI under ideal conditions, i.e., lossless channels and a generate-at-will traffic generation model at the IoT devices. When considering practical conditions, MAD is no longer optimal and hence we propose reinforcement learning (RL) based schedulers. The RL schedulers do well because they can

address the packet generation patterns and channel qualities in their scheduling decisions. Specifically, we use Deep Q Networks (DQN) and Proximal Policy Optimization (PPO) algorithms to design schedulers for general conditions. While the DQN-based scheduler works well for smaller networks, it cannot scale well for larger networks. This is due to an intrinsic drawback of DQN in converging for larger action spaces. The PPO-based scheduler doesn't have this drawback and hence can be applied to large networks. Another added advantage of the PPO-based scheduler is that it can also adapt to changing network conditions. It is desirable that a trained model is able to adapt to changing conditions. A natural extension of this work is to develop a decentralized learning model where each UAV is an agent that is responsible for scheduling the devices under its coverage. This will further reduce the action space and allow for collaboration among UAVs, using federated learning. Other possible areas for exploration are to add mobility to the IoT devices or users so that the handover of the devices/users between UAVs is considered. Finally, in this work, we consider the placement of the UAVs as fixed. However, given the location of the IoT devices, UAVs can be positioned for both best coverage and AoI minimization. Hence investigating the optimal placement along with the scheduling is another important direction for future work.

In Chapter 4, we focus on AoI minimization for distributed networks where the relationship between the source and destinations is not as direct as in the previous AoI studies. An example is the smart grid application. We develop a co-simulation platform that studies the performance of a smart grid in the presence of a 5G Radio Access Network (RAN). The smart grid is simulated in PSCAD and MATLAB, while the 5G RAN is modeled in Python. The results obtained show that 5G is able to reliably support smart grid operations. This is significant as current smart grid connectivity is done using fiber-optic cables which are expensive and also cannot scale easily as new components are added to the grid. Then, to investigate the relationship between improved AoI and stability in the smart grid, we evaluate different schedulers. Here we find that the MAD scheduler achieves the least AoI and that a reduced AoI directly correlates with improved smart grid stability. Therefore designing of AoI-centric 5G protocols becomes important as the smart grid is one of the key application areas of 5G. Additionally, we highlight the differences between distributed networks as compared to the networks traditionally considered in the literature, and the need to account for these differences in the scheduling process. While the MAD scheduler is proven to be optimal for a variety of networks, it is not the case for a distributed one. This provides a strong motivation to use reinforcement learning and we propose a DQN-based scheduler that outperforms all the other considered schedulers. The current results

consider only RB allocation and could be extended to incorporate modulation and coding scheme (MCS) selection based on changing channel conditions. Results obtained motivate schedulers to be developed that take into account factors specific to distributed networks with heterogeneous sources, like the number of destinations per source, RB requirements for the packets of different devices, sequential nature of uploading and downloading, channel conditions, and periodicity of packet generation. Such schedulers would be applicable for many modern applications like multi-robot coordination, networked control systems, etc.

Bibliography

[1] 3GPP R18 Launches 5G Smart Grid Research, Unifies Global Standards, and Leads Industry Development | Light Reading. https://www.lightreading.com/partner-perspectives-(sponsored-content)/3gpp-r18-launches-5g-smart-grid-research-unifies-global-standards-and-leads-industa/d-id/763422. (Accessed on 11/30/2020).

[2] 3GPP TS 38.211 version 15.3.0 Release 15. 5G NR Physical channels and modulation.

[3] 3GPP TS 38.306. User Equipment (UE) radio access capabilities. version 15.3.0 Rel 15.

[4] Connected Vehicles — Colorado Department of Transportation. https://www.codot.gov/programs/operations/intelligent-transportation-systems/innovation/connected-vehicles. (Accessed on 07/07/2020).

[5] Dedicated Short Range Communications (DSRC) Service | Federal Communications Commission. https://www.fcc.gov/wireless/bureau-divisions/mobility-division/dedicated-short-range-communications-dsrc-service. (Accessed on 07/13/2020).

[6] ETSI EN 302 637-2 V1.3.2 (2014-11) Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service.

[7] Ericsson Mobility Report November 2016. https://www.ericsson.com/en/mobility-report/reports. (Accessed on 12/06/2020).

[8] Five top 5G use cases for enterprises, according to Nokia. https://www.rcrwireless.com/20200612/5g/five-top-5g-use-cases-enterprises-nokia. (Accessed on 11/30/2020).

[9] GitHub - tensorflow/agents: TF-Agents is a library for Reinforcement Learning in TensorFlow. https://github.com/tensorflow/agents. (Accessed on 12/19/2020).

[10] How IoT Short-Range Connectivity Stacks Up | Mouser. https://www.mouser.com/applications/home-automation-how-iot-short-range-stacks-up/. (Accessed on 05/29/2021).

[11] Smart Grid Powered by 5G SA-based Network Slicing SGCC, China Telecom and Huawei. https://www.gsma.com/futurenetworks/wp-content/uploads/2020/03/2_Powered\protect\discretionary{\char\hyphenchar\font}{}{}by\protect\discretionary{\char\hyphenchar\font}{}{}SA_Smart\protect\discretionary{\char\hyphenchar\font}{}{}Grid\protect\discretionary{\char\hyphenchar\font}{}{}5G\protect\discretionary{\char\hyphenchar\font}{}{}Network\protect\discretionary{\char\hyphenchar\font}{}{}Slicing_China\protect\discretionary{\char\hyphenchar\font}{}{}Telecom_GSMA_v2.0.pdf. (Accessed on 11/30/2020).

[12] NR Scheduling Performance Evaluation - MATLAB & Simulink. https://www.mathworks.com/help/5g/ug/nr-tdd-symbol-based-scheduling-performance-evaluation.html. (Accessed on 01/11/2021).

[13] 3GPP TS 38.214. Physical layer procedures for data version. 15.3.0 Rel 15.

[14] The 5G Performance Management Revolution - SDxCentral. https://www.sdxcentral.com/monitoring/performance-management/definitions/5g-performance-management/. (Accessed on 05/09/2019).

[15] ns-3 | a discrete-event network simulator for internet systems. https://www.nsnam.org/. (Accessed on 08/01/2020).

[16] SUMO Documentation. https://sumo.dlr.de/docs/index.html. (Accessed on 08/01/2020).

[17] The Ongoing History of the Age of Information:. http://webhome.auburn.edu/~yzs0078/AoI.html. (Accessed on 06/04/2021).

[18] On-board system requirements for v2v safety communications. In *Soc. Autom. Eng.* SAE International, 2016.

[19] Mohamed A Abd-Elmagid, Aidin Ferdowsi, Harpreet S Dhillon, and Walid Saad. Deep reinforcement learning for minimizing age-of-information in UAV-assisted networks.

In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2019.

[20] Mohamed K Abdel-Aziz, Chen-Feng Liu, Sumudu Samarakoon, Mehdi Bennis, and Walid Saad. Ultra-reliable low-latency vehicular networks: Taming the age of information tail. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7. IEEE, 2018.

[21] Mohamed K Abdel-Aziz, Sumudu Samarakoon, Mehdi Bennis, and Walid Saad. Ultra-reliable and low-latency vehicular communication: An active learning approach. *IEEE Communications Letters*, 24(2):367–370, 2019.

[22] Ghafour Ahani, Di Yuan, and Yixin Zhao. Age-Optimal UAV Scheduling for Data Collection with Battery Recharging. *arXiv preprint arXiv:2005.00252*.

[23] Syed Amaar Ahmad, Abolfazl Hajisami, Hariharan Krishnan, Farid Ahmed-Zaid, and Ehsan Moradi-Pari. V2v system congestion control validation and performance. *IEEE Transactions on Vehicular Technology*, 68(3):2102–2110, 2019.

[24] Ahmed Arafa and Sennur Ulukus. Timely updates in energy harvesting two-hop networks: Offline and online policies. *IEEE Transactions on Wireless Communications*, 18(8):4017–4030, 2019.

[25] Ahmed Aziz, Karan Singh, Walid Osamy, and Ahmed M Khedr. Effective algorithm for optimizing compressive sensing in iot and periodic monitoring applications. *Journal of Network and Computer Applications*, 126:12–28, 2019.

[26] J. Aznar-Poveda, E. Egea-Lopez, A. Garcia-Sanchez, and P. Pavon-Mariáo. Time-to-collision-based awareness and congestion control for vehicular communications. *IEEE Access*, 7:154192–154208, 2019.

[27] Andrea Baiocchi and Ion Turcanu. A model for the optimization of beacon message age-of-information in a vanet. In *2017 29th International Teletraffic Congress (ITC 29)*, volume 1, pages 108–116. IEEE, 2017.

[28] Luca Baldesi, Leonardo Maccari, and Renato Lo Cigno. Keep it fresh: Reducing the age of information in v2x networks. In *Proceedings of the 1st ACM MobiHoc Workshop on Technologies, mOdels, and Protocols for Cooperative Connected Cars*, pages 7–12, 2019.

[29] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.

[30] F. Blaabjerg, R. Teodorescu, M. Liserre, and A.V. Timbus. Overview of control and grid synchronization for distributed power generation systems. *IEEE Trans. Ind. Electron*, 53(5):1398–1409, Oct. 2006.

[31] Annette Böhm and Magnus Jonsson. Real-time communication support for cooperative, infrastructure-based traffic safety applications. *International Journal of Vehicular Technology*, 2011, 2011.

[32] Baturalp Buyukates, Alkan Soysal, and Sennur Ulukus. Age of information in two-hop multicast networks. In *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pages 513–517. IEEE, 2018.

[33] Elif Tuğçe Ceran, Deniz Gündüz, and András György. A reinforcement learning approach to age of information in multi-user networks. In *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1967–1971. IEEE, 2018.

[34] Maohong Chen, Yong Xiao, Qiang Li, and Kwang-cheng Chen. Minimizing age-of-information for fog computing-supported vehicular networks with deep q-learning. *arXiv preprint arXiv:2004.04640*, 2020.

[35] B. Choudhury, V. K. Shah, A. Dayal, and J. H. Reed. Experimental analysis of safety application reliability in v2v networks. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5, 2020.

[36] Biplav Choudhury, Vijay K Shah, Avik Dayal, and Jeffrey H Reed. Joint age of information and self risk assessment for safer 802.11 p based v2v networks. *accepted to IEEE INFOCOM 2021*, 2020.

[37] Biplav Choudhury, Vijay K. Shah, Aidin Ferdowsi, Jeffrey H. Reed, and Y. Thomas Hou. Aoi-minimizing scheduling in uav-relayed iot networks. In *2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*, pages 117–126, 2021. doi: 10.1109/MASS52906.2021.00023.

[38] Maice Costa, Marian Codreanu, and Anthony Ephremides. On the age of information in status update systems with packet management. *IEEE Transactions on Information Theory*, 62(4):1897–1910, 2016. doi: 10.1109/TIT.2016.2533395.

[39] Erez Dagan, Ofer Mano, Gideon P Stein, and Amnon Shashua. Forward collision warning with a single camera. In *IEEE Intelligent Vehicles Symposium, 2004*, pages 37–42. IEEE, 2004.

[40] A. Dayal, E. Colbert, V. Marojevic, and J. Reed. Risk controlled beacon transmission in v2v communications. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pages 1–6, 2019.

[41] R. Dilli. Analysis of 5G Wireless Systems in FR1 and FR2 Frequency Bands. In *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, pages 767–772, March 2020. doi: 10.1109/ICIMIA48430.2020.9074973.

[42] Salman Durrani, Xiangyun Zhou, and Abhas Chandra. Effect of vehicle mobility on connectivity of vehicular ad hoc networks. In *2010 IEEE 72nd Vehicular Technology Conference-Fall*, pages 1–5. IEEE, 2010.

[43] Olakunle Elijah, Tharek Abdul Rahman, Igbafe Orikumhi, Chee Yen Leow, and MHD Nour Hindia. An overview of internet of things (iot) and data analytics in agriculture: Benefits and challenges. *IEEE Internet of Things Journal*, 5(5):3758–3773, 2018.

[44] Jakob Erdmann. Sumo's lane-changing model, Modeling Mobility with Open Data (Cham) (Michael Behrisch and Melanie Weber, eds.). *Springer International Publishing*, page 105–123, 2015.

[45] Yaser P Fallah, Ching-Ling Huang, Raja Sengupta, and Hariharan Krishnan. Analysis of information dissemination in vehicular ad-hoc networks with application to cooperative vehicle safety systems. *IEEE Transactions on Vehicular Technology*, 60(1): 233–247, 2010.

[46] Ben Foster and Susan Mazur-Stommen. Results from recent real-time feedback studies. In *American Council for an Energy-Efficient Economy.*, February 2012.

[47] Michele Garau, Matteo Anedda, Cristina Desogus, Emilio Ghiani, Maurizio Murroni, and Gianni Celli. A 5G cellular technology for distributed monitoring and control in

smart grid. In *2017 IEEE international symposium on broadband multimedia systems and broadcasting (BMSB)*, pages 1–6. IEEE, June 2017.

[48] Sneihil Gopal and Sanjit K Kaul. A game theoretic approach to dsrc and wifi coexistence. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 565–570. IEEE, 2018.

[49] Nishu Gupta, Arun Prakash, and Rajeev Tripathi. Medium access control protocols for safety applications in vehicular ad-hoc network: A classification and comprehensive survey. *Vehicular Communications*, 2(4):223–237, 2015.

[50] Q. He, G. Dan, and V. Fodor. Minimizing age of correlated information for wireless camera networks. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 547–552, 2018. doi: 10.1109/INFCOMW.2018.8406914.

[51] C. Hu, K. Lin, S. Huang, Y. Huang, H. Chiu, and L. Hui. Carview plus: A social-based live media distribution platform in vehicular networks. In *2019 8th International Conference on Innovation, Communication and Engineering (ICICE)*, pages 110–113, 2019.

[52] Huimin Hu, Ke Xiong, Gang Qu, Qiang Ni, Pingyi Fan, and Khaled Ben Letaief. AoI-Minimal Trajectory Planning and Data Collection in UAV-Assisted Wireless Powered IoT Networks. *IEEE Internet of Things Journal*, 8(2):1211–1223, 2021. doi: 10.1109/JIOT.2020.3012835.

[53] J. Hu, H. Zhang, K. Bian, L. Song, and Z. Han. Distributed Trajectory Design for Cooperative Internet of UAVs Using Deep Reinforcement Learning. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019. doi: 10.1109/GLOBECOM38437.2019.9014214.

[54] Jingzhi Hu, Hongliang Zhang, Lingyang Song, Robert Schober, and H Vincent Poor. Cooperative Internet of UAVs: Distributed Trajectory Design by Multi-agent Deep Reinforcement Learning. *IEEE Transactions on Communications*, 2020.

[55] C. Huang, Y. P. Fallah, R. Sengupta, and H. Krishnan. Adaptive intervehicle communication control for cooperative safety systems. *IEEE Network*, 24(1):6–13, 2010.

[56] Salama S Ikki and Sonia Aissa. Two-way amplify-and-forward relaying with gaussian imperfect channel estimations. *IEEE Communications Letters*, 16(7):956–959, 2012.

[57] Hadi S Jomaa, Josif Grabocka, and Lars Schmidt-Thieme. Hyp-rl: Hyperparameter optimization by reinforcement learning. *arXiv preprint arXiv:1906.11527*, 2019.

[58] Igor Kadota. *Age-of-Information in Wireless Networks:Theory and Implementation*. PhD thesis.

[59] Alexandros Karakasidis, Panos Vassiliadis, and Evaggelia Pitoura. Etl queues for active data warehousing. In *Proceedings of the 2nd international workshop on Information quality in information systems*, pages 28–39, 2005.

[60] Charles Jumaa Katila, Chiara Buratti, Melchiorre Danilo Abrignani, and Roberto Verdone. Neighbors-aware proportional fair scheduling for future wireless networks with mixed mac protocols. *EURASIP Journal on Wireless Communications and Networking*, 2017(1):1–12, 2017.

[61] S. Kaul, M. Gruteser, V. Rai, and J. Kenney. Minimizing age of information in vehicular networks. In *2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 350–358, 2011. doi: 10.1109/SAHCN.2011.5984917.

[62] S. Kaul, R. Yates, and M. Gruteser. On piggybacking in vehicular networks. In *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, pages 1–5, 2011.

[63] Sanjit Kaul, Marco Gruteser, Vinuth Rai, and John Kenney. Minimizing age of information in vehicular networks. In *2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 350–358. IEEE, 2011.

[64] JeongGil Ko, Chenyang Lu, Mani B Srivastava, John A Stankovic, Andreas Terzis, and Matt Welsh. Wireless sensor networks for healthcare. *Proceedings of the IEEE*, 98(11):1947–1960, 2010.

[65] Praveen Kumar Korrai, Eva Lagunas, Ashok Bandi, Shree Krishna Sharma, and Symeon Chatzinotas. Joint power and resource block allocation for mixed-numerology-based 5G downlink under imperfect CSI. *IEEE Open Journal of the Communications Society*, 1:1583–1601, Oct. 2020.

[66] Praveenkumar Korrai, Eva Lagunas, Shree Krishna Sharma, Symeon Chatzinotas, Ashok Bandi, and Björn Ottersten. A RAN resource slicing mechanism for multiplexing of eMBB and URLLC services in OFDMA based 5G wireless networks. *IEEE Access*, 8:45674–45688, Mar. 2020.

[67] Antzela Kosta. *Age of Information Aware Communication Systems: Modeling and Performance Analysis*, volume 2060. Linköping University Electronic Press, 2020.

[68] Antzela Kosta, Nikolaos Pappas, and Vangelis Angelakis. Age of information: A new concept, metric, and tool. *Foundations and Trends in Networking*, 12(3):162–259, 2017.

[69] Stefan Krauß, Peter Wagner, and Christian Gawron. Metastable states in a microscopic model of traffic flow. *Physical Review E*, 55(5):5597, 1997.

[70] Shiyang Leng and Aylin Yener. Age of information minimization for wireless ad hoc networks: A deep reinforcement learning approach. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2019.

[71] C. Li, Y. Huang, Y. Chen, B. Jalaian, Y. T. Hou, and W. Lou. Kronos: A 5g scheduler for aoi minimization under dynamic channel conditions. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 1466–1475, 2019. doi: 10.1109/ICDCS.2019.00146.

[72] Peiming Li and Jie Xu. Placement Optimization for UAV-Enabled Wireless Networks with Multi-Hop Backhauls. *Journal of Communications and Information Networks*, 3 (4):64–73, 2018. doi: 10.1007/s41650-018-0040-3.

[73] Juan Liu, Xijun Wang, Bo Bai, and Huaiyu Dai. Age-optimal trajectory planning for UAV-assisted data collection. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 553–558. IEEE, 2018.

[74] Ignacio Llatser, Andreas Festag, and Gerhard Fettweis. Vehicular communication performance in convoys of automated vehicles. In *2016 IEEE international conference on communications (ICC)*, pages 1–6. IEEE, 2016.

[75] F. Luo, C. Jiang, S. Yu, J. Wang, Y. Li, and Y. Ren. Stability of Cloud-Based UAV Systems Supporting Big Data Acquisition and Processing. *IEEE Transactions on Cloud Computing*, 7(3):866–877, 2019. doi: 10.1109/TCC.2017.2696529.

[76] Hien Phuong Luong, Manoj Panda, Hai L Vu, and Bao Quoc Vo. Beacon rate optimization for vehicular safety applications in highway scenarios. *IEEE Transactions on Vehicular Technology*, 67(1):524–536, 2017.

[77] N. Lyamin, B. Bellalta, and A. Vinel. Age-of-information-aware decentralized congestion control in vanets. *IEEE Networking Letters*, 2(1):33–37, 2020.

[78] F. Lyu, H. Zhu, N. Cheng, Y. Zhu, H. Zhou, W. Xu, G. Xue, and M. Li. Abc: Adaptive beacon control for rear-end collision avoidance in vanets. In *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9, 2018. doi: 10.1109/SAHCN.2018.8397130.

[79] Jiangbin Lyu, Yong Zeng, Rui Zhang, and Teng Joon Lim. Placement optimization of UAV-mounted mobile base stations. *IEEE Communications Letters*, 21(3):604–607, 2016.

[80] Ali Maatouk, Mohamad Assaad, and Anthony Ephremides. Minimizing the age of information in a CSMA environment. *arXiv preprint arXiv:1901.00481*, 2019.

[81] Ali Maatouk, Saad Kriouile, Mohamad Assaad, and Anthony Ephremides. The age of incorrect information: A new performance metric for status updates. *arXiv preprint arXiv:1907.06604*, 2019.

[82] Q. Mao, F. Hu, and Q. Hao. Deep learning for intelligent wireless networks: A comprehensive survey. *IEEE Communications Surveys Tutorials*, 20(4):2595–2621, 2018. doi: 10.1109/COMST.2018.2846401.

[83] A. Mehrizi-Sani and R. Iravani. Online set point modulation to enhance microgrid dynamic response: Theoretical foundation. *IEEE Transactions on Power Systems*, 27 (4):2167–2174, Nov. 2012. doi: 10.1109/TPWRS.2012.2190532.

[84] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518 (7540):529–533, 2015.

[85] Aya Moheddine, Fabio Patrone, and Mario Marchese. UAV and IoT Integration: A Flying Gateway. In *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pages 121–122, 2019. doi: 10.1109/ICECS46596.2019.8965135.

[86] Masoumeh Moradian and Aresh Dadlani. Age of information in scheduled wireless relay networks. In *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2020.

[87] Christian F Müller, Guillermo Galaviz, Ángel G Andrade, Irina Kaiser, and Wolfgang Fengler. Evaluation of scheduling algorithms for 5g mobile systems. In *Computer Science and Engineering—Theory and Applications*, pages 213–233. Springer, 2018.

[88] Gaurang Naik, Biplav Choudhury, and Jung-Min Park. IEEE 802.11 bd & 5G NR V2X: Evolution of radio access technologies for V2X communications. *IEEE Access*, 7:70169–70184, 2019.

[89] Neda Nasiriani, Roopa Ramachandran, Kaveh Rahimi, Yaser P Fallah, Parviz Famouri, Steve Bossart, and Keith Dodrill. An embedded communication network simulator for power systems simulations in pscad. In *2013 IEEE Power & Energy Society General Meeting*, pages 1–5. IEEE, 2013.

[90] Gam D Nguyen, Sastry Kompella, Clement Kam, Jeffrey E Wieselthier, and Anthony Ephremides. Impact of hostile interference on information freshness: A game approach. In *2017 15th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pages 1–7. IEEE, 2017.

[91] Hoa-Hung Nguyen and Han-You Jeong. Mobility-adaptive beacon broadcast for vehicular cooperative safety-critical applications. *IEEE Transactions on Intelligent Transportation Systems*, 19(6):1996–2010, 2017.

[92] Yuanzhi Ni, Lin Cai, and Yuming Bo. Vehicular beacon broadcast scheduling based on age of information (AoI). *China Communications*, 15(7):67–76, 2018.

[93] Taehyeun Park, Walid Saad, and Bo Zhou. Centralized and distributed age of information minimization with nonlinear aging functions in the internet of things. *IEEE Internet Things J.*, 8(10):8437–8455, May 2021.

[94] Doan Perdana, Aji Nur Sanyoto, and Yoseph Gustommy Bisono. Performance evaluation and comparison of scheduling algorithms on 5G networks using network simulator. *Int. Journal of Comp., Comm. & Cont.*, 14(4):530–539, Aug. 2019.

[95] Warren B Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*, volume 703. John Wiley & Sons, 2007.

[96] M. Prodanovic and T. C. Green. High-quality power generation through distributed control of a power park microgrid. *IEEE Transactions on Industrial Electronics*, 53 (5):1471–1482, Oct. 2006. doi: 10.1109/TIE.2006.882019.

[97] C Rajashekar Reddy, T Mukku, A Dwivedi, A Rout, S Chaudhari, K Vemuri, KS Rajan, and AM Hussain. Improving spatio-temporal understanding of particulate matter using low-cost iot sensors. In *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–7. IEEE, 2020.

[98] Javad Sabzehali, Vijay K. Shah, Harpreet S. Dhillon, and Jeffrey H. Reed. 3D Placement and Orientation of mmWave-based UAVs for Guaranteed LoS Coverage. *IEEE Wireless Communications Letters*, pages 1–1, 2021. doi: 10.1109/LWC.2021.3076463.

[99] Chiranjib Saha, Harpreet S. Dhillon, Naoto Miyoshi, and Jeffrey G. Andrews. Unified analysis of hetnets using poisson cluster processes under max-power association. *IEEE Transactions on Wireless Communications*, 18(8):3797–3812, 2019. doi: 10.1109/TWC.2019.2917904.

[100] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[101] Raja Sengupta, Shahram Rezaei, Steven E Shladover, Delphine Cody, Susan Dickey, and Hariharan Krishnan. Cooperative collision warning systems: Concept definition and experimental implementation. *Journal of Intelligent Transportation Systems*, 11 (3):143–155, 2007.

[102] Steven E Shladover and Swe-Kuang Tan. Analysis of vehicle positioning accuracy requirements for communication-based cooperative collision warning. *Journal of Intelligent Transportation Systems*, 10(3):131–140, 2006.

[103] S. Sofana Reka, Tomislav Dragičević, Pierluigi Siano, and SR Prabaharan. Future generation 5G wireless networks for smart grid: A comprehensive review. *Energies*, 12 (11):2140, 2019.

[104] NTT Smart Solutions. How Las Vegas is using data to boost public safety - Cities Today - Connecting the world's urban leaders. https://cities-today.com/industry/las-vegas-data-boost-public-safety/.

[105] Sunghwa Son and Kyung-Joon Park. Beat: Beacon inter-reception time ensured adaptive transmission for vehicle-to-vehicle safety communication. *Sensors*, 19(14):3061, 2019.

[106] Jaeyoung Song, Deniz Gunduz, and Wan Choi. Optimal scheduling policy for minimizing age of information with a relay. *arXiv preprint arXiv:2009.02716*, 2020.

[107] Mirko Stoffers and George Riley. Comparing the ns-3 propagation models. In *2012 IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 61–67. IEEE, 2012.

[108] Kamil H Suleiman, Tara Javidi, Mingyan Liu, and Somsak Kittipiyakul. The impact of mac buffer size on the throughput performance of ieee 802.11. *submitted to IEEE Transactions on Mobile Computing for publication on January*, 14, 2008.

[109] Yin Sun, Eiif Uysal-Biyikoglu, and Sastry Kompella. Age-optimal updates of multiple information flows. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 136–141. IEEE, 2018.

[110] M. H. Syed, E. Guillo-Sansano, D. Wang, A. Mehrizi-Sani, Y. Wang, G. M. Burt, and Y. Xu. Coordinated set point modulation of DERs for dynamically robust set point tracking as a virtual power plant. *IEEE Transactions on Power Systems*, Oct. 2020. submitted for review.

[111] Piotr Szczurek, Bo Xu, Ouri Wolfson, and Jie Lin. Estimating relevance for the emergency electronic brake light application. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1638–1656, 2012.

[112] Rajat Talak and Eytan Modiano. Age-delay tradeoffs in queueing systems. *IEEE Transactions on Information Theory*, 2020.

[113] Antony Tang and Alice Yip. Collision avoidance timing analysis of dsrc-based vehicles. *Accident Analysis & Prevention*, 42(1):182–195, 2010.

[114] Jayashree Thota, Nor Fadzilah Abdullah, Angela Doufexi, and Simon Armour. V2v for vehicular safety applications. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[115] Zhen Tong, Hongsheng Lu, Martin Haenggi, and Christian Poellabauer. A stochastic geometry approach to the modeling of dsrc for vehicular safety communication. *IEEE Transactions on Intelligent Transportation Systems*, 17(5):1448–1458, 2016.

[116] David Tse and Pramod Viswanath. *Fundamentals of wireless communication*. Cambridge university press, 2005.

[117] Alexey Vinel, Lin Lan, and Nikita Lyamin. Vehicle-to-vehicle communication in c-acc/platooning scenarios. *IEEE Communications Magazine*, 53(8):192–197, 2015.

[118] Xiaolin Wang, Cailian Chen, Jianping He, Shanying Zhu, and Xinping Guan. Aoi-aware control and communication co-design for industrial iot systems. *IEEE Internet of Things Journal*, 8(10):8464–8473, 2021. doi: 10.1109/JIOT.2020.3046742.

[119] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4): 279–292, 1992.

[120] G. Wikström, J. Torsner, J. Kronander, O. Al-Saadeh, F. Chernogorov, G. Bag, J. Neander, K. Landernäs, and P. Hovila. Wireless protection of power grids over a 5G network. In *2019 IEEE PES GTD Grand International Conference and Exposition Asia (GTD Asia)*, pages 976–981, Mar 2019. doi: 10.1109/GTDAsia.2019.8715864.

[121] Jie Xu, Cem Tekin, Simpson Zhang, and Mihaela Van Der Schaar. Distributed multi-agent online learning based on global feedback. *IEEE Transactions on Signal Processing*, 63(9):2225–2238, 2015.

[122] Roy D. Yates, Yin Sun, D. Richard Brown, Sanjit K. Kaul, Eytan Modiano, and Sennur Ulukus. Age of information: An introduction and survey. *IEEE Journal on Selected Areas in Communications*, 39(5):1183–1210, 2021. doi: 10.1109/JSAC.2021.3065072.

[123] M. Yi, X. Wang, J. Liu, Y. Zhang, and B. Bai. Deep Reinforcement Learning for Fresh Data Collection in UAV-assisted IoT Networks. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 716–721, 2020. doi: 10.1109/INFOCOMWKSHPS50562.2020.

[124] Haobo Yu, Lee Breslau, and Scott Shenker. A scalable web cache consistency architecture. *ACM SIGCOMM Computer Communication Review*, 29(4):163–174, 1999.

[125] Tom Zahavy, Matan Haroush, Nadav Merlis, Daniel J Mankowitz, and Shie Mannor. Learn what not to learn: Action elimination with deep reinforcement learning. *arXiv preprint arXiv:1809.02121*, 2018.

[126] S. Zhang and R. Zhang. Trajectory Optimization for Cellular-Connected UAV Under Outage Duration Constraint. *Journal of Communications and Information Networks*, 4(4):55–71, 2019. doi: 10.23919/JCIN.2019.9005434.

[127] Conghao Zhou, Hongli He, Peng Yang, Feng Lyu, Wen Wu, Nan Cheng, and Xuemin Shen. Deep RL-based trajectory planning for AoI minimization in UAV-assisted IoT. In *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6. IEEE, 2019.