Distributed Wireless Resource Management in the Internet of Things

Taehyeun Park

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

Walid Saad, Chair
Jeffrey H. Reed
T. Charles Clancy
Mantu K. Hudait
Douglas R. Bish

May 12, 2020
Blacksburg, Virginia

Keywords: Internet of Things, Learning, Game Theory

Distributed Wireless Resource Management in the Internet of Things

Taehyeun Park

ABSTRACT

The Internet of Things (IoT) is a promising networking technology that will interconnect a plethora of heterogeneous wireless devices. To support the connectivity across a massive-scale IoT, the scarce wireless communication resources must be appropriately allocated among the IoT devices, while considering the technical challenges that arise from the unique properties of the IoT, such as device heterogeneity, strict communication requirements, and limited device capabilities in terms of computation and memory. The primary goal of this dissertation is to develop novel resource management frameworks using which resource-constrained IoT devices can operate autonomously in a dynamic environment. First, a comprehensive overview on the use of various learning techniques for wireless resource management in an IoT is provided, and potential applications for each learning framework are proposed. Moreover, to capture the heterogeneity among IoT devices, a framework based on cognitive hierarchy theory is discussed, and its implementation with learning techniques of different complexities for IoT devices with varying capabilities is analyzed. Next, the problem of dynamic, distributed resource allocation in an IoT is studied when there are heterogeneous messages. Particularly, a novel finite memory multi-state sequential learning is proposed to enable diverse IoT devices to reallocate the limited communication resources in a self-organizing manner to satisfy the delay requirement of critical messages, while minimally affecting the delay-tolerant messages. The proposed learning framework is shown to be effective for the IoT devices with limited memory and observation capabilities to learn the number of critical messages. The results show that the performance of learning framework depends on memory size and observation capability of IoT devices and that the learning framework can realize low delay transmission in a massive IoT. Subsequently, the problem of one-to-one association between resource blocks and IoT devices is studied, when the IoT devices have partial information. The one-to-one association is formulated as Kolkata Paise Restaurant (KPR) game in which an IoT device tries to choose a resource block with highest gain, while avoiding duplicate selection. Moreover, a Nash equilibrium (NE) of IoT KPR game is shown to coincide with socially optimal solution. A proposed learning framework for IoT KPR game is shown to significantly increase the number of resource blocks used to successful transmit compared to a baseline. The KPR game is then extended to consider age of information (AoI), which is a metric to quantify the freshness of information in the perspective of destination. Moreover, to capture heterogeneity in an IoT, non-linear AoI is introduced. To minimize AoI, centralized and distributed approaches for the resource allocation are proposed to enable the sharing of limited communication resources, while delivering messages to the destination in a timely manner. Moreover, the proposed distributed resource allocation scheme is shown to converge to an NE and to significantly lower the average AoI compared to a baseline. Finally, the problem of dynamically partitioning the transmit power levels in non-orthogonal multiple access is studied when there are heterogeneous messages. In particular, an optimization problem is formulated to determine the number of power levels for different message types, and an estimation framework is proposed to enable the network base station to adjust power level partitioning to satisfy the performance requirements. The proposed framework is shown to effectively increase the transmission success probability compared to a baseline. Furthermore, an optimization problem is formulated to increase sum-rate and reliability by adjusting target received powers. Under different fading channels, the optimal target received powers are analyzed, and a tradeoff between reliability and sum-rate is shown. In conclusion, the theoretical and performance analysis of the frameworks proposed in this dissertation will prove essential for implementing an appropriate distributed resource allocation mechanisms for dynamic, heterogeneous IoT environments.

Distributed Wireless Resource Management in the Internet of Things

Taehyeun Park

GENERAL AUDIENCE ABSTRACT

The Internet of Things (IoT), which is a network of smart devices such as smart phones, wearable devices, smart appliances, and environment sensors, will transform many aspects of our society with numerous innovative IoT applications. Those IoT applications include interactive education, remote healthcare, smart grids, home automation, intelligent transportation, industrial monitoring, and smart agriculture. With the increasing complexity and scale of an IoT, it becomes more difficult to quickly manage the IoT devices through a cloud, and a centralized management approach may not be viable for certain IoT scenarios. Therefore, distributed solutions are needed for enabling IoT devices to fulfill their services and maintain seamless connectivity. Here, IoT device management refers to the fact that the system needs to decide which devices access the network and using which resources (e.g., frequencies). For distributed management of an IoT, the unique challenge is to allocate scarce communication resources to many IoT devices appropriately. With distributed resource management, diverse IoT devices can share the limited communication resources in a self-organizing manner. Distributed resource management overcomes the limitations of centralized resource management by satisfying strict service requirements in a massive, complex IoT.

Despite the advantages and the opportunities of distributed resource management, it is necessary to address the challenges related to an IoT, such as analyzing intricate interaction of heterogeneous devices, designing viable frameworks for constrained devices, and quickly adapting to a dynamic IoT. Furthermore, distributed resource management must enable IoT devices to communicate with high reliability and low delay. In this regard, this dissertation investigates these critical IoT challenges and introduces novel distributed resource management frameworks for an IoT. In particular, the proposed frameworks are tailored to realistic IoT scenarios and consider different performance metrics. To this end, mathematical frameworks and effective algorithms are developed by significantly extending tools from wireless communication, game theory, and machine learning. The results show that the proposed distributed wireless resource management frameworks can optimize key performance metrics and meet strict communication requirements while coping with device heterogeneity, massive scale, dynamic environment, and scarce wireless resources in an IoT.

To Qianqian Zhang and my parents

Acknowledgments

First, I would like to express my sincere gratitude to my PhD advisor, Dr. Walid Saad, for the continuous guidance and support throughout the course of my PhD study. His inspiration, knowledge, motivation and patience enable me to accomplish my PhD work including this dissertation. I am very grateful for all his time, dedication, and advice in mentoring my PhD work and for giving me the opportunity to benefit from his experience and brilliance. I am also thankful for his feedback and supervision that helped me to explore new research directions and allowed me to complete this dissertation. I would like to thank the members of my PhD advisory committee, Dr. Jeffrey H. Reed, Dr. T. Charles Clancy, Dr. Mantu K. Hudait, and Dr. Douglas R. Bish, for their insightful comments and invaluable guidance, which have helped me to greatly improve the quality of this dissertation.

I am grateful to all my friends and colleagues at the NEWS Lab and Wireless@VT for research discussions, support, and kind help. I am most grateful to Qianqian Zhang, who helped me throughout my time at Virginia Tech and supported me to overcome most difficult times.

I want to express my heartiest gratitude to my parents for their love, care, encouragement, and support.

This research been supported by the U.S. Office of Naval Research (ONR) under Grants N00014-15-1-2709 and N00014-19-1-2621.

Contents

List of Figures

List of Tables

Chapter 1

Motivation, Background, and Contributions

Internet of Things (IoT) will be an omnipresent network of billions of devices, such as smartphones, wearables, sensors, drones, and other machine type devices [1]. With massive network of devices, the IoT will enable many revolutionary applications that will affect many aspects of our daily lives and society. Applications of IoT include drone-based services [2–5], smart grid features [6–10], new healthcare applications [11], home automation [12], industrial monitoring [13, 14], and smart city [15]. To deliver a wide range of applications, a reliable wireless infrastructure is necessary to interconnect the devices. The IoT devices may be deployed using popular cellular networks or emerging long-range wide-area networks [16]. However, the deployment of IoT and the incorporation of IoT devices into wireless systems that may not be designed for them present many challenges, including coexistence with existing devices [17], device resource constraints [1], limited wireless communication resources [18], massive deployments, strict performance requirements [19, 20], and heterogeneity in message types. Many of these challenges lead to the next question of wireless resource management in IoT with specific requirements for effective operation. To this end, different approaches and their limitations to the question of wireless resource management in IoT are discussed.

1.1 Properties and Challenges of IoT

In this section, we present the defining properties, related challenges, and relevant applications of IoT.

1.1.1 IoT Devices

IoT devices include a wide range of devices from highly capable smartphones to low-cost machine type devices. This heterogeneity among the IoT devices poses a challenge when incorporating the devices into wireless networks, as it is unreasonable to expect low-cost sensors to operate at the same level as the smartphones. For instance, the battery life and the computational power between different IoT devices will vary greatly. As the computational capability of many low-cost IoT devices will be very limited, it is necessary to avoid computationally complex operations for the IoT devices.

The omnipresent IoT implies that many IoT devices are located at unfavorable locations for wireless communication, such as basement or remote locations [21,22]. Therefore, many IoT devices are located far from a reliable power source or a base station (BS). Since the IoT devices are expected to operate for many years on limited battery life [23, 24], the IoT devices cannot communicate frequently with a central entity, such as BS, due to their limited battery life or unfavorable location. Therefore, it is desirable that the IoT devices operate autonomously with minimal human or BS

intervention. Additionally, the IoT devices cannot communicate frequently with other nearby devices, and, thus, the IoT devices should be minimally dependent on nearby devices for information.

1.1.2 Performance Requirements

Taking advantage of omnipresent network of devices, many applications of IoT are related to monitoring and surveillance [11–14]. The IoT devices can be used to monitor health, environment, home, power usage, and industry. Typically, the monitoring applications are used to ensure that the activities are normal and to detect any extreme events quickly to respond. For instance, the smartwatches can be used to monitor and to record the heartbeat, but they can also be used to detect unusual levels of heartbeat. In a case where an IoT device detects extreme events, such as fire, critical health failure, and intrusion detection, it is necessary to report this finding as soon as possible. In certain monitoring applications of IoT, such events must be reported within a second or even in few milliseconds [20, 25]. Moreover, it is necessary that reporting the extreme events must be done reliably such that the packet delivery success rate must be very close to 1. This type of communication that requires to be very quick and highly successful is called *ultra-reliable low-latency communication* (URLLC) [25]. However, not all communications from the monitoring applications of the IoT require extreme performance levels as URLLC. For instance, the periodic meter reading and reporting in a smart meter application only require to be transmitted within a minute [20]. Therefore, there is a wide range of performance requirements, which depends on the message type, among the IoT devices.

With varying performance requirements from different message types, it is important to allocate the limited communication resources appropriately among the IoT devices such that they can satisfy the performance requirement of their message. Moreover, as the IoT environment is dynamic, there may suddenly be a message type that requires URLLC. In such a case, the allocation process must be quick to cope with low latency requirements of some message types.

1.1.3 Massive Scale

One of the most prominent features of the IoT is the massive scale. It is anticipated that there will be billions of devices worldwide, and there may be more than ten thousand machine type devices in a macrocell [26]. The massive scale of IoT implies that the IoT will require massive wireless connectivity for its operation. Therefore, the number of IoT devices will greatly outweigh the available communication resources. Moreover, if the wireless infrastructure is shared with pre-existing wireless network and its devices [1, 17], the availability of communication resources is even more limited. Therefore, due to massive scale of IoT and coexistence with pre-existing devices, the appropriate management of wireless communication resources is essential for the IoT.

1.2 Resource Management in IoT

As mentioned in previous section, many challenges presented by the IoT and its devices lead to the question of resource management in the IoT. Moreover, due to unique properties of the IoT and its devices, there are some desirable characteristics that the resource management should have to be effective in the IoT. As the resource management is one of the critical issues for the IoT, the resource management in IoT has attracted significant research in recent times. We present the existing works that can be categorized largely and that have addressed some of the key challenges of being effective resource management in the IoT.

1.2.1 Optimization

Optimization approach is widely used to manage wireless resources with specific objective, such as energy efficiency maximization, throughput maximization, and spectral efficiency maximization. Different multiple access designs have been proposed in order to maximize energy efficiency [23] or throughput [27]. Meanwhile, the work in [28] studies a resource allocation optimization for device-to-device communication using cellular networks by reusing the resource block of cellular user. The authors in [29] propose a resource allocation framework exploiting cloud radio access network for joint channel selection and power allocation. In [30], the authors study the multi-objective optimization problem of maximizing the coverage by adjusting the sensing range and, thus, energy consumption. The optimization approach is often centralized and computational complex in finding the optimal answer. The optimization cannot naturally handle the heterogeneity among the IoT devices without introducing significant overhead. One approach is to use multi-objective optimization approach [30], but this approach is often suboptimal and cannot scale well with massive IoT network.

1.2.2 Game Theory

One other promising resource management approach for the IoT is via the use of game-theoretic solutions [31]. However, to achieve such game-theoretic solutions, there is a need for distributed algorithms that can converge to an equilibrium in a timely manner. Indeed, the design of such algorithms is essential to practically deploy any game-theoretic construct within a real-world wireless system, such as the IoT.

1.2.3 Learning

Learning will allow the IoT devices to adapt to their dynamic environment, adjust their performance parameters, learn and process statistical information gathered from the users and other IoT devices, and optimize the overall system performance [32–35]. For the IoT, learning frameworks are particularly useful for managing limited resources and handling the aforementioned IoT properties [33, 36–39]. In [40], the authors consider a case in which the IoT is deployed using LTE and propose a reinforcement learning based eNB selection to avoid congestion. The authors in [33] presents different resource allocation problems, such as scheduling and time slot selection, that are solved with the machine learning approaches. For managing transmit powers, the states of reinforcement learning can be mapped as energy policies, the actions can be mapped as choosing the next best energy policy, and the rewards can be mapped as amount of energy saved [36]. As there are many types of learning frameworks with different properties, a learning approach for the distributed resource management scheme is very flexible. Furthermore, it is possible to expand and further develop a learning framework to use for the IoT. However, it is necessary to prove the convergence of a novel learning framework.

1.3 Limitations of Existing Works

The previous studies presented in Chapter 1.2 do not address the challenges arise from the properties of IoT and its devices as discussed in Chapter 1.1. The main limitations of the previous studies on the distributed resource management in the IoT are as follows.

- Even though the heterogeneity among the IoT devices in terms of the capabilities, message types, and performance requirements is a prominent property, the majority of the existing literature [19, 20, 33, 41–47] only

considers homogeneous message types from the IoT devices. In practice, there will be heterogeneous communication requirements, traffic patterns, and message types, and they will complicate the problem of the wireless resource management in the IoT.

- The majority of existing literature [19, 23, 29, 33, 40, 44] for the resource management in IoT relied on a centralized solution using which the BS gathers relevant information and then allocates the communication resources. However, the centralized solution may not be feasible to the IoT devices with limited battery life or at remote locations as frequent communication with BS is impractical. Moreover, the centralized solutions require significant overhead, which is not feasible with strict performance requirements of some message types. In contrast, a distributed solution to the resource management using which the devices can quickly reallocate the communication resources is needed.
- In the existing wireless resource management works in IoT [23,27,29,30,33,36,39–41], it is typically assumed that the devices and the BS are of unlimited computational and memory capabilities. Especially in a dynamic IoT environment, the wireless resources must be appropriately managed constantly and quickly to satisfy the performance requirements of IoT devices. However, many existing works are computationally complex for the IoT devices, and the resource management is not scalable even for BS with massive IoT network. However, in practice, the wireless resource management should be computationally simple considering the IoT devices with limited capabilities.
- The distributed resource management [11, 33, 36, 39] often requires the devices to know extensive information about its previous actions and corresponding results, or previous actions of all other devices in the IoT. As time passes, the amount of information necessary to appropriately allocate the wireless resources grows to infinity. Moreover, if information about all other devices is required, then the devices must go through long phase of sharing and gathering information, which is not feasible due to limited battery life and time this will take. In contrast, the distributed resource management can be of finite memory, which means that the resource management only requires a finite amount of information. There is a tradeoff between the amount of information and the accuracy of resource management, but the requirement of complete knowledge is not feasible for the IoT devices.

In summary, despite the significant body of work on wireless resource management in IoT, the existing literature is largely limited in scope and does not accurately capture the properties of IoT and its devices in implementing the distributed resource allocation.

1.4 Summary of Contributions

The main contribution of this dissertation is to develop the theoretical and performance analysis of the distributed wireless resource management mechanism for dynamic, heterogeneous IoT environment. In particular, we propose various distributed, low complexity resource management mechanisms tailored for dynamic, massive IoT environment and IoT devices of limited capabilities. By using the proposed frameworks, the IoT devices can dynamically adapt to the changing IoT environment and manage limited wireless resources autonomously in a balanced manner such that the performance requirements of different message types can be satisfied. To achieve these contributions, this dissertation will reply on a number of mathematical tools, including different types of learning and game theory. Furthermore, in taking these approaches, this dissertation will provide in-dept analytical foundations on the design of mechanism, convergence, and theoretical bounds on performance. In summary, our contributions are given as follows:

1.4.1 Learning in the IoT

As mentioned in Chapter 1.1, IoT and its devices have unique properties that pose as challenges when implementing an approach for the wireless resource management in the IoT. To be an effective resource management mechanism for the IoT devices, it should be distributed, low complexity, and finite in memory. However, many existing learning algorithms are not suitable to be used for the IoT and its devices as they are often computationally complex, centralized, slow in convergence, and inaccurate with limited information. Therefore, different learning frameworks with different suitable properties for the IoT are analyzed in terms of advantages, limitations, and potential IoT applications in chapter 2. In particular, machine learning, sequential learning, and reinforcement learning are analyzed. Moreover, for each learning, the inherent properties, computational complexity, and information requirement are analyzed to determine if a given learning is suitable for the IoT and its devices. Machine learning often requires extensive training data set to build a model. Additionally, the machine learning techniques typically are computationally complex and centralized. Despite these unfavorable properties, the machine learning techniques are useful for central entities, such as a BS or data aggregator, which can use for data analytic, such as data compression using principal component analysis. Sequential learning is distributed, flexible with information requirement, and effective in learning unknown parameters. However, the sequential learning relies on neighboring devices to obtain information and limited to binary state learning. Although it is limited to binary state learning, the sequential learning is effective in enhanced event detection and network operation adaption with learned information. Reinforcement learning is distributed, asynchronous, and low computational complexity. However, the reinforcement learning becomes computationally complex with incomplete information and has significant overhead in learning. In conclusion, many of the learning approaches are not readily applicable and require significant changes in algorithm design and then convergence proof.

1.4.2 Sequential Learning for Distributed Resource Management

As mentioned in Chapter 1.4.1, the sequential learning has suitable properties to be implemented in IoT, but it is limited to binary state learning. In Chapter 3, a novel finite memory multi-state sequential learning framework is proposed to reduce the expected delay of critical messages, while minimally affecting the throughput of the periodic messages, by IoT devices collectively reallocating the the random access preambles. This framework takes advantage of the fact that the sequential learning is very flexible with memory requirements, which means that the framework only requires small amount of information to learn and additional information can be used to increase the accuracy of learning. Additionally, the proposed framework has overcome the disadvantage of the sequential learning, which is limited to binary state learning, by expanding the learning to have an additional phase to consider multi-state. The proposed learning framework is proved to converge in probability, and the lower bound for the expected delay that can be achieved with our proposed framework is shown mathematically. The simulation results show that the effectiveness of the learning depends on the network density, memory size, and probability of detection.

1.4.3 Distributed Age of Information Minimization for IoT

Certain IoT applications require the most up-to-date information from devices, and the timely delivery of necessary information is crucial. A metric that can naturally quantify the freshness of the information in the perspective of the destination is the age of information (AoI). In Chapter 4, centralized and distributed approaches for the resource allocation minimizing non-linear AoI are proposed to enable the sharing of limited communication resources, while delivering IoT messages to the destination in a timely manner. Furthermore, non-linear aging functions are defined in terms of IoT devices types and message content, and the problem of AoI minimization is analyzed with non-linear aging functions. The proposed centralized resource allocation scheme uses priority scheduling with learning, while the distributed scheme has foundations in game theory. For the distributed scheme, the conditions to achieve a Nash equilibrium are shown, and a stochastic crowd avoidance algorithm is proposed and shown to converge to a Nash equilibrium. The two approaches are compared in terms of their overhead, implementation requirements, and

performance via simulations. Simulation results show that priority scheduling with learning achieves significantly lower average AoI when compared to simple priority scheduling without learning, while stochastic crowd avoidance achieves significantly lower average AoI when compared to random allocation. The results also show that centralized scheme outperforms distributed scheme in almost all cases, but the distributed scheme is more practical for a massive IoT.

1.4.4 Game Theoretic Approach for Distributed Resource Management

The wireless resource management can also be modeled via game theory. In Chapter 5, a massive IoT environment in which the devices compete over resource blocks is modeled as a Kolkata Paise Restaurant game. In this game, the IoT devices each have a preference ranking on the available resource blocks based on their channel gains and choose which resource block to use without cooperation every iteration. A device gets positive utility for choosing a resource block alone, while a deice get zero utility for duplicate selection of a resource block as it will result in transmission failure. In IoT Kolkata Paise Restaurant game, it is shown that the Nash equilibrium coincides with the socially optimal solution to the game. The socially optimal solution is when the service rate, which is the percentage of resource block that is chosen by only one device, is highest, and a learning framework is proposed to significantly increase the service rate without cooperation between the devices. Simulation results show that the service rate resulting from the proposed learning framework is a function of communication range, device density, and transmission probability.

1.4.5 Distributed Resource Management in NOMA

Non-orthogonal multiple access is a promising multiple access scheme that will allow massive connectivity of IoT devices to the BS. However, it is still true that the number of IoT devices greatly outweigh the available wireless communication resources. In Chapter 6, a novel estimation algorithm is proposed to enable the BS to accurately detect a change in traffic pattern among the IoT devices and then to adjust the power partitioning to satisfy the transmission success probability of the high-priority messages, while minimally affecting the throughput of the periodic message. An optimization problem is formulated to determine the power partitioning based on the estimated probability of having high-priority messages. Simulation results show that the transmission success probability is increased by 77% compared to the baseline case of without any estimation framework. In Chapter 7, a problem of uplink transmit power assignment is studied for a wireless network with power domain non-orthogonal multiple access under a random fading channel. This problem is posed as an optimization of target received signal powers at the base station to maximize the sum-rate and the reliability, where the received signal power at the base station is unknown to the devices due to a fading channel effect. To find the optimal allocation of the lower and higher target received powers, the expected sum-rate and the reliability are derived in terms of target received power values and power difference threshold. For Rayleigh fading channel, both theoretical analysis and simulation results show that the highest expected sum-rate and the highest reliability are achieved when the target received power values are highest. However, for Nakagami-m fading channel, simulation results show that there is a tradeoff between reliability and sum-rate

1.5 List of Publications

As a byproduct of the above contributions, thus far, this dissertation has led to the following key publications:

1.5.1 Journal Publications

1. **T. Park**, N. Abuzainab, and W. Saad, "Learning how to communicate in the internet of things: Finite resources and heterogeneity," IEEE Access, vol. 4, pp. 7063–7073, November 2016.
2. **T. Park**, W. Saad, "Distributed learning for low latency machine type communication in a massive internet of things," IEEE Internet of Things Journal, vol. 6, no. 3, pp. 5562-5576, June 2019.
3. **T. Park**, G. Lee, W. Saad, and M. Bennis, "Sum-Rate and reliability analysis for power-domain non-orthogonal multiple access (PD-NOMA)," submitted for publication, April 2020.
4. **T. Park**, W. Saad, and B. Zhou, "Centralized and distributed age of information minimization with non-linear aging functions in the internet of things," submitted for publication, April 2020.

1.5.2 Conference Publications

1. **T. Park** and W. Saad, "Learning with finite memory for machine type communication," in *Proc. of IEEE 50th Annual Conference on Information Science and Systems*, NJ, USA, March 2016, pp. 608-613.
2. **T. Park** and W. Saad, "Resource allocation and coordination for critical messages using finite memory learning," in *Proc. of IEEE Global Communications Conference Workshops*, Washington DC, USA, December 2016, pp. 1-6.
3. **T. Park** and W. Saad, "Kolkata paise restaurant game for resource allocation in the Internet of Things," in *Proc. of 51st Asilomar Conference on Signals, Systems, and Computers*, CA, USA, October 2017, pp. 1774-1778.
4. **T. Park**, G. Lee, and W. Saad, "Message-aware uplink transmit power level partitioning for non-orthogonal multiple access (NOMA)," in *Proc. of IEEE Global Communications Conference*, Abu Dhabi, UAE, December 2018, pp. 1-6.
5. **T. Park**, W. Saad, and B. Zhou, "On the minimization of non-linear age of information in the internet of things," submitted for publication, May 2020.

Chapter 2

Learning to Manage Wireless Resource in IoT

2.1 Background, Related Works, and Contributions

As mentioned in Chapter 1, the resource management in IoT to enable heterogeneous devices to operate autonomously has many challenges, including low capability IoT devices, strict performance requirements, massive scale, limited communication resources, and co-existence with previous wireless networks [1, 17, 19, 20]. To develop resource management mechanisms tailored to the IoT, different approaches, such as optimization theory [23, 28, 48] or game theory [31], have been applied. In this regard, optimization techniques are often centralized, which can incur unnecessary overhead and will thus require constant, energy-consuming communication between the energy limited IoT devices and BS. Moreover, most optimization frameworks cannot naturally handle the heterogeneity of IoT devices without significant additional overhead. One recently proposed solution was via the use of multi-objective optimization [30], however, such an approach is often suboptimal and may not scale well as the network size increases. One other promising resource management approach for the IoT is via the use of game-theoretic solutions [31]. However, to achieve such game-theoretic solutions, there is a need for distributed algorithms that can converge to an equilibrium in a timely manner. To this end, one effective approach to overcome the aforementioned challenges and enable self-organizing operation of the IoT is via the use of *learning frameworks*. Learning will allow the IoT devices to adapt to their dynamic environment, adjust their performance parameters, learn and process statistical information gathered from the users and other IoT devices, and optimize the overall system performance [32–35].

The main contribution of this chapter is to provide a comprehensive overview on the use of learning techniques within the scope of the IoT. Despite some recent surveys on the challenges of wireless communications in the IoT, such as in [49] and [17], none of these works have investigated the potential of developing learning techniques for the IoT. As such, to our best knowledge, this will be the first comprehensive tutorial on this topic. First, we discuss the various types of learning frameworks that are suitable to address a number of key challenges of the IoT. Then, for each learning approach, we outline the basic concepts, main challenges, potential applications, and key results. Subsequently, to handle system heterogeneity, we introduce the basics of cognitive hierarchy theory and its application to the IoT. Then, we draw the key connections between cognitive hierarchy theory and the different classes of learning algorithms overviewed. We conclude by summarizing the potential of learning in the IoT.

The rest of this chapter is organized as follows. In Section 2.2, we present the motivations and challenges of applying learning frameworks in the IoT. Section 2.3 discusses the advantages, limitations, and applications of the learning frameworks within the scope of the IoT. In Section 2.4, we introduce cognitive hierarchy theory and analyze its

implementation with learning frameworks in the IoT. Finally, we summarize our key findings in Section 2.5.

2.2 Learning in the IoT: Motivation and Challenges

2.2.1 Motivation

The sheer scale of the IoT makes it impractical to manage the devices manually, and, hence, it is essential to operate the devices in a self-organizing manner, with minimal human intervention. Moreover, the devices will be deployed in various environments depending on their applications. Examples include deployments in an urban area with many diverse devices, a distant forest with few sensors for environmental monitoring, an indoor environment for home automation, or a rural area with devices spread across a large field for agricultural applications. Furthermore, the deployment environment for the IoT will be highly dynamic due to unpredictable events, such as weather conditions, power outages, or medical emergencies. Such unpredictable events will trigger uplink transmission links that must be serviced with low latency and high reliability due to their urgent nature [37]. However, satisfying the QoS requirements of such uplink transmissions is challenging, because existing wireless networks have been designed primarily for H2H communications and the sheer scale of the IoT will strain their resources, thus requiring new approaches for enhancing network operation.

The adaption of learning techniques will provide an effective solution for enabling the IoT devices to adapt to dynamic environments, manage their limited resources, and satisfy the stringent QoS requirements. To manage its resource consumption, an IoT device can learn about its environment, devices, users, and their usage patterns. For instance, MTDs can learn about their usage patterns in terms of peak workload intervals to adjust their status, such as active or sleep, to minimize the energy consumption [36]. Moreover, learning can be used to coordinate the usage of limited radio resources among the IoT devices and human type devices (HTDs). For example, unpredictable events may trigger uplink communication links with stringent QoS requirements. Here, by using a proper learning framework, the uplink communication resources can be allocated for the IoT devices that are sending the data pertaining to the unpredictable events in a way to minimize the effects, such as throughput reduction or increased latency, on the normal uplink communications of IoT. Furthermore, learning can also enable the IoT to process and learn from the massive volume of data collected from the IoT devices. Indeed, learning is a key enabler for big data analytics that will be used to process the data related to the IoT, its human users, and its environment. Therefore, learning techniques in the IoT can be used to extend the lifetime of IoT devices, to mitigate the uplink communication bottleneck, to promote a coexistence of IoT and pre-existing systems, and to process the massive data from IoT.

2.2.2 Challenges

To effectively use learning for solving IoT-centric problems, several challenges that stem from the unique characteristics and requirements related to the IoT and its devices must be met, as described next.

- A key property of low-cost, low-capability MTDs is *low computational capability* as discussed in [1] and [33]. However, existing learning frameworks, such as decision trees [33] or reinforcement learning [48] and [50], can be computationally complex to be adapted by the resource-constrained MTDs.
- *Energy limited* MTDs are expected to have *extended lifetime* with *minimal human intervention* as pointed out in [11, 23], and [24]. Since a centralized learning framework will require a constant communication with the BS and potentially cause an excessive energy consumption, learning for the IoT devices must be distributed. Moreover, the MTDs may be deployed in adverse locations in which the BS is not always accessible to deploy a centralized framework. Therefore, a distributed learning technique is needed to lower energy consumption and enable the IoT devices to be self-organizing.

- To implement distributed learning methods in the IoT, the necessary information, such as the actions of other IoT devices, may need to be collected via resource consuming M2M communication links. However, with a lack of radio resources in the IoT and energy constraints of the IoT devices as pointed out in [1, 11], and [33], a frequent M2M communication to collect the necessary information for learning is not viable for the IoT devices, and thus they may have *limited information available*. Therefore, any deployed learning framework must be able to accurately adjust the performance parameters with limited information.
- Certain IoT applications, such as industrial control or health monitoring, are of critical importance, and thus such applications require *ultra-reliable, low-latency communication* as discussed in [19,20], and [9]. To satisfy such QoS requirements, the IoT devices must quickly learn to adjust their performance parameters. Therefore, the time necessary to converge to a steady state must be short.
- The IoT may be deployed using an existing wireless network, and thus *M2M communication will coexist with pre-existing communication links* as pointed out in [1] and [17]. In particular, the IoT is likely to be deployed over a wireless cellular network in which M2M communications will have to coexist with H2H communications. For harmonious coexistence, the learning techniques for the IoT must consider both existing traffic and the new traffic potentially stemming from the IoT. This is due to the fact that coexisting wireless networks will inevitably impact each other due to factors, such as interference, and it is necessary to not disrupt the existing network and its services.

To address the aforementioned challenges, next, we introduce different classes of learning techniques, while providing a key overview on their applicability to the IoT.

2.3 Classification of Learning Frameworks for the IoT

Learning has been developed for many disciplines ranging from socioeconomic to mathematical modeling which has led to a variety of learning frameworks having different properties, objectives, and capabilities, such as in [32, 33, 35, 37, 38], and [51]. For the IoT, there is no single learning framework that can be used to overcome all of its aforementioned challenges, and, thus, it is necessary to characterize the key learning frameworks that can be useful for various IoT applications. For each such framework, one needs to identify the main advantages and limitations. For example, a learning framework that is computationally simple but requires a long time to converge is suitable for the delay-tolerant IoT applications involving simple sensors, however, this will not be viable for critical applications requiring low latency.

To this end, we propose to classify the rich literature on learning into three classes of learning frameworks that include: a) machine learning, b) sequential learning, and c) reinforcement learning. This classification will allow us to discuss the main benefits of implementing each class in the IoT and to determine the learning frameworks that are suitable for a given aspect of the IoT. Indeed, machine learning, sequential learning, and reinforcement learning are proven to be particularly useful to address problems pertaining to big data analytics, enhanced event detection, and resource management, respectively, as in [33, 37], and [38]. For each class of learning, we will identify the main advantages, challenges, IoT applications, and key results. In addition, we will identify their inherent properties, in terms of computations, information requirement, and capabilities. Here, we note that, although learning has been very popular for wireless networks [50], there has not been yet any comprehensive overview on applicability and limitations of learning in the IoT.

2.3.1 Machine Learning

Key Concepts of Machine Learning

Machine learning (ML) techniques were originally developed for allowing computers to autonomously learn information from existing data sets and, subsequently, build suitable models to make a decision on future actions and behaviors as discussed in [32] and [33]. ML techniques are typically categorized into supervised and unsupervised learning [32]. Supervised learning requires a labeled training data, while unsupervised learning, such as clustering, does not require labels. However, unsupervised learning is more computationally complex than supervised learning. Furthermore, for certain ML techniques, there may be specific requirements for the training data set. For instance, decision trees require the data set to be linearly separable [32].

Using the training data set, ML techniques can build regression models to determine the relationship between the variables, divide the feature space to classify the unlabeled data points, cluster the data points to divide into different groups, or lower the dimensionality of feature space by removing the correlated features as discussed in [11, 32], and [39]. However, ML techniques may be computationally complex and implicitly require a central entity with significantly high computational capabilities. Therefore, ML techniques are often centralized.

Limitations and Opportunities of ML for the IoT

The biggest limitation of applying ML in IoT scenarios is that ML requires an extensive data set, such as the information of sensor locations and corresponding sensor measurements [33], for good performance. Such a data set needs to be quickly processed for the IoT devices to learn the environment, but the resource-constrained IoT devices may not be able to store and process the data set given that they have limited resources in terms of computation and memory [39]. Furthermore, supervised learning requires a labeled training data set, which can require human intervention to provide the correct labels. This is due to the fact that assuming that the IoT devices are able to label the data points correctly and autonomously would imply that their learning is already done a priori. Therefore, the use of ML must be properly tailored to such IoT features and device limitations.

Nonetheless, ML has been recently used in the IoT as a centralized framework as in [33, 52–55]. This is because a cloud-based centralized processing unit can be used to implement effective ML schemes. Since ML has a great potential in processing and analyzing the massive data collected in an IoT environment as discussed in [11] and [52], such a cloud-based processing unit will allow the IoT to run the ML techniques for big data analytics purposes [11]. The challenges of implementing a cloud-based processing unit for the IoT are designing a scalable wireless architecture with many cooperating wireless access points, developing advanced antenna technologies to boost throughput and reliability, and ensuring the confidentiality and security of data [52]. However, with a cloud-based processing unit, it is possible to determine spatial, temporal, and social correlations of the data traffics to reduce communication overheads, increase energy efficiency, and provide user-oriented services as discussed in [52] and [53].

Applications of ML in the IoT

One of the key applications of ML in the IoT is big data analytics as massive data will be accumulated at either BS or local data aggregators [11]. The key function of ML in the IoT is to eliminate the correlated information and to reduce the dimensionality of big data in the IoT [33]. This will make the data transmission from the data aggregators to BS less costly and the data processing at BS more efficient [33]. However, it is necessary to maintain the key information in the data.

Principle component analysis (PCA) has been used to intelligently compress the collected data by reducing the dimensionality, while keeping the key features of the data as discussed in [33, 52, 54], and [55]. PCA uses an orthogonal

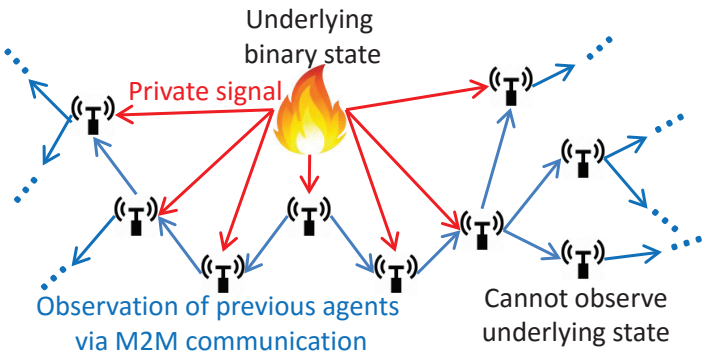


Figure 2.1: Sequential Learning in the IoT.

transformation to output a set of linearly uncorrelated data from possibly a correlated input data. PCA has been shown to be effective in reducing the resource consumption by compressing the data. For instance, PCA based query reduces the energy consumption by at least 25% compared to the normal query [53]. Therefore, PCA is particularly useful for the data aggregators and the BS for data processing. However, PCA is computationally complex and can require a significant time to process [33]. Although the high computational complexity may be feasible for a BS with sufficient resources, it will be challenging to deploy at the level of a data aggregator, which is often a typical IoT device with limited resources.

Since data aggregation is one of the popular solutions to mitigate the lack of radio resources in the IoT as discussed in [33,54] and [55], it would be worthwhile to investigate using different PCA-based approaches to analyze the tradeoff between computational complexity, data compression, and resource consumption. Moreover, analyzing different PCA-based approaches will help choosing an optimal data aggregation method for a given IoT application. For instance, PCA-based data aggregation with small delay can be used for the low latency IoT applications. Additionally, emerging ML techniques, such as echo state networks, can be useful to develop predictive approaches for resource allocation in the IoT as discussed in [56]. Finally, ML has been recently shown to be effective in deploying new security techniques for the IoT, such as IoT device authentication [57] or static malware analysis [58].

2.3.2 Sequential Learning

Key Concepts of Sequential Learning

In an environment with an underlying binary state, a number of autonomous agents can learn what the true underlying state of the environment is by using sequential learning (SL) as discussed in [34] and [59]. Here, such agents are intelligent entities that are capable of learning from a given information, and they would potentially correspond to the IoT devices if SL is implemented in the IoT. Here, the underlying state describes the current state of the IoT environment in which the agents are located, and it corresponds to events of interest, such as a medical status, fire alarm, or other environment-triggered events, within the IoT [37]. Using SL, the agents learn the state of the system by following a given order while observing the environment and the actions or observations of previous agents in the sequence. Moreover, the agents will eventually converge to a consensus on the true underlying state with repeated hypothesis testing as explained in [34] and [59]. For different agents, the observations of the environment, which are also known as the *private signals*, are independent and equally informative about the underlying state as pointed out in [34] and [51]. Such private signals are typically modeled as independent binary signals whose distribution depends on the underlying state such as in [37] and [51]. Furthermore, a private signal cannot fully reveal the underlying state of the system, which means that the likelihood ratio cannot be zero or infinity [51].

Another information necessary for SL is the observation of previous agents and specifically their estimates of the

underlying state. Depending on the number of previous estimates required, SL can be classified into *infinite* and *finite memory* as discussed in [34], [51], and [59]. In infinite memory SL, agents must observe all previous agents in the sequence, and thus the memory of previous estimates grows infinitely. In finite memory SL, agents only observe a fixed number of previous agents, and thus the memory of previous estimates is fixed [37]. SL with finite memory requires much less information than infinite memory SL at the expense of higher probability of error in estimating the underlying state [37].

One of the favorable properties for implementing SL approaches in the IoT is that SL can converge to the correct underlying state by only observing two previous agents as discussed in [34] and [51]. Unlike ML, SL does not require an extensive data set, but it relies on the sequential propagation of information among the agents. Moreover, SL can be implemented using distributed mechanisms, while the majority of ML techniques are centralized. Furthermore, SL is more suitable for enhancing event detection, while ML techniques are more suitable for data analytics.

Limitations and Opportunities of SL for the IoT

The majority of information necessary for SL stems from other agents, and, thus, SL must rely on the use of direct M2M communication links as illustrated in Fig. 2.1, which can consume additional network resources. Moreover, in SL, the IoT devices that are not able to communicate with any other device will not be able to learn. For infinite memory SL, the amount of necessary information will increase indefinitely as learning progresses. This implies that the size of the M2M communication packets may grow infinitely, which can become prohibitive in an IoT environment. However, finite memory SL will only require the packets of fixed size to be transmitted using M2M links, thus making it much more viable for IoT.

Another limitation of SL is that it is typically limited to learning a binary state. Although there are practical IoT scenarios that only have two states, such as whether to go to sleep mode to conserve energy or not in minimizing power consumption, many IoT scenarios will involve more than two states, such as multiple levels of transmit power. Furthermore, the private signals in the IoT may not be equally informative as the IoT devices can have different observations of the environment as shown in Fig. 2.1. However, one can overcome this private signal assumption by introducing additional information about previous IoT devices [37].

The biggest advantage of applying SL in the IoT is its flexibility with memory requirements. For finite memory SL, the IoT devices can observe different number of previous agents, and the SL will still converge [37]. Therefore, the IoT devices can optimize the amount of information to use for SL depending on the available resources. However, there is a lower bound on the number of observations of previous agents necessary for the convergence [51].

Applications of SL in the IoT

SL is particularly useful in enhancing the IoT event detection, and it can be used for distributed resource allocation in the IoT. For instance, there may be urgent events, such as a medical emergency, that trigger uplink communications requiring low latency. However, satisfying the QoS requirements of such communications is not trivial due to the uplink communication bottleneck caused by the IoT devices regularly reporting their data. In such a case, the IoT devices using SL can collectively and autonomously detect the urgent events and then allocate necessary radio resources for the event-triggered communications. Since periodic data reporting to the BS is also important, resource allocation for the event-triggered communications must be done in a way to minimize the amount of resources that need to be re-allocated from the devices that have periodic data to send.

Finite memory SL has been shown to be effective in autonomously allocating the radio resources to reduce the delay for the urgent, event-triggered communications in the IoT with limited uplink communication resources in [37]. Using our proposed approach based on finite memory SL in [37], the IoT devices transmitting delay-tolerant messages can collectively reallocate uplink communication resources for the IoT device transmitting urgent messages, while

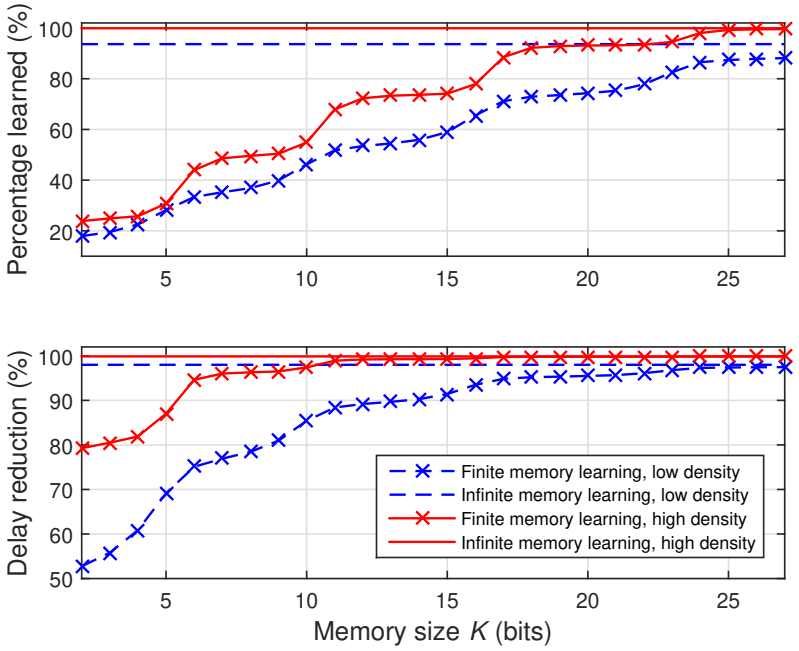


Figure 2.2: Effectiveness of SL for Different Memory Sizes and Device Densities.

minimally affecting the system throughput and significantly reducing the delay of the urgent message. In [37], it is shown that the delay of urgent communication can be further reduced at the cost of bigger memory size for SL and greater throughput reduction of the periodic communications. A key feature of finite memory SL is that the memory size can be optimized to satisfy a given QoS requirements. Fig. 2.2, based on our SL model in [37], shows that more devices will learn the correct state of the network and the delay will be reduced more with bigger memory sizes. However, bigger memory sizes will require more radio and energy resources to be consumed to transmit and to process the information. Therefore, the memory size must be chosen appropriately considering the performance improvements and available resources. Furthermore, Fig. 2.2 shows that the finite memory SL is more effective with higher device deployment density. This is due to the fact that, in a dense IoT environment, there will be a few devices that cannot communicate with any other device, which prevents such devices from learning. This also shows that SL-based techniques will generally be M2M communication reliant.

2.3.3 Reinforcement Learning

Key Concepts of Reinforcement Learning

Reinforcement learning (RL) is a learning technique in which a number of agents learn how to act by interacting with the environment as discussed in [33] and [35]. An RL algorithm is typically composed of a number of agents with their corresponding sets of actions, an environment with a set of states, a state transition function, an immediate reward function, and an initial observation function [35]. At the beginning of each time period, each agent observes the environment and takes an appropriate action to maximize the immediate or future reward. At the end of each time period, each agent will receive an immediate reward, and the state of the environment will change according to the state transition function. The agents iterate according to this process to learn and to converge to a steady state as explained in [33] and [35].

One of the unique features of RL is the use of action-reward combination as a feedback to the agents. For instance, using the future rewards as a feedback, RL can be used to maximize their long-term rewards as pointed out in [33, 35], and [38]. Furthermore, the function used for choosing the current action can be computationally simple. For instance, the well-known Q-function, which is purely algebraic, can be used [33].

Although RL can be computationally simple, it can take significant time to converge to a steady state as discussed in [38] and [60]. This is because RL learns by exploring different states. Although RL does not require an extensive training data, it requires its agents to know the state transition function as pointed out in [33] and [35]. The slow convergence and the requirement to know the state transition function are key challenges facing the use of RL in an IoT environment, however, the unique design of having actions and rewards between the agents and the environment makes RL very useful to treat a number of IoT problems as mentioned in [61–63].

Limitations and Opportunities of RL for the IoT

The assumption that the state transition function is known is generally needed for many RL algorithms, however, such an assumption may not hold true in the IoT. Indeed, the environment in which the IoT devices are deployed is subject to unpredictable events and uncertainties, such as abnormal sensor readings due to sudden increase in noise, device failures, and unexpected obstacles preventing normal device operation as discussed in [38] and [60]. As such, it is possible for the IoT devices to experience an incomplete state transition function. Moreover, it is impractical to account for all possible scenarios that the IoT devices may face in designing the state transition function. Therefore, there may be hidden or unknown states, which the IoT devices can reach at some point. These hidden states are critical to RL as they hinder with determining the reward maximizing action as pointed out in [38] and [60].

The hidden or unknown states in RL can be mitigated by introducing a memory of previous actions [38]. In this case, the IoT devices can combat uncertainties in choosing an action at the cost of higher computational complexity and slower convergence [38]. Therefore, RL may not be directly applicable to low latency IoT applications due to its potentially slow convergence.

Although RL may have a slower convergence than SL, it does not require significant interactions between the devices over M2M links, which will save significant amount of energy, and it can be used to find the equilibrium solutions for game-theoretic models [50]. Moreover, RL does not require the devices to be globally synchronized and to engage in the learning process at the same rate, which is advantageous in a distributed system such as the IoT [33]. Furthermore, RL can be computationally simple for low-capability devices in IoT [33]. Due to its aforementioned, desirable properties, RL has been one of the most popular learning frameworks in the IoT [61–63].

Applications of RL in the IoT

RL can be used for IoT resource management, because the components of RL, such as actions and states, can be easily mapped to the corresponding components of resource management in the IoT [61–63]. For example, for power management application, the states can be mapped as energy policies, the actions can be mapped as choosing the next best energy policy, and the rewards can be mapped as amount of energy saved [36]. More recently, the use of RL has been investigated in [61] as a mean to provide optimized resource management for an IoT system that relies on drones to deliver communications to disaster affected areas using both licensed and unlicensed bands. Here, the drones are the agents, and their actions pertain to choosing the proper duty cycle to enable a co-existence between cellular and WiFi communications via the drones [61]. Therefore, the key step in applying RL in IoT scenarios is finding appropriate counterparts to each component of RL.

The state transition function is the most challenging to define in the IoT, because it is typically required for all devices to know this function. In the given example of power management, the state transition function is defined such that the next state only depends on the current action [36]. However, for other resource management applications in the IoT,

Table 2.1: Summary of the Proposed Learning Frameworks

Learning frameworks	Advantages	Limitations	Applications
Machine Learning	• Diversity of available techniques with a multitude of applications. • Useful in processing the massive data collected in IoT to reduce resource consumption. •Enabler for big data analytics and predictive solutions.	• Implementation typically centralized. • Need for significant computational capabilities. • Need for extensive training data.	• Data aggregation and compression [54, 55]. • Query processing [53]. • Big data analytics [11, 33, 52]. • IoT security [57, 58].
Sequential Learning	• Distributed implementation. • Flexible memory and resource requirements. • Lack of a need for extensive knowledge of the system. • Ability to effectively learn unknown parameters.	• Reliance on M2M communication. • Limited to binary state learning. • Need for a private signal for learning.	• Enhanced event detection. • Dynamic resource management under uncertainty [37]. • Network operation adaptation.
Reinforcement Learning	• Distributed implementation. • Asynchronous operation. • Low computational complexity. • Ability to find game-theoretic equilibrium solutions.	• Significant overhead to reach steady state. • Need for complete information about state transition. • Higher computational complexity with incomplete information.	• Power control [36]. • Radio resource management for IoT environments [62–64]. • Dynamic scheduling for energy efficiency [62]. • Use of drones for enhanced communications [61].

the state transition function may be complex and dependent on other devices as well. For instance, in radio resource management, the actions may be choosing a channel to use for uplink communication, and the state can be signal-to-interference-plus-noise ratio of the selected channel [64]. In this application, the state transition function depends on other devices, and the next state is uncertain as the actions of other devices are unknown. Although the use of memory improves the performance of RL by allowing the uncertain states to exist [38], it is necessary to investigate the tradeoffs associated with having a memory.

There are different types of memory that RL algorithms can adapt to mitigate the effects of having unknown states [38]. Moreover, the different approaches of using memory require different amounts of information and introduce varying levels of computational complexity [38]. Furthermore, different approaches have different properties in terms of scalability, convergence time, and quality of steady state [38]. Therefore, similar to SL, it is necessary to choose a type of memory for RL that is most suitable considering the available resources and QoS requirements. This will make RL much more applicable for the IoT.

In summary, the IoT provides an environment that is ripe for applying a variety of learning algorithms. However, such algorithms must be properly designed and tailored to the intrinsic properties of the IoT. The various challenges, opportunities, and applications of learning are summarized in Table 2.1. Next, we discuss how learning approaches can be designed in a way to cope with the heterogeneity of the IoT environment.

2.4 Managing IoT Heterogeneity using Cognitive Hierarchy

2.4.1 Motivation

The aforementioned learning frameworks exhibit different resource requirements and can achieve varying levels of accuracy in the learned parameters. For instance, the agents using SL can have more observations of previous agents in the sequence with more memory, thus having more information about the underlying state [37]. Therefore, SL schemes with more memory will allow the IoT devices to more accurately learn the underlying state, thus achieving

lower probability of estimation error. On the other hand, the IoT devices can also have different available resources. For example, smartphones will be able to do advanced learning methods as they are computationally capable and have extensive memory, while simple sensors with limited resources available must resort to elementary learning methods. Therefore, the difference in the available resources among the IoT devices will cause them to reach different learning outcomes. Although most existing learning frameworks typically assume that the agents have the same available resources, it is necessary to capture and exploit the heterogeneity in available resources among the IoT devices.

In this regard, *cognitive hierarchy theory (CHT)* [65, 66], and [67], is a promising tool to accurately model the IoT, because it provides a modeling framework that can properly capture the heterogeneity among the agents. CHT divides a number of agents into different groups of varying rationality levels. Different rationality levels can be interpreted as different levels of available resources among the IoT devices. Thus, CHT will allow each group of agents (devices) at a given rationality level to choose a learning framework that is most appropriate based on their available resources. Therefore, CHT will make it possible to integrate different learning frameworks at different levels in the IoT, thus maximizing the overall usage of the resources and providing a more realistic model for a heterogeneous IoT system.

2.4.2 Preliminaries and Key Concepts

CHT is a branch of behavioral game theory and is based on the concept of bounded rationality [66, 67]. In general, bounded rationality means that each agent finds the best strategy based on its accessible information, its computational capacity, and the time available. In CHT, it is assumed that the agents are distributed into discrete levels of rationality. Agents belonging to the lowest level 0 are completely irrational and choose their strategy randomly. Agents at any higher level $k \geq 1$ believe that all others belong to the levels lower than k and choose their optimal strategy based on their beliefs. Therefore, to find the optimal strategy, any agent at level $k \geq 1$ starts by computing the strategy of level 1 agents then level 2 agents up to level $k-1$ agents and finally computes its optimal strategy, hence performing k levels of thinking.

The most popular CHT model is the Poisson model [67]. It relies on the following considerations. First, it considers that the agents are distributed into the rationality levels according to a Poisson distribution f with rate τ. The Poisson distribution has been shown in [65] to be a good model for the situations in which, as the rationality level k grows larger, fewer agents will be at a higher level than k. The second assumption is that level 0 agents choose their strategies randomly according to a uniform distribution. This is commonly known as the overconfidence assumption. The last assumption is that each agent at level $k \geq 1$ knows the true proportions $f(0), f(1), ..., f(k-1)$ of agents at lower rationality levels.

CHT is useful for the IoT because the different rationality levels of CHT can correspond to the different resources available for the IoT devices. Hence, by grouping the devices into the rationality levels based on their available resources, the CHT framework allows each IoT device to find its optimal strategy based on its own, individual computational capability. Note that the CHT model is different from classical hierarchical approaches, such as Stackelberg game and its variants. In such approaches, it is assumed that all agents are fully rational and that the hierarchy is defined based on the roles of the agents (leaders vs. followers) and not in terms of the rationality.

The concept of CHT can be further extended beyond the existing models, such as [65] and [66], to provide a learning scheme for the heterogeneous IoT devices. In this case, the devices at each rationality level can adapt a learning framework that matches their available resources and application. For example, a data aggregator will use ML to compress data, resource-constrained sensors will use finite memory SL with small memory for event detection, and smartphones will used sophisticated RL with relatively large memory for resource management. Moreover, resource-constrained sensors can use simple RL with small memory as they are low rationality, while smartphones would use SL with big memory as they are high rationality. Therefore, the memory size and the computational complexity associated with learning will depend on the rationality level of the device. Applying a hierarchical approach to learning was previously considered in [68], however, this approach cannot capture heterogeneity as it assumed that all the agents use the same learning framework.

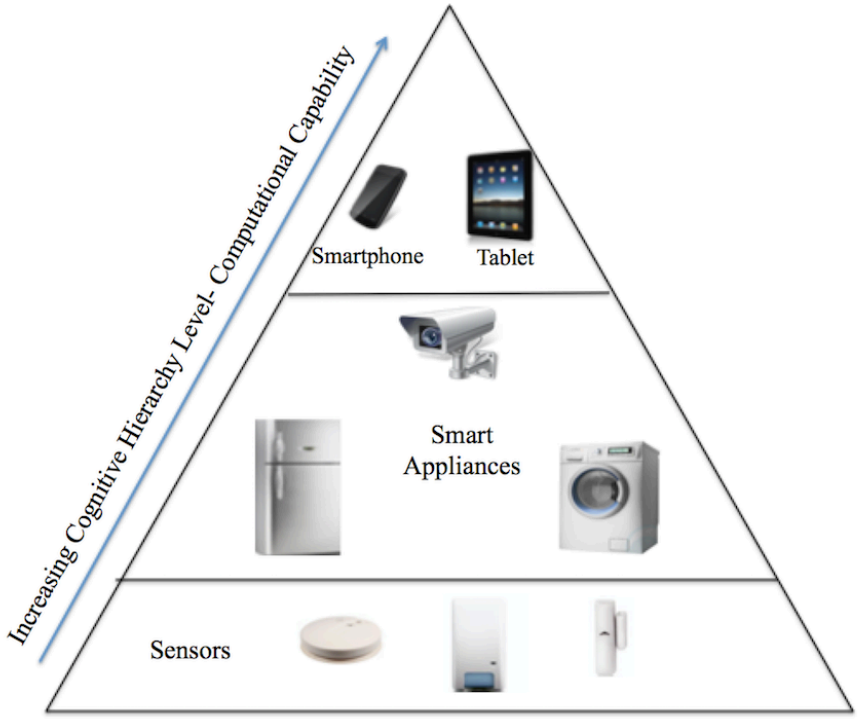


Figure 2.3: Distribution of IoT devices in different CHT levels.

2.4.3 Applications of CHT for the IoT

CHT can be applied to IoT problems that involve distributed optimization at the IoT devices, such as distributed multiple access, resource allocation, and secure transmissions. For example, we have proposed a CHT based approach in [69] to address the problem of distributed uplink multiple access in an IoT network composed of MTDs and HTDs. The MTDs have different characteristics, such as packet sizes, transmission powers, or queue buffer size. In this model, a random access is used for multiple access. In this scheme, a collision occurs if two or more devices transmit simultaneously. Furthermore, the QoS requirement of each device is defined depending on its type. For many IoT devices, such as health care sensors and alarm systems, it is necessary to deliver each packet within a strict deadline while minimizing their energy consumption. HTDs, on the other hand, are more interested in maximizing their transmission rates while keeping their energy consumed within a certain budget. Also due to collisions, the choice of transmission probability of each device affects the performance of all other devices, which motivates to use the game theoretic approaches.

Further, since the IoT devices have different resources available, different rationality levels are used to correspond to the heterogeneous resource availabilities of the IoT devices. A device at rationality level k does more thinking than devices at lower levels and hence should be more computationally capable. Fig. 2.3 shows an example on how some IoT devices can be distributed according to CHT rationality levels.

Here, the Poisson model is chosen as it is suitable to model the distribution of devices in the IoT network. This is because there is a limit on the computational capability of the devices especially in terms of memory and processing power. In fact, devices with higher computational capability are more expensive and hence are fewer in number. The second assumption of the Poisson model related to level-0 devices also holds in the IoT network for resource-constrained devices (e.g., MTDs), whose most feasible strategy is to pick randomly. However, in the approach that we proposed in [69], the overconfidence assumption is modified so that an agent at rationality level k is aware of the

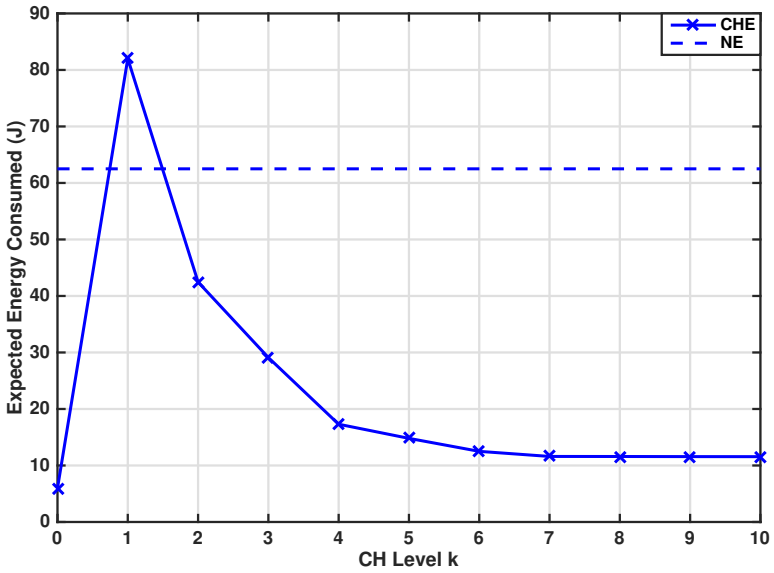


Figure 2.4: Expected energy consumed at the cognitive hierarchy equilibrium (CHE) solution vs. CHT level.

presence of agents at the same rationality level. The overconfidence assumption is suitable for games in which the agents are humans, however, in the studied IoT scenario, the agents are IoT devices that can observe other devices of the same computational capabilities.

Using this CHT model, in [69], we have shown that any device at level k needs to solve a nonlinear, nonconvex optimization problem to find its optimal strategy. Also, the strategy of the device at rationality level k is dependent on the strategies of agents at lower rationality levels. Hence, a device at level k needs to solve k nonlinear optimization problems to find its optimal strategy. This result illustrates the connection between the CHT rationality levels and the computational capabilities of the MTDs as an IoT device at level k needs to do more computations than the devices at lower levels. It is further shown that the proposed CHT approach brings considerable performance improvements. Fig. 2.4 shows that starting from CHT level 1, the energy consumed by an MTD decreases as its CHT level increases and becomes considerably lower than the energy consumed resulting from the classical Nash equilibrium (NE) approach, which shows that the proposed CHT approach provides a fair tradeoff between the achieved performance and the computational capability of the device.

Clearly, CHT is a promising modeling framework within which one can naturally capture the heterogeneity of the IoT and integrate different learning approaches that are directly mapped to the resource limitations of the IoT devices.

2.5 Summary

In this chapter, we have provided a comprehensive overview on the use of a variety of learning techniques within a wide range of applications in the IoT. First, we have discussed the general properties needed for learning techniques to be applied and developed for the purpose of IoT optimization and resource management. Then, we have reviewed the advantages and the limitations of three popular learning frameworks: machine learning, sequential learning, and reinforcement learning. For each such framework, we have identified the inherent properties, such as computational

complexity, memory requirements, and learning performance. Furthermore, we have introduced the fundamentals of cognitive hierarchy theory, which can be used to categorize the IoT devices into different levels of rationality, depending on their capabilities, thus providing a more realistic model of the IoT with heterogeneous devices. For each level of rationality, CHT makes it possible to determine which learning frameworks are most suitable given the requirements for learning and the available resources of the IoT devices in that level of cognitive hierarchy. In summary, this chapter provides a comprehensive reference on developing and applying several classes of learning techniques within well-defined IoT applications.

Chapter 3

Distributed Resource Management using Finite Memory Multi-State Sequential Learning

3.1 Background, Related Works, and Contributions

As discussed in Section 2.3.2, the sequential learning can be used for an effective, distributed resource allocation for the IoT. However, in implementing the sequential learning, many limiting properties of IoT must be carefully considered. The challenges that arise from unique properties of IoT include self-organizing operation, limited communication resources [18], heterogeneous device capabilities and traffic patterns [1], and strict performance requirements [9]. Despite these challenges to overcome, the sequential learning has favorable properties and presents many opportunities as an effective approach for distributed wireless resource management in IoT.

The sequential learning cannot be readily be applied to IoT. Although the sequential learning is flexible with memory requirements, it may still require impracticable amount of information for accurate learning [34]. Such sequential learning is also known as infinite memory sequential learning [34], and it is unfeasible for the IoT devices with limited computational and memory capabilities. Moreover, requiring extensive knowledge will take long time for the IoT devices to learn and react to the dynamic IoT environment. Furthermore, the sequential learning is limited to binary state learning, which greatly limits applicability of sequential learning in the IoT [34, 51]. To this end, it is necessary to design a sequential learning that overcomes its own limitations and is tailored for the IoT and its devices.

The main contribution of this chapter is a *multi-state sequential learning framework* [34] with finite memory that can enable the IoT devices to allocate their limited communication resources in a self-organizing manner while satisfying their heterogeneous latency requirements. In particular, we consider a massive IoT system in which there are IoT devices with limited capabilities transmitting either periodic or critical messages. *Periodic messages*, such as meter readings, are delay-tolerant messages with predictable traffic pattern, while *critical messages*, such as system failures, are randomly occurring, delay-intolerant messages. When some devices transmit critical messages, the communication resources that are normally allocated for the periodic messages must be reallocated for the critical messages for their timely transmission. We propose a novel multi-state sequential learning algorithm that allows the IoT devices to learn the number of existing critical messages and, then, to reallocate communication resources appropriately depending on the delay requirement and the number of critical messages. The proposed multi-state sequential learning is suitable for IoT devices as it only requires finite number of observations to learn in a distributed way. Then, we analyze the

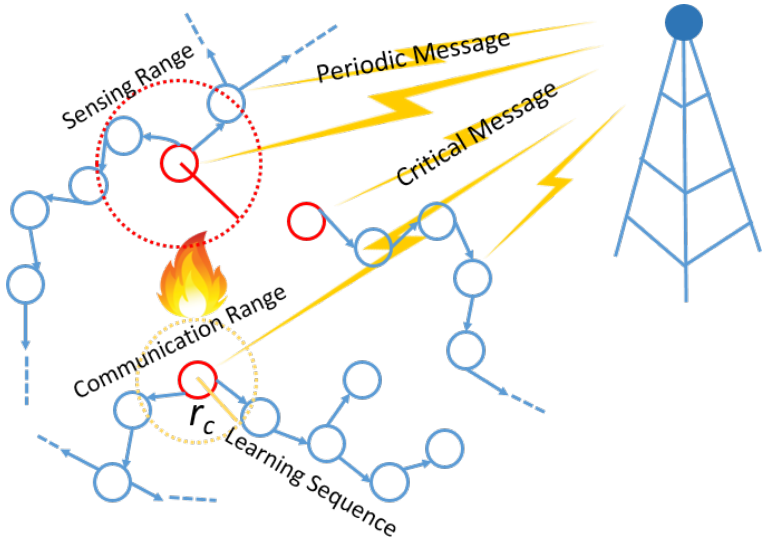


Figure 3.1: An overview of the problem setup showing communication range r_c, sensing range, learning sequence, and transmissions of critical and periodic messages from IoT device to BS.

memory, limited observation capabilities of devices, and expected delay resulting from our proposed algorithm. We show that our proposed learning framework converges to the true state of the environment (i.e., it autonomously learns the true number of existing critical messages). We also derive the lowest expected delay of a critical message that can be achieved with our learning framework. Simulation results show the learning effectiveness of the proposed framework as well as its ability to realize low-latency transmissions across a massive IoT network.

The rest of this chapter is organized as follows. Section 3.2 introduces the system model and Section 3.3 presents the proposed learning algorithm and analyzes its properties. Section 3.4 analyzes the simulation results, while Section 3.5 draws conclusions.

3.2 System Model

Consider the uplink of a wireless IoT system consisting of one BS serving N IoT devices. To transmit their messages to the BS, the IoT devices must first request uplink communication resources via a time-slotted random access channel (RACH) using one out of p random access (RA) preambles [70] that are typically divided into two types: p_c contention-based RA preambles (RAPs) and $p_f = p - p_c$ contention-free RAPs [70]. For using a contention-based RAP, the IoT devices choose one out of the p_c preambles randomly. However, if more than one IoT device choose a given contention-based RAP, there will be a collision resulting in a failure to request uplink resources. Therefore, a device will successfully be allocated uplink resources using contention-based RAP, only if it is the sole device using that preamble at that given time slot. For using contention-free RAPs, the devices are assigned one of the p_f preambles by the BS. The preamble usage and allocation among the IoT devices will depend on their message types as discussed later in this section.

One of the prominent features of the IoT is heterogeneity in terms of device types, functionalities, and transmitted messages as illustrated in Fig. 3.1. For instance, some IoT devices, such as smart meters and environment sensors, will be periodically transmitting small packets, such as meter readings, observation reports, and system status reports. However, such devices may need to also transmit urgent, critical messages, such as critical system state condition,

Table 3.1: Summary of notations.

N	Number of IoT devices
p	Number of RAPs
p_c, p'_c	Number of contention-based RAPs before and after reallocation
p_f, p'_f	Number of contention-free RAPs before and after reallocation
T	Period for periodic message transmissions
τ_i	Time slot for first periodic message transmission
T_{min}	Minimum feasible period for periodic messages
N_p	Expected number of IoT devices transmitting periodic message at any given time
$p_{p,s}$	Probability of an IoT device successfully transmitting periodic message using contention-based RAP
N_a	Number of IoT devices transmitting critical message
$p_{a,s}$	Probability of an IoT device successfully transmitting critical message using contention-based RAP
D_c, D'_c	Expected delay of critical message under contention-based RAP before and after reallocation
D_f, D'_f	Expected delay of critical message under contention-free RAP before and after reallocation
D, D'	Expected delay of a critical message
r_d, r'_d	Detection range for IoT devices before and after learning
p_{01}	Probability of missed detection within detection range
p'_{01}	Probability of missed detection outside of detection range
β	Number of contention-free RAPs that are reallocated
D_{th}	Required delay threshold for critical messages
r_c	Communication range for IoT devices
H_T	True underlying state
s_f	Favored state chosen during first phase
s'_f	Any other state that is not favored state
K	Number of observations necessary for first phase
α	Number of consecutive observations necessary to go back to first phase
m	Number of bits necessary to learn in second phase
x_i	Private belief of IoT device i based on observations

power outage, and fire detection reports. These message types differ in terms of quality-of-service (QoS) requirements and traffic patterns. To model heterogeneous messages, we consider the co-existence of *periodic messages*, which are infrequent, periodic transmissions, and *critical messages*, which are critical, urgent transmissions that require very small delay. Depending on the message type, the IoT devices will transmit using most appropriate RAP type.

3.2.1 Periodic Messages

Periodic messages are non-critical and recurring, and, thus, they are typically delay-tolerant with predictable traffic patterns [9]. Hence, the IoT devices transmitting periodic messages will request uplink resources using an RAP once every period T. We let τ_i for $i = 1, \cdots, T$ be the possible time slots when the IoT devices first transmit their periodic messages. The devices that first transmit in slot τ_i will transmit in slots $\tau_i + kT$ for $k \in \mathbb{Z}_+$. Since the IoT devices that have the same τ_i will always transmit simultaneously, it is sufficient to only consider RAP allocation among the IoT devices with same τ_i.

Since it is predictable when and which IoT devices will transmit the periodic messages, contention-free RAPs are more suitable for periodic message transmissions than contention-based RAPs as the contention-free RAPs can guarantee successful uplink resource allocation request for appropriate values of T. However, for given values of N and p_f, it

may be impossible to satisfy T, and the minimum period T_{min} of the periodic messages that can be supported will be:

$$T_{\text{min}} = \left\lceil \frac{N}{p_f} \right\rceil . \tag{3.1}$$

(3.1) results from the fact that at most p_f devices can use contention-free RAPs without collision in a time slot, and an IoT device must wait until all other IoT devices transmit their periodic messages to satisfy T of all periodic messages. If the required period of the periodic messages is less than T_{min}, there will be more than p_f IoT devices transmitting simultaneously in some time slots and collisions will occur. For $T \geq T_{\text{min}}$, the BS can determine T groups of IoT devices each group with at most p_f IoT devices with same τ_i. Moreover, the BS will allocate different contention-free RAPs to IoT devices with same τ_i. This allocation scheme helps avoiding RAP collision, while maximizing the number of IoT devices that can request uplink resources. The use of contention-free RAPs takes advantage of the predictable traffic patterns of the periodic messages and guarantees that the IoT devices successfully request their uplink resources.

Note that contention-based RAPs are not suitable for the periodic messages, because the probability of successful transmission is low, particularly for a massive IoT with large N. Assuming that τ_i are chosen uniformly randomly, if the IoT devices transmitting periodic messages use contention-based RAPs, the expected number N_p of IoT devices transmitting periodic messages at any given time slot will be $N_p = \frac{N}{T}$. The probability $p_{p,s}$ of an IoT device successfully transmitting using the contention-based RAP in a given time slot will then be:

$$p_{p,s} = \left(\frac{p_c - 1}{p_c} \right)^{N_p - 1} , \tag{3.2}$$

which represents the probability of all other $N_p - 1$ devices choosing some other contention-based RAP with probability $\frac{p_c - 1}{p_c}$.

Since the IoT will be massive (i.e., N is large), then, $p_{p,s}$ is small, and, thus, only few, if any, periodic messages will be successfully transmitted in a given time slot. For periodic messages with $T \geq T_{\text{min}}$, the use of contention-free RAPs guarantees that the IoT devices successfully request uplink resources, while the probability of successful request is low using contention-based RAPs. Therefore, it is more advantageous for the IoT devices transmitting periodic messages to use contention-free RAPs. Hereinafter, we assume that the period T of periodic messages is T_{min} and that N is a multiple of p_f for simplicity. This implies that all p_f contention-free RAPs are used in every time slot.

3.2.2 Critical Messages

Critical IoT device messages contain urgent information about an abnormal event and, thus, they must reach the BS with low latency. Therefore, critical messages are delay-intolerant messages that need retransmissions in case of RAP collision. Moreover, since critical messages are triggered by an unpredictable abnormal event, such as system failure or forest fire, the system cannot know beforehand when and which IoT devices will have critical messages. Furthermore, critical messages can be highly correlated since an abnormal event will typically trigger some critical messages simultaneously. For instance, a forest fire will trigger various devices monitoring different environmental parameters, such as temperature, humidity, or carbon monoxide. Since the network cannot predict when the critical messages will be triggered, how many will be triggered, and which IoT devices will have critical messages, the BS cannot assign pre-determined RAPs, and, thus, IoT devices with critical messages must use contention-based RAPs to request uplink transmission resources.

We let N_a be the number of IoT devices having critical messages to send. For IoT devices with critical messages using p_c contention-based RAPs, the probability $p_{a,s}$ of a device being successfully allocated an uplink resource for sending a critical message will be:

$$p_{a,s} = \left(\frac{p_c - 1}{p_c} \right)^{N_a - 1} , \tag{3.3}$$

which is probability of all other $N_a - 1$ devices choosing some other contention-based RAP with probability $\frac{p_c - 1}{p_c}$. Given $p_{a,s}$, the expected delay D_c of a critical message under a contention-based RAP is defined as the expected number of time slots needed until the first successful acquisition of an uplink communication resource using contention-based RAP, as follows:

$$D_c = p_{a,s}^{-1} = \left(\frac{p_c}{p_c - 1} \right)^{N_a - 1}, \tag{3.4}$$

since D_c is a mean of geometric distribution with probability of success of $p_{a,s}$.

While using contention-based RAPs, an IoT device having a critical message may be assigned with contention-free RAP for its next periodic message transmission. In this case, instead of using the assigned contention-free RAP for the periodic message, IoT devices with critical messages will use the assigned contention-free RAP for sending their critical messages. Since $T = T_{\text{min}}$ and τ_i are chosen uniformly randomly for all devices, the expected delay D_f of a critical message under a contention-free RAP is defined as the expected number of time slots needed until an IoT device with critical message is allocated with contention-free RAP, as follows:

$$D_f = \frac{1}{T_{\text{min}}} \sum_{j=0}^{T_{\text{min}}-1} j = \frac{T_{\text{min}} - 1}{2}. \tag{3.5}$$

Since an IoT device having a critical message just needs one successful transmission using either RAP, the expected delay of a critical message is $D = \min(D_c, D_f)$.

To minimize the expected delay D of the critical messages, D_c or D_f must be minimized. However, decreasing the value of one of them will directly increase the value of the other as D_c and D_f are related with $p_c + p_f = p$. In a massive IoT network with $N \gg p$, T_{min} in (3.1) will be large even if $p_f = p$, and, thus, D_f will also be large. Since the number of critical messages N_a will typically be small, it is more effective to minimize D_c to minimize D. Since the value of N_a cannot be controlled, the value of p_c must be maximized to minimize D_c. Since the number of RAPs p is fixed, the only way to increase the value of p_c is to reallocate RAPs from contention-free to contention-based. The method used for such reallocation cannot be centralized since a centralized method requires the BS to have already received the critical messages and to be aware of the abnormal event, which is impractical. Therefore, RAP reallocation must be done in a distributed manner.

To reallocate an appropriate number of contention-free RAPs, the IoT devices must be able to observe the number of critical messages N_a in the system. We assume that the overhead signaling for the periodic messages and critical messages are different such that the IoT devices can observe the number of critical messages N_a being transmitted in a given time slot [71]. However, the IoT devices may not be able to accurately observe N_a, since they cannot observe all of the critical messages. For instance, an IoT device that is located far from the abnormal event may not be able to observe the event and its corresponding critical messages. IoT devices may also be prone to missed detection even when they are in proximity to the abnormal event. To capture the limited observation capability, we assume that IoT devices within *a detection range* r_d have a probability of missed detection p_{01}, while IoT devices outside of the detection range r_d have a much higher probability of missed detection, p'_{01}.

In order for the IoT devices to accurately know N_a despite their limited observation capability, they can employ a distributed learning process [51] using which the devices can collectively learn the number of critical messages N_a in a self-organizing manner. For instance, IoT devices with limited observation capabilities can use the observations of other IoT devices to have a more accurate estimate of N_a. For the reallocation of RAPs, we assume that there is *a globally known order of contention-free RAPs that will be reallocated to contention-based RAPs.* When an IoT device having a periodic message learns that there are N_a critical messages, it will stop using the first β contention-free RAPs of the order. This means that even if the IoT device with periodic message is assigned to use one of the β RAPs by the BS, it will not transmit. When an IoT device having a critical message learns that there are N_a critical messages, it will use one of the first β contention-free RAPs of the order or one of the contention-based RAPs. Therefore, the number of contention-based RAPs will increase from p_c to $p'_c = p_c + \beta$, and the number of contention-free RAPs will decrease from p_f to $p'_f = p_f - \beta$. The expected delay D'_c from using contention-based RAP, the expected delay D'_f

from using contention-free RAP, and the expected delay D' of critical message *after learning* will, respectively, be given by:

$$D' = \min(D'_c, D'_f) = \min\left(\left(\frac{p'_c}{p'_c - 1}\right)^{N_a - 1}, \frac{\left\lceil \frac{N}{p'_f} \right\rceil - 1}{2}\right). \tag{3.6}$$

For a given design parameter threshold delay D_{th} that the critical messages should satisfy, the value of β is determined to satisfy $D' \leq D_{\text{th}}$. Moreover, the value of β should be appropriate as an excessive value of β will greatly disturb the transmission of periodic messages.

To develop an effective learning approach, one must take into account specific characteristics of the IoT devices. As the IoT devices are small, low cost devices, their computational capability will be limited. Moreover, they are constrained by very limited battery life, and, thus, they cannot communicate frequently with the BS or with other IoT devices. Further, the IoT devices may also be limited in terms of how much information they can store and process. Therefore, they may not have sufficient information for accurate learning, and their learning method must account for the lack of memory. Also, the IoT devices may have inaccurate information about N_a due to their limited observability of the environment. For applications that require low latency, the expected delay of critical messages must be reduced in relatively real time, and, thus, the IoT devices must quickly and accurately learn N_a. Therefore, the learning scheme used by the IoT devices must be computationally simple, distributed, quick, and accurate without requiring excessive memory and frequent communication. To this end, one can use the framework of *finite memory sequential learning* [34]. Even though this framework has been used before in decentralized, binary decision making, this cannot be readily applied to the IoT as it is only limited to binary state learning. Moreover, the resource constraints of IoT devices in terms of computation and memory, the limited communication resources, and the unique properties of IoT system must be considered. Hence, next we develop a new sequential learning framework tailored to the studied IoT system.

3.3 Finite Memory Multi-state Sequential Learning

For our model, the IoT devices with their inherent limited memory and computation capabilities must learn to satisfy their heterogeneous QoS requirements, in presence of both known periodic messages and unknown critical messages. To this end, we propose a novel multi-state sequential learning framework for enabling the IoT devices to learn the true value of N_a to determine the number of RAPs to reallocate β to satisfy the delay threshold D_{th} of critical messages. The IoT devices learning the true value of N_a can be mapped to agents learning the true state of a system with multiple states. For instance, the different values of N_a that the IoT devices can learn can be mapped to different system states, while the true value of N_a is the true, underlying state that the IoT devices should learn. By mapping different values of N_a to different states, we propose *a multi-state learning framework for dynamic reallocation of RAPs*, while taking into account the requirements of the IoT devices. Furthermore, our learning method will require an IoT device to only communicate with one neighboring device within the communication range r_c to obtain a finite memory of observations of other devices as illustrated in Fig. 3.1.

As shown in Fig. 3.2, we propose a learning algorithm composed of two phases, which are used to choose a state and to test whether the chosen state is the true underlying state H_T. A state is chosen out of all possible states during the first phase, and the chosen state is called a *favored state*. During the second phase, it is repeatedly tested whether the favored state should be changed by going back to the first phase. In our model, the IoT devices learn the number of critical messages N_a based on their observations and the observations of neighboring devices during the first phase. During the second phase, the IoT devices will be informed about the favored state N_a learned during the first phase and will repeatedly test whether the learned value of N_a is a true value or not. If it is determined that the learned value is not the true value of N_a, then the learning method reverts back to the first phase. Moreover, an IoT device belongs to either first phase or second phase and only learns once to prevent the learning sequence from oscillating.

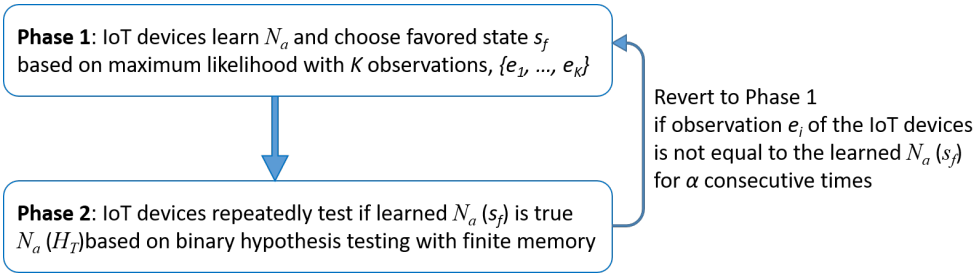


Figure 3.2: Flowchart of the proposed finite memory multi-state learning.

The sequential learning is implemented *individually by all devices*, and the IoT devices will learn N_a as they get necessary information. In other words, the IoT devices do not converge to the final decision on N_a simultaneously, and the learned values of N_a may be incorrect with limited observability. As learning propagates sequentially, the IoT devices adjust their RAP usage as soon as their learn N_a, and, thus, the reallocation of β contention-free RAPs is done gradually. Moreover, β depends on the learned value of N_a and the given value of design parameter D_{th} as further discussed in Subsection 3.3.4.

For both phases, the observations are critical in learning N_a correctly, and more observations typically result in higher probability of learning correctly. However, it is unrealistic to assume that an IoT device is able to know, store, and process observations of all other IoT devices, which corresponds to *infinite memory* sequential learning in [34]. Therefore, our proposed learning framework has *finite memory* [51] as it assumes that an IoT device only utilizes some of the observations of other IoT devices. In other words, the finite memory learning requires finite number of observations, while the infinite memory learning requires infinite number of observations. The convergence of our finite memory learning framework is shown in Subsection 3.3.3.

3.3.1 First Phase of Sequential Learning

For an appropriate reallocation of contention-free RAPs, the IoT devices must learn the true value of the number of critical messages N_a. However, the devices may learn wrong values of N_a due to limited observability. We let $\mathcal{S}$ be the set of all possible values of the state N_a. Here, the true value of N_a is the true underlying state $H_T \in \mathcal{S}$. During the first phase, the IoT devices chose a favored state $s_f \in \mathcal{S}$ after K observations, which provides information about H_T. s_f is a value of N_a that the devices learn to be true, and $k \ll N$ as we assume that the IoT devices are limited in terms of memory and computational capability. In our model, the IoT devices observe how many critical messages exist depending on their limited observation range r_d. For simplicity, we model an observation of the environment e to be an element of $\mathcal{S}$ such that, for any given $H_T \in \mathcal{S}$,

$$0 < \frac{\Pr(e = j \mid H_T)}{\Pr(e = k \mid H_T)} < \infty \; \forall j, k \in \mathcal{S}, \tag{3.7}$$

$$\Pr(e = H_T \mid H_T) \gg \Pr(e \neq H_T \mid H_T),$$

$$\Pr(e = i \mid i) \gg \Pr(e = i \mid j) \; \forall i, j \in \mathcal{S}, \tag{3.8}$$

$$\Pr(e = j \mid i) \approx \Pr(e = k \mid i) \; \forall i, j, k \in \mathcal{S}, \tag{3.9}$$

where $\Pr(e = i|j)$ is probability of observing i when j is the true underlying state. (3.7) is necessary to ensure that the learning process is not trivial by preventing the likelihood ratio of observations from being 0 or ∞. In (3.8), we assume that a given device is more likely to observe the true underlying state H_T than any other state in $\mathcal{S}$ and that it is most likely to observe $i \in \mathcal{S}$ when the true underlying state is i. Furthermore, in (3.9), the probabilities of observing the states that are not true underlying state are approximately equal. We assume that the environment observations, which correspond to the observed numbers of critical messages, are independent, because the observability of an IoT

device depends only on the observation range r_d.

The first phase is initiated by an abnormal event, and the devices that are first to observe the abnormal event will have to transmit critical messages. Furthermore, as illustrated in Fig. 3.1, those devices propagate the learning sequence to their neighboring devices within r_c by transmitting their observations of the environment. Those neighboring devices constitute the next set of devices that learn in the first phase, and these devices learn N_a based on their observations and the information from the device that learned first. Therefore, the first phase propagates sequentially to the various IoT devices within r_c. When a device obtains K observations, it determines s_f based on K observations and initiates the second phase. If an IoT device that has not yet learned receives the necessary information for learning from more than one neighboring device simultaneously, then it will arbitrarily choose one of the learning sequences to continue.

We assume that $r_c < r_d$ as the communication links between IoT devices are usually short-ranged. Moreover, it is desirable for K to be small so that s_f is chosen quickly and that the learning method converges faster to H_T. For small K, the memory and computational requirements for the IoT devices will also be small due to the reduced amount of information that must be stored and processed, and, thus, the critical messages will be successfully transmitted with small delay as the learning converges faster to H_T. Hence, the IoT devices during the first phase will most likely be within r_d. Since the probability of missed detection p_{01} within r_d is usually close to 0 [72], the IoT devices participating in the first phase are most likely to observe $e = H_T$. Therefore, (3.8) holds true for the IoT devices within r_d, where the first phase always occurs.

While the IoT devices accumulate K independent observations of the environment, they can make an intelligent decision on which state is most likely to be the true underlying state H_T (i.e., the true value of N_a). This decision will be based on *maximum likelihood*, whose computational complexity can be simplified given (3.8). For a set of K observations $\{e_1, \cdots, e_K\}$, the favored state s_f will be:

$$s_f = \underset{s \in \mathcal{S}}{\operatorname{argmax}} \Pr(e_1, \cdots, e_K \mid s) = \underset{s \in \mathcal{S}}{\operatorname{argmax}} \prod_{j=1}^{K} \Pr(e_j \mid s), \tag{3.10}$$

$$= \underset{s \in \mathcal{S}}{\operatorname{argmax}} \prod_{j \in \mathcal{S}} \Pr(j \mid s)^{k_j} = \underset{s \in \mathcal{S}}{\operatorname{argmax}} \sum_{j \in \mathcal{S}} k_j \ln(\Pr(j \mid s)), \tag{3.11}$$

where k_j is the number of times that state $j \in \mathcal{S}$ is observed out of K observations. If the likelihoods $\Pr(e = i \mid j)\ \forall i, j \in \mathcal{S}$ are known by the devices, then s_f can be determined easily using (3.11). However, when the likelihoods $\Pr(e = i \mid j)\ \forall i, j \in \mathcal{S}$ are unknown, the maximum likelihood can be simplified using (3.8) and (3.9) and, then, the favored state s_f will be $a \in \mathcal{S}$ if:

$$\frac{\sum_{j \in \mathcal{S}} k_j \ln(\Pr(j \mid a))}{\sum_{j \in \mathcal{S}} k_j \ln(\Pr(j \mid b))} \approx \frac{\sum_{j \in \mathcal{S}, j \neq a} k_j}{\sum_{j \in \mathcal{S}, j \neq b} k_j} = \frac{K - k_a}{K - k_b} \leq 1\ \forall\, b \in \mathcal{S}. \tag{3.12}$$

Therefore, with simplifications, the decision process in the first phase will be such that the devices will be able to designate one of the observed states as a favored state s_f.

3.3.2 Second Phase of Sequential Learning

Once the IoT devices learn s_f to be the true value of N_a after K observations, it must be repeatedly tested if s_f is indeed H_T, given the possibility of learning error due to limited observability. If it is found that $s_f \neq H_T$, then s_f must be changed to some other state in $\mathcal{S}' = \mathcal{S} \backslash \{s_f\}$. Since the second phase tests whether s_f must be changed or not, this can be modeled as binary hypothesis testing between $s_f = H_T$ and $s = H_T$ for some state $s \neq s_f$, $s \in \mathcal{S}'$, and the IoT devices in second phase each test with their available information. This binary hypothesis testing uses binary environmental observations to test whether the favored hypothesis should be changed, and the favored hypothesis will converge to the underlying truth by design.

As our learning method is multi-state, the devices will observe more than two states. However, if a device observes any state in $\mathcal{S}'$, it will be equivalent to observing an unfavored state s'_f. Therefore, the number of possible observations reduces to two: s_f and s'_f, during the second phase. Here, observing s_f is equivalent to 1 and observing s'_f is equivalent to 0. Similar to the observations in first phase, an environment observation e in the second phase is such that, for an underlying true state $H_T \in \mathcal{S}$,

$$\Pr(e = s_f \mid H_T) + \Pr(e \in \mathcal{S}' \mid H_T) = 1 \tag{3.13}$$

$$0 < \frac{\Pr(e = s_f \mid H_T)}{\Pr(e = s'_f \mid H_T)} < \infty. \tag{3.14}$$

If $s_f \neq H_T$ is chosen during the first phase and the environment observations repeatedly indicate s'_f, then the favored state must be changed. Therefore, a mechanism for returning to the first phase to adjust the choice of s_f is necessary. We set a threshold α such that if the environment observations indicate s'_f for a total of α consecutive times, then the learning method will go back to the first phase to choose s_f again with some probability p_b, which will be further discussed later. Using a threshold, our learning method may go back to the first phase both when $s_f = H_T$ and when $s_f \neq H_T$, and the probabilities of returning to the first phase for both cases are critical for the convergence of the proposed learning method.

The probability of observing s'_f for α consecutive times when $s_f \neq H_T$ is

$$\Pr(\{e_{n_1-\alpha+1}, \cdots, e_{n_1}\} = \{s'_f, \cdots, s'_f\} \forall n_1 \geq \alpha \mid s_f \neq H_T) = (1 - \Pr(e = s_f \mid s_f \neq H_T))^\alpha, \tag{3.15}$$

where e_j is j-th observation of the environment and $n_1 \in \mathbb{Z}_+$ is the number of observations necessary for α consecutive observations of s'_f to occur when $s_f \neq H_T$. n_1 is a random variable, and it is essential that the learning method quickly reverts to the first phase if $s_f \neq H_T$. Therefore, the expected value $\mathbb{E}[n_1 \mid s_f \neq H_T]$ of n_1 when $s_f \neq H_T$ is of high interest. If s_f is observed after s'_f is observed $j < \alpha$ consecutive times, then the number of consecutive observations of s'_f is reset to 0, and this is equivalent to increasing $\mathbb{E}[n_1 \mid s_f \neq H_T]$ by $j + 1$. With $p_1 = \Pr(e = s_f \mid s_f \neq H_T)$, the aforementioned case occurs with probability $p_1(1 - p_1)^j$. Therefore, the expected value of n_1, for any environment observation e, will be:

$$\mathbb{E}[n_1 \mid s_f \neq H_T] \tag{3.16}$$

$$= \sum_{j=0}^{\alpha-1} p_1(1-p_1)^j \left(\mathbb{E}[n_1 \mid s_f \neq H_T] + j + 1\right) + \alpha(1-p_1)^\alpha, \tag{3.17}$$

$$= \mathbb{E}[n_1 \mid s_f \neq H_T] p_1 \left(\sum_{j=0}^{\alpha-1} (1-p_1)^j\right) + p_1 \left(\sum_{j=0}^{\alpha-1} (1-p_1)^j (j+1)\right) + \alpha(1-p_1)^\alpha, \tag{3.18}$$

$$= \mathbb{E}[n_1 \mid s_f \neq H_T] p_1 \frac{1-(1-p_1)^\alpha}{1-(1-p_1)} + p_1 \left(\sum_{j=0}^{\alpha-1} (1-p_1)^j (j+1)\right) + \alpha(1-p_1)^\alpha, \tag{3.19}$$

$$= \frac{p_1 \left(\sum_{j=0}^{\alpha-1} (1-p_1)^j (j+1)\right) + \alpha(1-p_1)^\alpha}{(1-p_1)^\alpha}. \tag{3.20}$$

Similarly, the probability of observing s'_f for α consecutive times when $s_f = H_T$ is

$$\Pr(\{e_{n_1-\alpha+1}, \cdots, e_{n_1}\} = \{s'_f, \cdots, s'_f\} \forall n_2 \geq \alpha \mid s_f = H_T) = (1 - \Pr(e = s_f \mid s_f = H_T))^\alpha, \tag{3.21}$$

where $n_2 \in \mathbb{Z}_+$ is the number of observations necessary for α consecutive observations of s'_f to occur when $s_f = H_T$. Similar to n_1, n_2 is a random variable, but it must be highly unlikely that the learning method goes back to the first

phase if $s_f = H_T$. In other words, the expected value $\mathbb{E}[n_2 \mid s_f = H_T]$ should be very large. If s_f is observed after s'_f is observed $j < \alpha$ consecutive times, then the number of consecutive observations of s'_f is reset to 0, and this is equivalent to increasing $\mathbb{E}[n_2 \mid s_f = H_T]$ by $j + 1$. With $p_2 = \Pr(e = s_f \mid s_f = H_T)$, the aforementioned case occurs with probability $p_2(1-p_2)^j$. Therefore, the expected value of n_2, for any environment observation e, will be:

$$\mathbb{E}[n_2 \mid s_f = H_T] \tag{3.22}$$

$$= \sum_{j=0}^{\alpha-1} p_2(1-p_2)^j \left(\mathbb{E}[n_2 \mid s_f = H_T] + j + 1\right) + \alpha(1-p_2)^\alpha, \tag{3.23}$$

$$= \mathbb{E}[n_2 \mid s_f = H_T] p_2 \left(\sum_{j=0}^{\alpha-1} (1-p_2)^j\right) + p_2 \left(\sum_{j=0}^{\alpha-1} (1-p_2)^j (j+1)\right) + \alpha(1-p_2)^\alpha, \tag{3.24}$$

$$= \mathbb{E}[n_2 \mid s_f = H_T] p_2 \frac{1-(1-p_2)^\alpha}{1-(1-p_2)} + p_2 \left(\sum_{j=0}^{\alpha-1} (1-p_2)^j (j+1)\right) + \alpha(1-p_2)^\alpha, \tag{3.25}$$

$$= \frac{p_2 \left(\sum_{j=0}^{\alpha-1} (1-p_2)^j (j+1)\right) + \alpha(1-p_2)^\alpha}{(1-p_2)^\alpha}. \tag{3.26}$$

The threshold α can be seen as a design parameter to adjust the expected number of necessary iterations in (3.20) and (3.26). Moreover, there are advantages and disadvantages of choosing a high value and a low value for α. Given (3.8), a high value of α will imply that it is highly unlikely to incorrectly go back to the first phase. However, when choosing $s_f \neq H_T$, it will take many observations to go back to the first phase, and, thus, the learning method converges slowly to $s_f = H_T$. A low value of α will imply that the learning method can quickly go back to the first phase to choose s_f again if $s_f \neq H_T$. Meanwhile, a small α also implies that it is more likely to incorrectly go back to the first phase even when $s_f = H_T$.

Similar to the first phase, the IoT devices in the second phase will individually learn after receiving m bits of information from any neighboring IoT device that learned and, then, they will transmit m bits of updated information to neighboring IoT devices. The *memory* that IoT device i needs to learn N_a is essentially the size of the received information set $\{e_{i-m+2}, \cdots, e_{i-1}, F_{i-1}, Q_{i-1}\}$, where e_{i-1} is the environment observation of IoT device $i-1$, F_{i-1} is the currently favored hypothesis, and Q_{i-1} indicates whether F_{i-1} should be changed. The value of m must be at least two bits to capture $\{F_{i-1}, Q_{i-1}\}$, which are necessary for learning [34], and the second phase reduces to the finite memory sequential learning introduced in [34] if $m = 2$. However, the finite memory sequential learning in [34] assumes that (3.8) holds true for all environment observations, which is not true for the IoT devices with limited observability. Therefore, the memory of previous observations is used to mitigate the limited observability.

For $m > 2$, an IoT device i computes its *private belief* x_i based on maximum likelihood using the observations of previous devices in learning sequence $\{e_{i-m+2}, \cdots, e_{i-1}\}$ and its own observation e_i. The private belief x_i of IoT device i is:

$$x_i = \underset{s \in \{s_f, s'_f\}}{\operatorname{argmax}} \Pr(e_{i-m+2}, \cdots, e_i \mid s) = \underset{s \in \{s_f, s'_f\}}{\operatorname{argmax}} \prod_{j=i-m+2}^{i} \Pr(e_j \mid s), \tag{3.27}$$

$$= \underset{s \in \{s_f, s'_f\}}{\operatorname{argmax}} \Pr(s_f \mid s)^{k_{s_f}} \Pr(s'_f \mid s)^{k_{s'_f}} = \underset{s \in \{s_f, s'_f\}}{\operatorname{argmax}} k_{s_f} \ln(\Pr(s_f \mid s)) + k_{s'_f} \ln(\Pr(s'_f \mid s)), \tag{3.28}$$

where k_{s_f} is the number of times that state s_f is observed and $k_{s'_f}$ is the number of times that state s'_f is observed out of $m-1$ observations. If the likelihoods $\Pr(e = i|j)\ \forall i, j \in \{s_f, s'_f\}$ are known by the devices, then x_i can be determined easily using (3.28). However, when the likelihoods are unknown, the maximum likelihood can be

simplified using (3.8) and (3.9), and the private signal is:

$$x_i = \begin{cases} s_f & \text{if } \frac{k_{s_f}\ln(\Pr(s_f|s_f))+k_{s'_f}\ln(\Pr(s'_f|s_f))}{k_{s_f}\ln(\Pr(s_f|s'_f))+k_{s'_f}\ln(\Pr(s'_f|s'_f))} \leq 1, \\ s'_f & \text{otherwise,} \end{cases} = \begin{cases} s_f & \text{if } k_{s_f} \geq 1, \\ s'_f & \text{otherwise.} \end{cases} \tag{3.29}$$

Therefore, the private belief is s_f if any of the environment observations is s_f.

When $s_f \neq H_T$, the k_{s_f} will most likely be 0, because the observations from both within r_d and outside of r_d will highly unlikely be s_f. However, when $s_f = H_T$, k_{s_f} will most likely be greater than 0 if any of the observations is from within r_d. This is because the observations outside of r_d is subject to high probability of missed detection, and, thus, the (3.8) does not hold. As the oldest observation e_{i-m+2} is replaced by the newest observation e_i and there are $m-2$ previous observations, all of the observations are from outside of r_d after the learning sequence has progressed $m-2$ times outside of r_d. Therefore, assuming that the network is dense and most IoT devices have at least one neighboring IoT device, the furthest IoT device from the abnormal event with an observation from within r_d is located $r_d + (m-2)r_c$. Therefore, by using x_i, k_{s_f} of the IoT devices within $r_d + (m-2)r_c$ will most likely be greater than 0 and have $x_i = s_f$ when $s_f = H_T$. In other words, (3.8) holds for more IoT devices, and more IoT devices will learn correctly. Moreover, the use of x_i instead of e_i is very effective in the second phase, because the learning progresses sequentially away from the abnormal event and the IoT devices participating in the second phase will most likely be outside of r_d. The effective detection radius r'_d within which (3.8) holds can therefore be defined as follows:

$$r'_d = r_d + (m-2)r_c, \tag{3.30}$$

where m is the memory size of a device (in bits). For a larger memory size m, r'_d is bigger, and, thus, (3.8) holds for more IoT devices. However, as the deployment region is finite, it is unnecessary to have very large values of m, which can render r'_d much greater than the dimensions of the deployment region. Moreover, for larger m, more energy is required to transmit, which may be impractical for IoT devices with energy constraints. Therefore, it is necessary to choose an appropriate value of m depending on network dimensions, energy constraints, and delay requirements.

An IoT device i in the second phase learns using x_i, F_{i-1}, and Q_{i-1} by repeatedly testing whether the currently favored hypothesis $F_{i-1} \in \{0, 1\}$ should be changed. Moreover, a number of consecutive devices in the learning sequence need to collectively test and decide whether F_{i-1} should be changed. Here, we define the notion of an *S-block* as a set consecutive devices testing whether F_{i-1} should be changed from 0 to 1 and an *R-block* as a set of consecutive devices testing whether F_{i-1} should be changed from 1 to 0. Every device in a learning sequence belongs to either an S-block or an R-block, and both blocks are alternating in a learning sequence to repeatedly test F_{i-1}. As shown in [34], the lengths of the S and R-blocks must be chosen to ensure that F_{i-1} will converge to true hypothesis.

Since observing s_f is equivalent to 1 and observing s'_f is equivalent to 0, the IoT devices belonging to an S-block decide that F_{i-1} should be changed from s'_f to s_f if all devices in that S-block observe s_f. Similarly, the IoT devices belonging to an R-block decide that F_{i-1} should be changed from s_f to s'_f if all devices in that R-block observe s'_f. Q_{i-1} is used to track if all of the observations are s_f in S-block or s'_f in R-block, and F_{i-1} is changed accordingly by the IoT devices. After learning, IoT device i updates m bits of information by replacing $\{F_{i-1}, Q_{i-1}\}$ with $\{F_i, Q_i\}$ and the oldest observation e_{i-m+2} with its own observation e_i. Moreover, IoT device i propagates the updated information to its neighboring devices for them to learn.

3.3.3 Convergence of the Proposed Learning Method

Our proposed learning method consists of two phases during which a favored state s_f is chosen and then is repeatedly tested whether s_f should be changed. For the first learning phase, it must be shown that the favored state $s_f = H_T$ can be chosen correctly during the first phase with nonzero probability. Moreover, for the second phase, it must be shown that the learning method will not revert back to first phase with probability approaching 1 if $s_f = H_T$. As $s_f \neq H_T$ can be chosen during the first phase, it must also be shown that the learning method will revert back to first phase with

probability 1 when $s_f \neq H_T$. If the aforementioned properties of both first and second phases of the learning method can be shown, then the learning method will be guaranteed to converge to the correct underlying state H_T, as shown next.

Theorem 1. *For a memory size m such that $(m-2) \geq \alpha$, if (3.8) holds, then the proposed finite memory learning method will converge in probability to the true, underlying state H_T.*

Proof. See Appendix 3.6.1. □

The result of Theorem 1 applies whenever (3.8) holds true for all agents participating in the learning method and none of the agents suffers from the limited observation capability. However, the IoT devices in our model have limited observation capabilities, and, thus, (3.8) may not hold true for all of them. However, with sufficiently large memory size m such that r'_d in (3.30) is large enough to circumscribe the entire deployment region, (3.8) will hold true for all IoT devices. Therefore, with an additional condition on the value of m, the proposed finite memory learning method will converge in probability to H_T for the IoT devices. Furthermore, the IoT devices are limited in memory, and, thus, the effect of having small or big memory sizes on the performance and convergence of learning will be further analyzed via simulations in Section 3.4.

3.3.4 Delay of Critical Messages

For larger sizes m of the devices' memory, (3.8) holds true for more IoT devices that can learn β correctly during the second phase, and, thus, the delay of the IoT devices with critical messages with be smaller. As the learning propagates during t time slots, an IoT device, which learned and is located furthest from a device with critical messages, is located at a distance of tr_c from the abnormal event. In other words, after t time slots, only the IoT devices that are located closer than tr_c are part of a learning sequence. Therefore, whether an IoT device was able to learn correctly depends on both m and t as the IoT devices that are located within $r_t = \min(tr_c, r'_d)$ are likely to learn correctly.

Assuming a massive IoT network with limited number of RAPs with $N \gg p$, $T_{\min}$ is very large, and, thus, D and D' are dictated by D_c and D'_c respectively. Since D'_c approaches 1 as β increases, D_{th} must be greater than 1 for a feasible value of β. For a given threshold delay $D_{\text{th}} < D_c$, the value of β is, given system parameters p_c and N_a:

$$D'_c = \left(\frac{p_c+\beta}{p_c+\beta-1}\right)^{N_a-1} \leq D_{\text{th}}, \tag{3.31}$$

$$p_c + \beta \leq \sqrt[N_a-1]{D_{\text{th}}}(p_c+\beta-1) \tag{3.32}$$

$$\beta(1-\sqrt[N_a-1]{D_{\text{th}}}) \leq \sqrt[N_a-1]{D_{\text{th}}}(p_c-1) - p_c, \tag{3.33}$$

$$\frac{\sqrt[N_a-1]{D_{\text{th}}}(p_c-1)-p_c}{1-\sqrt[N_a-1]{D_{\text{th}}}} \leq \beta = \left\lceil \frac{\sqrt[N_a-1]{D_{\text{th}}}(p_c-1)-p_c}{1-\sqrt[N_a-1]{D_{\text{th}}}} \right\rceil. \tag{3.34}$$

For the value of β in (3.34), the delay threshold D_{th} will be satisfied for the critical messages for m and t approaching ∞, while minimally affecting the periodic transmissions. However, it is impractical to have very large m considering the restricted resources of IoT devices. Furthermore, it is of interest to analyze the transient phase of our learning method, because the IoT devices with critical messages try to transmit as other devices are still learning. Therefore, the maximum number of reallocated RAPs β_t at time t, for finite values of m and t, is of interest to determine the expected delay of critical messages for realistic values of m and t.

Theorem 2. *For finite values of t and m, the maximum expected number of reallocated RAPs β_t at time t with average n_t IoT devices that learned correctly is:*

$$\mathbb{E}[\beta_t] = \frac{(p_f-\beta)!}{p_f!}\sum_{b=0}^{\beta} bP(n_t,b)P(p_f-n_t,\beta-b)C(\beta,b), \tag{3.35}$$

where $P(n,k)$ is k-permutations of n and $C(n,k)$ is k-combinations of n.

Proof. See Appendix 3.6.2. □

Theorem 2 derives the maximum expected number of contention-free RAPs that will be reallocated to contention-based RAPs to reduce the delay of critical messages in a realistic IoT with finite t and m. Moreover, it is interesting to note that when $m = \infty$ and $t = \infty$, $n_t = p_f$ as r'_d will be large enough to include the entire deployment region. Therefore, the probability of having β_t reallocated RAPs at time t in (3.41) is 1 for $\beta_t = \beta$ and zero otherwise, and, thus, $\mathbb{E}[\beta_t] = \beta$. Moreover, for higher values of m, t, and, thus, n_t, $\mathbb{E}[\beta_t]$ will approach β and the critical messages are more likely to satisfy D_{th}. With finite values of m and t, the lowest expected delay that can be achieved with our proposed learning method is $\left(\frac{p_c+\mathbb{E}[\beta_t]}{p_c+\mathbb{E}[\beta_t]-1}\right)^{N_a}$.

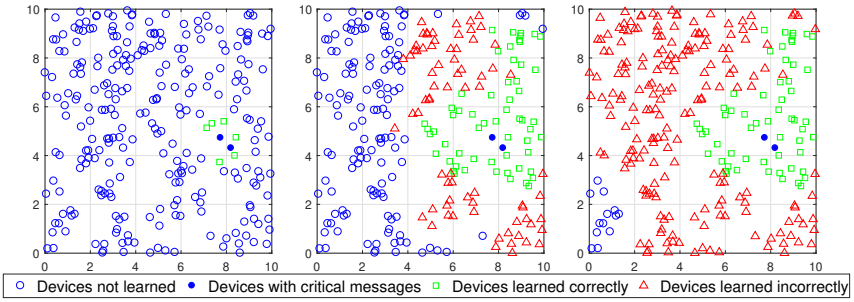
3.4 Simulation Results and Analysis

For our simulations, we consider a rectangular area of width w and length l within which the IoT devices are deployed following a Poisson point process of density λ and an expected number of IoT devices $wl\lambda$. We let $w = l = 100$ m, $r_c = 2$ m, and $r_d = 10$ m, and we choose a time slot duration of 0.25 ms [73]. A random location within the deployment region is chosen to be the location of an abnormal event, and the IoT devices within a distance of 1 m will have critical messages and initiate the learning procedure. The number of RAPs p is 64 [70], which includes $p_f = 63$ contention-free RAPs and $p_c = 1$ contention-based RAPs. We also set $K = 3$ and $\alpha = 5$. Moreover, the probability of missed detection p'_{01} outside of r'_d is set to 0.9 [72]. All statistical results are averaged over a large number of independent runs.

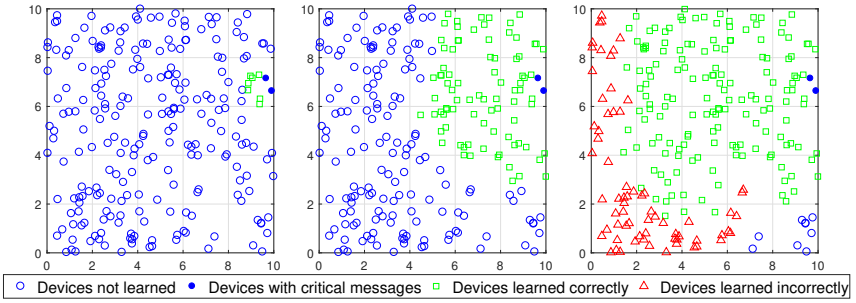
Fig. 3.3 shows snapshots of learning with different memory sizes progressing in a small-scale network at $t = 0.25$ ms, 1.75 ms, and 4.25 ms. For both scenarios, learning converges by $t = 4.25$ ms. At $t = 0.25$ ms, for both values of m, only the neighboring devices of the IoT devices with critical messages had a chance to learn in the first phase, and they all learned correctly. At $t = 1.75$ ms, for learning with $m = 3$ bits shown in Fig. 3.3(a), IoT devices outside of r'_d started to learn incorrectly due to the limited observability. However, for learning with $m = 10$ bits shown in Fig. 3.3(b), IoT devices are still within r'_d and are learning correctly. At $t = 4.25$ ms, even with larger memory size of $m = 10$ bits, some IoT devices are outside of r'_d and are unable to learn correctly. For both memory sizes, there are few IoT devices that did not have a chance to learn as they are not within r_c of any other IoT devices.

Fig. 3.4 shows the cumulative distribution function for the delay of critical messages for different delay thresholds D_{th} with $\lambda = 2$, $p_{11} = 0.9$, and $m = 5$ bits. Fig. 3.4 shows that the delay threshold is satisfied, which means that the achieved delay is less than or equal to D_{th}, with probability 0.80 for $D_{\text{th}} = 0.75$ ms, 0.87 for $D_{\text{th}} = 1.25$ ms, 0.95 for $D_{\text{th}} = 2.5$ ms, and 0.99 for $D_{\text{th}} = 5$ ms. Moreover, the average achieved delay is 0.59 ms for $D_{\text{th}} = 0.75$ ms, 0.72 ms for $D_{\text{th}} = 1.25$ ms, 0.99 ms for $D_{\text{th}} = 2.5$ ms, and 1.35 ms for $D_{\text{th}} = 5$ ms. Although the largest $D_{\text{th}} = 2.5$ ms has the longest average delay, it has the highest probability of satisfying the delay threshold. This is because the IoT devices with critical messages have more time available to satisfy D_{th} for larger D_{th}, and more IoT devices can learn β with more time available. Moreover, with more time, the IoT devices with critical messages are more likely to be allocated contention-free RAP by the BS for successful RA request. Here, it is important to note that D_{th} may not be satisfied, because t and m are finite.

Fig. 3.5 shows the cumulative distribution function for the delay of critical messages for different network densities λ with $D_{\text{th}} = 2.5$ ms, $p_{11} = 0.9$, and $m = 5$ bits. As our proposed learning method relies on the IoT devices communicating and propagating information, it is important that the IoT devices have at least one neighboring device to be able to learn β. Therefore, a massive IoT network with higher values of λ is better for learning. However, with smaller value of λ, T_{min} is smaller, and, thus, D_f is smaller. The value of D_f is 19.75 ms for $\lambda = 1$, 39.63 ms for $\lambda = 2$, and 79.25 ms for $\lambda = 4$. Therefore, a small value of λ increases the delay by increasing the number of isolated



(a) Learning with smaller memory of $m = 3$ bits.



(b) Learning with larger memory of $m = 10$ bits.

Figure 3.3: Snapshots showing how the proposed learning scheme progresses in small scale networks at $t = 0.25$ ms, 1.75 ms, and 4.25 ms, respectively.

IoT devices and decreases the delay by decreasing the value of D_f. From Fig. 3.5, we can see that the probability of achieving a delay less than or equal to D_{th} is 0.97 for $\lambda = 1$, 0.95 for $\lambda = 2$ and 0.92 for $\lambda = 4$. Therefore, it is more likely to satisfy D_{th} with lower λ, and, thus, the effect of having smaller D_f is more significant than having more isolated IoT devices. Furthermore, this shows the effectiveness of our learning method in a massive network with devices that have limited resources and capabilities.

Fig. 3.6 shows the cumulative distribution function for the delay of critical messages for different probabilities of detection p_{11} with $D_{\text{th}} = 2.5$ ms, $\lambda = 2$, and $m = 5$ bits. Since the observation of an IoT device affects its learning as well as that of all IoT devices later in the sequence, the performance of our proposed method significantly depends on the quality of observations. For low p_{11}, the IoT devices are prone to false alarms as $p_{01} = 1 - p_{11}$, and, thus, they do not learn N_a correctly, which causes higher critical message delays. The delay threshold is satisfied with probability 0.98 for $p_{11} = 0.99$, 0.96 for $p_{11} = 0.9$, 0.85 for $p_{11} = 0.7$, and 0.57 for $p_{11} = 0.5$. Unlike D_{th} and λ, which indirectly affect the performance of learning, p_{11} affects the performance of learning method directly and more significantly, because the probability of a successful RA request achieves 0.99 after 11 time slots.

Fig. 3.7 shows the cumulative distribution function for the delay of critical messages for different memory sizes m with $D_{\text{th}} = 2.5$ ms, $\lambda = 2$, and $p_{11} = 0.9$. The effect of having larger values of m is not significant in reducing the delay of critical messages even though the memory size m is critical for the convergence of proposed learning method. This is because the RA request is successfully done during first few time slots as shown in Fig. 3.7, when the learning method is still in an early transient phase. The effect of having larger values of m is more clear when the learning has progressed to the IoT devices at the edge of detection range r_d as the memory size m extends r_d to r'_d. Therefore, the effect of m on the performance of the learning method will be more pronounced after more devices learned as is later shown in Fig. 3.8.

The convergence of the proposed learning technique is analyzed via simulations by studying the percentage of IoT

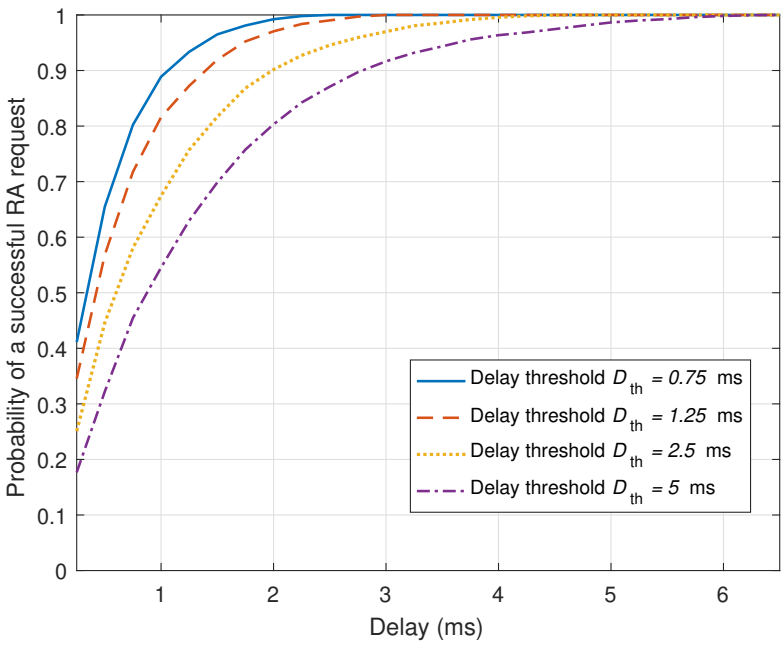


Figure 3.4: Cumulative distribution function of delay of critical messages for different D_{th}.

devices that learn correctly after all devices had a chance to learn N_a. The parameters that affect the performance of the proposed learning technique include memory size m, network density λ, and probability of detection p_{11} within r'_d. The network density λ affects the number of IoT devices that can learn such that an IoT device is more likely to have no neighboring device within r_c to get necessary information for learning for lower values of λ. Such isolated device will not be able to learn using our proposed learning method. However, if the network is dense enough, the performance of learning is limited not by λ but by memory size m and probability of detection p_{11}. The memory size m determines the effective detection radius r'_d and p_{11} controls how accurately the devices can observe within r'_d. For the subsequent simulations, we take $D_{\text{th}} = 3$, while varying m and p_{11}.

Fig. 3.8 shows the average percentage of devices that learned N_a correctly out of all N IoT devices for different memory sizes m with $\lambda = 2$. As the IoT devices can learn N_a correctly within r'_d and r'_d depends on m, the average percentage of devices that learned correctly is clearly limited for finite memories $m = \{3, 10, 15, 30\}$ bits, while the average percentage approaches 100% for infinite memory $m = \infty$ bits. Moreover, the average percentage of devices that learned correctly is an increasing function of m as r'_d is an increasing function of m. The maximum average percentage of devices that learned correctly is 4.67% for $m = 3$ bits, 18.79% for $m = 10$ bits, 29.84% for $m = 15$ bits, and 66.07% for $m = 30$ bits. This is because as the learning progresses further, the IoT devices outside of r'_d will learn incorrectly, and, thus, the percentage of devices that learn correctly out of all N will no longer increase. Moreover, the maximum average percentages for different values of m are achieved at different time slots since a larger m requires more time to achieve the maximum average percentage.

Fig. 3.9 shows the average percentage of devices that learned N_a correctly out of all N IoT devices for different probabilities of detection p_{11} with $m = 15$ bits and $\lambda = 2$. For lower p_{11}, the devices are more likely to observe and learn N_a incorrectly. Learning incorrectly will affect subsequent devices in their learning as the proposed approach relies on the observation of previous IoT devices. For $p_{11} = 0.5$, less than 1% of IoT devices learn correctly, and about 17.5% of IoT devices learn correctly even for $p_{11} = 0.9$. Even for a p_{11} close to 1, Fig. 3.9 shows that the average percentage of devices that learn N_a correctly does not approach 100%. Hence, we can conclude that m is much more critical in increasing the percentage of devices that learned correctly than p_{11}.

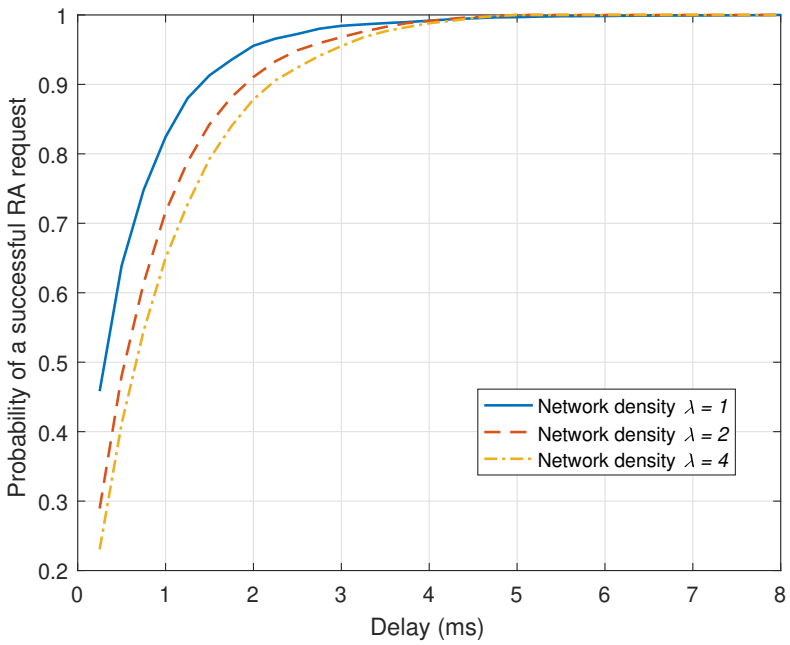


Figure 3.5: Cumulative distribution function of delay of critical messages for different λ.

The percentage of devices that learned correctly does not approach 100% in Fig. 3.9, and the proposed learning method does not seem to converge to H_T. This is because the value of m is not large enough to let r'_d circumscribe the deployment region. For small m, some IoT devices do not satisfy (3.8) due to their limited observation range. In other words, for those devices, the effective detection range r'_d is not extended enough to include the IoT devices located far away from the abnormal event, and the IoT devices located outside of r'_d will not learn correctly. However, it is unnecessary to have all IoT devices learn correctly to satisfy a given delay threshold as shown in Fig. 3.4, Fig. 3.5, and Fig. 3.6. Furthermore, since the memory size m will be finite or even small in practical IoT systems, it is important to analyze the convergence of learning method with finite values of m as shown in Fig. 3.8.

3.5 Summary

In this chapter, we have proposed a novel learning framework for enabling IoT devices to reallocate limited communication resources depending on the types and number of messages existing in the IoT. We have introduced a finite memory multi-state sequential learning using which the heterogeneous IoT devices can reallocate RAPs appropriately to reduce the delay of critical messages. We have shown that our proposed learning framework is suitable for the IoT devices with limited computational capability and finite memory and effective against the limited observation capability of IoT devices. Furthermore, we have proved the convergence of our proposed learning framework and derived the lowest expected delay of critical messages that can be achieved with our proposed learning framework. Simulation results have shown that the cumulative distribution function of delay of critical messages and average percentage of devices that learned correctly are functions of memory size, network density, and probability of detection. In particular, our proposed learning framework has shown to be effective in massive network with low delay threshold.

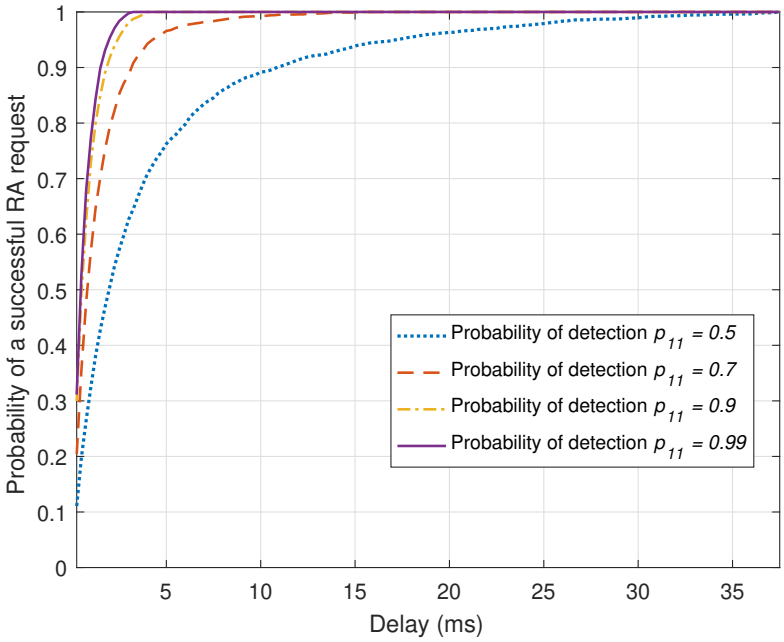


Figure 3.6: Cumulative distribution function of delay of critical messages for different p_{11}.

3.6 Appendix

3.6.1 Proof of Theorem 1

The first phase of the proposed learning method is based on maximum likelihood estimation, and, thus, the chosen s_f depends on the K observations. Given the assumption in (3.8), if all of the K observations are H_T, then $s \in \mathcal{S}$ that will maximize (3.11) will be H_T and, thus, s_f will be H_T. Moreover, given (3.7), $\Pr(e = H_T|H_T)$ cannot be zero. Since the observations are independent, the probability of observing K H_T is $\Pr(e = H_T|H_T)^K$, and, thus, the probability of having K observations of H_T is not zero. Therefore, the favored state s_f will be chosen correctly such that $s_f = H_T$ with nonzero probability from the first learning phase.

If $s_f = H_T$ is chosen from first phase, then the true underlying state is s_f and, thus, second phase will converge to H_T. However, if $s_f \neq H_T$ is chosen from the first phase, then the true underlying state is s'_f and, thus, the second phase will not converge to H_T. Therefore, there is a returning condition, which is α consecutive observations of s'_f, using which the learning method can go back to the first phase if $s_f \neq H_T$ seems to be chosen from first phase.
When $s_f \neq H_T$, the probability of observing s'_f for α consecutive times, which may cause the learning method to go back to the first phase, is:

$$\Pr(n_e|s_f \neq H_T) = \Pr(\{e_{n'-\alpha+1}, \cdots, e_{n'}\} = \{s'_f, \cdots, s'_f\}\ \forall n' \leq n_e \mid s_f \neq H_T) \tag{3.36}$$

$$= \begin{cases} 0 & \text{if } n_e < \alpha, \\ q_1^\alpha & \text{if } n_e = \alpha, \\ q_1^\alpha + (N-\alpha)p_1 q_1^\alpha & \\ \quad -p_1 q_1^\alpha \left(\sum_{i=0}^{n_e-(\alpha+1)} \Pr(i|s_f \neq H_T) \right) & \text{if } n_e > \alpha, \end{cases} \tag{3.37}$$

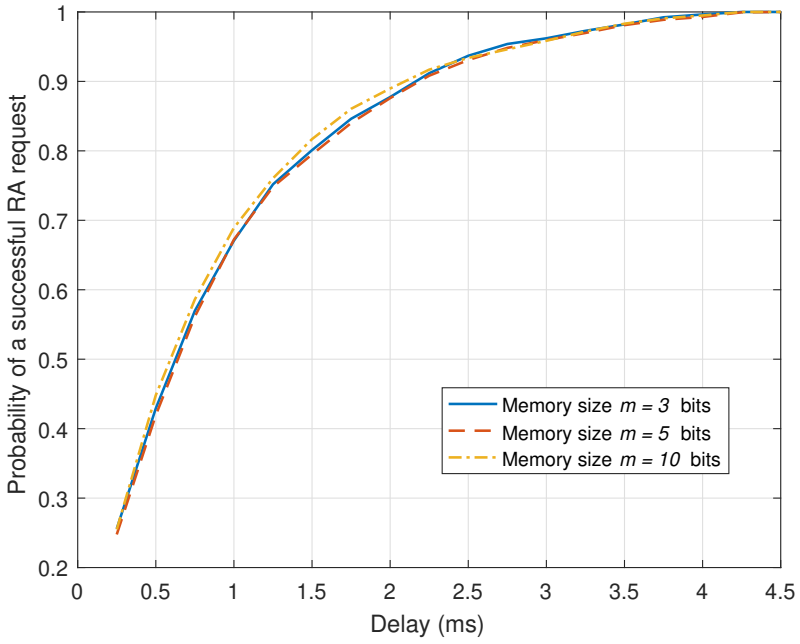


Figure 3.7: Cumulative distribution function of delay of critical messages for different m.

where q_1 is $(1-p_1)$ and n_e is number of observations. Similarly, for the case when $s_f = H_T$, the probability of observing s'_f for α consecutive times after n_e observations is:

$$\Pr(n_e|s_f = H_T) = \Pr(\{e_{n'-\alpha+1}, \cdots, e_{n'}\} = \{s'_f, \cdots, s'_f\}\ \forall n' \leq n_e \mid s_f = H_T) \tag{3.38}$$

$$= \begin{cases} 0 & \text{if } n_e < \alpha, \\ q_2^\alpha & \text{if } n_e = \alpha, \\ q_2^\alpha + (N-\alpha)p_2 q_2^\alpha & \\ \quad -p_2 q_2^\alpha \left(\sum_{i=0}^{n_e-(\alpha+1)} \Pr(i|s_f = H_T) \right) & \text{if } n_e > \alpha, \end{cases} \tag{3.39}$$

where q_2 is $(1-p_2)$. Here, we note that $\Pr(n_e|s_f \neq H_T)$ and $\Pr(n_e|s_f = H_T)$ are cumulative probability distributions. Since $p_2 > p_1$, $\Pr(n_e|s_f \neq H_T)$ increases to 1 faster than $\Pr(n_e|s_f = H_T)$. However, as n_e increases to infinity, $\Pr(n_e|s_f \neq H_T)$ and $\Pr(n_e|s_f = H_T)$ will both approach 1. Thus, as n_e increases to infinity, the learning method will go back to the first phase regardless of s_f. Therefore, the probability density function p_b, which determines the probability of going back to Phase 1 once s'_f is observed α consecutive times, must be designed appropriately to ensure that learning will only go back to Phase 1 when $s_f \neq H_T$.

An appropriate choice for p_b is such that p_b is high for lower values of n_e, while p_b approaches 0 as n_e increases. Since $\Pr(n_e|s_f \neq H_T)$ increases to 1 faster than $\Pr(n_e|s_f = H_T) \forall n_e$ as $p_2 > p_1$, $\Pr(n_e|s_f \neq H_T)$ is much greater than $\Pr(n_e|s_f = H_T)$ for small values of n_e, and, thus, it is much more likely to observe s'_f for α consecutive times when $s_f \neq H_T$ than when $s_f = H_T$ for small values of n_e. Therefore, for small n_e, p_b must be high for small values of n_e so that the learning method is likely to go back to the first phase when $s_f \neq H_T$ as $\Pr(n_e|s_f \neq H_T)$ is big, while the learning method is unlikely to go back to the first phase as $\Pr(n_e|s_f = H_T)$ is small. However, as $\Pr(n_e|s_f = H_T)$ approaches 1 as n_e goes to infinity, p_b must approach 0 as n_e goes to infinity to prevent going back to first phase when $s_f = H_T$. One such choice for p_b will be a sigmoid function that quickly decreases to 0 for $n_e > a$ for some value a.

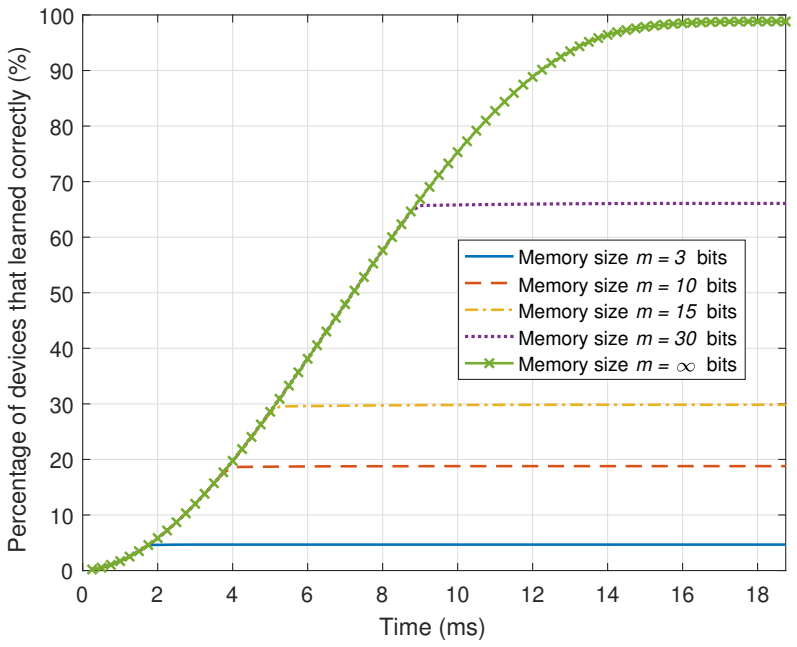


Figure 3.8: Average percentage of devices that learned correctly out of N devices with varying m.

For instance, one such sigmoid function is:

$$C\left(1-\frac{n_e-a}{1+|n_e-a|}\right), \tag{3.40}$$

where C is a scaling factor to make (3.40) an appropriate probability distribution. Similar to α, the value of a is a design parameter that determines the value of n_e at which it becomes unlikely for the learning method to go back to the first phase. Furthermore, the value of a should be chosen between the values of n_e where $\Pr(n_e \mid s_f \neq H_T)$ is close to 1, while $\Pr(n_e \mid s_f = H_T)$ is close to 0, and where $\Pr(n_e \mid s_f = H_T)$ starts to significantly increase to 1. With appropriate choice for p_b, when $s_f \neq H_T$, the learning method is highly likely to go back to the first phase even when n_e is low, because both p_b and $\Pr(n_e \mid s_f \neq H_T)$ are high. However, when $s_f = H_T$, the learning method is highly unlikely to go back to the first phase, because either p_b or $\Pr(n_e \mid s_f = H_T)$ is high, while the other is low. Therefore, as learning progresses and the number of observations n_e increases to infinity, the learning method goes back to the first phase to change s_f with probability approaching 0 when $s_f = H_T$. However, n_e increasing to infinity does not require the learning method to have infinite memory as our finite memory learning method updates the memory by replacing the oldest observation by newest observation. With m such that $(m-2) \geq \alpha$, the finite memory will have enough observations to check if α consecutive observations of s'_f have occurred. Since the first phase will choose $s_f = H_T$ with nonzero probability, the learning method will converge to H_T in probability.

3.6.2 Proof of Theorem 2

For finite values of t and m, the maximum ratio of IoT devices that learn correctly is $\frac{\pi r_t^2}{A}$, where A is the area of geographical region on which the IoT devices are deployed and $r_t = \min(tr_c, r'_d)$ is a range within which the devices are likely to learn correctly. It assumes that r_t is entirely within the network and all IoT devices learn correctly within r_t, which is best scenario for our learning method. In a given time slot at time t, the expected number of IoT devices

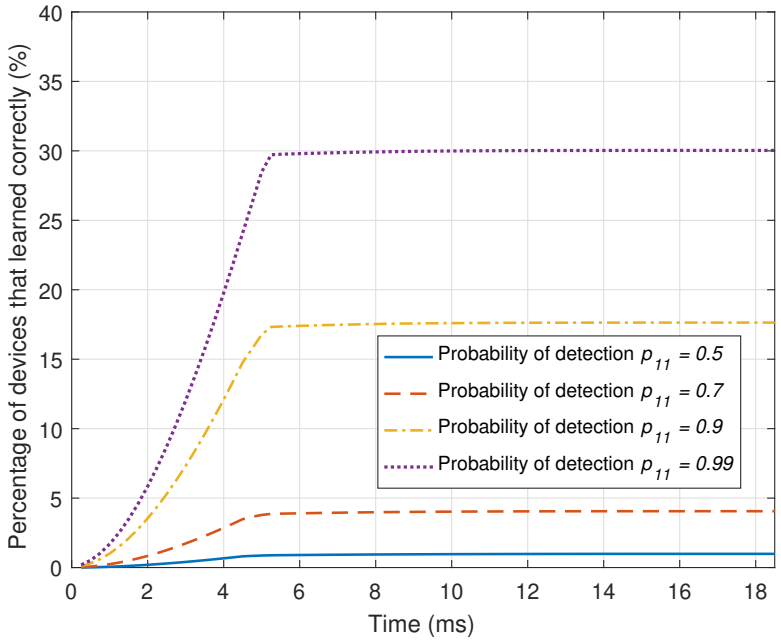


Figure 3.9: Average percentage of devices that learned correctly out of N devices with varying p_{11}.

with periodic messages transmitting in that time slot that learned correctly is $n_t = \frac{p_f \pi r_t^2}{A}$ and, the expected number of IoT devices with periodic messages transmitting in that time slot that did not learn correctly is $p_f - n_t$.

Since there are IoT devices outside of r_t that do not learn correctly with finite m and t, there will be varying number of reallocated RAPs β_t in different time slots, and the value of β_t may range from 0 to β. For a value of $\beta_t \in [0, \beta]$, the probability of having β_t reallocated RAPs at any given time t is:

$$\frac{(p_f - \beta)!}{p_f!} P(n_t, \beta_t) P(p_f - n_t, \beta - \beta_t) C(\beta, \beta_t), \tag{3.41}$$

where $P(n, k)$ is k-permutations of n and $C(n, k)$ is k-combinations of n. The probability in (3.41) considers different cases at which β_t IoT devices that learned correctly are assigned to use the first β contention-free RAPs, which are to be allocated first, for their periodic messages. Furthermore, the expected number $\mathbb{E}[\beta_t]$ of contention-free RAPs that will be reallocated to contention-based RAPs is:

$$\mathbb{E}[\beta_t] = \frac{(p_f - \beta)!}{p_f!} \sum_{b=0}^{\beta} b P(n_t, b) P(p_f - n_t, \beta - b) C(\beta, b). \tag{3.42}$$

Chapter 4

Distributed Age of Information Minimization in IoT

4.1 Background, Related Works, and Contributions

The effective operation of several IoT services, such as industrial monitoring [74], health monitoring [75], drones [76], virtual reality [77], and vehicular network [78], requires timely and frequent communications. To maintain the proper performance of such diverse IoT applications, the base station (BS) must maintain the most relevant information gathered from the IoT devices at any given time.

In addition to timely transmissions from the devices to the BS, another key challenge is to account for the distinctive characteristics of an IoT and its devices. One prominent property of an IoT is its massive scale as the number of devices greatly outnumbers the available communication resources [1]. Therefore, an appropriate allocation of the limited communication resources among numerous IoT devices is necessary for the deployment of an IoT and its services [79]. Furthermore, the IoT exhibits a high heterogeneity in terms of device types, functions, messages, transmission requirements, and resource constraints [80]. The aforementioned IoT properties pose challenges for timely uplink transmission in an IoT. To ensure the performance of time-sensitive IoT applications despite the aforementioned challenges, a new information timeliness performance metric is needed as an alternative to conventional delay, reliability, and data rate.

To evaluate the communication between the BS and the devices, the *age of information* (AoI), which is a metric that can quantify the relevance and the freshness of the information, is used [81, 82]. However, the AoI has different characteristics compared to delay [83], because it explicitly considers packet generation time. The problem of AoI minimization in an IoT has unique challenges due to the characteristics of an IoT, including massive scale, limited communication resources, and IoT device heterogeneity. Largely, AoI minimization can be done in a centralized way or in a distributed way. However, a centralized AoI minimization approach is not always viable for an IoT, because the energy constrained IoT devices may not be able to communicate frequently with BS. On the other hand, a distributed AoI minimization approach may require extensive device-to-device communication and could perform worse than a centralized solution for some IoT scenarios. Therefore, both centralized and distributed AoI minimization must be investigated to compare their applicability and performances in an IoT.

4.1.1 Existing Works

A number of recent works studied the problem of AoI minimization in wireless networks [82–102]. These studies use various approaches to minimize the AoI under different constraints and conditions. For instance, the works in [82–101] study a variety of scheduling policies for AoI minimization in different networks, including single hop broadcast network [84], single-hop uplink communication [85], multi-hop uplink communication [86], and ad-hoc networks [87]. The authors in [88] and [89] propose and analyze scheduling policies for the wireless networks with known and unknown channel state information. The works in [83], [85], [87], and [90] introduce effective scheduling policies to minimize the average AoI with network constraints, such as throughput requirement, physical constraint for sensing, spectrum sharing, and varying packet sizes. In [87], [91], and [92], the authors use online techniques, such as reinforcement learning to perform AoI-minimal scheduling. Moreover, the authors in [92] and [93] analyze the performance of user scheduling for minimizing the average AoI in presence of multiple sources of information and propose a hybrid queueing system.

The works in [94–97] address the problem of AoI minimization using queueing-theoretic approaches. In [94], the authors analyze the peak AoI in a multi-class queueing system with packets having heterogeneous service times and requirements. The authors in [95–97] derive closed-form solutions for the average AoI and the peak AoI for different queueing models, including M/M/1, M/G/1, and M/G/1/1. In [98], the authors consider a vehicular network with ultra-reliable low-latency communication and minimize the tail of the AoI distribution. The peak AoI considering the packet delivery failure is analyzed in [99]. Moreover, the non-orthogonal multiple access is compared against the conventional orthogonal multiple access in terms of AoI minimization in [100]. The authors study the sampling policies to minimize the average AoI with the joint status sampling in IoT [82] or with the non-linear aging functions [101].

Despite being interesting, existing solutions in [82–101] are not suitable for an IoT and cannot be readily adapted for IoT devices, because they do not consider the prominent properties and the key challenges of IoT. One of the key challenges in IoT is the massive scale of IoT coupled with highly limited available communication resources. However, the works in [83–87, 91], and [99–101] do not investigate the realistic IoT scenario in which the number of devices greatly outnumbers the communication resources. Furthermore, the inherent heterogeneity among IoT devices and the presence of non-linear aging functions are not considered in [82, 84–91], and [95–100]. Most of the prior works for AoI minimization in [83–89, 92–97], and [101] only consider a centralized approach. Moreover, in [102], we studied centralized AoI minimization with non-linear aging functions and proposed a centralized resource allocation scheme to enable the BS to consider different aging functions. However, the work in [102] only investigates a centralized approach for AoI minimization and does not introduce a distributed resource allocation framework for an IoT with non-linear aging functions. A centralized approach may not always be suitable for an IoT, because the frequent communication with BS is not viable for the energy constrained IoT devices. These important challenges for enhancing the AoI in an IoT have been largely overlooked in prior works [82–102].

4.1.2 Contributions

The main contributions of this paper are novel centralized and distributed resource allocation frameworks that can be used to minimize the average AoI for a massive IoT with heterogeneous devices and non-linear aging functions. In particular, we capture the heterogeneity among IoT devices using *non-linear aging functions*. Typically, the AoI is defined only in terms of time, and it is assumed to increase linearly with a slope of 1 [81]. However, the definition of the AoI can be broader such that the AoI can be a function of completeness, validity, accuracy, currency, and utility [103, 104]. Under such a broader definition of the AoI, the aging function can be defined as an age penalty function or an age utility function [101], which can be an exponential, linear, or step function [103]. As such, we propose to capture the heterogeneity among IoT devices and messages by assigning different aging functions based on the devices types, the IoT application, the message content, and the transmission requirement.

For centralized AoI minimization, we propose a new priority scheduling scheme with a learning perspective using

which the device types and the aging functions can be determined. For non-linear aging functions, we show that using the future AoI for priority scheduling achieves a lower average AoI than using the current AoI. Simulation results for the centralized approach show that the proposed priority scheduling scheme achieves 26.7% lower average AoI with high activation probability and 31.7% lower average AoI with high outage probability than a simple priority scheduling. In particular, our approach outperforms a simple priority scheduling and performs similar to a priority scheduling with complete information on device types and aging functions.

For the distributed AoI minimization, we formulate a minority game [105], using which massive number of IoT devices can share the limited available communication resources autonomously. Furthermore, a payoff function is designed to allow the messages with the highest AoI to transmit first in a self-organizing manner. We then show the conditions that a resource allocation among IoT devices must satisfy to achieve a Nash equilibrium (NE). We propose a stochastic crowd avoidance algorithm for the resource allocation game and prove that the resource allocation using our proposed algorithm converges to an NE with sufficient information and under certain network parameters. Simulation results for the distributed case show that the proposed algorithm is effective in minimizing the AoI even if the devices only have the partial information about other devices. The results show that our game-based approach achieves 63.6% lower average AoI with limited information and 45.8% lower average AoI with high outage probability than a random resource allocation. Moreover, after convergence, our game-based approach performs similar to the predetermined resource allocation with complete information.

The centralized and the distributed AoI minimization schemes are compared in terms of overhead, implementation, and requirements. In particular, the centralized AoI minimization has an overhead of uplink communication request, while the distributed AoI minimization has an overhead of device-to-device communication. Simulation results show that the distributed AoI minimization achieves 40-fold higher average AoI than the centralized AoI minimization in a massive IoT, where the communication resources are highly limited and the devices only have partial information. In a less constrained IoT, simulation results show that the distributed AoI minimization achieves 8-fold higher average AoI than the centralized AoI minimization. Although the centralized AoI minimization outperforms the distributed AoI minimization in terms of average AoI, the distributed approach may be more suitable for an IoT, because the centralized approach may not be practical or viable for an IoT. As such, our analysis clearly showcases the contrasts between the two solutions.

The rest of this paper is organized as follows. Section II introduces the system model and the non-linear aging functions. Section III analyzes the AoI minimization with coexistence of linear and non-linear aging functions. Section IV presents the centralized and the distributed resource allocations in an IoT. Section V analyzes the simulation results, while Section VI draws conclusions.

4.2 System Model

Consider the uplink of a wireless IoT system consisting of one BS serving N IoT devices. The IoT devices can transmit their messages to the BS using the communication resources allocated by either a centralized or distributed resource allocation scheme. To transmit to the BS, the IoT devices use time-slotted orthogonal frequency-division multiple access (OFDMA). Here, R time-frequency resource blocks (RBs) are allocated to the IoT devices at each time slot. If more than one IoT device use a given RB, none of the messages transmitted using the given RB can be successfully decoded, which leads to transmission failures. This implies that at most R devices can transmit successfully to the BS at a time slot. In an IoT where the number of devices N greatly outnumbers the number of RBs R, RB allocation is critical for the operation of IoT, and RB allocation can be done in a centralized or in a distributed way.

Under a centralized resource allocation scheme, the BS allocates the RBs to the IoT devices such that a given RB is used by only one device. Therefore, duplicate RB usage will not occur when using a centralized resource allocation. However, centralized resource allocation incurs an overhead related to the need that the devices request their uplink communication resources via a random access channel (RACH) [106]. Furthermore, the uplink communication re-

source request using RACH can fail resulting in transmission failure. In contrast, when using a distributed resource allocation scheme, the devices decide which RB to use autonomously without any intervention from the BS and without RACH. Although there is no overhead related to the need for requesting uplink communication resources, distributed resource allocation incurs an overhead related to the devices cooperating to avoid duplicate RB usage. Since there is no RACH request when performing distributed resource allocation, no RACH request failures will happen. However, the uplink transmission may fail because of a duplicate RB usage.

For both centralized and distributed resource allocations, the uplink transmission can also fail because of the RB outage. The RB outage is based on the signal-to-noise ratio (SNR) such that the transmission is considered to be a failure if the SNR is less than a given threshold $\epsilon \geq 0$. We consider a stationary Rayleigh fading channel with additive white Gaussian noise (AWGN), such that the statistical properties of channel do not change over time. Therefore, the SNR outage probability is $\Pr\left({}^{S^2}/_{\sigma^2} \leq \epsilon\right)$, where S^2 is a random variable for the received signal power and σ^2 is the variance of the AWGN. We assume that the IoT devices only know the distributional properties of the channel and the AWGN at the receiver. Furthermore, we assume that all devices transmit with the same transmit power as the devices do not know exact channel gain. Since we consider a Rayleigh fading channel, the received signal power S^2 is exponentially distributed, and the devices know the mean λ^{-1} of S^2. We assume that the transmit powers of all IoT devices are equal, and, thus, the mean of the received signal power will be the same for all devices. When an IoT device uses multiple RBs simultaneously, then the transmit power will be equally divided among those RBs. For instance, if an IoT device i uses $R_{i,t}$ RBs simultaneously at time slot t, then the received signal power is ${}^{S^2}/_{R_{i,t}}$. For an IoT device i using $R_{i,t}$ RBs simultaneously at time slot t, the outage probability $p_{i,t}$ for device i at time slot t will be:

$$p_{i,t} = \Pr\left(\frac{{}^{S^2}/_{R_{i,t}}}{\sigma^2} \leq \epsilon\right). \tag{4.1}$$

Since S^2 is exponentially distributed with mean λ^{-1}, $\frac{{}^{S^2}/_{R_{i,t}}}{\sigma^2}$ is exponentially distributed with mean $(R_{i,t}\sigma^2\lambda)^{-1}$ for a given $R_{i,t}$ and a known σ^2. Moreover, the outage probability $p_{i,t}$ can be interpreted as a cumulative distribution function of an exponential distribution. Since an exponential random variable S^2 with mean λ^{-1} has a cumulative distribution function of $\Pr\left(S^2 \leq \epsilon\right) = 1 - \exp\left(-\lambda\epsilon\right)$ for $\epsilon \geq 0$, the outage probability $p_{i,t}$ in (4.1) will be:

$$p_{i,t} = 1 - \exp\left(-\left(R_{i,t}\sigma^2\lambda\right)\epsilon\right). \tag{4.2}$$

For a successful uplink transmission, an RB must be used by only one device, and the SNR must be higher than a given threshold ϵ.

One prominent feature of an IoT is its massive scale. In particular, the number of IoT devices N greatly outnumbers the number of RBs R. For an IoT scenario where $N > R$, the problem of RB allocation among the IoT devices becomes more challenging. Another prominent feature of an IoT is the heterogeneity among the IoT devices. The IoT devices are heterogeneous in terms of message types, transmission requirements, and packet content. A metric that can be used to determine which R out of the N devices will transmit and to quantify the freshness of the information in perspective of the destination is AoI. Furthermore, the heterogeneity among the IoT devices and their messages can be captured by extending the definition of AoI to include the quality of information and introducing non-linear aging functions.

4.2.1 Age of Information

The AoI is a metric that quantifies the freshness of the information in the perspective of a destination [82]. The definition of the AoI is the time elapsed since the generation of a message that is most recently received at the BS, and, thus, the AoI considers both the packet generation time at the source and the packet delay for the BS to receive the new packet.

In prior art on the AoI, devices are commonly assumed to generate the messages at will [107] and to update BS with a new message just in time [108]. The generate-at-will model for AoI implies that IoT devices can have messages to

transmit to the BS at any time, and the just-in-time model for AoI implies that IoT devices transmit new message to the BS immediately after the successful transmission of the current message. However, for an IoT, the devices are not always *active* and do not always have the messages to transmit to the BS. In our model, a device has a message to transmit to the BS at a given time slot with an activation probability v_a. If a device transmits to the BS unsuccessfully at a given time slot, then the device retransmits the message immediately at following time slot without a random backoff time. For the distributed RB allocation, the proposed game is designed to give incentive to devices with low AoI to not transmit. The RACH phase of the centralized RB allocation can use the game formulated for distributed RB allocation to achieve a lower average AoI. The random backoff time [109] does not consider the AoI or the aging function, when the devices are avoiding collision. For instance, a long backoff time can be assigned to a device with higher AoI and exponential aging function, while a short backoff time can be assigned to a device with low AoI and linear aging function.

Minimizing the AoI implies that the destination maintains fresh information from the source. However, minimizing the AoI is different from simply minimizing delay [83]. The AoI is measured by an aging function, and it is typically assumed that all devices have the same, linear aging function with a slope of 1 [81]. However, by using different aging functions for different messages, the AoI can naturally capture the inherent heterogeneity among IoT devices and messages. For instance, depending on the device type and the message characteristics, the aging function can be assigned appropriately. If the device is a simple sensor transmitting a time-insensitive update messages, the appropriate aging function is a linear aging function. On the other hand, if the device is an industrial monitoring sensor transmitting a critical, time-sensitive status report, the appropriate aging function is an exponential aging function. By using different aging functions, the AoI captures both the freshness of information and the value of information [101, 103, 104]. Furthermore, with different aging functions, minimizing AoI reduces the delay more for time-sensitive messages than for time-insensitive messages.

To model the heterogeneous messages, we consider the coexistence of linear aging function and exponential aging function. In other words, some IoT devices have linearly aging messages, while other IoT devices have exponentially aging messages. In particular, with a linear aging function, the AoI from IoT device i at the beginning of time slot $t \in \mathbb{Z}_+$ is:

$$a_i(t) = t - \delta_i(t), \tag{4.3}$$

where $\delta_i(t)$ is time slot at which the most recent message from device i received by the BS was generated. With an exponential aging function, the AoI from IoT device i at the beginning of time slot $t \in \mathbb{Z}_+$ is:

$$b_i(t) = 2^{t-\delta_i(t)-1}. \tag{4.4}$$

Although a specific linear and exponential aging functions are considered, our proposed centralized and distributed approaches to minimize the AoI are not limited to these aging functions only, and any type of aging function can be used.

In a scenario where all IoT devices have the same linear aging function with slope 1, the IoT devices and the BS can easily determine the AoI of the messages by simply counting the number of time slots passed since the most recently received message was generated [81]. However, in our system model where different aging functions coexist, the aging function of a given message is determined by the content of the message [103]. For instance, the aging function of a message whose content is pertinent and critical would be the exponential aging function $b_i(t)$, while the aging function of a message whose content is normal and time-insensitive would be the linear aging function $a_i(t)$ [101]. This implies that the BS cannot determine the aging function directly before receiving it, and, thus, the BS cannot compute the AoI of the messages. Therefore, the devices determine the aging function of their own message and compute the current AoI.

To capture the heterogeneity among the IoT devices, the IoT devices are classified into different types based on the probabilistic properties of their messages. A typical IoT device would not always have a time-sensitive, urgent message to send to BS. Additionally, a device that usually sends time-insensitive messages may sometimes have a critical message to send. In our model, we consider two types of devices. Type 1 devices are more likely to have linearly aging messages than exponentially aging messages. In other words, type 1 devices have the messages with

aging function $a_i(t)$ with probability m_1 and have messages with aging function $b_i(t)$ with probability $(1 - m_1)$ with $1 > m_1 > 0.5$. Type 2 devices are more likely to have exponentially aging messages than linearly aging messages. In other words, type 2 devices have messages with aging function $b_i(t)$ with probability m_2 and have messages with aging function $a_i(t)$ with probability $(1 - m_2)$ with $1 > m_2 > 0.5$. We assume that the characteristics of devices types, such as m_1 and m_2, are known to the BS, but the BS does not know the type of a given device. Although having different types of messages realistically models the heterogeneity of the IoT devices, this makes the RB allocation more challenging, because the messages transmitted by a given device may have different aging functions. With non-linear aging functions and heterogeneous device types, the problem of AoI minimization is different from AoI minimization with only linear aging function and homogeneous devices. Hence, next we investigate the problem of non-linear AoI minimization with coexistence of different aging functions and heterogeneous devices.

4.3 Non-linear AoI Minimization

To minimize the average AoI, the devices with highest AoI are permitted to transmit to the BS, and the AoI of different devices are compared to decide which devices are allocated the RBs in a massive IoT. Without coexistence of different aging functions, comparing AoI from different devices to allocate the limited RBs is simple. If all devices have same linear aging function $a_i(t)$, then $a_i(\tau) > a_h(\tau)$ implies $a_i(\tau + \beta) > a_h(\tau + \beta)$ for any positive integers τ and β given that devices i and h do not transmit successfully to the BS. Therefore, in a massive IoT with $N > R$, comparing current AoI with $t = \tau$ at time slot τ can be used to decide which R out of N devices transmit and to minimize the average AoI. Furthermore, using current AoI with $t = \tau$ or using future AoI with $t = \tau + \beta$ at time slot τ is equivalent, when all devices have same linear aging function $a_i(t)$.

When a linear aging function $a_i(t)$ and an exponential aging function $b_i(t)$ coexist, it is insufficient to only compare the AoI of different devices to minimize the average AoI, and RBs must be allocated considering the aging functions to minimize the average AoI. For instance, even if a device i with $b_i(t)$ has lower AoI than a device h with $a_h(t)$, device i will eventually have higher AoI than device h, because $b_i(t)$ increases faster than $a_h(t)$. Therefore, the AoI comparison must take aging functions into account, and one way to consider the aging functions is to compare the future AoI. Next, we prove that performing RB allocation based on the future AoI achieves a lower average AoI than an alternative that is based on the current AoI.

Proposition 1. In a massive IoT where $N > R$, if there are different aging functions $a_i(t)$ and $b_i(t)$, comparing the future AoI with $t = \tau + \beta$ for some positive integer β at time slot τ to determine the RB allocation achieves a lower average AoI than comparing the current AoI with $t = \tau$.

Proof. See Appendix 4.7.1. □

From Proposition 1, we observe that allocating RBs to the devices with highest future AoI results in lower average AoI than allocating RBs to the devices with highest current AoI. The value of positive integer β in determining which future AoI to use for the RB allocation is a design parameter. Furthermore, it is important to note that the current AoI at the time slot of successful transmission is used to compute the average AoI, and the future AoI is only used to determine the RB allocation. If higher values of β are used, devices with exponentially aging messages are more likely to be allocated the RBs than devices with linearly aging messages. This implies that devices with exponentially aging messages are allocated the RBs even when their current AoI is low, while devices with linearly aging messages are not allocated the RBs even when their current AoI is high. Therefore, for higher values of β, the average AoI of IoT devices with exponentially aging messages is lower, while the average AoI of IoT devices with linearly aging messages is higher. Furthermore, using higher values of β for the future AoI does not necessarily achieve a lower average AoI. In our system model, the centralized and the distributed RB allocations use future AoI with $\beta = 1$.

We consider that the IoT devices may have messages requiring multiple RBs to successfully transmit to the BS. When the messages take multiple RBs to successfully transmit, $\delta_i(t)$ in (4.3) and (4.4) will represent the time slot during

which the most recent message from device i, which is fully received by BS, was generated. Typically for the AoI, it is assumed that the messages take one RB to successfully transmit to the BS. However, to minimize the average AoI in a massive IoT, the devices must determine how to transmit the messages taking multiple RBs. In particular, if a device i has a message that requires n_i RBs to transmit, then the message may be transmitted simultaneously by using n_i RBs at a given time slot, consecutively by using 1 RB each time for n_i time slots, or jointly by using both simultaneous and consecutive transmissions. The simultaneous transmission may complete the transmission at once reducing the AoI, but it has high $R_{i,t}$ and outage probability $p_{i,t}$ (4.2). A consecutive transmission achieves its lowest outage probability $p_{i,t}$ with $R_{i,t} = 1$, but it takes the largest number of time slots to completely transmit increasing the AoI. Therefore, the problem of AoI minimization must consider the outage probability. The optimization problem to minimize the average AoI for a device i with a linearly aging message requiring n_i RBs at time slot τ is:

$$\min_{\boldsymbol{R}} \; a_i \left(\tau + \sum_{j=\tau}^{\tau+n_i-1} (1-p_{i,j})^{-1} \right), \tag{4.5}$$

$$\text{s.t.} \sum_{j=\tau}^{\tau+n_i-1} R_{i,j} = n_i, \tag{4.6}$$

$$0 \leq R_{i,j} \leq R \; \forall \, j, \tag{4.7}$$

where $\boldsymbol{R} = [R_{i,\tau}, ..., R_{i,\tau+n_i-1}]$. When a device uses $R_{i,t}$ RBs simultaneously, $(1-p_{i,t})$ is the probability of successful transmission given that duplicate RB selection did not occur. Taking $(1-p_{i,t})$ as success probability in a geometric distribution, the expected number of time slots needed for the successful transmission when using $R_{i,t}$ RBs simultaneously is $(1-p_{i,t})^{-1}$. Therefore, $\sum_{j=\tau}^{\tau+n_i-1}(1-p_{i,t})^{-1}$ is the expected number of time slots needed to transmit a message requiring n_i RBs. If a device i has an exponentially aging message, $b_i(t)$ replaces $a_i(t)$ in (4.5). The solution to the optimization problem (4.5) determines the number of RBs $R_{i,\tau}$ that a device i should be allocated with at time slot τ so that the average AoI is minimized. Furthermore, the solution to the optimization problem is used for centralized and distributed approaches for the RB allocation.

4.4 Resource Block Allocation

In a massive IoT with $N > R$, RB allocation is a challenging problem especially given the high heterogeneity among IoT devices and the messages. RB allocation in an IoT can be done in a centralized way or in a distributed way. Moreover, RB allocation schemes can achieve a lower average AoI by allocating the limited RBs to IoT devices with higher future AoI. Centralized RB allocation scheme is based on a priority scheduling improved with maximum likelihood to determine the aging functions and to learn the device types. The proposed distributed RB allocation scheme is designed to enable only the devices with sufficiently high future AoI to transmit, and the proposed stochastic crowd avoidance algorithm is proved to converge to an NE of the formulated game.

4.4.1 Centralized RB Allocation

For centralized RB allocation, the BS allocates the RBs to the IoT devices. In a time slot. the centralized approach has two phases. The active devices request for the RB using RACH in the first phase. If an active device is allocated an RB, the active device transmits its message to the BS in the second phase. Although there will be no duplicate RB usage causing a transmission failure in the second phase, there may be RACH preamble collision causing an RB request failure in the first phase. Therefore, using centralized RB allocation scheme, a device fails to transmit to the BS because of the RACH preamble collision, the outage based on SNR, and the lack of RB allocation.

We let P be the number of RACH preambles and N_t be the number of active devices at time slot t. The probability of the RACH preamble collision c_t at time slot t is:

$$c_t = 1 - \left(\frac{P-1}{P}\right)^{N_t - 1}. \tag{4.8}$$

If a device fails to request for an RB at time t with probability c_t, then this is equivalent to a transmission failure. However, if a device i successfully requests for an RB at time t, IoT device i sends the information about its current AoI C_i and the necessary number of RBs $R_{i,t}$ at time slot t. After gathering the information from the devices, the BS determines the RB allocation based on the future AoI of devices to minimize the average AoI.

In the second phase, the BS allocates the RBs to the active devices using a priority scheduling based on the future AoI. The priority scheduling allocates the R RBs to at most R active devices with the highest future AoI, minimizing the average AoI. If a device i with high future AoI has $R_{i,t} > 1$, then IoT device i may be allocated more than 1 RB at time slot t. The problem in the second phase is determining the future AoI using the received current AoI because of the coexistence of different aging functions. In other words, the BS does not know the aging function of an active device i, and the BS must determine the aging function to compute the future AoI F_i, which is used for the priority scheduling scheme to achieve a lower average AoI as shown in Proposition 1. However, the BS can determine the current aging function of an active device i from C_i.

The BS can determine that the aging function of an active device i is $a_i(t)$, when C_i cannot be derived using $b_i(t)$. The possible values of AoI using $a_i(t)$ are $\{1, 2, 3, 4, ...\}$, and the possible values of the AoI using $b_i(t)$ are $\{1, 2, 4, 8, ...\}$. Therefore, the values of AoI that are only possible using $a_i(t)$ are $\{3, 5, 6, 7, ...\}$. If the received current age C_i from an active device i is one of $\{3, 5, 6, 7, ...\}$, then the aging function is $a_i(t)$, and, thus, the future AoI F_i of device i is $C_i + 1$. The BS can also determine the aging functions of active devices by using regression. For an active device i, the BS can use the AoI from the most recently received uplink request from device i and C_i to determine if the aging function is $a_i(t)$ or $b_i(t)$. After determining the aging functions of active devices, the BS can compute the future AoI F_i of active devices and learn the device types, which are used for the priority scheduling scheme with learning. The BS can use either of the two methods to accurately determine the current aging functions, but it is not always possible to use these methods. When both methods are not possible to use in a given time slot due to RACH preamble collision, the BS uses the expected value of F_i for the priority scheduling scheme.

The BS can compute the expected value of F_i of an active device i by learning the device type of device i. The BS can learn the type of a device i by using the previous data from the instances that the BS was able to determine the aging function of messages from device i. In particular, the BS can use a maximum likelihood to determine the device types. We let $\mathcal{S}$ be a set of all device types and $\boldsymbol{O}_i$ be a vector of aging functions that a device i had that the BS was able to determine exactly. Furthermore, we let $k_{i,f}$ be the number of times that the BS determined device i to have aging function f. Assuming that the aging functions of a device i are determined independently, the learned type $H_i \in \mathcal{S}$ of a device i is:

$$H_i = \underset{s \in \mathcal{S}}{\operatorname{argmax}} \Pr(\boldsymbol{O}_i \mid s) = \underset{s \in \mathcal{S}}{\operatorname{argmax}} \prod_{f \in \mathcal{F}} \Pr(f \mid s)^{k_{i,f}}, \tag{4.9}$$

$$= \underset{s \in \mathcal{S}}{\operatorname{argmax}} \sum_{f \in \mathcal{F}} k_{i,f} \ln(\Pr(f \mid s)), \tag{4.10}$$

where $\mathcal{F}$ is a set of all aging functions. For our model, the values of $\Pr(f \mid s)$ for any $f \in \mathcal{F}$ and $s \in \mathcal{S}$ are known. In particular, $\Pr(a_i \mid s = 1) = m_1$, $\Pr(b_i \mid s = 1) = 1 - m_1$, $\Pr(a_i \mid s = 2) = 1 - m_2$, and $\Pr(b_i \mid s = 2) = m_2$. Therefore, the maximum likelihood in (4.10) can be solved by the BS directly.

Once the device type of a device i is learned, the expected future AoI $\mathbb{E}[F_i]$ of device i can be computed. For our model, if $H_i = 1$, then the expected future AoI $\mathbb{E}[F_i \mid H_i = 1]$ is:

$$\mathbb{E}[F_i \mid H_i = 1] = m_1(C_i + 1) + (1 - m_1)(2C_i). \tag{4.11}$$

Algorithm 4.1 Priority scheduling based on future AoI at time slot t.

```
Receive values C_i and R_{i,t}, and initialize R.
Compute F_i or E[F_i] for each C_i.
for j = 1, 2, ···
  Z_1 ← set of type 1 devices with j-th highest value among F_i.
  Z_2 ← set of type 2 devices with j-th highest value among F_i.
  for all i ∈ Z_2
    if R > 0,
      Allocate min(R, R_{i,t}) RBs to device i.
      R ← R − min(R, R_{i,t}). end if
  end for
  for all i ∈ Z_1
    if R > 0,
      Allocate min(R, R_{i,t}) RBs to device i.
      R ← R − min(R, R_{i,t}). end if
  end for
end for
```

If $H_i = 2$, then the expected future AoI $\mathbb{E}[F_i \mid H_i = 2]$ is:

$$\mathbb{E}[F_i \mid H_i = 2] = (1 - m_2)(C_i + 1) + m_2(2C_i). \tag{4.12}$$

The expected value of F_i is used for the priority scheduling scheme only if the exact value of F_i cannot be determined.

Priority scheduling determines the RB allocation among the active IoT devices based on the future AoI F_i. In a time slot τ, the RBs are allocated first to the devices with the highest F_i. Furthermore, for a device i with highest F_i at time slot τ, the number of RBs allocated to device i is $R_{i,\tau}$. If some of the active devices have same F_i, then the RBs are allocated first to the devices whose type is more likely to have a faster aging function. For instance, if a device i is device type 1, a device j is device type 2, and $F_i = F_j$, then device j has a priority over device i, because device j is more likely to have exponentially aging messages. The RBs are allocated to the active devices until all RBs are allocated or all active devices are allocated the RBs. One of the major problems with priority scheduling is the infinite blocking of low-priority tasks. However, since the priority depends on the future AoI, the priority of low-priority messages increases with time. Therefore, regardless of the aging function or the device type, the messages will eventually be allocated the RBs to transmit to the BS. One of the limitations of centralized RB allocation scheme is the overhead related to the RB request via RACH. With the higher number of active devices N_t at time t, the probability of the RACH preamble collision c_t becomes significant, and, thus, more transmission failures occur. Furthermore, with centralized RB allocation scheme, a frequent communication between the devices and the BS is required, which may not be viable for the IoT devices [110]. However, the main advantage of centralized RB allocation scheme is that the RBs are fully utilized.

4.4.2 Distributed RB Allocation

Distributed RB allocation enables the devices to allocate the RBs in a self-organizing manner without any intervention from the BS. Since a duplicate RB selection results in transmission failures, the active devices must choose the RBs such that no other device is choosing the same RB. Furthermore, the distribution RB allocation can be modeled as

one-to-one association between the RBs and the devices. The behavior of the devices wanting to choosing an RB alone can be formulated as a minority game [111].

A suitable minority game for distributed RB allocation in an IoT is the Kolkata paise restaurant (KPR) game [105]. The KPR game is a repeated game in which the customers simultaneous go to one of the restaurants, which can only serve one customer each. Additionally, the cost of going to a restaurant is the same for all restaurants, and a customer can only go to one restaurant at any given time. In the KPR game, the players are the customers, whose action in each iteration is to choose one of the restaurants. The payoff of a given player depends on the utility of the chosen restaurants and the number of players choosing the same restaurant. Given players, actions, and payoffs in a game, one important stable solution is an NE. A vector of actions is an NE if no player can achieve a higher payoff by a unilateral change of action. For the KPR game, the existence of an NE depends on the utilities of the restaurants [105], and an NE is when all customers go to different restaurants and none of the customers have the utility of 0. If an NE exists in the KPR game, it coincides with the socially optimal solution, which is when all restaurants are being utilized.

The fundamental structure of the KPR game can be readily extended for our IoT model. The customers can be modeled as IoT devices, and the restaurants can be modeled as the RBs. Furthermore, the cost of using any of the RBs is same. However, there are significant differences between the KPR game and the IoT game. In the KPR game, the number of customers and the number of restaurants are the same, and each customer goes to one of the restaurants at every iteration. In the IoT game, the number of devices N and the number of RBs R may not be the same, and not all devices are active and need to use the RBs at each time slot. The most significant difference is the payoff in the case of duplicate RB selection. When multiple customers choose the same restaurant in the KPR game, one of those customers is randomly chosen to get the full payoff, while other customers with duplicate selection get a zero payoff. However, in the IoT game, all devices that choose the same RB get a zero payoff because of the transmission failures.

For our AoI minimization, the players are the N IoT devices, and their action is to transmit using the RBs or not to transmit. We let $\mathcal{R}$ be the set of R RBs and let $x_i(t)$ be the action of device i at time slot t. If $x_i(t) \in \mathcal{R}$, then device i transmits using the RB in $x_i(t)$ at time slot t. If $x_i(t) = 0$, then device i does not transmit at time slot t. at each time slot, the payoff of each device depends on the actions of all devices. If a device transmits successfully by using an RB alone, then the payoff is ρ. Furthermore, a successful transmission using any of the RBs has the same payoff of ρ. If a device transmits unsuccessfully due to a duplicate RB usage, then the payoff is $-\gamma$. The transmission failure has a negative payoff, because the energy is consumed for the transmission without success. Additionally, the transmission failure using any of the RBs has the same payoff of $-\gamma$, and ρ and γ are positive numbers such that $\rho > \gamma$.

To minimize the AoI of the devices, active devices with lower AoI must not transmit, while the active devices with high AoI need to transmit. Using a distributed RB allocation scheme, the devices must know the AoI of other devices to determine if their own AoI is high enough to transmit. We assume that the active devices broadcast their own future AoI F_i to other devices within the communication range r_c. Moreover, we assume that device-to-device communication links are orthogonal to the uplink communication. Similar to the overhead related to the RACH uplink request for centralized RB allocation scheme, the communication between the devices to share the AoI can be seen as the overhead for distributed RB allocation scheme. However, depending on r_c, the active devices may not know F_i of all other active devices. We let α_i be the active status of a device i such that $\alpha_i = 0$ implies that device i is inactive and $\alpha_i = 1$ implies that device i is active. We let $\boldsymbol{A}$ be a vector that captures the future AoI F_i of all active devices, and $\boldsymbol{A}_i$ be a vector of the future AoI F_i of the active devices within r_c of an active device i. With r_c sufficiently large, $|\boldsymbol{A}_i| = |\boldsymbol{A}|$ for all devices, where $|\boldsymbol{A}|$ is the cardinality of $\boldsymbol{A}$. For an active device i, if F_i is higher than κ-th highest AoI in $\boldsymbol{A}_i$, then the active device i transmits. κ determines if F_i is sufficiently higher than the future AoI of other active devices, and we let $\boldsymbol{A}_i(\kappa)$ be the κ-th highest AoI in $\boldsymbol{A}_i$. Moreover, κ should ensure that the number of transmitting devices T_t at time slot t is equal to R such that all transmitting devices can be allocated with the RB. If T_t is higher than R, then there are at least two active devices with transmission failure, which causes the average AoI to increase. If T_t is less than R, then there are some of the RBs not used by any of the active devices, which may cause the average AoI to increase. However, $T_t = R$ is difficult to achieve when r_c is not sufficiently large and $|\boldsymbol{A}_i| < |\boldsymbol{A}|$.

When $|\boldsymbol{A}_i| = |\boldsymbol{A}|$, the devices have full information on the future AoI F_i of the active devices. Moreover, the devices know all active devices. Since the devices have full information of F_i, an active device i transmits if $F_i \geq \boldsymbol{A}_i(R)$, and

an active device i does not transmit if $F_i < \boldsymbol{A}_i(R)$. Therefore, $\kappa = R$ when $|\boldsymbol{A}_i| = |\boldsymbol{A}|$ for all i. This ensures that R active devices transmit, while $|\boldsymbol{A}| - R$ devices do not transmit. Therefore, the number of transmitting devices T_t at time slot t is equal to R. To formulate this decision to transmit or not to transmit into the payoff, the payoff y_{full} when an active device i does not transmit is:

$$y_{\text{full}}(\boldsymbol{A}_i) = (\rho + \eta)\theta_{\mathbb{R}_+}(\boldsymbol{A}_i(R) - F_i - \eta) - (\gamma + \eta)\theta_{\mathbb{R}_+}(F_i - \boldsymbol{A}_i(R)), \tag{4.13}$$

where η is a real number in $(0, 1)$ and $\theta_{\mathbb{R}_+}$ is an indicator function such that:

$$\theta_{\mathbb{R}_+}(x) = \begin{cases} 1 & \text{if } x \in [0, \infty), \\ 0 & \text{if } x \notin [0, \infty). \end{cases} \tag{4.14}$$

It is important to note that $y_{\text{full}}(\boldsymbol{A}_i)$ does not depend on actions of the players, because the decision to transmit or not to transmit only depends on future AoI. Given the payoff in (4.13), the R active devices with $F_i \geq \boldsymbol{A}_i(R)$ transmit, because their payoff of not transmitting is $-(\gamma + \eta)$. The $|\boldsymbol{A}| - R$ active devices with $F_i < \boldsymbol{A}_i(R)$ do not transmit, because their payoff of not transmitting is $\rho + \eta$. Therefore, IoT devices with R highest future AoI transmit, while other devices do not transmit.

In a more realistic scenario where IoT devices do not have full information of F_i, $|\boldsymbol{A}_i| < |\boldsymbol{A}|$, and the devices do not know all active devices. It is difficult to make only R active devices to transmit at time slot t, and, thus, κ is designed to make $T_t \approx R$. For the case of $|\boldsymbol{A}_i| < |\boldsymbol{A}|$, κ is:

$$\kappa = \left\lceil \frac{R}{|\boldsymbol{A}|} |\boldsymbol{A}_i| \right\rceil, \tag{4.15}$$

where $\kappa = R$ in the case of $\boldsymbol{A} = \boldsymbol{A}_i$. However, with partial information, the active devices do not know $\boldsymbol{A}$ in (4.15). $|\boldsymbol{A}|$ is the number of active devices, and the expected number of newly active devices is Nv_a. However, with previously active devices yet to transmit successfully, $|\boldsymbol{A}|$ is typically greater than Nv_a. Therefore, in (4.15), $|\boldsymbol{A}|$ can be estimated with $Nv_a\zeta$, where ζ is a design parameter to consider the number of previously active devices yet to transmit successfully. With higher values of ζ, the number of transmitting devices T_t at time slot t is smaller, while T_t is bigger with smaller values of ζ. Therefore, with approximation for $|\boldsymbol{A}|$, κ is:

$$\kappa = \left\lceil \frac{R}{Nv_a\zeta} |\boldsymbol{A}_i| \right\rceil. \tag{4.16}$$

For an active device i, κ in (4.16) approximates if F_i is sufficiently higher than the future AoI of other active devices based on the percentile of F_i on the known vector of future AoI $\boldsymbol{A}_i$. With $|\boldsymbol{A}_i| < |\boldsymbol{A}|$, the payoff y_{act} when an active device i does not transmit is:

$$y_{\text{act}}(\boldsymbol{A}_i) = (\rho + \eta)\theta_{\mathbb{R}_+}(\boldsymbol{A}_i(\kappa) - F_i - \eta) - (\gamma + \eta)\theta_{\mathbb{R}_+}(F_i - \boldsymbol{A}_i(\kappa)). \tag{4.17}$$

It is important to note that $y_{\text{act}}(\boldsymbol{A}_i)$ is equal to $y_{\text{full}}(\boldsymbol{A}_i)$, when the active devices have full information with $|\boldsymbol{A}_i| = |\boldsymbol{A}|$ and $\kappa = R$. Similar to $y_{\text{full}}(\boldsymbol{A}_i)$ (4.13), with the payoff $y_{\text{act}}(\boldsymbol{A}_i)$, the active devices with sufficiently high F_i satisfying $F_i \geq \boldsymbol{A}_i(\kappa)$ transmit, while the active devices with F_i such that $F_i < \boldsymbol{A}_i(\kappa)$ do not transmit. Moreover, $y_{\text{act}}(\boldsymbol{A}_i)$ also does not depend on actions of the players.

In the IoT game, the payoff needs to consider the inactive devices. The inactive devices with $\alpha_i = 0$ do not transmit as they do not have messages to transmit. The payoff y_{nt} when an device i does not transmit is:

$$y_{\text{nt}}(\boldsymbol{A}_i, \alpha_i) = (1 - \alpha_i)(\rho + \eta) + \alpha_i y_{\text{act}}(\boldsymbol{A}_i). \tag{4.18}$$

With payoff $y_{\text{nt}}(\boldsymbol{A}_i, \alpha_i)$ for not transmitting, the inactive devices with $\alpha_i = 0$ do not transmit as they get payoff of $\rho + \eta$. The active devices with $\alpha_i = 1$ decide to transmit or to not transmit based on κ and future AoI. We let

Table 4.1: IoT game with $N = R = 2$.

	$x_1(t) = 1$	$x_1(t) = 2$	$x_1(t) = 0$
$x_2(t) = 1$	$(-\gamma, -\gamma)$	(ρ, ρ)	$(-(\gamma + \eta), \rho)$
$x_2(t) = 2$	(ρ, ρ)	$(-\gamma, -\gamma)$	$(-(\gamma + \eta), \rho)$
$x_2(t) = 0$	$(\rho, -(\gamma + \eta))$	$(\rho, -(\gamma + \eta))$	$(-(\gamma + \eta), -(\gamma + \eta))$

$\boldsymbol{x}(t) = [x_1(t), x_2(t), \cdots, x_N(t)]$ be a vector of all actions of N devices at time slot t. For a given $\boldsymbol{x}(t)$, the payoff function $y_i(\boldsymbol{x}(t), \boldsymbol{A}_i, \alpha_i)$ for a device i at time slot t is:

$$y_i(\boldsymbol{x}(t), \boldsymbol{A}_i, \alpha_i) = \begin{cases} \rho & \text{if } x_i(t) \neq x_j(t) \; \forall \; j \neq i, x_i(t) \neq 0, \\ -\gamma & \text{if } \exists \; j \neq i \text{ s.t. } x_j(t) = x_i(t) \neq 0, \\ y_{\text{nt}}(\boldsymbol{A}_i, \alpha_i) & \text{if } x_i(t) = 0. \end{cases} \tag{4.19}$$

In a simple game where $N = R = 2$ with $v_a = 1$, the payoffs of two devices at each time slot is summarized in Table 4.1. NE in this IoT game is $\boldsymbol{x}(t) = [1, 2]$ or $\boldsymbol{x}(t) = [2, 1]$. For simple IoT game, NE is when two devices choose different RBs and get the payoff of ρ. If one device deviates from NE and the other device does not deviate from NE, the device deviating from NE gets the lower payoff of $-\gamma$ or $-(\gamma + \eta)$. Furthermore, in the IoT game, NE implies that a duplicate RB selection does not occur. An NE for a more general case of the IoT game can be found with certain conditions.

Theorem 3. *For the IoT game with N players with action $x_i(t) \in \{0, \mathcal{R}\}$, payoff function $y_i(\boldsymbol{x}(t))$, and $|\boldsymbol{A}_i| = |\boldsymbol{A}|$ for all i, any vector of actions $\boldsymbol{x}(t)$ such that at most R active devices with $F_i \geq \boldsymbol{A}_i(R)$ transmit, the rest of the devices do not transmit, and each of the RBs is used by at most one device is an NE.*

Proof. See Appendix 4.7.2. □

There are many sets of actions that satisfy the conditions described in Theorem 3, and, thus, NE in the IoT game is not unique. For instance, when the number of devices is equal to the number of RBs with $v_a = 1$ in an IoT, an NE is when each RB is used by one device, and, thus, the number of NEs in that particular IoT game is $N!$. There are many NEs, because the payoff of successful transmission does not depend on which RB is used. *Although there are many NEs, the expected payoffs of devices at any given NE are the same, and, thus, one of those NEs is chosen with distributed RB allocation algorithm discussed in Section* 4.4.2. Similar to the simple IoT game in Table 4.1, an NE for our IoT game implies that the number of transmitting devices T_t is equal to the number of RBs R and that a duplicate RB selection does not occur. Furthermore, at an NE, a transmitting device has one of the R highest AoI, and a device that is not transmitting is inactive or has low AoI. This is because the payoff function with $|\boldsymbol{A}_i| = |\boldsymbol{A}|$ is designed to only allow the messages having the R highest AoI to transmit. Therefore, the convergence of a distributed RB allocation algorithm to an NE reduces the average AoI. However, when the devices only have partial information such that $|\boldsymbol{A}_i| < |\boldsymbol{A}|$, $\boldsymbol{x}(t)$ described in the Theorem 3 is not necessarily an NE, because κ is not necessarily equal to R.

In addition to the NE, another solution concept is a socially optimal solution in which the overall payoff of an IoT game is maximized. In other words, a vector of actions is a socially optimal solution when the sum of payoffs of all devices is maximized. In an IoT game, a socially optimal solution is when the devices with highest future AoI F_i fully utilize RBs without any duplicate RB selection. Therefore, similar to an NE in KPR game [105], an NE in IoT game coincides with a socially optimal solution. A performance metric that can be used to describe an NE and a socially optimal solution is a service rate s_r, which is the percentage of RBs that are used by one device. NE in Theorem 3 has a service rate of 1, which implies that all RBs are used by one device, or the highest possible service rate of T_t/R.

With the number of transmitting devices T_t approximately equal to R via IoT game design, a distributed RB allocation algorithm is necessary to enable the transmitting devices to share RBs autonomously. Moreover, with existence of a

Algorithm 4.2 SCA for device i at time t.

Receive $\boldsymbol{X}_i$, $\boldsymbol{P}_i$, $\boldsymbol{A}_i$, and $\mathcal{L}$.
if $x_i(t-1) \in \mathcal{R}$, $y_i(\boldsymbol{x}(t-1)) = \rho$, and $\alpha_i = 1$,
  $x_i(t) \leftarrow x_i(t-1)$.
else if $x_i(t-1) \in \mathcal{R}$, $y_i(\boldsymbol{x}(t-1)) = \rho$, and $\alpha_i = 0$,
  $j \leftarrow$ one neighboring device chosen with $\frac{F_j}{\sum_{F_h \in \boldsymbol{A}_i} F_h}$.
  $x_j(t) \leftarrow x_i(t-1)$.
else if $x_i(t-1) \in \mathcal{R}$ and $y_i(\boldsymbol{x}(t-1)) = -\gamma$,
  $z \leftarrow 1$ with probability $\boldsymbol{X}_i(x_i(t-1))^{-1}$.
  if $z = 1$, $x_i(t) \leftarrow x_i(t-1)$.
  else $x_i(t) \leftarrow$ randomly chosen from $\mathcal{L}$. **end if**.
else if $x_i(t-1) = 0$,
  $x_i(t) \leftarrow$ randomly chosen from $\mathcal{L}$.
end if.

socially optimal NE in our IoT game, the convergence of a distributed RB allocation algorithm to an NE is crucial in minimizing the average AoI. To evaluate different RB allocation algorithms, the service rates are compared. The service rate is an important metric to determine convergence to an NE, because a vector of actions must achieve service rate of 1 or highest service rate to be an NE. Furthermore, the service rate is an important performance metric for the AoI, because the higher service rate implies that more devices are transmitting successfully at each time slot, reducing the average AoI. Therefore, a distributed RB allocation algorithm must achieve a service rate of 1 or increase the service rate as high as possible, because a high service rate is required to achieve a low average AoI.

Stochastic Crowd Avoidance

We propose a stochastic crowd avoidance (SCA) algorithm that enables the devices to avoid using the RBs that are used by many devices stochastically and to choose an RB for a successful transmission. For SCA algorithm, the devices need to share more information in addition to their F_i and can perform a channel sensing to determine the RBs that were not used at previous time slot. At time slot t, the devices that transmitted at the time slot $t-1$ share their previous actions $x_i(t-1) \in \mathcal{R}$ and the previous payoff $y_i(\boldsymbol{x}(t-1))$ of their transmission. We let $\boldsymbol{X}_i$ be the vector of actions of the transmitting devices at time slot $t-1$ that a device i knows and $\boldsymbol{P}_i$ be the vector of payoffs of the transmitting devices at time slot $t-1$ that a device i knows. Furthermore, we let $\boldsymbol{X}_i(x)$ with $x \in \mathcal{R}$ be the number of x in $\boldsymbol{X}_i$. In other words, $\boldsymbol{X}_i(x)$ is the number of devices that chose the RB x at time slot $t-1$ that a device i knows. Learning from $\boldsymbol{X}_i$ and $\boldsymbol{P}_i$, the proposed SCA algorithm enables a transmitting device i at time slot t to not use the RBs that are being used successfully by other devices and to avoid using the contended RBs stochastically.

Using SCA algorithm, at time slot t, a transmitting device i determines its RB usage based on $x_i(t-1)$ and $y_i(\boldsymbol{x}(t-1))$. If the transmission at time slot $t-1$ is successful such that $x_i(t-1) \in \mathcal{R}$ and $y_i(\boldsymbol{x}(t-1)) = \rho$, then transmitting device i uses the same RB $x_i(t) = x_i(t-1)$. If the transmission at time slot $t-1$ is unsuccessful such that $x_i(t-1) \in \mathcal{R}$ and $y_i(\boldsymbol{x}(t-1)) = -\gamma$, then transmitting device i uses the same RB $x_i(t) = x_i(t-1)$ with probability $\boldsymbol{X}_i(x_i(t-1))^{-1}$ or chooses an RB from a set $\mathcal{L}$ uniformly randomly. $\mathcal{L}$ is a set of the RBs that were not used by any of the device at time slot $t-1$ determined with channel sensing. If there is no transmission at time slot $t-1$ such that $x_i(t-1) = 0$, then the transmitting device i chooses an RB from $\mathcal{L}$ uniformly randomly. In the case where $v_a < 1$, a device that transmitted successfully at time slot $t-1$ may no longer be active at time slot t. In this case, a device i with $\alpha_i = 0$ and $y_i(\boldsymbol{x}(t-1)) = \rho$ chooses a neighboring device j with $\alpha_j = 1$ with a probability proportional to F_j, such that the

probability is ${}^{F_j}/\sum_{F_h \in \boldsymbol{A}_i} F_h$.

In our SCA algorithm, the RB that is used successfully at time slot $t-1$ is also used successfully at time slot t if $v_a = 1$ or if there is an active neighboring device. Moreover, with r_c sufficiently large, the expected number of transmitting devices using an RB that is used by more than one device at the previous time slot $t-1$ is 1 at the current time slot t. This is because the probability of choosing the same RB after a transmission failure is ${}^{1}/\boldsymbol{X}_{i(x_i(t-1))}$. The devices avoid to use the same RB stochastically even after a transmission failure, because the strict crowd avoidance causes the crowding in other RBs resulting in more transmission failures. Furthermore, when a device i chooses some other RB such that $x_i(t) \neq x_i(t-1)$, device i chooses from a set of RBs $\mathcal{L}$ that were not used at time slot $t-1$. This is to avoid using the RBs that are either used successfully or crowded, both of which cause the transmission failures. With SCA algorithm design to avoid duplicate RB selection, next we prove that the proposed SCA algorithm converges to an NE under certain IoT system parameters.

Theorem 4. *When N devices are always active with full information and use 1 out of N RBs to transmit with negligible outage probability $p_{i,t}$ at each time slot, the vector of actions $\boldsymbol{x}(t)$ converges to an NE using SCA.*

Proof. See Appendix 4.7.3. □

Under the conditions in Theorem 4, $\boldsymbol{x}(t)$ converges to an NE, and this implies that the service rate increases to 1. In general, SCA algorithm increases the service rate, because an RB, which is used by 1 device at previous time slot $t-1$, is still used by 1 device at current time slot t. Therefore, SCA algorithm is effective in reducing the average AoI. However, SCA is susceptible to the high outage probability $p_{i,t}$ based on the SNR. This is because SCA cannot distinguish the transmission failure due to the duplicate RB selection and the transmission failure due to the outage based on the SNR. Furthermore, with partial information $|\boldsymbol{A}_i| < |\boldsymbol{A}|$, the devices also have partial information of $\boldsymbol{X}_i$ and $\boldsymbol{P}_i$, and, thus, the devices cannot choose $x_i(t)$ accurately in the case of transmission failure.

In a massive IoT with $N > R$ and partial information, the vector of actions $\boldsymbol{x}(t)$ does not converge to an NE using SCA, because it is not possible to achieve service rate of 1. The service rate cannot be 1, because the transmitting devices are changing every time slot and a duplicate RB selection is inevitable. Since the service rate cannot be 1, the average AoI in a massive IoT is higher than the average AoI in an ideal IoT described in Theorem 4. However, in a massive IoT, the proposed SCA algorithm still enables the transmitting devices to stochastically avoid duplicate RB selection with available information. Therefore, the proposed SCA algorithm still increases the service rate and reduces the average AoI. However, in that case, it does not reach an NE but rather a sub-optimal, heuristic solution. To evaluate performance of the proposed SCA algorithm, next we study a random RB selection for distributed RB allocation scheme.

Random RB Selection

One way for the transmitting devices to determine their RB usage is via random selection. In other words, the actions $x_i(t)$ of the transmitting devices are chosen uniformly random in $\mathcal{R}$. This random RB selection is used as a baseline. Even with $|\boldsymbol{A}_i| = |\boldsymbol{A}|$ for all i, the random RB selection is highly unlikely to achieve $\boldsymbol{x}(t)$ such that each of the RBs is used by at most one device, which is the requirement of $\boldsymbol{x}(t)$ to be an NE. Furthermore, the service rate s_r using random RB selection for the transmitting devices is low.

Proposition 2. At a time slot t, the service rate s_r with T_t transmitting devices using random RB selection is:

$$s_r = \frac{T_t}{R}\left(\frac{R-1}{R}\right)^{T_t-1}, \tag{4.20}$$

and, for a massive IoT with N increasing to infinity, the service rate s_r is:

$$\lim_{N\to\infty} s_r = \frac{T_t}{R-1}\exp\left(\frac{-T_t}{R}\right) \tag{4.21}$$

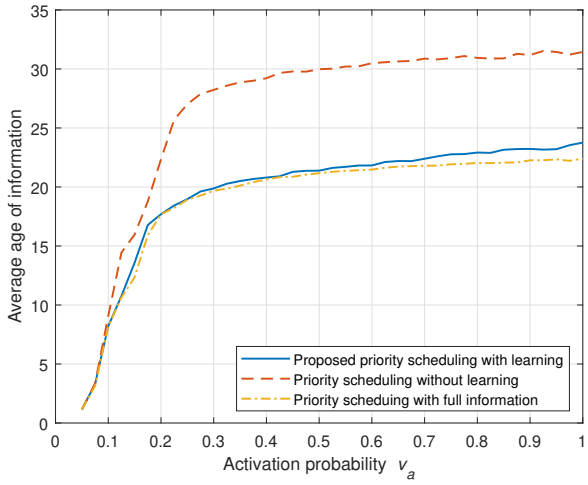


Figure 4.1: Average AoI using centralized RB allocation schemes while varying v_a.

Proof. See Appendix 4.7.4. □

Under the conditions in Theorem 4, when N devices are always active and use 1 out of N RBs, the number of transmitting devices T_t is equal to N. In this case, the service rate using random RB selection is always less than 1 even with $N = R \geq 2$. On the other hand, for a massive IoT with $N > R$, the service rate using random RB selection exponentially decreases to 0 as N increases. Therefore, the random RB selection is not suitable for a massive IoT in which the number of devices N outnumbers the number of RBs R, Furthermore, with low service rate, the probability of a successful transmission is low for a device, and, thus, the average AoI is high.

4.5 Simulation Results and Analysis

For our simulations, we consider a rectangular area with width w and length l within which the N devices are deployed following a Poisson point process. We let $w = l = 10$ m and $R = 50$ with a 10 MHz frequency band [112], while the number of devices N will be varied for analysis. We choose a time slot duration of 1 ms [113] and expected value of SNR of 20 dB with $\lambda = .1$ and $\sigma^2 = 0.001$. To vary outage probability $p_{i,t}$, different values of ϵ are used. Moreover, a device is assumed to be of type 1 with probability 0.6 with $m_1 = 0.75$, while a device is assumed to be of type 2 with probability 0.4 with $m_2 = 0.75$. The average of the current AoI C_i at the time slot of successful transmission is the performance metric for different RB allocation schemes, and their performances are analyzed with varying v_a and $p_{i,t}$.

For the centralized RB allocation scheme, the number of RACH preambles P for the uplink transmission request is 64 [70]. Three different kinds of priority scheduling are analyzed. Priority scheduling without learning does not learn the device types, while priority scheduling with learning learns the device types using maximum likelihood and information on F_i. Priority scheduling with full information assumes that the BS knows the types of all devices all the time. All three priority scheduling algorithms are analyzed with $N = 500$, while varying v_a and $p_{i,t}$.

Fig. 4.1 shows the average AoI of the devices using centralized RB allocation schemes for different values of the activation probability v_a with $p_{i,t} = 0.01$ and $\epsilon = 1$. The average AoI for the no learning case quickly increases to 27.88 and then increases slowly above 30, while the average AoI for both learning and full information quickly increases

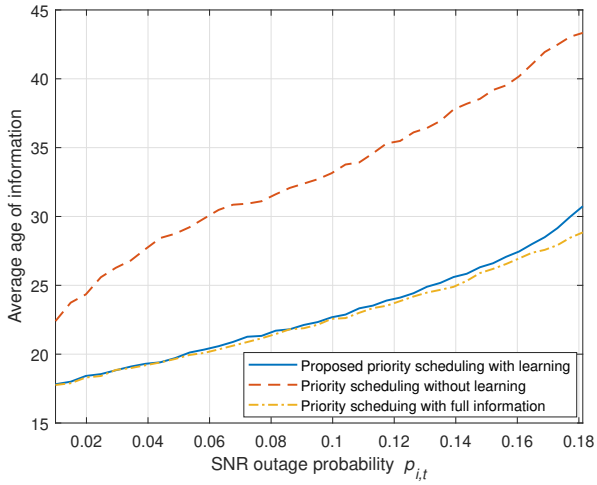


Figure 4.2: Average AoI using centralized RB allocation schemes while varying $p_{i,t}$.

to 19.62 and then increases slowly above 20. As v_a increases, the average AoI increases for all priority scheduling algorithms. However, for $v_a > 0.2$, the average AoI flattens and increases at a much slower rate with increasing v_a for all priority scheduling algorithms, because all RBs are fully saturated. Moreover, there is a significant difference between priority scheduling with learning and without learning. After the average AoI flattens, the difference of the average AoI between priority scheduling with learning and without learning is constantly about 8. This implies that the proposed priority scheduling scheme with learning achieves about 26.7% lower average AoI when compared to simple priority scheduling scheme. Hence, learning the device types is important to decrease the average AoI for priority scheduling. However, there is an insignificant difference between priority scheduling with learning and with full information, which implies that the learning is effective in learning the device types.

Fig. 4.2 shows the average AoI of the devices using centralized RB allocation schemes for different values of the SNR outage probability $p_{i,t}$ with $v_a = 0.2$ and varying ϵ from 1 to 20. Unlike in Fig. 4.1, the average AoI does not flatten and increases at about the same rate. With $p_{i,t} = 0.02$, the average AoI without learning is 24.34, and the average AoI with learning is 18.41. With $p_{i,t} = 0.16$, the average AoI without learning is 40.17, and the average AoI with learning is 27.45. Therefore, the difference between the average AoI with learning and without learning increases as $p_{i,t}$ increases. Furthermore, with high $p_{i,t}$, the proposed priority scheduling scheme with learning achieves about 31.7% lower average AoI when compared to simple priority scheduling scheme. For a higher $p_{i,t}$ and more frequent transmission failures, the AoI of the exponentially aging messages becomes much higher than the AoI of the linearly aging messages. In this case, the learning scheme, which enables the BS to accurately identify the messages aging faster, becomes more crucial in reducing the average AoI. Furthermore, the difference between the average AoI with learning and with full information also increases as $p_{i,t}$ increases. This is because it becomes increasingly difficult to learn the device types as $p_{i,t}$ increases.

For the distributed RB allocation scheme, the communication range r_c determines the information $\boldsymbol{A}_i$ that the devices have. For the given dimensions of the deployment area, $r_c \geq 15$ m is sufficiently large such that $|\boldsymbol{A}_i| = |\boldsymbol{A}|$ for any device i. SCA algorithm is compared against two baselines, which are the random RB selection and the predetermined RB selection. The predetermined RB selection scheme is known as the dictator's solution in the KPR game, and the RB usage for a device i is predetermined based on the rank of F_i. For instance, if F_i is κ-th highest in $\boldsymbol{A}_i$ for $\kappa \leq R$, device i uses a specific RB as previously agreed among IoT devices. The predetermined RB allocation scheme requires full information $|\boldsymbol{A}_i| = |\boldsymbol{A}|$ for all i and always achieves a service rate s_r of 1. To analyze different distributed RB allocation schemes, the activation probability v_a, the SNR outage probability $p_{i,t}$, and the communication range r_c are varied. In addition to the average AoI, the service rate s_r is evaluated for different distributed RB allocation schemes.

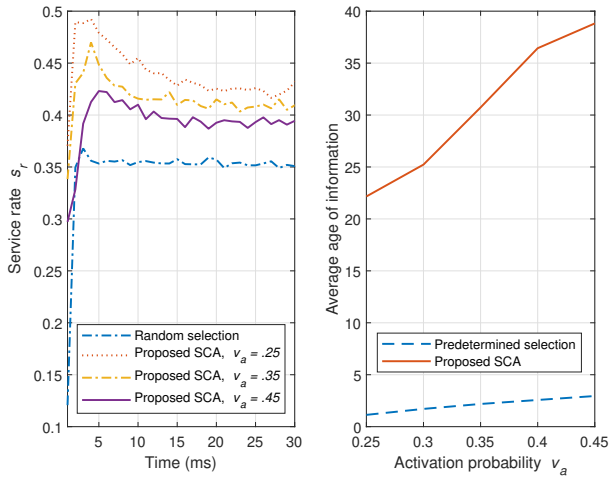


Figure 4.3: Average AoI and service rate using distributed RB allocation schemes while varying v_a.

Fig. 4.3 shows the average AoI and the service rate of the devices using distributed RB allocation schemes for different values of the activation probability v_a with $p_{i,t} = 0.01$, $\epsilon = 1$, $r_c = 10$ m, and $N = 200$. It is important to note that the expected number of newly active devices at a given time slot is Nv_a, and, thus, the number of active devices N_t outnumbers the number of RBs R for high values of v_a, simulating a massive IoT. Moreover, with $r_c = 10$ m, SCA only has partial information such that $|\boldsymbol{A}_i| < |\boldsymbol{A}|$ for all i. The service rate converges to 0.42 for SCA with $v_a = 0.25$, 0.41 for SCA with $v_a = 0.35$, 0.39 for SCA with $v_a = 0.45$, and 0.35 for random RB allocation with $v_a = 0.35$. As v_a increases from 0.25 to 0.45, the average AoI increases from 22.16 to 38.82 using SCA. As N_t increases with increasing v_a, the transmission failure is more likely to occur due to the duplicate RB selection, because R is fixed. Therefore, as v_a increases, s_r decreases, and the average AoI increases. The predetermined RB allocation scheme achieves a much lower average AoI compared to SCA, because the predetermined RB allocation scheme requires and uses the full information. Furthermore, the average AoI with random RB allocation is multiple orders of magnitude higher than the average AoI with SCA. Therefore, in a massive IoT with partial information, the proposed SCA algorithm is the most suitable algorithm to achieve low average AoI as it balances between having full information and performing arbitrary allocations.

Fig. 4.4 shows the average AoI and the service rate of the devices using distributed RB allocation schemes for different values of the SNR outage probability $p_{i,t}$ with $r_c = 10$ m, $v_a = 1$, $N = R = 50$, and varying ϵ from 1 to 5. It is important to note that the number of active devices N_t is equal to R with $N = R$ and $v_a = 1$, and this is the condition considered in Theorem 4. The service rate converges to 0.98 for SCA with $p_{i,t} = 0.01$, 0.92 for SCA with $p_{i,t} = 0.03$, 0.88 for SCA with $p_{i,t} = 0.05$, and 0.37 for random RB allocation. As $p_{i,t}$ increases from 0.01 to 0.05, the average AoI increases from 1.25 to 4.14 using SCA, while the average AoI increases from 5.73 to 7.64 using random RB allocation. With high $p_{i,t}$, the proposed SCA algorithm achieves about 45.8% lower average AoI when compared to random RB allocation. Since $T_t = N$ with $v_a = 1$ and $N = R$, the theoretical value of s_r (4.20) for random RB allocation case matches the simulated value of s_r in Fig. 4.4. As $p_{i,t}$ increases, the converged value of s_r for SCA decreases, because the proposed SCA assumes that the transmission failures are caused by duplicate RB selection. Therefore, using SCA, a device i stochastically avoids to use an RB even when there was no duplicate RB selection and the transmission failure is caused by SNR outage. Furthermore, the difference between the average AoI using SCA and random RB allocation decreases as $p_{i,t}$ increases, because increasing $p_{i,t}$ has a more negative impact on SCA than on random RB allocation. However, it is important to note that SCA with low $p_{i,t}$ converges quickly to the service rate of 1 with low $p_{i,t}$ as discussed in Theorem 4.

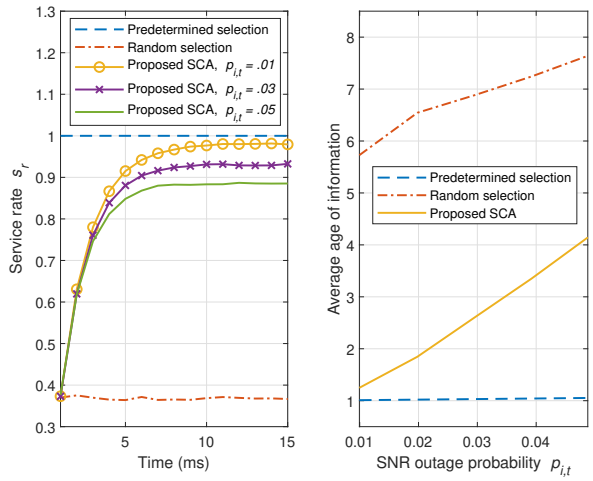


Figure 4.4: Average AoI and service rate using distributed RB allocation schemes while varying $p_{i,t}$.

Fig. 4.5 shows the average AoI and the service rate of the devices using distributed RB allocation schemes for different values of the communication range r_c with $p_{i,t} = 0.01$, $\epsilon = 1$, $v_a = 1$, and $N = R = 50$. The communication range r_c determines the amount of information $\boldsymbol{A}_i$ that the devices have, and $r_c = 15$ m implies that the devices have full information $|\boldsymbol{A}_i| = |\boldsymbol{A}|$ for all i. The service rate converges to 0.981 for SCA with $r_c = 15$ m, 0.978 for SCA with $r_c = 10$ m, 0.967 for SCA with $r_c = 5$ m, 0.939 for SCA with $r_c = 1$ m, and 0.374 for random RB allocation. As r_c increases from 2 m to 15 m, the average AoI decreases from 2.62 to 1.21 using SCA, while the average AoI decreases from 7.19 to 5.07 using random RB allocation. With low r_c, the proposed SCA algorithm achieves about 63.6% lower average AoI when compared to random RB allocation. Similar to the Fig. 4.4. the theoretical and simulated values of s_r for random RB allocation are matched. As r_c increases, the convergence value of s_r for SCA increases, because the devices have more information $\boldsymbol{A}_i$ with higher r_c. As r_c increases sufficiently such that $|\boldsymbol{A}_i| = |\boldsymbol{A}|$ for all i, the service rate converges to 1 as discussed in Theorem 4. However, SCA with only partial information can still achieve s_r close to 1. Moreover, as r_c increases, the average AoI using SCA converges to the average AoI using predetermined RB allocation scheme.

Next, the proposed centralized and distributed RB allocation schemes are compared in a massive IoT with $N > R$ and in an ideal IoT described in Theorem 4. To analyze different RB allocation schemes, the activation probability v_a and the SNR outage probability $p_{i,t}$ are varied,

Fig. 4.6 shows the average AoI of the devices using centralized and distributed RB allocation schemes for different values of the activation probability v_a with $p_{i,t} = 0.01$, $\epsilon = 1$, and $N = 200$. This simulates a massive IoT as the number of devices N greatly outnumbers the number of RBs R. As v_a increases, the average AoI converges to 200 for SCA with $r_c = 5$ m, 157 for SCA with $r_c = 10$ m, and 145 for SCA with $r_c = 15$ m. On the other hand, for priority scheduling, the average AoI increases from 1.14 to 6.03 as v_a increases from 0.1 to 0.5. With high v_a, the proposed SCA algorithm with $r_c = 15$ m achieves about 24-fold higher average AoI when compared to the proposed priority scheduling with learning. For almost any values of v_a, centralized RB allocation with priority scheduling performs much better than distributed RB allocation with SCA in terms of the average AoI. However, centralized RB allocation scheme requires the BS to dictate the RB allocation for all devices, and, thus, centralized RB allocation scheme may not be viable for some of the IoT. Moreover, similar to Fig. 4.1, the average AoI flattens after a certain value of v_a, because all RBs are fully saturated. There is a performance gap between SCA with different values of r_c, because r_c is directly related to the amount of information that the devices have. With higher r_c and more information for the devices, SCA is more effective in reducing the average AoI.

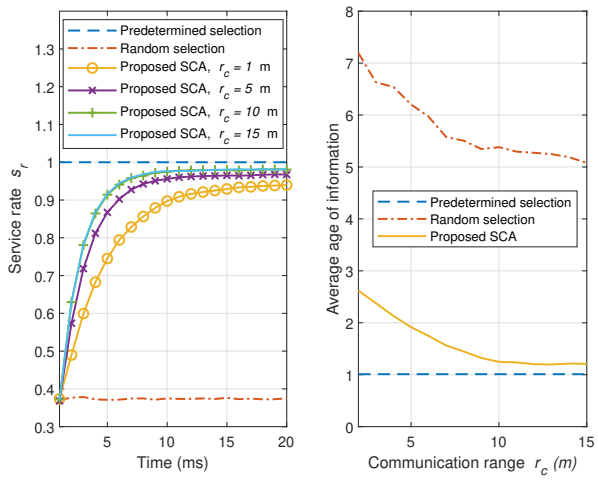


Figure 4.5: Average AoI and service rate using distributed Rb allocation schemes while varying r_c.

Fig. 4.7 shows the average AoI of the devices using centralized and distributed RB allocation schemes for different values of the SNR outage probability $p_{i,t}$ with $v_a = 0.5$, $N = 200$, and varying ϵ from 1 to 20. Similar to Fig. 4.6, this simulates a massive IoT. When $p_{i,t} = 0.18$. the average AoI is 585.59 using SCA with $r_c = 5$ m, 529.94 using SCA with $r_c = 10$ m, 472.81 using SCA with $r_c = 15$ m, and 11.90 using priority scheduling. With high $p_{i,t}$, the proposed SCA algorithm with $r_c = 15$ m achieves about 40-fold higher average AoI when compared to the proposed priority scheduling with learning. Similar to Fig. 4.6, centralized RB allocation with priority scheduling performs much better than distributed RB allocation with SCA in terms of the average AoI for all values of $p_{i,t}$. It is interesting to note that the difference in the average AoI between SCA algorithms increases as $p_{i,t}$ increases. This implies that SCA with less information is more severely affected by increasing $p_{i,t}$ than SCA with more information.

Fig. 4.8 shows the average AoI of the devices using centralized and distributed RB allocation schemes for different values of the SNR outage probability $p_{i,t}$ with $v_a = 1$, $N = R = 50$, and varying ϵ from 1 to 20. This simulates an ideal IoT for SCA as some of the conditions for NE convergence in Theorem 4 are satisfied. When $p_{i,t} = 0.18$. the average AoI is 25.53 using SCA with $r_c = 5$ m, 19.96 using SCA with $r_c = 10$ m, 17.27 using SCA with $r_c = 15$ m, and 2.21 using priority scheduling. With high $p_{i,t}$, the proposed SCA algorithm with $r_c = 15$ m achieves about 8-fold higher average AoI when compared to the proposed priority scheduling with learning. It is interesting to note that even in an ideal IoT for SCA, priority scheduling with learning performs better than SCA in terms of the average AoI for most values of $p_{i,t}$. Moreover, similar to Fig. 4.7, the difference in the average AoI between SCA algorithms increases as $p_{i,t}$ increases.

From our simulations, we observe that both priority scheduling and SCA are susceptible to high SNR outage probability $p_{i,t}$ as the average AoI increases without flattening as $p_{i,t}$ increases. This is because the SNR outage probability is directly related to the transmission failures. However, the average AoI increases slowly after a certain value of the activation probability v_a, because the RBs are fully saturated. Since increasing v_a with fixed N is equivalent to increasing N with fixed v_a, the average AoI also flattens for the case in which only the number of devices N increases. Although centralized RB allocation scheme outperforms distributed RB allocation scheme in most cases, SCA can still achieve a high service rate s_r and low average AoI only with partial information. Furthermore, communication range r_c and information availability are critical to the performance of SCA.

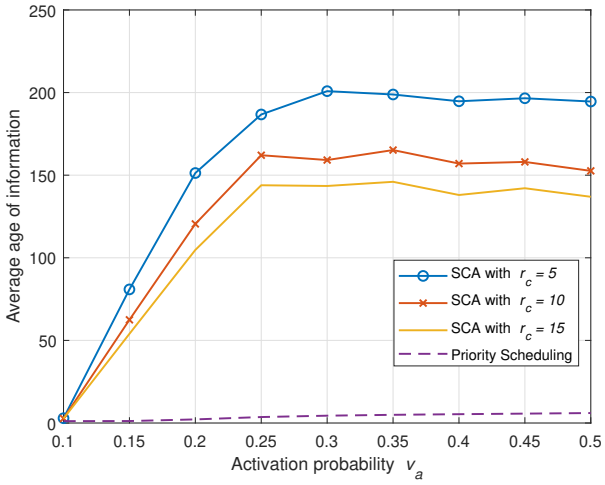


Figure 4.6: Average AoI using centralized and distributed RB allocation schemes while varying v_a in a massive IoT.

4.6 Summary

In this paper, we have proposed centralized and distributed approaches for allocating the limited communication resources based on the aging function and the current AoI of IoT devices. In the presence of both linear and exponential aging functions, we have shown that comparing the future AoI achieves a lower average AoI at the BS than comparing the current AoI. For the centralized approach, we have introduced a priority scheduling scheme with learning, which enables the BS to allocate the limited RBs to the heterogeneous devices based on their future AoI. For the distributed approach, we have formulated the problem of autonomously allocating the limited RBs to the devices using game theory, and we have designed payoff functions to encourage the devices with high AoI to transmit, while discouraging the devices with low AoI to not transmit. Furthermore, we have proposed a novel SCA algorithm using which the heterogeneous devices can allocate the RBs in a self-organizing manner to avoid the duplicate RB selection and to minimize the AoI. We have proved the conditions that a vector of actions in the IoT game must satisfy to achieve an NE. Furthermore, we have proved that the actions of devices using our proposed SCA algorithm converge to an NE, if the devices have sufficient information under certain network parameters. Simulation results have shown that the average AoI is an increasing function of the activation probability and the SNR outage probability. Moreover, the simulation results have shown that the service rate is an increasing function of the communication range and a decreasing function of the activation probability and the SNR outage probability. We have compared our centralized and distributed RB allocation schemes, and we have shown that our centralized RB allocation scheme outperforms our distributed RB allocation scheme in most cases. However, our proposed SCA algorithm has shown to be effective in reducing the AoI and increasing the service rate only with partial information. With high SNR outage probability, the proposed priority scheduling scheme with learning has shown to achieve about 31.7% lower average AoI when compared to simple priority scheduling scheme. Furthermore, with high SNR outage probability, the proposed SCA algorithm has shown to achieve about 45.8% lower average AoI when compared to random RB allocation.

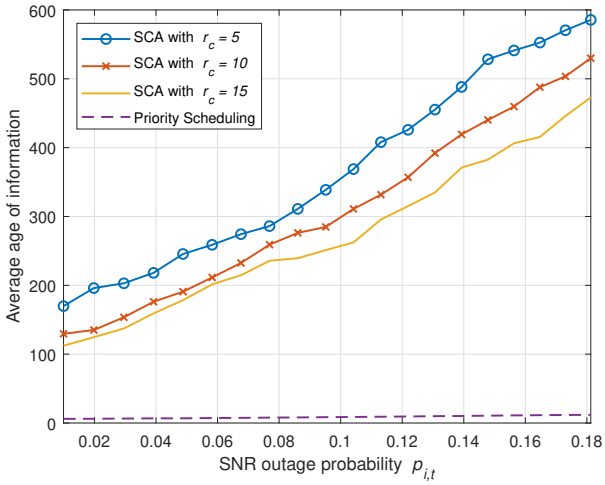


Figure 4.7: Average AoI using centralized and distributed RB allocation schemes while varying $p_{i,t}$ in a massive IoT.

4.7 Appendix

4.7.1 Proof of Proposition 1

Without a loss of generality, let there be an IoT device i with aging function $a_i(t)$ and an IoT device h with aging function $b_h(t)$. Moreover, the proof only considers the case with $N = 2$ and $R = 1$. It is sufficient to only consider $N = 2$, because comparing the AoI of $N > 2$ devices is equivalent to doing pairwise AoI comparison $N(N-1)/2$ times. It is unnecessary to consider the cases of $R = 0$ and $R \geq 2$. If $N = 2$ and $R = 0$, none of the devices can be allocated an RB. If $N = 2$ and $R \geq 2$, all devices can be allocated the RBs. Therefore, the AoI comparison to determine RB allocation is unnecessary.

At time slot τ, the current AoI of devices i and h must be one of the following cases: $a_i(\tau) > b_h(\tau)$, $a_i(\tau) = b_h(\tau)$, or $a_i(\tau) < b_h(\tau)$. If $a_i(\tau) \leq b_h(\tau)$, then $a_i(\tau+\beta) \leq b_h(\tau+\beta)$ for any positive integer β. At time slot τ, the current AoI comparison with $t = \tau$ and the future AoI comparison with $t = \tau+\beta$ are equivalent, because device h is allocated with an RB for both. However, if $a_i(\tau) > b_h(\tau)$, the future AoI of devices i and h can be any of the following cases: $a_i(\tau+\beta) > b_h(\tau+\beta)$, $a_i(\tau+\beta) = b_h(\tau+\beta)$, or $a_i(\tau+\beta) < b_h(\tau+\beta)$. For the case of $a_i(\tau) > b_h(\tau)$ and $a_i(\tau+\beta) \geq b_h(\tau+\beta)$, the current AoI comparison with $t = \tau$ and the future AoI comparison with $t = \tau+\beta$ are equivalent, because device i is allocated with an RB for both.

Comparing the current AoI and the future AoI are different if $a_i(\tau) > b_h(\tau)$ and $a_i(\tau+\beta) < b_h(\tau+\beta)$, because comparing the current AoI allocates the RB to device i, while comparing the future AoI allocates the RB to device h. When one device is allocated an RB at time slot τ and the other device is allocated an RB at time slot $\tau+\beta$, the RB allocation based on current AoI achieves the average AoI of $0.5(a_i(\tau) + 2^\beta b_h(\tau))$, and the RB allocation based on future AoI achieves the average AoI of $0.5(a_i(\tau) + \beta + b_h(\tau))$. For any $\beta \in \mathbb{Z}_+$, comparing the average AoI of two cases is:

$$\frac{a_i(\tau) + 2^\beta b_h(\tau)}{2} > \frac{a_i(\tau) + \beta + b_h(\tau)}{2}, \tag{4.22}$$

$$(2^\beta - 1) b_h(\tau) > \beta, \tag{4.23}$$

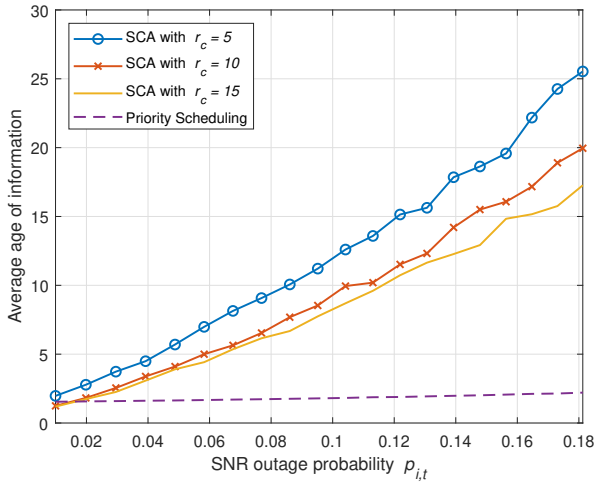


Figure 4.8: Average AoI using centralized and distributed RB allocation schemes while varying $p_{i,t}$ in an ideal IoT.

$$b_h(\tau) > \frac{\beta}{2^{\beta}-1}. \tag{4.24}$$

Even when one device is allocated with an RB at time slot τ and the other device is allocated with an RB one time slot later at time slot $\tau+1$, the current AoI comparison yields higher average AoI than the future AoI comparison. Since $\frac{\beta}{2^{\beta}-1} = 1$ with $\beta = 1$, $b_h(\tau)$ cannot be less than or equal to 1, because the condition of $a_i(\tau) > b_h(\tau)$ and $a_i(\tau+1) < b_h(\tau+1)$ cannot be satisfied. Therefore, at time slot τ, comparing the future AoI with $t = \tau + \beta$ to determine the RB allocation achieves lower average AoI than comparing the current AoI with $t = \tau$.

4.7.2 Proof of Theorem 3

With $\boldsymbol{x}(t)$ satisfying the given conditions, then at most R active devices with sufficiently high values of F_i are transmitting successfully, while rest of the devices are not transmitting. Assuming that all other devices do not change their action, an active device i that is transmitting successfully with sufficiently high F_i cannot change its action $x_i(t)$ to get higher payoff than its current payoff of ρ. If device i uses some other RB, then its transmission may fail due to the duplicate RB usages, getting the payoff of $-\gamma$, or its transmission may succeed, getting the same payoff of ρ. If device i does not transmit, then the payoff is $-(\gamma+\eta)$ as F_i is greater than $\boldsymbol{A}_i(R)$. Therefore, the active devices that are transmitting successfully with sufficiently high F_i do not change their action.

The devices that are not transmitting may be active or inactive. An inactive device that is not transmitting has a payoff of $(\rho+\eta)$, which is higher than the payoff of transmitting successfully. Therefore, the inactive devices do not transmit. With $\boldsymbol{x}(t)$ satisfying the given conditions, then active devices with $F_i < \boldsymbol{A}_i(R)$ are not transmitting. With $y_i(\boldsymbol{x}(t))$, the active devices with $F_i < \boldsymbol{A}_i(R)$ have the payoff of $(\rho+\eta)$, which is higher than the payoff of transmitting successfully. Therefore, the active devices with $F_i < \boldsymbol{A}_i(R)$ do not change their action from not transmitting.

With the design of payoff function $y_i(\boldsymbol{x}(t))$ (4.19), inactive devices and active devices with $F_i < \boldsymbol{A}_i(R)$ have the highest payoff of $(\rho+\eta)$ by not transmitting. Moreover, with $|\boldsymbol{A}_i| = |\boldsymbol{A}|$, at most R active devices with $F_i \geq \boldsymbol{A}_i(R)$ have higher payoff from transmitting successfully than from not transmitting. With $\boldsymbol{x}(t)$ such that each of the RBs is used by at most one device, the active devices with $F_i \geq \boldsymbol{A}_i(R)$ have the highest payoff of ρ by transmitting successfully. Therefore, with $y_i(\boldsymbol{x}(t))$, any vector of actions $(x)(t)$ such that at most R active devices with $F_i \geq$

$\boldsymbol{A}_i(R)$ transmit, rest of the devices do not transmit, and each of the RBs is used by at most one device is an NE.

4.7.3 Proof of Theorem 4

At time slot $t = 1$, $\boldsymbol{x}(t)$ is initialized as a random RB selection. For $N = R$ and $v_a = 1$, $T_t = N$ and the service rate s_r is $\left({}^{(R-1)}/_{R}\right)^{N-1}$. Therefore, from the initial RB allocation, the expected number of RBs that are used by one device is Rs_r. With SCA and $v_a = 1$, an RB that is used by one device at time slot $t = 1$ is used by the same device at time slot $t = 2$. Furthermore, with SCA, the RBs that are used by more than one device at the time slot $t = 1$ are expected to be used by one device at the time slot $t = 2$. Therefore, at time slot $t = 2$, $\mathbb{E}[|\mathcal{L}|] = R\left({}^{(R-1)}/_{R}\right)^{N}$, which is the expected number of RBs that are used by none of the devices at time slot $t = 1$. $\mathbb{E}[|\mathcal{L}|]$ also is the expected number of devices that are competing to use the RBs in $\mathcal{L}$, because the devices use at most 1 RB at each time slot. Since the devices choose the RBs in $\mathcal{L}$ randomly, the RB selection at time slot $t = 2$ is equivalent to the random RB selection with the number of devices and RBs equal to $R\left({}^{(R-1)}/_{R}\right)^{N}$. Moreover, the same analysis done for $t = 1$ can be done with $t = 2$.

Expanding to the general case, at time slot t, the expected number of RBs that are used by none of the devices and the expected number of devices competing to use the RBs in $\mathcal{L}$ is $R\left({}^{(R-1)}/_{R}\right)^{N(t-1)}$. As t increases to infinity, the expected number of RBs that are used by none of the devices and the expected number of devices competing to use the RBs in $\mathcal{L}$ decrease to 0. With SCA, this implies that the number of RBs each used by one device increases to R as t increases to infinity. Furthermore, with $N = R$ and $|\boldsymbol{A}_i| = |\boldsymbol{A}|$ for all i, any active device i satisfies $F_i \geq \boldsymbol{A}_i(R)$. The action space $\boldsymbol{x}(t)$ converged using SCA is such that all R active devices transmit and each of the RBs is used by one device. Therefore, $\boldsymbol{x}(t)$ converged using SCA is an NE.

4.7.4 Proof of Proposition 2

When the T_t transmitting devices use random RB selection, the service rate is equivalent to the probability of an RB being used by only one transmitting device. Therefore, the service rate s_r is:

$$s_r = \binom{T_t}{1}\frac{1}{R}\left(1-\frac{1}{R}\right)^{T_t-1} = \frac{T_t}{R}\left(\frac{R-1}{R}\right)^{T_t-1}. \tag{4.25}$$

When the number of IoT devices N increases to infinity in a massive IoT, the number of transmitting devices T_t also increases to infinity with fixed v_a, and the service rate s_r is:

$$\lim_{T_t\to\infty} s_r = \lim_{T_t\to\infty} \frac{T_t}{R}\left(1+\frac{-{}^{T_t}/_{R}}{T_t}\right)^{T_t-1}, \tag{4.26}$$

$$= \lim_{T_t\to\infty} \frac{T_t}{R-1}\left(1+\frac{-{}^{T_t}/_{R}}{T_t}\right)^{T_t} = \frac{T_t}{R-1}\exp\left(\frac{-T_t}{R}\right). \tag{4.27}$$

Chapter 5

Distributed Resource Management as Kolkata Paise Restaurant Game

5.1 Background, Related Works, and Contributions

In addition to learning approach, to develop a distributed resource management for the IoT devices, a game-theoretic approach can be used to formulate, analyze and solve the distributed resource problem in the IoT in a self-organizing manner. In particular, the *Kolkata paise restaurant (KPR) game* is a suitable framework for allocating multiple resources to multiple users with individual preferences on the resources. The authors in [105] introduce and discuss the basics of a KPR game. Such a game typically studies decision-making scenarios in which there are customers with common preference ranking of restaurants, and the goal of the customers is to go to a more preferred restaurant, while avoiding to choose the same restaurant as other customers. The work in [105] shows that the socially optimal solution of a KPR game is a Nash equilibrium (NE) and investigates different learning methods that converge to the socially optimal NE. Moreover, the learning methods introduced in [105] do not require a complete knowledge of other players and can converge to the NE only with partial knowledge, which is suitable for the IoT devices. However, the game design and the learning methods of [105] are tailored towards social applications, and, thus, they are not directly applicable to the IoT. In particular, the existing KPR model in [105] does not consider different preference ranking for different players, varying number of players, and zero utility for overlapping resource usage.

The main contribution of this chapter is to study and analyze the IoT resource allocation problem using a KPR game that is tailored toward the unique features of the IoT. In particular, we consider a massive IoT in which IoT devices with imperfect knowledge compete over limited communication resources. We show that the socially optimal solution for our resource allocation game is an NE and we propose a learning framework that can be used to converge to the NE that better utilizes the scarce communication resources. The proposed learning framework allows the IoT devices to learn and intelligently choose the communication resources, such as resource blocks, for higher system throughput. Moreover, using the proposed learning framework, the IoT devices can learn to better choose the communication resource with only local information about the resource usage of neighboring IoT devices. Simulation results show that the percentage of communication resources that are used to successfully transmit is significantly increased and the learning framework is more effective in denser IoT. In particular, the results show that the percentage of communication resources that are used to successfully transmit is increased by 7% in an IoT having 2.5 devices per square meter and by threefold in a denser IoT with 5 devices per square meter compared to a baseline case without any learning.

The rest of this chapter is organized as follows. Section 5.2 presents the system model. In Section 5.3, we introduce the IoT resource allocation game and propose a learning framework for the IoT devices. Section 5.4 analyzes and

discusses the simulation results, while Section 5.5 draws some conclusions.

5.2 System Model

Consider the uplink of a wireless IoT system having one BS that serves N IoT devices transmitting short packets [20, 114]. For short packet transmissions, it is known that random access transmission is more suitable than scheduled channels as it can provide lower latency [19, 23], and [115]. Hence, we consider time-slotted random access over a frequency division multiple access (FDMA) system [23]. The channel is divided into b resource blocks (RBs) each of which can be used by only one IoT device, at a given time [116]. Whenever multiple IoT devices use the same RB, their transmissions will fail due to collisions. During a time slot, each device will have a short packet to transmit with probability p, and, thus, the expected number of packets to be transmitted $\mathbb{E}[N_t]$ in a slot is $N \cdot p$, where N_t is a random variable capturing the number of transmitting IoT devices in slot t.

In an IoT, the number of IoT devices transmitting simultaneously greatly outweigh the available communication resources such that $\mathbb{E}[N_t] \gg b$. Therefore, the efficient allocation of limited communication resources among the IoT devices is critical. However, a centralized solution using which the BS allocates the communication resources to the IoT devices is impractical. This is because the short packets randomly arrive at the IoT devices, and, thus, the BS cannot predict and schedule the RB usage. A trivial solution to resource allocation would be to enforce the cooperation of all IoT devices to coordinate RB usage. However, this can be impractical due to overhead and energy constraints [1]. Furthermore, such coordination would assume that the IoT devices have perfect knowledge of all other IoT devices.

One simple solution would be to allow the IoT devices to randomly choose RBs whenever they transmit. For the case in which b RBs are chosen with equal probability, we let S_n be a random variable that captures the number of successfully transmitted short packets. The support of S_n, supp(S_n), depends on both N_t and b. When $N_t \leq b$, the support of S_n is $[0, N_t]$, while the support of S_n is $[0, b-1]$ when $N_t > b$. With $N_t = n_t$, the probability of having at least s successful transmissions for $s \in \text{supp}(S_n)$ is:

$$\Pr(S_n \geq s | N_t = n_t) = \prod_{i=0}^{s-1} \frac{b-i}{b} \left(\frac{b-s}{b} \right)^{(n_t - s)}. \tag{5.1}$$

From (5.1), for given $N_t = n_t$, the probability of having s successful transmissions for $s \in \text{supp}(S_n)$ is:

$$\Pr(S_n = s | N_t = n_t) = \begin{cases} \Pr(S_n \geq s | N_t = n_t) - \sum_{i=1}^{n_t - s} \binom{n_t - s}{i} \Pr(S_n = s+i | N_t = n_t), & \text{if } n_t \leq b, \\ \Pr(S_n \geq s | N_t = n_t) - \sum_{i=1}^{b-1-s} \binom{n_t - s}{i} \Pr(S_n = s+i | N_t = n_t), & \text{if } n_t > b. \end{cases} \tag{5.2}$$

Note that $\Pr(S_n = s | N_t = n_t) = \Pr(S_n \geq s | N_t = n_t)$ when $s = n_t \leq b$ and when $s = b - 1$ and $n_t > b$. Since $N_t \sim B(N, p)$, the probability of having n_t IoT devices transmitting in a given slot is:

$$P(N_t = n_t) = \binom{N}{n_t} p^{n_t} (1-p)^{N - n_t}. \tag{5.3}$$

Using (5.2) and (5.3), the probability of having s successful transmissions will be:

$$\Pr(S_n = s) = \sum_{i=s}^{N} \binom{i}{s} \Pr(S_n = s | N_t = i) \Pr(N_t = i). \tag{5.4}$$

For all possible values of N_t, the maximum number of successfully transmitted packets is b, and, thus, the expected number of successfully transmitted packets in a slot is:

$$\mathbb{E}[S_n] = \sum_{s=1}^{b} s \Pr(S_n = s). \tag{5.5}$$

In a massive IoT with very large N, the probability of having s successful transmissions in (5.4) will be very low, and, thus, the expected number of successful transmissions will also be low. In other words, the scarce communication resources will not be used efficiently, because very few RBs are used to successfully transmit. Therefore, there is a need for a distributed resource allocation for the IoT devices to intelligently choose the RBs given their incomplete knowledge on the RB usage of other devices.

5.3 Game-Theoretic Resource Allocation

The RB allocation among the IoT devices can be seen as one-to-one association between RBs and IoT devices, because one RB must be allocated to one device for a successful transmission. Furthermore, the IoT devices will decide which RB to use based on the incomplete knowledge about the RB usages of other devices. This competition among the IoT devices to choose an RB without overlap can be formulated as a *game* [117, 118] since the successful transmission of an IoT device depends on the actions of other devices due to collisions. One suitable framework is the so-called *Kolkata paise restaurant game* [105] which is a repeated game in which n customers (players) simultaneously go to one of n restaurants (actions), each of which can only serve one customer. Although the restaurants have the same price, there is a preference ranking (utilities) of restaurants commonly known among the customers based on food quality. A customer can only visit one restaurant at any given iteration, and a restaurant will randomly select to serve one of the visiting customers. In the KPR game, when the utility of the least preferred restaurant is at least half of the utility of most preferred restaurant, the NE is achieved when the *service rate*, which is the fraction of restaurants with customers, is 1 [105]. Moreover, the NE of the KPR game coincides with the socially optimal solution. Therefore, various learning methods that achieve higher service rate than random selection are analyzed [105].

We model the one-to-one association between RBs and IoT devices as a KPR game whose players are N IoT devices in a set $\mathcal{P}$. Each player $i \in \mathcal{P}$ has a set of actions $\mathcal{A}_i$, which represents the set of b RBs. Furthermore, we assume that the IoT devices transmit via Rayleigh fading channel. Moreover, the channel gains for different RBs are different for the IoT devices, and the IoT devices prefer to transmit using an RB with higher channel gain. We let $a_{i,t}$ be an action by IoT device i at time t and $g_{i,j}\ \forall\ i \in \mathcal{P}, j \in \mathcal{A}_i$ be the fading channel gain of RB j when used by IoT device i. Therefore, the utility function is:

$$u(a_{i,t}) = \begin{cases} g_{i,a_{i,t}} & \text{if } a_{i,t} \neq a_{k,t}\ \forall\ k \in \mathcal{P}, k \neq i, \\ 0 & \text{otherwise.} \end{cases} \tag{5.6}$$

The IoT resource allocation game differs from the classical KPR game in various aspects. For instance, the preference ranking of the RBs based on their channel gains will be different for different IoT devices. Unlike the classical KPR game, here, the number of players is random at each iteration, and the number of RBs may not be equal to the number of devices. Moreover, if multiple devices choose an RB, they will all fail to transmit resulting in zero utility. Despite the significant differences from the original KPR, next, we show that the proposed IoT resource allocation game has an NE that may also be a socially optimal solution.

Proposition 3. If the number of transmitting IoT devices in a time slot is greater than b, then the Nash equilibrium corresponds to the case in which all RBs are chosen by at least one IoT device, and the socially optimal solution achieving the highest possible service rate is Nash equilibrium.

Proof. In a massive IoT with $N \gg b$, the number of transmitting IoT devices n_t in time slot t will most likely be larger than the limited communication resources, and, thus, we assume that $n_t > b$. In such a case, any solution that has at least one IoT device choosing each RB is an NE. For those solutions, some IoT devices will transmit successfully, and other IoT devices will not transmit successfully due to collision. The IoT devices that are transmitting successfully will not deviate and choose some other RB to transmit even if they are transmitting with least preferred RB. This is because if they choose some other RB to transmit, then they will fail to transmit due to collision, resulting in zero utility. Since

the successful transmission using any RB has positive utility, the IoT devices that are transmitting successfully will not deviate. On the other hand, the IoT devices that are not transmitting successfully are indifferent about deviating. This is because those IoT devices will not transmit successfully due to collision even if they deviate and choose some other RB to transmit. Since changing and not changing the RB selection will both result in zero utility for the IoT devices transmitting unsuccessfully, such IoT devices are indifferent. Therefore, the solutions such that all RBs are chosen by at least one IoT device are NE.

In a massive IoT, the socially optimal solution is when $b-1$ IoT devices transmit successfully, while $n_t-(b-1)$ IoT devices do not transmit successfully. This achieves the highest service rate, which is ${}^{b-1}/_{b}$. All RBs are chosen by at least one IoT devices in socially optimal solution, and, thus, the socially optimal solution is an NE. □

In IoT resource allocation game, there exist many NE, but most NE will have very low service rates due to RB collision caused by overlapping RB selection. It is desirable to achieve NE with high service rate, such as the socially optimal NE. However, the IoT devices only have partial information about the RB usages of other IoT devices, which makes achieving the socially optimal solution challenging. Therefore, a novel framework to coordinate the RB usage of the IoT devices in a self-organizing manner using only partial information on RB usage is necessary. To this end, we propose a learning framework using which the IoT devices learn from the partial information to significantly increase the service rate compared to the baseline case of random RB selection.

5.3.1 Learning Framework

The IoT devices only have partial information about the RB usage of other IoT devices to intelligently choose an RB. For partial information, we model the IoT devices to be able to only communicate with neighboring IoT devices within a range r_c to learn about their RB usage. We let $\mathcal{T}_t$ be the set of IoT devices that transmitted during time slot t, $r_{t,i}$ be the RB used by device i in $\mathcal{T}_t$, and $\mathcal{N}_i$ be a set of neighbors of IoT device i within r_c. We let $\mathcal{S}_t$ be the set of IoT devices that transmitted successfully at slot t and $\mathcal{F}_t$ the set of IoT devices that did not transmit successfully at slot t such that $\mathcal{S}_t \cup \mathcal{F}_t = \mathcal{T}_t$ and $\mathcal{S}_t \cap \mathcal{F}_t = \varnothing$.

Using limited information on RB usage within r_c, the IoT devices can intelligently choose RBs to increase the service rate, and one such approach is to learn from the most recent actions of neighboring devices. Since the transmission failures result in zero utility and the objective is to increase the service rate, an IoT device i in $\mathcal{T}_t$ will learn from the IoT devices in $\mathcal{F}_{t-1} \cap \mathcal{N}_i$ and learn from the IoT devices in $\mathcal{S}_{t-1} \cap \mathcal{N}_i$ only if $\mathcal{F}_{t-1} \cap \mathcal{N}_i = \varnothing$. If $\mathcal{F}_{t-1} \cap \mathcal{N}_i \neq \varnothing$, an IoT device $i \in \mathcal{T}_t$ will choose $r_{t,i}$ randomly among the RBs that are preferred no more than the RBs used by devices in $\mathcal{F}_{t-1} \cap \mathcal{N}_i$. If $\mathcal{F}_{t-1} \cap \mathcal{N}_i = \varnothing$, an IoT devices $i \in \mathcal{T}_t$ will choose $r_{t,i}$ randomly among the RBs that are preferred no less than the RBs used by devices in $\mathcal{S}_{t-1} \cap \mathcal{N}_i$. In a case where $\mathcal{T}_{t-1} \cap \mathcal{N}_i = \varnothing$, an IoT device i will choose $r_{t,i}$ randomly from $\mathcal{A}_i$ with higher probability for more preferred RBs. This proposed learning method primarily increases the probability of successful transmission for the IoT devices transmitting unsuccessfully by using less preferred RBs, while allowing the IoT devices transmitting successfully to use more preferred RBs. Furthermore, whether the proposed learning framework converges to the NE depends on r_c, and this will be further discussed in the next section. Moreover, it is important to note that this method only uses the immediate history of RB usage of the IoT devices in $\mathcal{T}_{t-1}$ as the IoT devices are limited in memory. The performance of the learning framework will be analyzed and compared against the baseline case of random RB selection, whose service rate is ${}^{\mathbb{E}[S_n]}/_{b}$.

5.4 Simulation Results and Analysis

For our simulations, we will analyze the performance of the proposed learning framework in the resource allocation game under different communication ranges r_c, device densities λ, and transmission probabilities p. We assume that the IoT devices are deployed within a square field of dimension R meters based on Poisson point process with device

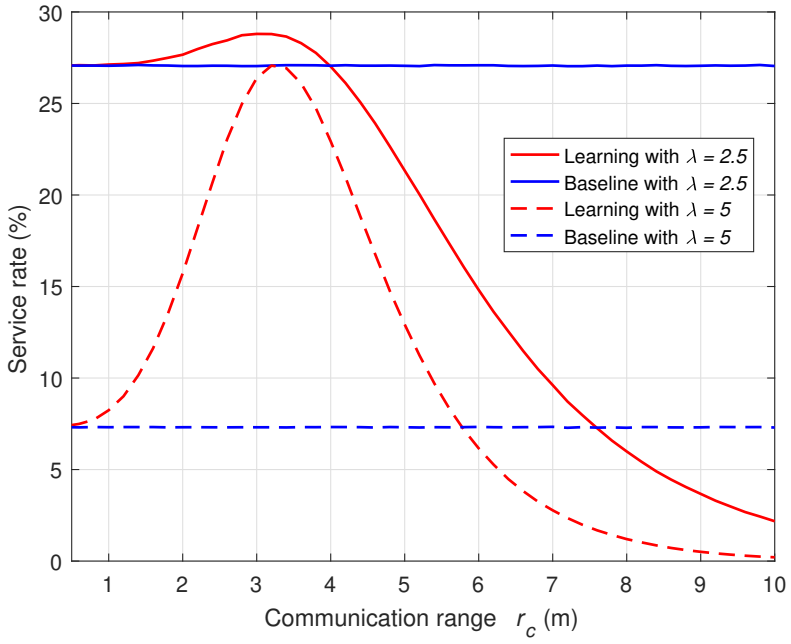


Figure 5.1: Average service rate of RBs with different λ and $p = 0.01$.

density λ per square meter. In our simulations, we set the dimension R of the deployment region to be 20 meters and the number of resource blocks b to be 5. We assume that the channel gains and the preference rankings of the RBs of neighboring IoT devices are similar. Primarily, the learning framework with varying values of r_c will be compared against the baseline case of random RB selection. Furthermore, we analyze the performance of the learning framework with low device density of $\lambda = 2.5$ and high device density of $\lambda = 5$. Moreover, we study the effect of the transmission probabilities p on the performance of the proposed learning framework.

Fig. 5.1 shows the average service rate of RBs resulting from the proposed learning framework with $p = 0.01$ and varying r_c and λ. With lower λ, the highest service rate that can be achieved using our learning framework is 28.8% when $r_c = 3$ meters, which is about 7% higher than the corresponding baseline. However, with higher λ, the highest service rate that can be achieved with learning framework is 27.1% when $r_c = 3.2$ meters, which is almost threefold greater than that of the corresponding baseline. Therefore, it can be concluded that the proposed learning framework is more effective in an IoT with higher λ, and it is particularly suitable for practical, massive IoT environments [1]. For both values of λ, there is an optimal value of r_c using which the learning framework increases the service rate the most.

The learning framework learns from both transmission failures and successes such that an IoT device learning from failure chooses less preferred RB, while an IoT device learning from success chooses more preferred RB. To learn from failures and successes, r_c can be interpreted as the amount of available information. For instance, r_c close to 0 implies that the IoT devices have almost no information, and, thus, they will choose RB randomly. Therefore, the learning framework achieves a service rate similar to the baseline, as seen in Fig. 5.1. The IoT devices with large r_c will almost always learn from the failures and mostly use less preferred RBs, because they first learn from the failures. Therefore, the IoT devices will be competing over fewer RBs as they all try to use less preferred RBs. This can be observed in Fig. 5.1 where the service rate decreases below the baseline for high values of r_c. Furthermore, the NE is not achieved with large r_c, because the RB selection is biased and not all RBs are used by at least one IoT device. There is an optimal value of r_c such that the IoT devices learn from both transmission failures and successes in choosing their RBs. Using such a value of r_c, the IoT devices do not compete over fewer RBs, and the service rate

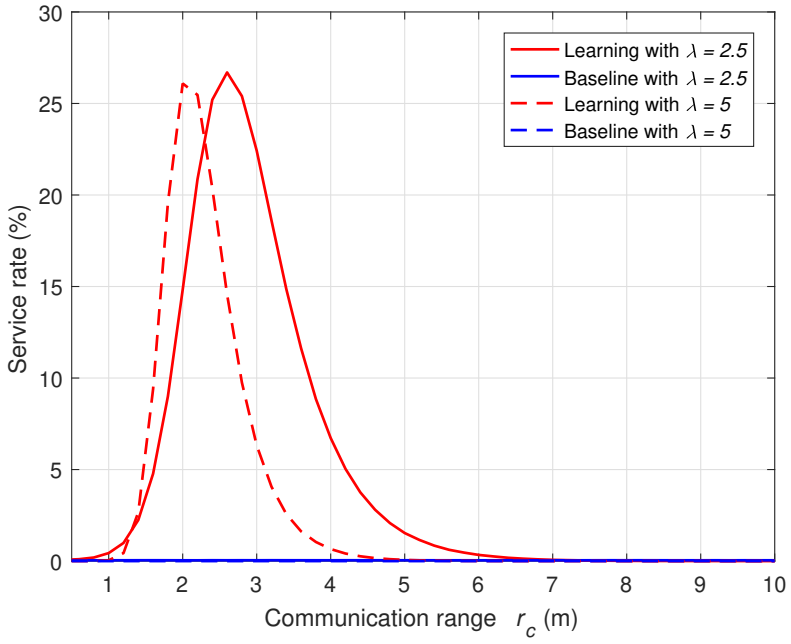


Figure 5.2: Average service rate of RBs with different λ and $p = 0.05$.

is improved compared to the baseline, as seen in Fig. 5.1. Furthermore, with an optimal value of r_c, the IoT devices will learn from successes and failures, and their RB selection will not be biased toward less preferred RBs. Since our learning framework enables the devices to avoid overlapping RB selections in a massive IoT to increase the service rate, all RBs will be chosen by at least one IoT device, and, thus, the NE is achieved with optimal value of r_c.

Fig. 5.2 shows the average service rate of RBs resulting from the proposed learning framework with $p = 0.05$ and varying r_c and λ. In an IoT, a higher value of p can be interpreted as an increase of IoT device activity due to user requests or some abnormal event. For a higher value of p, the baseline has service rates very close to 0% for both values of λ, because the number of transmissions per time slot increased greatly. For lower λ, the highest achievable service rate is 26.7% when $r_c = 2.6$ meters. For higher λ, the highest achievable service rate is 26% when $r_c = 2$ meters. Similar to Fig. 5.1, there is an optimal value of r_c that yields the highest service rate. Furthermore, the effect of increasing r_c on the performance of the proposed learning framework is similar to Fig. 5.1.

5.5 Summary

In this chapter, we have formulated the problem of allocating limited communication resources among a large number of IoT devices as a KPR game and proposed a novel approach for increasing the service rate with imperfect information. In particular, we have modeled the IoT devices as players choosing which RB to use based on their preference ranking and partial information about neighboring devices. We have shown that the socially optimal solution for this game is an NE, when the utility of successful transmission is positive. Furthermore, we have introduced a learning framework that allows the IoT devices to learn from the transmission failures and successes of neighboring IoT devices and significantly increases the service rate compared to the baseline of random RB selection. Simulation results have shown that the service rate resulting from the proposed learning framework is a function of the communication range, device density, and transmission probability. Furthermore, there is an optimal value of communication range

that maximizes the service rate, and the NE is achieved with certain values of communication range. Moreover, the results have shown that the learning framework is more effective in denser IoT.

Chapter 6

Distributed Resource Management in Non-Orthogonal Multiple Access

6.1 Background, Related Works, and Contributions

NOMA can potentially enable massive wireless connectivity and achieve a higher spectral efficiency compared to conventional orthogonal multiple-access (OMA), which makes it a suitable candidate for emerging 5G systems and applications [119–121]. As a key feature of NOMA, receivers exploit successive interference cancellation (SIC) to recover the transmitted signals from the received signals [122]. To decode the received signals at the receiver, their power levels must be different. Indeed, if the signals transmitted by multiple devices are received with different power levels at the receiver using NOMA, a large number of users can be simultaneously supported over the same frequency resources at the same time [123]. Thus, when devices transmit their uplink packets using NOMA, designing effective power level allocation schemes for coordinated, scheduled transmissions and uncooridnated, random access type transmissions is essential to take advantage of SIC and support massive connectivity.

In this regard, the power-level allocation problem in uplink NOMA is studied in [43–47]. In particular, coordinated NOMA uplink systems using a centralized scheduler are considered in [43] and [44]. In [43], an uplink power allocation scheme is proposed by using proportional fair scheduling, and the fractional frequency reuse scheme is applied to NOMA so that the throughput of the cell-edge users is improved. The authors in [44] propose a subcarrier and power allocation algorithm requiring a lower receiver complexity. Also, the problem of uplink random access with NOMA is studied for uncoordinated systems in [45–47]. The work in [45] considers a random access scheme using NOMA and proposes multi-channel ALOHA to achieve a higher throughput. In [46], multi-channel ALOHA is proposed to improve the throughput when users do not know the channel state information. The authors in [47] propose a random access scheme exploiting the different time of arrivals of the PRACH signals to identify multiple users having the same preamble.

However, the works in [43–47] are not applicable for scenarios in which devices need to transmit both *non-critical* and *high-priority* messages. In particular, using coordinated power levels as done in [43] and [44] to transmit high-priority messages can incur an additional cost of scheduling delay, and using uncoordinated power levels, e.g., using random access as done in [45–47], for all message types, can degrade the system throughput due to duplicate selection of devices' power levels. Therefore, one can consider a hybrid power-level allocation scheme for uplink NOMA in which non-critical messages are transmitted by using coordinated power levels, but high-priority messages are transmitted by using the power levels determined by random access. In particular, when non-critical and high-priority messages must be transmitted by using a limited number of total power levels, a learning framework can be used to allocate wireless

resources to devices [124]. Thus, *to the best of our knowledge, this work is the first to study how a network can use uplink NOMA to transmit non-critical and high-priority messages through exclusively partitioned power levels.*

The main contribution of this chapter is a novel framework for optimal, message-aware uplink transmit power level partitioning that allows the transmission of high-priority messages without scheduling in a NOMA cellular system. This framework allows a base BS to automatically partition the available uplink transmit power levels between the non-critical and high-priority messages in its network. To this end, we formulate an optimization problem whose goal is to maximize the throughput of non-critical messages while satisfying a reliability constraint, defined as the minimum throughput of high-priority messages. We assume that the exact number of devices transmitting each message type is unknown to the BS. Since the BS is unable to know the exact number of devices having high-priority messages, our framework estimates it by minimizing the mean-squared error between the observed and theoretically derived failure probability distributions. After the number of devices transmitting the high-priority messages is estimated, the derived failure probability is used to partition the power levels so as to satisfy the performance requirements of high-priority messages. Simulation results show that the proposed approach achieves a reliability of 77% for the uplink transmission of high-priority messages and significantly improves the reliability by 75% compared to the baseline case without optimal power level partitioning, which achieves a reliability of only 44%.

The rest of this chapter is organized as follows. In Section 6.2, we present the system model. In Section 6.3, we formulate the problem of allocating uplink transmit power levels for different types of messages and propose an algorithm to estimate the number of devices and to partition the power levels accordingly. Simulation results are analyzed in Section 6.4 while conclusions are drawn in Section 6.5.

6.2 System Model

Consider the uplink of a wireless system consisting of one BS located at the center of a geographical area in which N wireless devices are deployed as shown in Figure 6.1. The devices communicate with the BS using a time-slotted system with time slot duration τ using a non-orthogonal multiple access scheme with power division [43–47]. NOMA is chosen here since it has been shown to be capable of supporting high connectivity in 5G wireless network [123]. We focus on the uplink transmissions during which the devices transmit their messages to the BS using one of l different power levels. A transmission at a given power level fails if multiple devices choose the same power level simultaneously, because NOMA uses SIC to decode the received signal. Moreover, this duplicate power level selection will also cause the transmissions using the lower power levels to fail. Therefore, a transmission at a power level is successful only if one device uses that power level and there is no duplicate selection for the higher power levels.

In our model, we assume that the devices may have one of two message types. When devices transmit messages such as delay-tolerant system update reports, it is called *non-critical message*. We let p_p be the probability that a device has a non-critical message to transmit, and, thus, the expected number of devices with non-critical messages in a time slot is $N_p = Np_p$. Since non-critical messages are delay-tolerant and predictable, the BS can schedule the power levels for devices in order to avoid duplicated power-level selections. Therefore, once the power levels are allocated to the devices, they can transmit their non-critical messages without failure in NOMA uplink. However, due to unexpected events, the devices will often need to transmit mission-critical system failure messages, that are *high-priority messages*. We let p_c be the probability that a device has a high-priority message to transmit, and, thus, the expected number of devices with high-priority messages in a time slot is $N_c = Np_c$. We assume that the majority of the messages in a time slot will be non-critical messages such that $N_p > N_c$. The high-priority messages must be delivered to the BS with minimal latency and high reliability. To this end, random access type transmission can be used [45]. Random access NOMA uplink transmission has an advantage in that devices can immediately transmit messages without incurring scheduling delay. However, as a tradeoff of instant transmission, uncoordinated random access can result in duplicated power-level selection.

Given two types of messages, a total of l power levels in the NOMA uplink channel are partitioned into two sets of

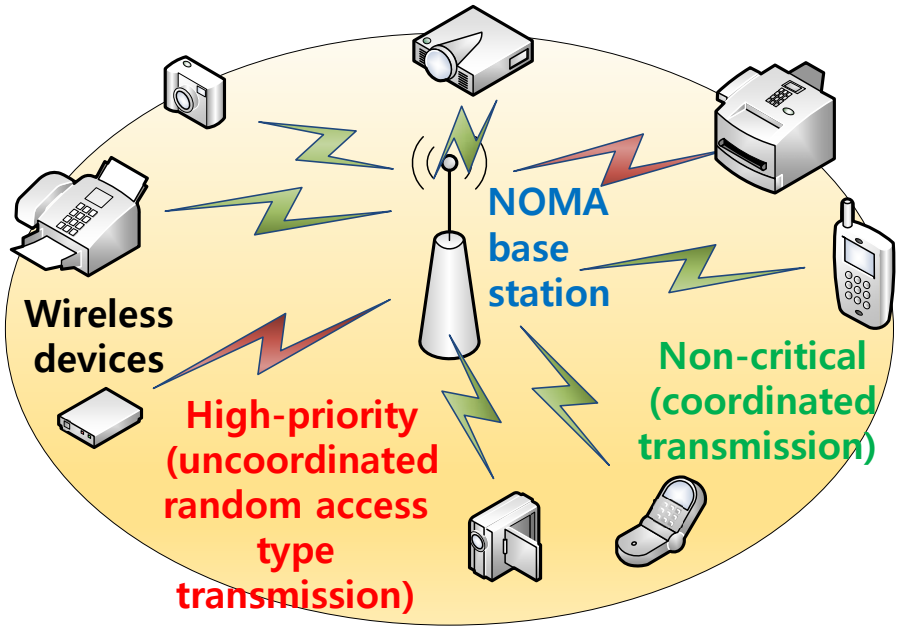


Figure 6.1: A NOMA uplink system when each device transmit a non-critical or high-priority message.

power levels as shown in Fig. 6.2. We assume that l_p and $l_c = l - l_p$ power levels are chosen for non-critical and high-priority messages, respectively. In NOMA uplink, if duplicate selection happens for a certain power level, then all power levels below that power level cannot be decoded [45], thus degrading the system throughput. If high-priority messages with uncoordinated transmissions use top l_c power levels, then transmission failures caused by duplicate selection among high-priority messages may occur for the non-critical messages using the bottom l_p power levels, greatly reducing throughput. However, if non-critical messages with coordinated transmissions use top l_p power levels, then transmission failures caused by duplicate selection among non-critical messages will not occur for the high-priority messages using the bottom l_c power levels. Therefore, we assume that high-priority messages use lower l_c power levels and non-critical messages use higher l_p power levels to prevent transmission failures for the non-critical messages.

Since duplicate power level selection is possible for high-priority message transmissions, the probability of transmission failure is important to ensure that l_c is large enough to satisfy the reliability requirement r for the high-priority messages. Here, *reliability* is defined as the probability of successful transmission in a time slot. For given N_c and l_c, the probability p_k with which the k-th highest power level of l_c power levels is the highest power level with a duplicate selection is:

$$p_k = \begin{cases} 1 - \frac{N_c}{l_c}\left(\frac{l_c-1}{l_c}\right)^{N_c-1} - \left(\frac{l_c-1}{l_c}\right)^{N_c}, & \text{if } k = 1, \\ \left(\frac{1}{l_c}\right)^{N_c} \sum_{i=0}^{\min(k-1,N_c-2)} C(k-1,i)P(N_c,i) \\ \quad \times \sum_{j=2}^{N_c-i} C(N_c-i,j)(l_c-k)^{N_c-i-j}, & \text{if } 1 < k < l_c, \\ \left(\frac{1}{l_c}\right)^{N_c} \sum_{i=0}^{\min(k-1,N_c-2)} C(k-1,i)P(N_c,i), & \text{if } k = l_c, \end{cases} \tag{6.1}$$

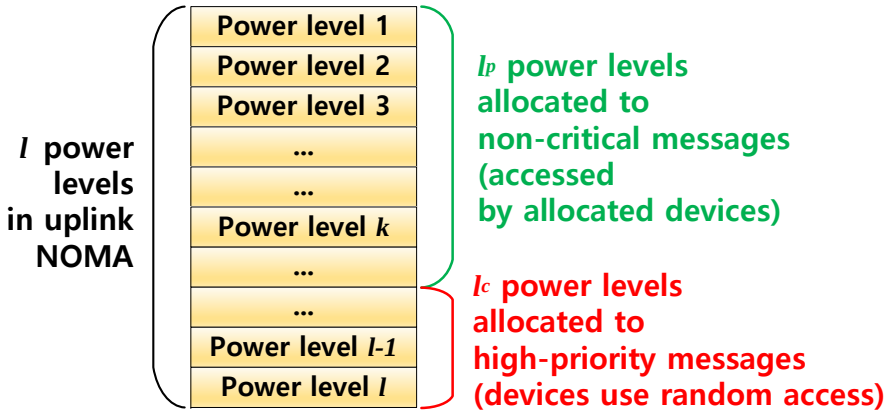


Figure 6.2: Total power levels are partitioned into two sets for non-critical and high-priority messages, respectively.

where $P(n,k)$ represents k-permutations of n and $C(n,k)$ represents k-combinations of n. We let $\boldsymbol{P}_k(N_c, l_c) = [p_1, \cdots, p_{l_c}]$ be a vector of p_k for each power level k. From p_k, the probability that an device fails to transmit using the k-th highest power level can be computed, as follows.

Theorem 5. When N_c devices use one of l_c power levels chosen uniformly and randomly, the probability $p_{f,k}$ with which a device will fail to transmit using the k-th highest power level is:

$$p_{f,k} = \begin{cases} p_k & \text{if } k = 1, \\ p_{f,k-1} + p_k & \text{if } k > 1. \end{cases} \tag{6.2}$$

Proof. At highest power level $k = 1$, the only cause of the transmission failure would be duplicate selection, and, thus, $p_{f,k} = p_k$ when $k = 1$. When $k > 1$, a device will fail to transmit using the k-th highest power level if there is a duplicate power level selection on any higher power level from 1 to $k-1$, because a duplicate selection causes the transmissions using the lower power levels to fail. The probability of having a duplicate selection on any power level from 1 to $k-1$ is equivalent to the probability of a transmission failing at the $(k-1)$-th highest power level, which is $p_{f,k-1}$. Furthermore, the transmission also fails when a duplicate selection occurs on the k-th highest power level. Since p_k is the probability of having first duplicate selection on the k-th highest power level, $p_{f,k} = p_{f,k-1} + p_k$ when $k > 1$. □

Using (6.2), it is possible to determine l_c for given p_c and N_c to satisfy a given reliability requirement by computing the average probability of transmission failure. For given N_c and l_c, the average probability p_f of transmission failure when N_c devices use one of l_c power levels chosen uniform randomly is:

$$p_f(N_c, l_c) = \frac{1}{l_c} \sum_{i=1}^{l_c} p_{f,i} = \frac{1}{l_c} \sum_{i=1}^{l_c} \sum_{j=1}^{i} p_j. \tag{6.3}$$

It is important to note that the average probability p_s of transmission success when N_c devices use one of the l_c power levels chosen uniformly randomly is not $(1 - p_f(N_c, l_c))$, because there is a probability that a power level is not used by any of the devices. However, one can still approximate it as $p'_s(N_c, l_c) \approx (1 - p_f(N_c, l_c))$, because the probability of a power level not being chosen by any of the N_c devices is low especially with large N_c. Based on the estimated $p'_s(N_c, l_c)$, we assume that l_c and l_p are initially determined to satisfy the reliability requirement r for the high-priority messages for known values of p_c and N_c.

Table 6.1: Summary of notations.

N	Number of devices
τ	Time slot duration
l	Number of total power levels
l_p	Power levels for non-critical messages
l_c	Power levels for high-priority messages
p_p	Probability that a device has a non-critical message
N_p	Expected number of non-critical messages
p_c	Probability that a device has a high-priority message
N_c	Expected number of high-priority messages
p_k	Probability that k-th highest power level of l_c power levels is the highest power level with a duplicate selection
$p_{f,k}$	Probability that device fails to transmit using k-th highest power level
p_f	Average probability of transmission failure when N_c devices using one of l_c power levels

It is possible for the occurrence of high-priority messages N_c to increase due to events, such as natural disaster or system failure. For a case in which p_c and N_c increase and l_c is not increased accordingly, the reliability of the high-priority messages is no longer satisfied as the average probability of transmission failure p_f in (6.3) will be higher. Therefore, there is a need to develop a new approach that allows the BS to determine if p_c has changed so as to update l_c accordingly to satisfy the reliability requirements of high-priority messages, while minimally affecting the throughput of the non-critical messages. The list of variables defined is shown in Table 6.1.

6.3 Problem Formulation and Power Level Partitioning Strategy

6.3.1 Problem Formulation

Our goal is to optimize the number of power levels for high-priority messages in order to minimize the transmission success probability such that the reliability threshold is satisfied and the throughput of other message types is minimally affected. Here, the *throughput* of non-critical messages is defined as the number of successful transmissions in a time slot. Since the transmissions of non-critical messages are coordinated, the maximum throughput of non-critical messages is l_p. To determine an appropriate partitioning of power levels, the BS must know the number of devices transmitting each type of messages. In practice, since the urgent system failure messages are generated at an uncertain moment, the number of devices transmitting high-priority messages will dynamically change. As a result, the BS will be unable to know a priori the information on the number of devices that are transmitting high-priority messages. Therefore, optimizing the number of power levels becomes challenging for the BS. This is because the number of high-priority messages may change at any moment, and the BS must quickly update the number of power levels for high-priority messages so that the reliability requirement can be satisfied. Furthermore, it is undesirable to allocate all l power levels for the high-priority message transmissions, because this may unnecessarily reduce the throughput of non-critical messages. Therefore, a method to accurately estimate N_c and to update l_c accordingly to satisfy r is necessary.

The total number of power levels is limited to l, and l_p and l_c are related such that $l = l_p + l_c$. Moreover, $l_c(N_c, r)$ depends on the number of devices having high-priority messages N_c that need to satisfy reliability requirement r. Since $p'_s(N_c, l_c)$ is an increasing function of l_c, it is necessary to increase l_c so that $p'_s(N_c, l_c) \geq r$. However, the throughput of non-critical messages l_p must also be as high as it can, while the reliability requirement is satisfied. Therefore, we can formulate the following power level optimization problem whose goal is to maximize the number of power levels for non-critical messages l_p, while satisfying the reliability constraint of the high-priority messages, when previous transmission failure history can be accessible:

$$\max_{N_c} \; l_p = l - l_c(N_c, r), \tag{6.4}$$

$$\text{s.t. } p'_s(N_c, l_c) \geq r. \tag{6.5}$$

The optimal solution of this problem would be a value of l'_c and $l'_p = l - l'_c$ such that $p'_s(N_c, l'_c) \geq r$ in (6.5), while $p'_s(N_c, l'_c - 1) < r$. In other words, the optimal solution is the smallest value of l_c that satisfies (6.5), and, thus, the throughput of non-critical messages l_p is maximized. To calculate $p'_s(N_c, l_c)$, the BS needs to know N_c, and, therefore, the number of high-priority messages is estimated in:

$$N_c = \underset{x}{\operatorname{argmin}} \, \mathbb{E}[(P'_k - P_k(x, l_c))^T (P'_k - P_k(x, l_c)]. \tag{6.6}$$

The proposed Algorithm 6.3 finds the optimal solution by first estimating N_c in line 7 based on the previous transmission failure history $\{\mathcal{H}_{t-h+1}, \cdots, \mathcal{H}_t\}$ and P'_k. Using the estimated value of N_c, the proposed algorithm finds the smallest value of l'_c such that $p'_s(N_c, l'_c) \geq r$ in lines 9, 10, and 11. However, if l'_c satisfying (6.5) is greater than the total number of power levels l, then the optimization problem does not have any feasible solution. Therefore, all power levels are used by the high-priority message transmissions such that $l_c = l$, which achieves highest possible reliability for given l.

6.3.2 Estimating N_c and Optimizing l_c

To determine whether the values of p_c and N_c have changed, we assume that the BS can determine which power levels have transmission failures. However, the BS does not know the cause of transmission failure, which can be a duplicate power selection on either the k-th power level or any one of the higher power levels. However, it can be determined that the highest power level with a transmission failure is the highest power level with a duplicate selection at that power level. After observing the highest power level with transmission failure multiple times, the BS can estimate the number of high-priority messages N_c using (6.1) and known value of l_c based on minimum mean-squared error (MMSE). After estimating the changed value of N_c, the BS can determine the appropriate value of l_c to satisfy the reliability requirement using (6.2) and the estimated value of N_c.

We let $\mathcal{H}$ be the set of observations with h_i being the i-th observation. The number of observations h needed to accurately determine the probabilities of having the first duplicate selection at each power level is important in estimating the N_p using (6.1). If h is large, then the BS will be able to compute the probabilities more accurately. However, large h also implies that the BS will be slow to react to the change in p_c and N_c. If h is small, then the BS will be able to quickly update l_c, but the updated value of l_c may be inaccurate. Furthermore, the BS estimates N_p and updates l_c every h time slots using most recent h observations of the transmission failures. The BS does not use the complete history of observations and, instead, uses only the most recent h observations, because p_c and N_c may have changed multiple times and older observations may be inaccurate.

In updating l_c, the estimated probability of success $p'_s(N_c, l_c)$ is used instead of the actual probability of success p_s. p'_s is an underestimate of p_s, because there is a possibility that a power level is not selected by any of the devices with high-priority messages. Therefore, the reliability requirement r should be set higher than the actual value of reliability requirement. However, for given p_c and N_c, there is an upper bound to the achievable reliability, and the upper bound can be achieved when all power levels are used by the high-priority messages $l_c = l$. Therefore, with limited l, r may not be achievable depending on the value of N_c.

Algorithm 6.3 Proposed algorithm to estimate N_c and update l_c.

1 : Initialize values l_c, h, r.
2 : **for** $t = 1, 2, \cdots$
3 : $\mathcal{H}_t \leftarrow$ highest power level with transmission failure.
5 : **if** $t = h, 2h, \cdots$
6 : Compute probability P'_k of being highest power level with transmission failure for each power level using $\{\mathcal{H}_{t-h+1}, \cdots, \mathcal{H}_t\}$.
7 : $N_c = \underset{x}{\operatorname{argmin}}\ \mathbb{E}[(P'_k - P_k(x, l_c))^T (P'_k - P_k(x, l_c)]$.
8 : $l'_c \leftarrow 0$.
9 : **while** $p'_s(N_c, l'_c) < r$
10 : $l'_c \leftarrow l'_c + 1$.
11 : **end while**
12 : **if** $l'_c \leq l$, $l_c \leftarrow l'_c$.
13 : **else**, $l_c \leftarrow l$. **end if**
14 : **end if**
15: **end for**

6.4 Simulation Results

For our simulations, we focus on analyzing the performance of the proposed algorithm. The initial number of power levels assigned for high-priority messages is $l_c = 2$, and the total number of power levels is $l = 30$. We set the number of devices to $N = 1000$, while the probability of high-priority message transmission is given by $p_c = 0.2\%$. Therefore, the expected number of high-probability messages is $N_c = 2$. We choose the time slot duration to be $\tau = 1$ ms and the number of observations to be $h = 40$. Also, the results of high-priority message transmissions are averaged over $100,000$ independent simulation runs.

Fig. 6.3 shows how the success probability changes over time when having a target reliability of $r = 0.95$. The baseline is the case in which l_c is fixed as 2. In Fig. 6.3, we observe that the success probability increases at the time when the proposed algorithm updates the power levels for high-priority messages after h observations. We can observe a stair-like increment in success probability after 40 observations from 44% to 74%. The success probability further increases to 77% after 40 more observations. This is because the value of N_c is better estimated after more observations. However, the target reliability of 0.95 is not achieved. This is due to the fact that the proposed algorithm uses the estimated probability of successful transmission $p'_s(N_c, l_c)$. Moreover, it is interesting to note that the majority of available power levels are allocated for the high-priority messages such that $l_c \approx l$. This implies that the target reliability of 95% is higher than the highest achievable reliability for the given values of N_c and l.

Fig. 6.4 shows how the success probability changes over time when having a target reliability of $r = 0.7$, when p_c is increased from 0.2% to 0.3% at 80 ms. Similar to Fig. 6.3, the baseline is the case in which l_c is fixed as 2. In Fig. 6.4, we can also see the stair-like increment in success probability after 40 observations from 44% to 59% as the proposed algorithm updates l_c. The achieved probability of success after 40 observations is lower compared to Fig. 6.3, because the target reliability is lower. When p_c is increased from 0.2% to 0.3% after 80 time slots, the success probability for baseline and the proposed algorithm will significantly decrease to 30% and 50%, respectively. However, after additional observations of transmission failures, the success probability of the proposed algorithm increases to 64%. This shows that the proposed algorithm can quickly react to the change in p_c as the success probability is significantly increased after only one more set of observations.

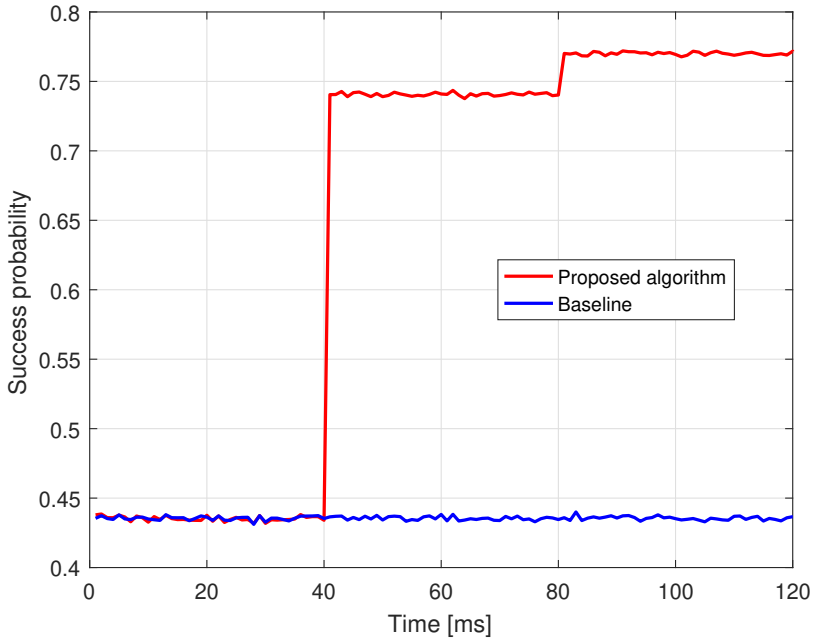


Figure 6.3: Success probability when l_c is updated by the proposed algorithm.

Fig. 6.5 shows how the success probability changes over time when having varying target reliability $r = 0.7, 0.8, 0.9,$ and 0.95. While l_c is fixed at 2 for the baseline, l_c is increased to 3 for $r = 0.7$, 4 for $r = 0.8$, 6 for $r = 0.9$, and 11 for $r = 0.95$ after 80 ms. For a larger l_c, the achieved success probability is 53% for $r = 0.7$, 59% for $r = 0.8$, 68% for $r = 0.9$, and 77% for $r = 0.95$ after 80 ms. Although a higher success probability is achieved with higher r, the target reliability is not achieved, because the estimated probability of success $p'_s(N_c, l_c)$ is used. Therefore, r must be set higher to achieve a desirable reliability requirement.

In Figs. 6.6 and 6.7, the failure probability is shown for different numbers of high-priority messages N_c varying from 6 to 16, when the number of power levels for high-priority messages l_c is 10 and 30, respectively. Here, all power levels are for the high-priority message transmissions, and the x-axis shows the index of the k-th highest power level. Figs. 6.6 and 6.7 show that the failure probability increases as k increases and the power level decreases. This is due to the fact that the transmissions at the lower power levels with large k will fail if any of the higher power levels with small k fails. Therefore, the failure probability of lower power levels is the accumulated failure probability as derived in Theorem 5. For example, from Fig. 6.6, we can see that the failure probability of power level $k = 6$ is two times the failure probability of power level $k = 1$ when $N = 16$. Moreover, the transmission failure probability can be reduced if the number of total power levels increases. As the number of total power levels increases, the devices have less probability to choose the same power level to transmit high-priority messages. For instance, Figs. 6.6 and 6.7 show that the failure probability of power level 1 can be reduced by up to 80% if the number of total power levels increases from 10 to 30 with $N = 16$.

6.5 Summary

In this chapter, we have proposed a NOMA uplink system that supports both non-critical and high-priority messages. In the proposed uplink system, the NOMA power levels are partitioned into two sets that are used by the devices

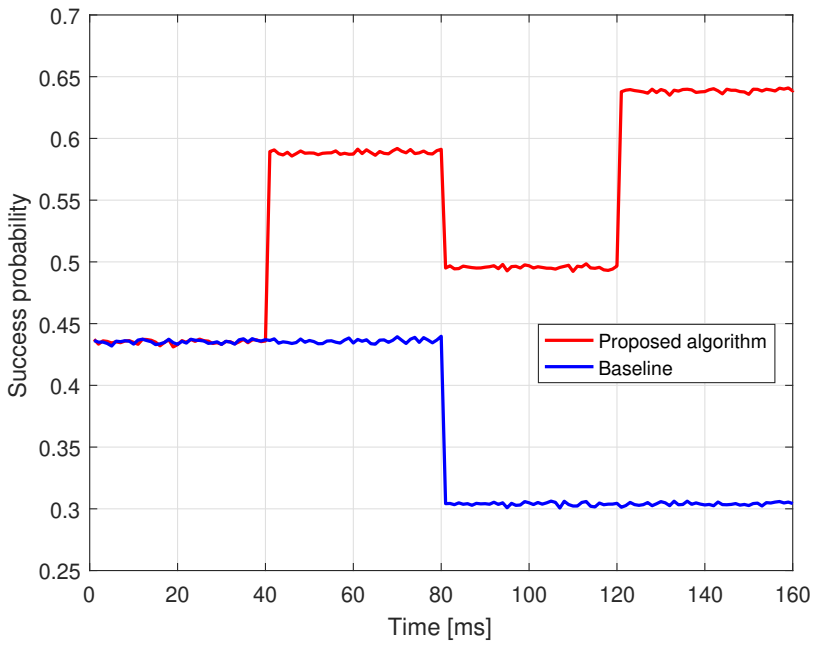


Figure 6.4: Success probability when l_c is updated by the proposed algorithm and p_c is changed at 80 ms.

transmitting the non-critical and high-priority messages, respectively. By doing so, the non-critical messages can be transmitted by the scheduled power levels without duplicate selection whereas the high-priority messages are transmitted by randomly selected power levels. We have formulated an optimization problem to determine the number of power levels for high-priority messages. We have proposed an estimation framework that suitably adjusts the number of power levels to achieve the targeting transmission success probability. Simulation results have shown that the proposed algorithm can increase the transmission success probability by up to 77% when the number of devices transmitting high-priority messages is properly estimated.

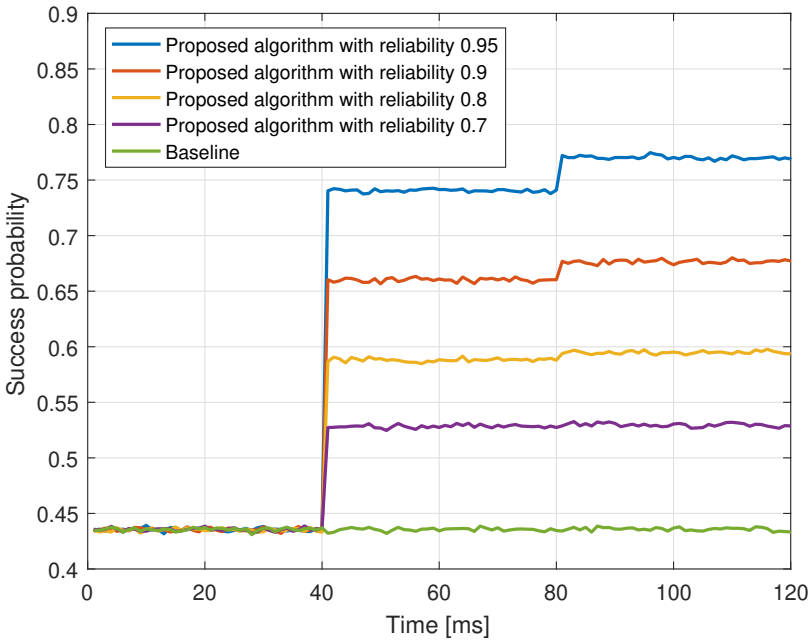


Figure 6.5: Success probability when l_c is updated by the proposed algorithm for varying target reliability r.

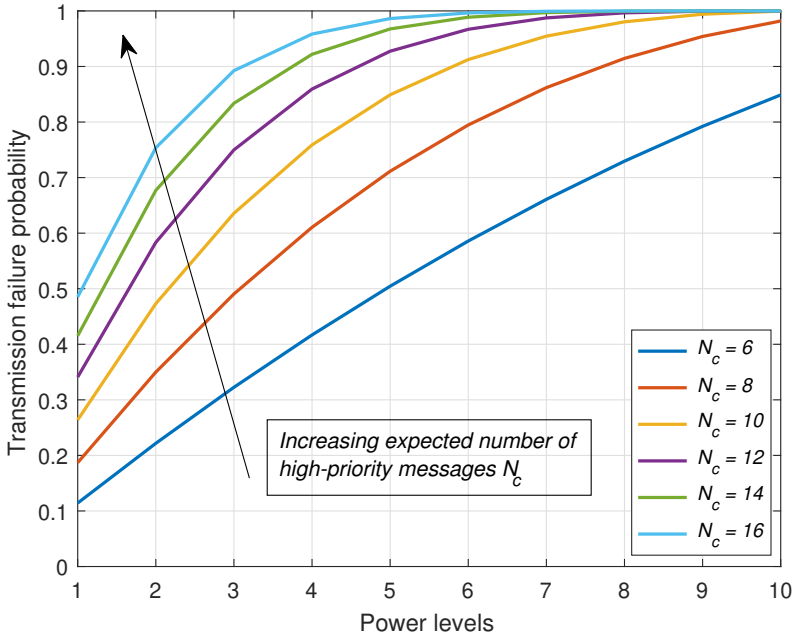


Figure 6.6: Failure probability for varying N_c with $l_c = 10$.

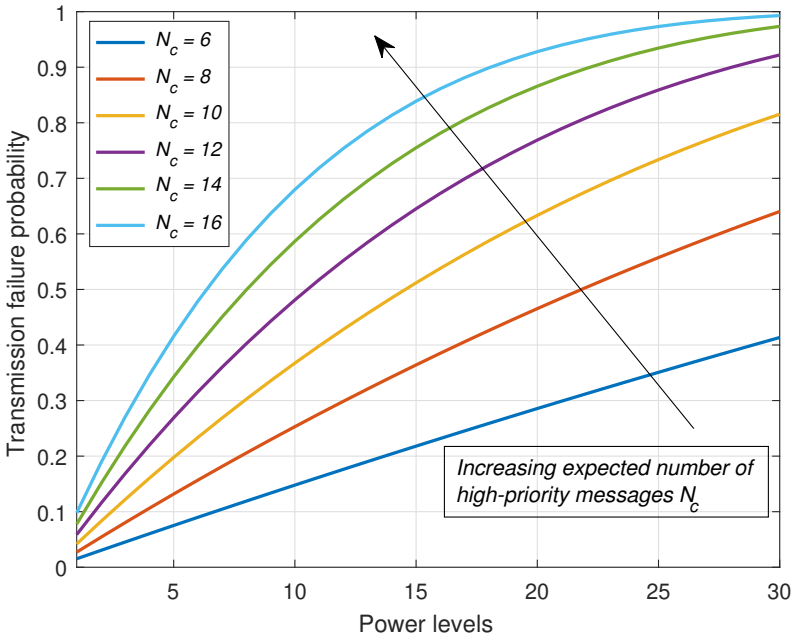


Figure 6.7: Failure probability for varying N_c with $l_c = 30$.

Chapter 7

Sum-rate and Reliability Analysis for Non-Orthogonal Multiple Access

7.1 Background, Related Works, and Contributions

Non-orthogonal multiple access (NOMA) can potentially improve spectral efficiency and, thus, it can be a suitable multiple access technology for emerging 6G systems and Internet of Things (IoT) [125–127]. In power-division (PD) NOMA, successive interference cancellation (SIC) is used by receivers to recover the transmitted signals from the superimposed, received signals [122]. To decode the received signals at the receiver, the value of the received signal powers should be different for successful SIC operation [128]. Therefore, if the signals from different devices are received with different power levels at the receiver, these multiple devices can be simultaneously supported over the same frequency spectrum at the same time. To determine the transmit power of the devices using PD-NOMA, designing effective transmit power allocation schemes for uplink radio resource management is essential to take advantage of SIC. In designing the transmit power allocation scheme for PD-NOMA, the sum-rate and the reliability must be considered for PD-NOMA to be applicable for ultra-reliable low latency communication (URLLC), particularly for IoT services [127, 129].

7.1.1 Related Works

In this regard, the transmit power allocation problem in the uplink of a NOMA system is studied in [130–142]. In particular, the data rate maximization problems of NOMA uplink systems are considered in [130, 132–135]. The work in [130] investigates a power allocation problem and shows the trade-off between orthogonal multiple access (OMA) and NOMA. In [131], the outage probability and the achievable average rate are investigated in downlink and uplink NOMA scenarios. The authors in [132] propose an uplink transmit power allocation scheme in which two users can simultaneously transmit their signals in each orthogonal subchannel. The work in [133] compares the sum-rate in uplink of a NOMA base station (BS) with the sum-rate of a conventional orthogonal multiple access (OMA) system when the BS is equipped with the multiple-input multiple-output (MIMO) infrastructure and uses millimeter-wave spectrum. The data rate of a two-user uplink NOMA system using a relay station is investigated in [134] when the relay has knowledge about its instantaneous fading channel. The authors in [135] show that the computational complexity of the average data rate maximization problem jointly optimizing subcarrier and power allocation in multi-carrier NOMA systems becomes NP-hard. Also, a number of problems related to the reliability of uplink NOMA systems are studied in [136] and [137]. The work in [136] investigates an uplink reliability improvement scheme that saves the

time by using unscheduled transmission, and the saved time is used to carry out hybrid automatic repeat request retransmissions in NOMA systems In [137], the theoretical transmission failure probability in NOMA uplink is derived to investigate the power level partitioning problem of finding the minimum number of power levels used to transmit high-priority messages with minimum quality-of-service constraints. Furthermore, implementation and performance of NOMA for an IoT and its devices are studied in [138–142]. The authors in [138] propose a medium access control protocol for an IoT using NOMA to increase throughput. Moreover, the energy efficient resource allocation schemes for NOMA are analyzed for IoT devices with energy harvesting [139] and for mobile IoT devices [140]. The work in [141] proposes an intelligent clustering of mobile IoT devices in a cellular network with NOMA. Moreover, the authors in [142] discuss limitations and stability of deploying an IoT in cellular network with NOMA in terms of packet arrive rate, delay constraint and throughput.

In all of these existing NOMA works [130–142], it is generally assumed that the channel gains are known to the transmitting devices in a network. However, in practice, the uplink devices cannot have complete information on the instantaneous channels over the transmission time period, and, hence, the actual channel information can be uncertain particularly in a fading environment. This is particularly true in IoT system. Indeed, it is challenging for NOMA devices to control their uplink transmit power so that the received signal power at the BS is the target received power level. Further, most of existing works [130–142] investigate energy efficiency [139, 140], data rate [130–135, 138, 141, 142] or reliability [136, 137] as the main metrics interest of uplink NOMA systems. To achieve URLLC in PD-NOMA, the data rate and reliability can be jointly optimized. Consequently, unlike the existing literature [130–137] which assumes full information knowledge for channel in NOMA uplink and optimizes a single objective, our goal is to design an uplink transmit power allocation scheme for NOMA, under uncertainty on fading channels, while maximizing both the sum-rate and the reliability of the uplink devices in a wireless PD-NOMA network. These considerations are particularly suitable for IoT systems.

7.1.2 Key Contributions

The main contribution of this paper is a novel analytical framework for optimal uplink transmit power allocation that maximizes the sum-rate and the reliability of the devices in a NOMA wireless system that must meet reliability targets for IoT-like services. This framework allows a BS to allocate the optimal target received power values to the devices using PD-NOMA. The BS needs to allocate the uplink transmit power values to the devices so that the separation of actual received power levels at the BS is greater than the power difference threshold required for SIC operation. When the transmitting devices do not know the exact channel gains due to random channel fading, we formulate a multi-objective optimization problem whose objective is to maximize the expected sum-rate and the reliability, which are critical performance metrics for URLLC and IoT services. To solve this problem, we analyze the expected sum-rate and the reliability in terms of target received power values and power threshold. Also, for Rayleigh fading channel, we show that both the expected sum-rate and the reliability are increasing functions of the target received power values, and the reliability is a decreasing function of the power difference threshold. Therefore, for the case in which there are two devices using PD-NOMA under Rayleigh fading channel, mathematical analysis shows that the highest expected sum-rate and the highest reliability are achieved when the target received power values are highest. Simulation results confirm the theoretical results and show that the proposed transmit power allocation scheme maximizes the sum-rate and reliability. Therefore, with Rayleigh fading channel, the proposed framework is shown to be able to achieve the optimal sum-rate and reliability of the devices by allocating the highest value of the target received power values of two devices. However, with Nakagami-m fading channel, simulation results show that there is a tradeoff between sum-rate and reliability depending on the target received power values, and, thus, an uplink transmit power allocation is necessary to satisfy the communication requirements of IoT-like services. For a specific simulation setting, simulation results also show that decreasing the power difference threshold is more effective in increasing the reliability than increasing the target received power values.

The rest of this paper is organized as follows. In Section 7.2, we present the system model. In Section 7.3, we analyze the performance of allocating uplink transmit power for the uplink users and prove that the derived transmit power

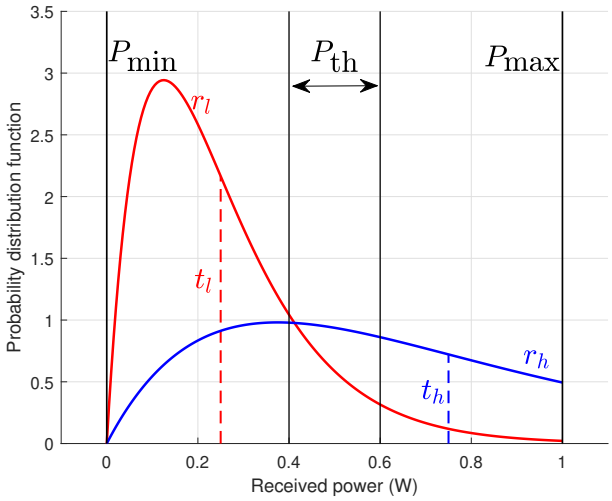


Figure 7.1: A model explaining the relations between the key parameters of the system model with $t_l = 0.25$ W, $t_h = 0.75$ W, $m = 2$, $P_{\text{min}} = 0$ W, $P_{\text{max}} = 1$ W, and $P_{\text{th}} = 0.2$ W.

allocation scheme achieves the optimal sum-rate and reliability. Simulation results are analyzed in Section 7.4 while conclusions are drawn in Section 7.5.

7.2 System Model

Consider the uplink of a wireless system consisting of one BS located at the center of a geographical area in which two devices are active. Those devices can be seen as IoT-like devices that need to transmit uplink data reliably and within a short deadline. For the communication between a device and the BS, we consider a time-slotted system with time slot duration of τ using a PD-NOMA scheme with two adjustable power levels. The two adjustable power levels include a lower target received power t_l and a higher target received power t_h such that $t_l \leq t_h$. In PD-NOMA, the two devices choose different target received power levels and transmit simultaneously so that the BS can decode the received signals using SIC. We assume that target received power values t_l and t_h are assigned to the two devices by the BS and that the two devices know their target received power values. Without loss of generality, we let device l be a device assigned with t_l and device h be a device assigned with t_h.

We consider a Nakagami-m fading channel and path loss with parameters α_1 and α_2. We assume that the devices only know the parameter of the Nakagami fading channel m and their distances to the BS. We let d_l be the distance between the BS and device l and d_h be the distance between the BS and device h. The actual received power r_i by device i for $i \in \{l, h\}$ will be:

$$r_i = L_i p_i h, \tag{7.1}$$

$$L_i = \alpha_2 d_i^{-\alpha_1}, \tag{7.2}$$

where h is the Nakagami fading channel gain, L_i is the path loss of device i, and p_i is transmit power of device i. Since the devices know m and their distances d_i to the BS, the devices can adjust their transmit power values p_i such that the expected received power values $\mathbb{E}[r_i]$ match their assigned target received power levels t_i.

The mathematical analysis of expected sum-rate and reliability is done for a Nakagami-m fading channel with $m = 1$, which is equivalent to a Rayleigh fading channel. Since the received power with Nakagami fading channel is gamma distributed, the received power with Rayleigh fading channel is exponentially distributed with mean λ^{-1}. Considering p_i to be constant and with known d_i, the actual received power r_i is random due to h, and the expected value of r_i should be equal to t_i such that:

$$\mathbb{E}[r_i] = t_i = \mathbb{E}[L_i p_i h] = \frac{p_i \alpha_2 d_i^{-\alpha_1}}{\lambda}, \tag{7.3}$$

$$p_i = \frac{\lambda t_i}{\alpha_2 d_i^{-\alpha_1}}, \tag{7.4}$$

$$r_i = \alpha_2 d_i^{-\alpha_1} \left(\frac{\lambda t_i}{\alpha_2 d_i^{-\alpha_1}} \right) h = \lambda t_i h. \tag{7.5}$$

The long term transmit power allocation p_i in (7.4) is a value of transmit power to ensure that $\mathbb{E}[r_i] = t_i$ by considering the known λ and d_i. Since h is exponentially distributed with mean λ^{-1}, r_i in (7.5) is also exponentially distributed with mean t_i. Figure 7.1 shows an example of r_l and r_h with given t_l, t_h, m, P_{min}, P_{max}, and P_{th}.

The target received power values t_l and t_h are within the bounds of P_{min} and P_{max} such that $t_l, t_h \in [P_{\text{min}}, P_{\text{max}}]$. We let P_{range} be a range of power that t_l and t_h are in such that $P_{\text{range}} = (P_{\text{max}} - P_{\text{min}})$. The values of t_l and t_h must be chosen carefully as the received signals are decoded using SIC. The SIC cannot decode the received signals if the difference of the received power values Δ_r is less than or equal to a power difference threshold P_{th}, which can be affected by the SIC and the device rate requirement [128]. In other words, for PD-NOMA, a transmission will fail if the difference between the actual received power values $\Delta_r = |r_h - r_l|$ is less than or equal to P_{th}. Typically, P_{th} is within the bounds of 0 and $(P_{\text{max}} - P_{\text{min}})$, and a smaller P_{th} implies better performing SIC. Moreover, the probability of transmission failure is defined as $\Pr(\Delta_r \leq P_{\text{th}})$. Moreover, in our model, the *reliability* S is defined as the probability of successful transmission, which is $S = 1 - \Pr(\Delta_r \leq P_{\text{th}})$. This is a key parameter for IoT-like services that must deliver data reliably to the BS. Furthermore, the expected sum-rate R of the two devices each targeting t_l and t_h is:

$$R = \mathbb{E}\left[\sum_{i \in \{l,h\}} B \log_2 \left(1 + \frac{r_i}{\sigma^2} \right) \right], \tag{7.6}$$

where B is the bandwidth and σ^2 is the additive white Gaussian noise power. Since PD-NOMA is a promising multiple access scheme for URLLC, the performance of PD-NOMA in terms of sum-rate and reliability will be further analyzed. Therefore, a multi-objective optimization problem maximizing the expected sum-rate R and the reliability S can be formulates such that:

$$\max_{t_l, t_h} \ (S, R), \tag{7.7}$$

$$\text{s.t.} \ \ P_{\text{min}} \leq t_l \leq t_h \leq P_{\text{max}}. \tag{7.8}$$

In the following section, we will find the optimal solution that maximizes both the reliability S and the expected sum-rate R.

7.3 Performance Analysis

For PD-NOMA to be used for URLLC or IoT-like services, a highly reliable communication and a timely transmission are required. Therefore, it is essential to determine how the sum-rate and the reliability are affected by the system parameters. In particular, the sum-rate and the reliability of PD-NOMA will be analyzed as functions of the lower target received power t_l, the higher power received power t_h, and the power difference threshold P_{th}.

7.3.1 PD-NOMA Reliability

In our system model, the transmissions using PD-NOMA from both devices fail if $\Delta_r \leq P_{\text{th}}$. Therefore, the reliability S is defined as the probability that the transmissions do not fail, which is:

$$S = 1 - \Pr(\Delta_r \leq P_{\text{th}}). \tag{7.9}$$

Proposition 4. For two devices using PD-NOMA with target received power values t_l and t_h and SIC failing to decode if the difference of the received power values is less than or equal to a power difference threshold P_{th}, the reliability S is:

$$S = \frac{t_h}{t_l + t_h} e^{-P_{\text{th}}/t_h} + \frac{t_l}{t_l + t_h} e^{-P_{\text{th}}/t_l}. \tag{7.10}$$

Proof. The probability of transmission failure $\Pr(\Delta_r \leq P_{\text{th}})$ due to SIC can be divided into two cases such that:

$$\begin{aligned} &\Pr(\Delta_r \leq P_{\text{th}}) = \Pr(|r_h - r_l| \leq P_{\text{th}}), \\ &= \Pr(r_h - r_l \leq P_{\text{th}} | r_h \geq r_l) + \Pr(r_l - r_h \leq P_{\text{th}} | r_l \geq r_h), \end{aligned} \tag{7.11}$$

to consider the two cases where $r_h \geq r_l$ and $r_l \geq r_h$.

For the first case where $r_h \geq r_l$, we let s_1 be $r_h - r_l$ and $s_1 \geq 0$. s_1 is a difference between two exponential distributions r_h and r_l with means t_h and t_l, respectively. Therefore, the probability density function of s_1 is:

$$f_{s_1}(x) = \frac{1}{t_l + t_h} \begin{cases} e^{-x/t_h} & \text{if } x \geq 0, \\ e^{x/t_l} & \text{if } x < 0. \end{cases} \tag{7.12}$$

As a result, the probability of transmission failure for the first case is:

$$\Pr(r_h - r_l \leq P_{\text{th}} | r_h \geq r_l) = \Pr(s_1 \leq P_{\text{th}} | s_1 \geq 0), \tag{7.13}$$

$$= \int_0^{P_{\text{th}}} f_{s_1}(x) dx = \int_0^{P_{\text{th}}} \frac{1}{t_l + t_h} e^{-x/t_h} dx, \tag{7.14}$$

$$= \frac{t_h}{t_l + t_h} \left(1 - e^{-P_{\text{th}}/t_h}\right). \tag{7.15}$$

Similarly, for the second case where $r_l \geq r_h$, we let s_2 be $r_l - r_h$ and $s_2 \geq 0$. The probability density function of s_2 is:

$$f_{s_2}(x) = \frac{1}{t_l + t_h} \begin{cases} e^{-x/t_l} & \text{if } x \geq 0, \\ e^{x/t_h} & \text{if } x < 0, \end{cases} \tag{7.16}$$

which is the probability density function of s_1 reflected about y-axis. Moreover, the probability of transmission failure for the second case is:

$$\Pr(r_l - r_h \leq P_{\text{th}} | r_l \geq r_h) = \Pr(s_2 \leq P_{\text{th}} | s_2 \geq 0), \tag{7.17}$$

$$= \int_0^{P_{\text{th}}} f_{s_2}(x)dx = \int_0^{P_{\text{th}}} \frac{1}{t_l + t_h} e^{-x/t_l} dx, \tag{7.18}$$

$$= \frac{t_l}{t_l + t_h} \left(1 - e^{-P_{\text{th}}/t_l}\right). \tag{7.19}$$

The total probability of transmission failure is:

$$\begin{aligned} \Pr(\Delta_r \leq P_{\text{th}}) &= \Pr(s_1 \leq P_{\text{th}} | s_1 \geq 0) + \Pr(s_2 \leq P_{\text{th}} | s_2 \geq 0), \\ &= 1 - \frac{t_h}{t_l + t_h} e^{-P_{\text{th}}/t_h} - \frac{t_l}{t_l + t_h} e^{-P_{\text{th}}/t_l}. \end{aligned} \tag{7.20}$$

Therefore, the reliability S of our system model is:

$$S = \frac{t_h}{t_l + t_h} e^{-P_{\text{th}}/t_h} + \frac{t_l}{t_l + t_h} e^{-P_{\text{th}}/t_l}. \tag{7.21}$$

□

Since P_{th} is a given system parameter and the reliability S in (7.10) is a function of t_l, t_h, and P_{th}, it is essential to analyze S in terms of t_l and t_h to find an optimal solution to the maximization problem in (7.7).

Proposition 5. For some fixed values of $P_{\text{min}} \geq 0$, $P_{\text{max}} > 0$, $t_h \in [P_{\text{min}}, P_{\text{max}}]$, and $P_{\text{th}} \geq 0$, the highest achievable reliability is:

$$\max_{t_l} S = e^{-P_{\text{th}}/t_h}, \tag{7.22}$$

and this is achieved when $t_l = 0$ or $t_l = t_h$ if $P_{\text{min}} = 0$ and only when $t_l = t_h$ if $P_{\text{min}} > 0$.

Proof. By the definition of t_l, $t_l \in [P_{\text{min}}, P_{\text{max}}]$ and $t_h \geq t_l$. At the bounds of t_l, when $t_l = 0$ and $t_l = t_h$, the value of reliability function is e^{-P_{th}/t_h}. For any value of $t_l = \beta$ such that $\beta \in (0, t_h)$, the value of reliability function is always less than the value of reliability function at the bounds of t_l such that:

$$e^{-P_{\text{th}}/t_h} > \frac{t_h}{\beta + t_h} e^{-P_{\text{th}}/t_h} + \frac{\beta}{\beta + t_h} e^{-P_{\text{th}}/\beta}, \tag{7.23}$$

$$e^{-P_{\text{th}}/t_h} > e^{-P_{\text{th}}/\beta}. \tag{7.24}$$

Since $t_h > \beta$, e^{-P_{th}/t_h} is always less than $e^{-P_{\text{th}}/\beta}$. Therefore, the highest value of reliability is e^{-P_{th}/t_h} when $t_l = 0$ or $t_l = t_h$ if $P_{\text{min}} = 0$ and only when $t_l = t_h$ if $P_{\text{min}} > 0$. □

When $t_l = 0$, the reliability is highest. However, this also implies that device l is not served, and the transmission from device h fails only due to severe Rayleigh fading. Moreover, the sum-rate of the devices will be greatly reduced when $t_l = 0$ and device l is not served.

For a fixed value of $t_h \in [P_{\text{min}}, P_{\text{max}}]$, the highest achievable reliability is e^{-P_{th}/t_h} (7.22). This is an increasing function of t_h, and the highest reliability S for any $t_l, t_h \in [P_{\text{min}}, P_{\text{max}}]$ is:

$$\max_{t_l, t_h} S = e^{-P_{\text{th}}/P_{\text{max}}}, \tag{7.25}$$

which occurs when $t_l = t_h = P_{\text{max}}$. Next, the second part of the optimization problem maximizing the expected sum-rate in (7.7) will be analyzed.

7.3.2 PD-NOMA Sum-Rate

The sum-rate of PD-NOMA is of interest as the sum-rate will affect the latency of the transmission. In determining the sum-rate of the devices, only the signal-to-noise ratio is considered. The expected sum-rate R of the two devices each targeting t_l and t_h is:

$$R = \mathbb{E}\left[\sum_{i\in\{l,h\}} B\log_2\left(1+\frac{r_i}{\sigma^2}\right)\right]. \tag{7.26}$$

Proposition 6. For two devices using PD-NOMA with target received power values t_l and t_h, the expected sum-rate R is:

$$R = \frac{-B}{\ln(2)}\sum_{i\in\{l,h\}} \exp\left(\frac{\sigma^4}{t_i}\right)\mathrm{Ei}\left(\frac{-\sigma^4}{t_i}\right), \tag{7.27}$$

where $\mathrm{Ei}(x)$ is an exponential integral function.

Proof. The actual received power values r_l and r_h are exponentially distributed with means t_l and t_h, respectively. Therefore, for $i \in \{l,h\}$, $\frac{r_i}{\sigma^2}$ is exponentially distributed with mean $\frac{t_i}{\sigma^2}$. Therefore, the expected sum-rate is:

$$\sum_{i\in\{l,h\}}\int_0^\infty B\log_2\left(1+\frac{r_i}{\sigma^2}\right)\frac{\sigma^2}{t_i}\exp\left(\frac{-r_i\sigma^2}{t_i}\right)dr_i$$

$$= \frac{B}{\ln(2)}\sum_{i\in\{l,h\}}\int_0^\infty \ln\left(1+\frac{r_i}{\sigma^2}\right)\frac{\sigma^2}{t_i}\exp\left(\frac{-r_i\sigma^2}{t_i}\right)dr_i.$$

Substituting $t = 1+\frac{r_i}{\sigma^2}$, the expected sum-rate is:

$$\frac{B\sigma^2}{\ln(2)}\sum_{i\in\{l,h\}}\frac{1}{t_i}\int_1^\infty \ln(t)\exp\left(\frac{(1-t)\sigma^4}{t_i}\right)\sigma^2 dt$$
$$= \frac{B\sigma^4}{\ln(2)}\sum_{i\in\{l,h\}}\frac{\exp(\sigma^4/t_i)}{t_i}\int_1^\infty \ln(t)\exp\left(\frac{-\sigma^4 t}{t_i}\right)dt.$$

Using integral solution from [143, Equation (4.331-2)], the expected sum-rate becomes:

$$\frac{B\sigma^4}{\ln(2)}\sum_{i\in\{l,h\}}\frac{\exp(\sigma^4/t_i)}{t_i}\left(\frac{-t_i}{\sigma^4}\mathrm{Ei}\left(\frac{-\sigma^4}{t_i}\right)\right)$$
$$= \frac{-B}{\ln(2)}\sum_{i\in\{l,h\}}\exp\left(\frac{\sigma^4}{t_i}\right)\mathrm{Ei}\left(\frac{-\sigma^4}{t_i}\right).$$

□

Since $\mathrm{Ei}(0) = -\infty$ and $\mathrm{Ei}(x) = 0$ as x approaches $-\infty$, $\mathrm{Ei}\left(-\sigma^4/t_i\right)$ is a strictly decreasing function of t_i. As t_i increases, $\exp\left(\sigma^4/t_i\right)$ approaches 1, while $\mathrm{Ei}\left(-\sigma^4/t_i\right)$ approaches $-\infty$. Therefore, the expected sum-rate is an increasing function of t_i.

Similar to the highest reliability in (7.25), the highest expected sum-rate is:

$$\max_{t_l,t_h} R = \frac{-2B}{\ln(2)}\exp\left(\frac{\sigma^4}{P_{\max}}\right)\mathrm{Ei}\left(\frac{-\sigma^4}{P_{\max}}\right), \tag{7.28}$$

which occurs when $t_l = t_h = P_{\text{max}}$. Since the highest expected sum-rate and the highest reliability are both achieved when $t_l = t_h = P_{\text{max}}$, the optimal solution to the maximization problem in (7.7) is $t_l = t_h = P_{\text{max}}$. One implication of this result is that the two devices should choose the target received power of P_{max} for highest expected sum-rate and reliability for the case of two devices using PD-NOMA. Moreover, it is unnecessary for the BS to allocate the lower and higher target received power values t_l and t_h to the devices.

The solution of $t_l = t_h = P_{\text{max}}$ to optimization problem (7.7) is only limited to Rayleigh fading channels. This is because the received power (7.5) is exponentially distributed with Rayleigh fading channel, and the variance of received power increases with higher target received power. Therefore, with higher variance, it is more likely for the actual received signal powers to be separated by more than P_{th} for successful transmission. Furthermore, with a higher target received power, the actual received power increases, and, thus, the sum-rate increases.

For a general Nakagami-m fading channel with $m > 1$, the optimization problem in (7.7) does not admit the solution of $t_l = t_h = P_{\text{max}}$, because the received power is gamma distributed. Unlike an exponential distribution that has a peak at 0, a gamma distribution with $m > 1$ does not have a peak at 0 as shown in Figure 7.1, and a peak of gamma distribution changes depending on the target received power. As t_l increases to t_h, the actual received powers r_l and r_h eventually have same nonzero peaks, and, thus, it is less likely for r_l and r_h to be separated by more than P_{th} for successful transmission. In other words, as t_l increases to t_h, the sum-rate increases, while the reliability decreases. Therefore, there is a tradeoff between sum-rate and reliability as t_l increases to t_h, and $t_l = t_h = P_{\text{max}}$ is not a solution to optimization problem with $m > 1$. The tradeoff between sum-rate and reliability with $m > 1$ is further analyzed via simulations in Section 7.4.

7.4 Simulation Results and Analysis

For our simulations, we focus on analyzing the rate and the reliability of the PD-NOMA for various lower target received power t_l, higher target received power t_h, power range P_{range}, and power different threshold P_{th}. For SIC, the maximum power P_{max} is set to 1 W, and the minimum power P_{min} is set to 0 W, which means that the power range P_{range} is set to 1 W. For all simulations, we determine the values of t_l, t_h, and P_{th} as the percentages of either P_{max} and P_{range}. Since the expected sum-rates (7.27) and the reliability (7.10) are functions of t_l, t_h, and P_{th}, the simulation results are valid for different values of P_{min} and P_{max} as long as the relative values of t_l, t_h, and P_{th} with respect to P_{min} and P_{max} are the same. For instance, the simulation result for $P_{\text{th}} = 0.1$ W with $P_{\text{max}} = 1$ W is equivalent to the simulation result for $P_{\text{th}} = 0.01$ W with $P_{\text{max}} = 0.1$ W. Therefore, the simulation results can be readily used to analyze the expected sum-rate and the reliability of PD-NOMA with various values of P_{min} and P_{max}.

In our simulations, we set the noise power to -174 dBm/Hz and the bandwidth to 1 MHz. The path loss parameters α_1 and α_2 are to be free space path loss with $\alpha_1 = 2$ and $\alpha_2 = \frac{1}{4\pi}$. Since we assume that the devices know their distances to the BS and their transmit power values p_i are adjusted accordingly, it is unnecessary to specify the deployment region and device deployment process. The Rayleigh fading channel parameter λ is 1, while the Nakagami fading channel parameter m is varied.

Figure 7.2 shows the simulated reliability for various values of t_l and m, while $t_h = P_{\text{max}}$ and $P_{\text{th}} = 25\%$ of P_{range}. It is important to note that $m = 1$ is Rayleigh fading, and the simulated reliability with $m = 1$ matches with corresponding theoretical result in (7.10). For $m > 1$, the actual received powers are gamma distributed, and, thus, the simulated values of reliability for $t_l = 0$ and $t_l = t_h$ are not the same. The simulated reliability decreases and then increases as t_l increases. The simulated reliability decreases, because the peaks in probability density functions of r_l and r_h start to coincide and the likelihood of transmission failure increases as t_l increases. The simulated reliability increases for higher values of t_l, because the variance of gamma distribution separates the actual received powers and the likelihood of transmission failure decreases as t_l increases. For Nakagami-m fading with $m > 1$, reliability is higher for lower values of t_l, which implies that there is a tradeoff between reliability and sum-rate. Furthermore, for higher m, separating t_l and t_h is more important to achieve higher reliability.

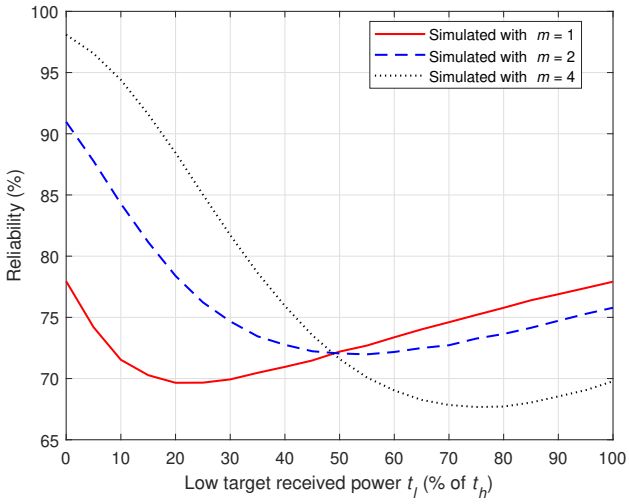


Figure 7.2: Simulated reliability for various values of t_l and m with $t_h = P_{\text{max}}$ and $P_{\text{th}} = 25\%$ of P_{range}.

Figure 7.3 shows the simulated reliability and sum-rate for various values of m, while $t_l = t_h$ and $P_{\text{th}} = 25\%$ of P_{range}. For Rayleigh fading channel, $t_l = t_h = P_{\text{max}}$ is the solution that maximizes both reliability and sum-rate. For any value of m, reliability and sum-rate both increase as $t_l = t_h$ increases from P_{min} to P_{max}. The simulated reliability increases, because the variance of gamma distribution increases as $t_l = t_h$ increases and it is more likely for Nakagami-m fading channel to separate the actual received powers. The simulated sum-rate increases, because the actual received powers increase as $t_l = t_h$ increases. Although there is no tradeoff between reliability and sum-rate when $t_l = t_h$, $t_l < t_h = P_{\text{max}}$ achieves a higher reliability than $t_l = t_h = P_{\text{max}}$ as shown in Figure 7.2.

Figure 7.4 shows the simulated reliability and sum-rate for various values of m, while $t_h = P_{\text{max}}$ and $P_{\text{th}} = 25\%$ of P_{range}. For any m, the simulated sum-rate increases as t_l increases from P_{min} to P_{max} with fixed $t_h = P_{\text{max}}$. However, for $m > 1$, $t_l = t_h = P_{\text{max}}$ does not maximize both reliability and sum-rate. For $m > 1$, the simulated reliability can increase higher by lowering sum-rate. Therefore, there is a tradeoff between reliability and sum-rate for Nakagami-m fading channel when $m > 1$ and $t_l < t_h = P_{\text{max}}$. Furthermore, the solution of $t_l = t_h = P_{\text{max}}$ is not applicable to Nakagami-m fading channel, and t_l can be chosen to satisfy given reliability and sum-rate requirements.

Figure 7.5 shows the theoretical and simulated reliability of PD-NOMA for various values of P_{th}, while $t_l = t_h = P_{\text{max}}$ and $m = 1$. It is important to note that P_{th} is defined as a percentage of P_{range}, and $t_l = t_h = P_{\text{max}}$ yields the highest achievable reliability. Moreover, $P_{\text{th}} = 0\%$ of P_{range} implies perfect SIC i.e., any received signals can be decoded successfully, while increasing values of P_{th} implies a worse SIC performances. For Rayleigh fading channel, the theoretical values of reliability calculated using (7.10) coincides closely with the simulated values of reliability. The reliability is 100% with a perfect SIC with $P_{\text{th}} = 0\%$ of P_{range}, and the reliability is 36.8% with SIC with $P_{\text{th}} = 100\%$ of P_{range}. Furthermore, the reliability decreases as the value of P_{th} increases. This is because it is less likely for the actual received power values r_l and r_h to be more than P_{th} apart for higher values of P_{th} with fixed values of t_l and t_h. As it can be seen in (7.25), which expresses the reliability as a function of only P_{th} when $t_l = t_h = P_{\text{max}}$, the reliability is an exponentially decreasing function of P_{th}.

Figure 7.6 shows the theoretical and simulated reliability of PD-NOMA for various values of P_{th} and t_l, while $t_h = P_{\text{max}}$ and $m = 1$. It is important to note that t_l is defined as a percentage of t_h. The highest reliability is 77.9% for $P_{\text{th}} = 25\%$ of P_{range}, 60.7% for $P_{\text{th}} = 50\%$ of P_{range}, and 47.2% for $P_{\text{th}} = 75\%$ of P_{range}. Moreover, for Rayleigh fading channel, the highest reliability occurs when $t_l = 0\%$ and $t_l = 100\%$ of t_h as proved in Proposition 5. Similar

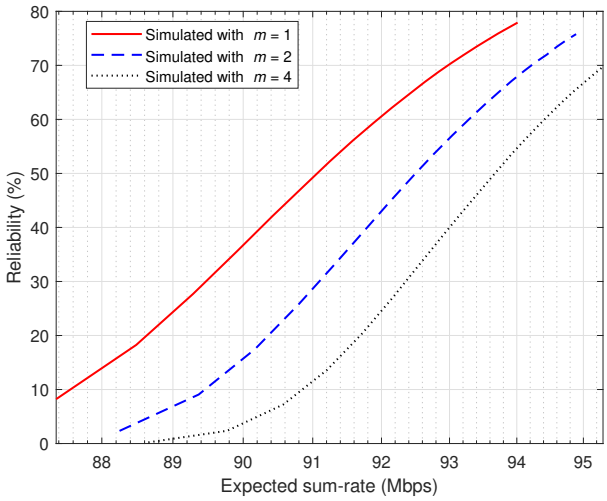


Figure 7.3: Simulated reliability and sum-rate for various values of m with $t_l = t_h$ and $P_{\text{th}} = 25\%$ of P_{range}.

to Figure 7.2, the reliability initially decreases for small values of t_l, because a possibility of transmission failure due to $\Delta_r \leq P_{\text{th}}$ is introduced from device l not transmitting with $t_l = 0$. However, the reliability increases for higher values of t_l, because the variance of exponentially distributed r_l increases with increasing mean t_l. Therefore, with fixed $t_h = P_{\text{max}}$, it is more likely for actual received powers r_i to be separated by more than P_{th}. As shown in Figure 7.5, the reliability increases as P_{th} decreases for corresponding values of t_l. Moreover, decreasing the value of P_{th} more effectively increases the value of reliability than changing the value of t_l.

Figure 7.7 shows the simulated reliability and sum-rate of PD-NOMA for various values of the expected sum-rate and P_{th}, while $t_l = t_h$ for the highest reliability and $m = 1$. The expected sum-rate increases as $t_l, t_h \in [P_{\text{min}}, P_{\text{max}}]$ increase. The highest expected sum-rate is 94 Mbps for all values of P_{th}, and this occurs when $t_l = t_h = P_{\text{max}}$. As shown in Figure 7.6, the corresponding reliability for the highest expected sum-rate is 77.9% for $P_{\text{th}} = 25\%$ of P_{range}, 60.7% for $P_{\text{th}} = 50\%$ of P_{range}, and 47.2% for $P_{\text{th}} = 75\%$ of P_{range}. The reliability in (7.10) and the expected sum-rate in (7.27) are both increasing functions of t_l and t_h. Therefore, as shown in Figure 7.7, there is no tradeoff between the expected sum-rate and the reliability, when Rayleigh fading channel is considered.

Figure 7.8 shows the theoretical and simulated reliability for various values of t_l, t_h, and P_{th}, while $t_l = t_h$ for the highest reliability and $m = 1$. It is important to note that t_l and t_h are defined as percentages of P_{range}, since $P_{\text{min}} = 0$ W. If $P_{\text{min}} > 0$ W, t_l and t_h would be defined as a sum of P_{min} and percentage of P_{range} to ensure t_l and t_h are in the bounds $[P_{\text{min}}, P_{\text{max}}]$. Moreover, t_l and t_h increase up to $2P_{\text{max}}$ to analyze the reliability at higher values of target received power values. When $t_l = t_h = 2P_{\text{max}}$, the value of reliability is 88.3% for $P_{\text{th}} = 25\%$ of P_{range}, 77.9% for $P_{\text{th}} = 50\%$ of P_{range}, and 68.74% for $P_{\text{th}} = 75\%$ of P_{range}. As t_l and t_h increase from P_{min} to $2P_{\text{max}}$, the value of reliability increases for all P_{th}. This is because the target received power values t_i are means of the exponentially distributed actual received power values r_i with $m = 1$. When the target received power values t_i increase, the variance of the exponentially distributed actual received power also increases, because the variance is t_i^2. Therefore, with a higher t_i, it is more likely for the actual received power values r_i to be more than P_{th} for successful transmission. In other words, the power separation of P_{th} required for successful decoding using SIC is done naturally by Rayleigh fading. Furthermore, Rayleigh fading is more likely to separate the actual received powers by more than P_{th} with higher target received power values. Therefore, with Rayleigh fading, it is unnecessary to allocate lower and higher target received power values. Moreover, both devices should choose the target received power values of P_{max} for the highest expected sum-rate and reliability.

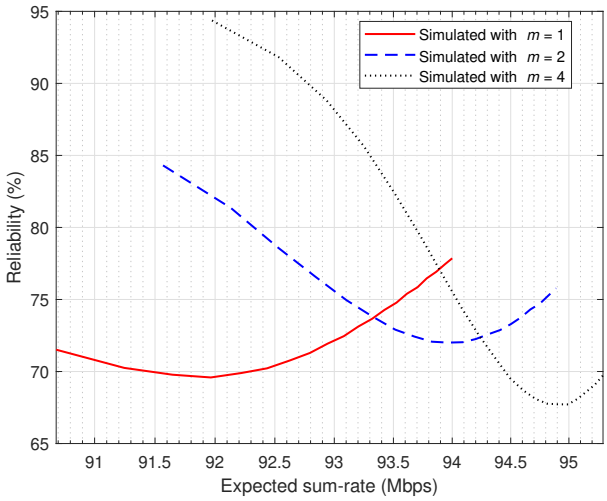


Figure 7.4: Simulated reliability and sum-rate for various values of m with $t_l = P_{\text{max}}$ and $P_{\text{th}} = 25\%$ of P_{range}.

7.5 Summary

In this paper, we have proposed a novel approach for allocating the lower and higher target received power values to IoT devices using PD-NOMA with different fading channels. In particular, we have analyzed the expected sum-rate and the reliability in terms of target received power values and power difference threshold for Rayleigh and Nakagami-m fading channels. Using both mathematical analysis and simulations, we have shown that the highest expected sum-rate and the highest reliability are achieved when the target received power values are highest for Rayleigh fading channel. Moreover, for Rayleigh fading channel, the simulation results have shown that the reliability is an increasing function of the target received power values and a decreasing function of the power difference threshold. However, we have shown that decreasing the power difference threshold is more effective in increasing the reliability than increasing the target received power values. Furthermore, we have derived the expected sum-rate and shown that it is an increasing function of target received power values. For Rayleigh fading channel, it is unnecessary to allocate different target received power values to the devices, because the target received power values must be highest to maximize both reliability and sum-rate. However, for Nakagami-m fading channel, we have shown that there is a tradeoff between reliability and sum-rate via simulations, and an allocation of lower and higher target received power values is necessary to satisfy the communication requirements of IoT-like services.

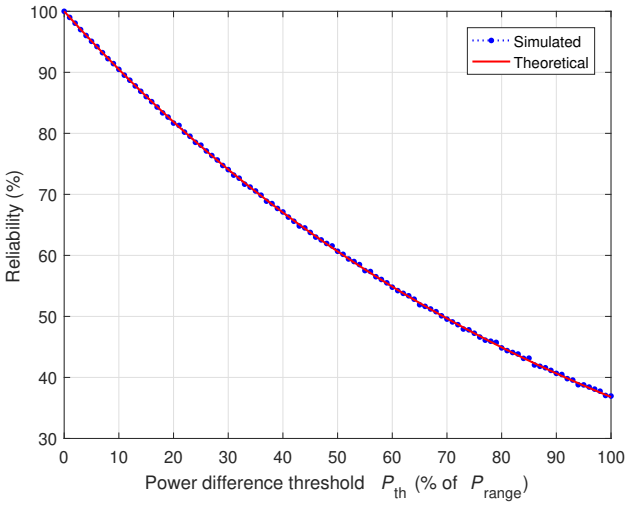


Figure 7.5: Theoretical and simulated reliability for various values of P_{th} with $t_l = t_h = P_{\text{max}}$ and $m = 1$.

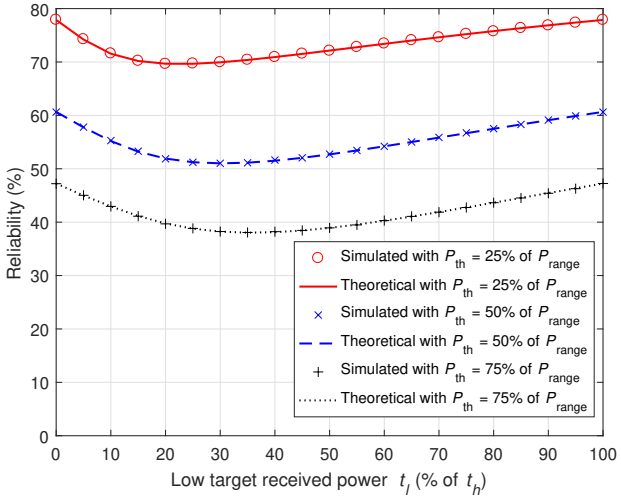


Figure 7.6: Theoretical and simulated reliability for various values of t_l and P_{th} with $t_h = P_{\text{max}}$ and $m = 1$.

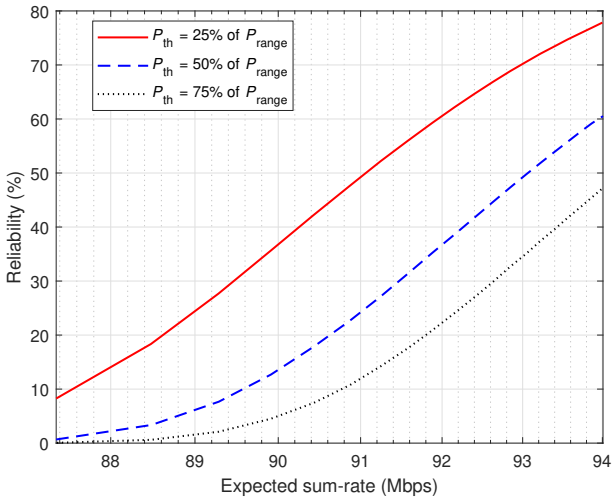


Figure 7.7: Simulated reliability and sum-rate for various values of P_{th} with $t_l = t_h$ and $m = 1$.

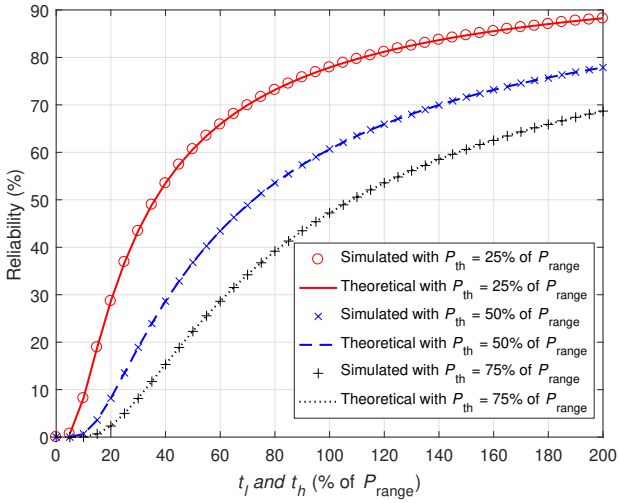


Figure 7.8: Theoretical and simulated reliability for various values of t_l, t_h, and P_{th} with $t_l = t_h$ and $m = 1$.

Chapter 8

Conclusions and Open Problems

The main body of our work in this dissertation can be summarized as follows. In the first chapter, we have presented the prominent properties of an IoT and the key challenges of distributed resource management for an IoT. For instance, the defining properties of an IoT are massive scale, high heterogeneity among IoT devices, and strict communication requirement, and these properties of an IoT lead to a question of communication resource management. We have discussed different approaches for communication resource management, including optimization, game theory, and machine learning. In particular, we have analyzed advantages and challenges of a distributed wireless resource management. A distributed wireless resource management can autonomously allocate the communication resources based on heterogeneous communication requirements, which enables critical IoT applications. Despite the major advantage and practicality of a distributed wireless resource management in an IoT, we need to account for key challenges related to IoT devices, such as lack of information, complexity, and capturing heterogeneity.

In the following sections, we have analyzed a number of mathematical tools that can be adapted for distributed wireless resource management, and we have investigated different approaches to accurately capture the heterogeneity among IoT devices. In Chapter 2, we have provided a comprehensive overview on the use of a variety of learning techniques within a wide range of applications in the IoT. In particular, we have discussed the general properties needed for learning techniques to be applied and developed for the purpose of IoT optimization and resource management. Furthermore, we have introduced the fundamentals of cognitive hierarchy theory, and we have investigated how it can be used to categorize the IoT devices into different levels of rationality, depending on their capabilities, thus providing a more realistic model of the IoT with heterogeneous devices. In Chapter 3, we have investigated applying sequential learning discussed in Chapter 2 for a distributed wireless resource management framework for an IoT. We have proposed a novel learning framework for enabling IoT devices to reallocate limited communication resources depending on the types and number of messages existing in the IoT. We have introduced a finite memory multi-state sequential learning, which is suitable for the IoT devices with limited computational capability and finite memory and effective against the limited observation capability of IoT devices. In particular, our proposed learning framework has shown to be effective in massive network with low delay threshold.

In Chapter 4 and Chapter 5, we have investigated the game theoretic approaches for distributed wireless resource management. In Chapter 4, we have introduced age of information, which measures the freshness of information in the perspective of base station. We have proposed both centralized and distributed approaches for resource allocation to minimize the AoI. Moreover, the heterogeneity among IoT devices is captured with non-linear aging functions. For the centralized approach, we have introduced a priority scheduling with learning. However, for the distributed approach, we have formulated a minority game for one-to-one association between RBs and devices. Moreover, we have designed payoff functions to encourage the devices with high AoI to transmit, while discouraging the devices with low AoI to not transmit. Furthermore, we have proposed a novel distributed algorithm using which the heterogeneous devices can allocate the communication resources in a self-organizing manner to minimize the AoI. We have shown

the conditions for existence of an NE, and we have proved that the proposed distributed algorithm converges to an NE. Simulation results show that the proposed distributed algorithm effectively accounts for heterogeneity among aging functions in reducing the AoI. Although the simulation results show that our centralized resource allocation scheme outperforms our distributed resource allocation scheme in most cases, our proposed distributed resource allocation is more practical for an IoT. Furthermore, the proposed distributed resource allocation outperforms a baseline of random allocation. In Chapter 5, we have formulated the problem of allocating limited communication resources among a large number of IoT devices as a minority game and proposed a novel approach for increasing the service rate with imperfect information. When the utility of successful transmission is positive, we have shown that the socially optimal solution coincides with an NE. Simulation results show that the service rate achieved with learning framework is a function of communication range, device density, and transmission probability. Furthermore, there is an optimal value of communication range that maximizes the service rate, and the NE is achieved with certain values of communication range.

In Chapter 6 and Chapter 7, we have investigated the non-orthogonal multiple access to support a massive IoT with heterogeneous devices. In Chapter 6, we have proposed a NOMA uplink system that supports both non-critical and high-priority messages by partitioning the power levels. We have formulated an optimization problem to determine the power level allocation for high-priority messages, and we have proposed an estimation framework to determine the number of high-priority messages. Simulation results show that the proposed algorithm can greatly improve the transmission success probability with proper estimation. In Chapter 7, we have proposed a novel approach for allocating the target received powers to the devices using PD-NOMA under different fading channels. In particular, we have analyzed the expected sum-rate and the reliability in PD-NOMA with Rayleigh fading channel and Nakagami-m fading channel. Both numerical analysis and simulation results show that sum-rate and reliability are maximized when the target received powers are highest. However, the simulation results show that there is a tradeoff between sum-rate and reliability for Nakagami-m channel, and the target received power allocation is necessary to satisfy the communication requirements.

While we have addressed the key challenges for distributed wireless resource allocation in an IoT, some of our works can be further extended. Therefore, in this chapter, we present a brief overview of the research in this dissertation, and we introduce a number of new research directions.

8.1 Summary

8.1.1 Distributed Wireless Resource Management in IoT: Motivation and Challenges

In Chapter 1, we have presented the unique properties of an IoT and the related challenges have been presented. To overcome the IoT challenges, we have provided a comprehensive overview on different approaches for distributed wireless resource management. In particular, we have discussed the distributed wireless resource management frameworks based on optimization, game theory, and machine learning. We have analyzed the advantages and the disadvantages of each framework for an IoT and its devices. For instance, we have evaluated the distributed wireless resource management frameworks in their ability to account for massive scale, high heterogeneity, and strict communication requirements. Finally, we have shown the distributed wireless resource management frameworks that are particularly suitable in mitigating massive scale, capturing heterogeneity, and satisfying strict communication requirements.

8.1.2 Learning to Manage Wireless Resource in IoT

In Chapter 2, we have provided a comprehensive overview on adapting the learning frameworks for distributed wireless resource management in an IoT. In particular, we have reviewed the advantages and the limitations of three popular learning frameworks: machine learning, sequential learning, and reinforcement learning. For each such framework, we have identified the inherent properties, such as computational complexity, memory requirements, and learning performance. Moreover, we have investigating on adapting cognitive hierarchy theory to capture the inherent heterogeneity among IoT devices. To this end, we have proposed using cognitive hierarchy theory with learning to determine which learning frameworks are most suitable given the requirements for learning and the available resources of IoT devices in a given level of cognitive hierarchy. In summary, this chapter has provided a comprehensive reference on developing and applying several classes of learning techniques within well-defined IoT applications.

8.1.3 Distributed Resource Management using Finite Memory Multi-State Sequential Learning

In Chapter 3, we have investigated a problem of communication resource allocation in presence of heterogeneous device types and communication requirements. We have introduced a finite memory multi-state sequential learning using which the heterogeneous IoT devices can reallocate RAPs appropriately to reduce the delay of critical messages. We have shown that our proposed learning framework is suitable for IoT devices, and we have proved the convergence of our proposed learning framework. The theoretical analysis have shown lowest expected delay of critical messages that can be achieved with our proposed learning framework. Moreover, simulation results have shown that the cumulative distribution function of delay of critical messages and average percentage of devices that learned correctly are functions of memory size, network density, and probability of detection.

8.1.4 Distributed Age of Information Minimization in IoT

In Chapter 4, we have proposed centralized and distributed approaches for allocating the limited communication resources to minimize AoI in presence of non-linear aging functions. When linear and non-linear aging functions coexist, we have shown that the problem of AoI minimization is different compared to a case where only linear aging function exists. For the centralized approach, we have introduced a priority scheduling scheme with learning, which enables the BS to allocate the limited resources to the heterogeneous devices based on their AoI and aging function. For the distributed approach, we have formulated the problem of AoI minimization as a minority game, and we have proposed a novel stochastic crowd avoidance algorithm using which the heterogeneous devices can share limited communication resources to avoid transmission failures and to minimize AoI. We have shown the conditions for an NE to exist, and we have proved that the proposed distributed algorithm converges to an NE. With simulations, we have compared our centralized and distributed RB allocation schemes, and we have shown that our centralized RB allocation scheme outperforms our distributed RB allocation scheme in most cases. However, our proposed distributed algorithm has shown to be effective in reducing the AoI only with partial information.

8.1.5 Game Theoretic Approach for Distributed Resource Management

In Chapter 5, we have formulated the problem of allocating limited communication resources among IoT devices as a Kolkata paise restaurant game. In this game, we have shown that the socially optimal solution is a Nash equilibrium. We have proposed a novel learning framework that allows the IoT devices to learn from transmission failures and successes of other IoT devices to avoid their own transmission failures. Simulation results have shown that the probability

of successful transmission is a function of communication range, device density, and transmission probability. Moreover, the results have shown that the learning framework is more effective in increasing the probability of successful transmission when compared to a baseline, when an IoT has higher device density.

8.1.6 Distributed Resource Management in Non-Orthogonal Multiple Access

In Chapter 6, we have proposed a NOMA uplink system that supports both non-critical and high-priority messages. We have formulated an optimization problem to determine the number of power levels for high-priority messages and non-critical messages. We have proposed an estimation framework that suitably adjusts the number of power levels to achieve a target transmission success probability for high-priority messages. Simulation results have shown that the proposed algorithm can greatly increase the transmission success probability for high-priority messages when the number of high-priority messages is properly estimated.

8.1.7 Sum-rate and Reliability Analysis for Non-Orthogonal Multiple Access

In Chapter 7, we have proposed a novel approach for allocating the lower and higher target received power values to the devices using PD-NOMA. In particular, we have analyzed the expected sum-rate and the reliability in terms of target received power values and power difference threshold for Rayleigh and Nakagami-m fading channels. Moreover, we have formulated a joint optimization problem to maximize the sum-rate and reliability. For Rayleigh fading channel, we have shown that the highest expected sum-rate and the highest reliability are achieved when the target received power values are both highest. Therefore, we have concluded that it is unnecessary to allocate different target received power values to the devices for Rayleigh fading channel. However, for Nakagami-m fading channel, we have shown that there is a tradeoff between reliability and sum-rate via simulations, and an allocation of lower and higher target received power values is necessary.

8.2 Open Problems

Despite numerous studies on distributed wireless resource management in an IoT, there are still many and important open questions and problems that must be investigated further. Furthermore, while we have addressed the key challenges for implementing the distributed wireless resource management in an IoT, some of our approaches can be extended further and new research directions can be explored. Therefore, we present an overview of the future research directions of this dissertation.

8.2.1 Non-orthogonal Multiple Access for Massive IoT

Non-orthogonal multiple access (NOMA) is a promising multiple access scheme to support a massive IoT, because more devices can be served simultaneously compared to an orthogonal multiple access. Moreover, NOMA may be a key to accelerate the deployment of IoT and its devices. There are several types of NOMA, such as code-domain NOMA and power-domain NOMA, and a type of NOMA may be more suitable for an IoT than other types of NOMA depending on the number of devices or communication requirements. Therefore, in terms of open problems implementing NOMA for IoT, there is a need for development of: a) analyzing reliability and latency of using different

types of NOMA, b) evaluating the performance of different types of NOMA in a dynamic IoT with high heterogeneity, and c) determining which type of NOMA is most suitable in a given IoT.

8.2.2 Finite Memory Learning in an IoT

In implementing different learning frameworks for IoT devices, a key challenge is gathering the information to learn from and overcoming the finite memory of IoT devices. The devices using distributed learning framework often have finite memory of information, while a centralized learning framework incurs excessive overhead to IoT devices. However, a hybrid learning framework for IoT devices can greatly improve the availability of information without being impractical for IoT devices. Therefore, in terms of improving finite memory learning in an IoT, a performance analysis is necessary to address: a) tradeoff between accuracy of learning and overhead of using a hybrid learning framework, b) extent to which a base station can intervene to improve finite memory learning, and c) dynamic availability of a base station to improve learning.

8.2.3 Distributed Framework in a Mobile IoT

A mobility of devices introduces high uncertain in terms of information availability. If the devices can move, the information that a device can access changes dynamically, and, thus, the performance of learning also changes. Moreover, an environment in which a device is located changes, and this implies that the objective of learning may also be changing as the devices move. Therefore, in terms of introducing mobility to an IoT, a performance analysis is necessary to account for: a) mobility model of IoT devices, b) mathematical model to capture dynamic availability of information, and c) change in underlying parameter that devices are learning.

Bibliography

[1] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, "Toward massive machine type cellular communications," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 120–128, Feb. 2017.

[2] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3949–3963, Jun. 2016.

[3] ——, "Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage," to appear in *IEEE Communications Letters*, 2016. [Online]. Available: https://arxiv.org/pdf/1606.01962v1.pdf

[4] P. Zhan, K. Yu, and A. L. Swindlehurst, "Wireless relay communications with unmanned aerial vehicle: Performance and optimization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 3, pp. 2068–2085, July 2011.

[5] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1046–1061, May 2017.

[6] W. Saad, A. Glass, N. Mandayam, and H. Poor, "Towards a consumer-centric grid: A behavioral perspective," to appear in *Proceedings of the IEEE*, 2016. [Online]. Available: http://arxiv.org/pdf/1511.01065v1.pdf

[7] H. Farhangi, "The path of the smart grid," *IEEE Power and Energy Magazine*, vol. 8, no.1, pp. 18–28, Jan. 2010.

[8] V. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. Hancke, "Smart grid technologies: Communication technologies and standards," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 529–539, Nov. 2011.

[9] J. J. Nielsen, G. C. Madueno, N. K. Pratas, R. B. Sorensen, C. Stefanovic, and P. Popovski, "What can wireless cellular technologies do about the upcoming smart metering traffic?" *IEEE Communications Magazine*, vol. 53, no. 9, pp. 41–47, Sep. 2015.

[10] E. Spanò, L. Niccolini, S. D. Pascoli, and G. Iannacconeluca, "Last-meter smart grid embedded in an internet-of-things platform," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 468–476, Jan. 2015.

[11] J. Gubbi, R. Buyya, S. Marusic, and M. palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[12] S. Kelly, N. Suryadevara, and S. Mukhopadhyay, "Towards the implementation of IoT for environmental condition monitoring in homes," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3846–3853, Oct. 2013.

[13] M. Weiner, M. Jorgovanovic, A. Sahai, and B. Nikolic, "Design of a low-latency, high-reliability wireless communication system for control applications," in *Proc. of IEEE International Conference on Communications*, Sydney, Australia, June 2014.

[14] L. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[15] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[16] K. Mikhaylov, J. Petaejaejaervi, and T. Haenninen, "Analysis of capacity and scalability of the lora low power wide area network technology," in *Proc. of 22nd European Wireless Conference*, Oulu, Finland, May 2016.

[17] J. Kim, J. Lee, J. Kim, and J. Yun, "M2m service platforms: Survey, issues, and enabling technologies," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 61–76, Feb. 2014.

[18] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: The internet of things architecture, possible applications and key challenges," in *Proc. of 10th International Conference on Frontiers of Information Technology*, Islamabad, Pakistan, Dec. 2012.

[19] G. C. Madueno, C. Stefanovic, and P. Popovski, "Reliable reporting for massive M2M communications with periodic resource pooling," *IEEE Wireless Communications Letters*, vol. 3, no. 4, pp. 429–432, May 2014.

[20] G. Durisi, T. Koch, and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets," *Proc. of the IEEE*, vol. 104, no. 9, pp. 1711–1726, Sep. 2016.

[21] G. Naddafzadeh-Shirazi, L. Lampe, G. Vos, and S. Bennett, "Coverage enhancement techniques for machine-to-machine communications over LTE," *IEEE Communications Magazine*, vol. 53, no. 7, pp. 192–200, July 2015.

[22] C. Hagerling, C. Ide, and C. Wietfeld, "Coverage and capacity analysis of wireless m2m technologies for smart distribution grid services," in *Proc. of IEEE International Conference on Smart Grid Communications*, Venice, Italy, Nov. 2014.

[23] H. Dhillon, H. Huang, H. Viswanathan, and R. Valenzuela, "Power-efficient system design for cellular-based machine-to-machine communications," *IEEE Trans. Wireless Commun.*, vol. 12, no.11, pp. 5740–5753, Nov. 2013.

[24] R. Ratasuk, N. Mangalvedhe, and A. Ghosh, "Extending LTE coverage for machine type communications," in *IEEE Second World Forum on Internet of Things*, Milan, Italy, Dec. 2015.

[25] J. J. Nielsen, R. Liu, and P. Popovski, "Ultra-reliable low latency communication using interface diversity," *IEEE Transactions on Communications*, vol. 66, no. 3, pp. 1322–1334, March 2018.

[26] R. Ratasuk, N. Mangalvedhe, A. Ghosh, and B. Vejlgaard, "Narrowband LTE-M system for M2M communication," in *Proc. of IEEE 80th Vehicular Technology Conference*, Vancouver, Canada, Sep. 2014.

[27] H. Dhillon, H. Huang, H. Viswanathan, and R. Valenzuela, "Fundamentals of throughput maximization with random arrivals for M2M communications," *IEEE Trans. Commun.*, vol. 62, no.11, pp. 4094–4109, Nov. 2014.

[28] B. Wang, L. Chen, X. Chen, X. Zhang, and D. Yang, "Resource allocation optimization for device-to-device communication underlaying cellular networks," in *Proc. of the 73rd IEEE Vehicular Technology Conference*, Budapest, Hungary, May 2011.

[29] Z. Zhou, M. Dong, K. Ota, G. Wang, and L. T. Yang, "Energy-efficient resource allocation for D2D communications underlaying cloud-RAN-based LTE-A networks," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 428–438, June 2016.

[30] J. Jia, J. Chen, G. Chang, Y. Wen, and J. Song, "Multi-objective optimization for coverage control in wireless sensor network with adjustable sensing radius," *Computers and Mathematics with Applications*, vol. 57, no. 11-12, pp. 1767–1775, 2009.

[31] G. Zhang, K. Yang, and H. Chen, "Resource allocation for wireless cooperative networks: A unified cooperative bargaining game theoretic framework," *IEEE Wireless Communications*, vol. 19, no.2, pp. 38–43, Apr. 2012.

[32] C. Bishop, *Pattern Recognition and Machine Learning*. Secaucus, NJ: Springer, 2006.

[33] M. A. Alsheikh, S. Lin, D. Niyato, and H. P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 4, pp. 1996–2018, Apr. 2014.

[34] T. Cover, "Hypothesis testing with finite statistics," *The Annals of Mathematical Statistics*, vol. 40, no. 3, pp. 828–835, Jun. 1969.

[35] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[36] A. Paul, "Real-time power management for embedded m2m using intelligent learning methods," *ACM Transactions on Embedded Computing Systems*, vol. 13, no. 5s, pp. 148:1–148:22, Nov. 2014.

[37] T. Park and W. Saad, "Learning with finite memory for machine type communication," in *Proc. of the IEEE 50th Annual Conference on Information Science and Systems*, Princeton, USA, Mar. 2016.

[38] S. W. Hasinoff, "Reinforcement learning for problems with hidden state," University of Toronto, Toronto, Canada, Tech. Rep., Sep. 2003.

[39] A. Forster, "Machine learning techniques applied to wireless ad-hoc networks: Guide and survey," in *Proc. of 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, Melbourne, Austrailia, Dec. 2007.

[40] M. Hasan, E. Hossain, and D. Niyato, "Random access for machine-to-machine communication in LTE-Advanced networks: Issues and approaches," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 86–93, June 2013.

[41] V. Venkatasubramanian, M. Hesse, P. Marsch, and M. Maternia, "On the performance gain of flexible UL/DL TDD with centralized and decentralized resource allocation in dense 5G deployments," in *Proc. of IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication*, Washington DC, USA, Sep. 2014.

[42] M. Chen, Y. Miao, Y. Hao, and K. Hwang, "Narrow band internet of things," *IEEE Access*, vol. PP, no. 99, pp. 1–1, Sep. 2017.

[43] X. Chen, A. Benjebbour, A. Li, and A. Harada, "Multi-user proportional fair scheduling for uplink non-orthogonal multiple access (NOMA)," in *Proc. IEEE Veh. Technol. Conf.*, Seoul, Korea, May 2014, pp. 1–5.

[44] M. Al-Imari, P. Xiao, M. A. Imran, and R. Tafazolli, "Uplink non-orthogonal multiple access for 5G wireless networks," in *Proc. Int. Symposium on Wireless Commun. Systems*, Barcelona, Spain, Aug. 2014, pp. 781–785.

[45] J. Choi, "NOMA-based random access with multichannel ALOHA," *IEEE J. Sel. Areas in Commun.*, vol. 35, no. 12, pp. 2736–2743, Dec. 2017.

[46] ——, "Layered non-orthogonal random access with SIC and transmit diversity for reliable transmissions," *IEEE Trans. Commun.*, vol. 66, no. 3, pp. 1262–1272, Mar. 2018.

[47] Y. Liang, X. Li, J. Zhang, and Z. Ding, "Non-orthogonal random access for 5G networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4817–4831, Jul. 2017.

[48] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile internet of things: Can uavs provide an energy-efficient mobile architecture?" in *Proc. of the IEEE Global Communications Conference, Internet of Things Symposium*, Washington DC, USA, Dec. 2016.

[49] L. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[50] L. Rose, S. Perlaza, S. Lasaulce, and M. Debbah, "Learning equilibria with partial information in decentralized wireless networks," *IEEE Communications Magazine*, vol. 49, no.8, pp. 136–142, Aug. 2011.

[51] K. Drakopoulos, A. Ozdaglar, and J. N. Tsitsiklis, "On learning with finite memory," *IEEE Transactions on Information Theory*, vol. 59, no.10, pp. 6859–6872, May 2013.

[52] S. Bi, R. Zhang, Z. Ding, and S. Cui, "Wireless communications in the era of big data," *IEEE Communications Magazine*, vol. 53, no. 10, pp. 190–199, Oct. 2015.

[53] H. Malik, A. Malik, and C. Roy, "A methodology to optimize query in wireless sensor networks using historical data," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2, no. 3, pp. 227–238, 2011.

[54] R. Masiero, G. Quer, D. Munaretto, M. Rossi, J. Widmer, and M. Zorzi, "Data acquisition through joint compressive sensing and principal component analysis," in *Proc. of the IEEE Global Communications Conference*, Honolulu, HI, Dec. 2009.

[55] G. Quer, R. Masiero, G. Pillonetto, M. Rossi, and M. Zorzi, "Sensing, Compression, and Recovery for WSNs: Sparse Signal Modeling and Monitoring Framework," *IEEE Transactions on Wireless Communications*, vol. 11, no. 10, pp. 3447–3461, Oct. 2012.

[56] M. Chen, W. Saad, C. Yin, and M. Debbah, "Echo State Networks for Proactive Caching and Content Prediction in Cloud Radio Access Networks," in *Proc. of the IEEE Global Communications Conference, Workshop on 5G RAN Design*, Washington DC, USA, Dec. 2016.

[57] Y. Sharaf-Dabbagh and W. Saad, "On the authentication of devices in the internet of things," in *Proc. of 17th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Coimbra, Portugal, Jun. 2016.

[58] H. Nath and B. Mehtre, "Static malware analysis using machine learning methods," in *Recent Trends in Computer Networks and Distributed Systems Security*. Springer, 2014, pp. 450 – 450.

[59] S. Bikhchandani, D. Hirshleifer, and I. Welch, "A theory of fads, fashion, custom, and cultural changes informational cascade," *Journal of Political Economy*, vol. 100, no. 5, pp. 992–1026, Oct. 1992.

[60] L. Kaelbling, M. Littman, and A. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, May 1996.

[61] D. Athukoralage, I. Guvenc, W. Saad, and M. Bennis, "Regret based learning for uav assisted lte-u/wifi public safety networks," in *Proc. of the IEEE Global Communications Conference, Mobile and Wireless Networks Symposium*, Washington DC, USA, Dec. 2016.

[62] Z. Liu and I. Elhanany, "Rl-mac: A reinforcement learning based mac protocol for wireless sensor networks," *International Journal of Sensor Networks*, vol. 1, no. 3-4, pp. 117–124, 2006.

[63] K. Shah and M. Kumar, "Distributed independent reinforcement learning (dirl) approach to resource management in wireless sensor networks," in *Proc. of the IEEE International Conference on Mobile Adhoc and Sensor Systems*, Pisa, Italy, Oct. 2007.

[64] Y. Li, H. Ji, X. Li, and V. Leung, "Dynamic channel selection with reinforcement learning for cognitive wlan over fiber," *International Journal of Communication Systems*, vol. 25, no. 8, pp. 1077–1090, Aug. 2012.

[65] C. Camerer, T. Ho, and J. Chong, "A cognitive hierarchy model of games," *Quarterly Journal of Economics*, vol. 119, no. 3, pp. 861–898, 2004.

[66] B. Rogers, T. Palfrey, and C. Camerer, "Heterogeneous quantal response equilibrium and cognitive hierarchies," *Quarterly Journal of Economics*, vol. 144, no. 4, pp. 1440–1467, 2009.

[67] J. Chong, C. Camerer, and T. Ho, "Cognitive hierarchy: A limited thinking theory in games," in *Experimental Business Research*. Springer, 2005, pp. 203–228.

[68] W. Saad, Q. Zhu, T. Başar, Z. Han, and A. Hjørungnes, "Hierarchical network formation games in the uplink of multi-hop wireless network," in *Proc. of the IEEE Global Communications Conference, Wireless Networking Symposium*, Honolulu, HI, Dec. 2009.

[69] N. Abuzainab and W. Saad, "Cognitive hierarchy theory for heterogeneous uplink multiple access in the internet of things," to appear in *Proc. IEEE Symposium of Information Theory*, Barcelona, Spain, July 2016.

[70] M. Ali, E. Hossain, and D. Kim, "LTE/LTE-A random access for massive machine-type communications in smart cities," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 76–83, Jan. 2017.

[71] C. Yu, L. Yu, Y. Wu, Y. He, and Q. Lu, "Uplink scheduling and link adaptation for narrowband internet of things systems," *IEEE Access*, vol. 5, pp. 1724–1734, Feb. 2017.

[72] B. C. Arrue, A. Ollero, and J. R. M. de Dios, "An intelligent system for false alarm reduction in infrared forest-fire detection," *IEEE Intelligent Systems and their Applications*, vol. 15, no. 3, pp. 64–73, June 2000.

[73] D. Catania, M. Sarret, A. Cattoni, F. Frederiksen, G. Berardinelli, and P. Mogensen, "The potential of flexible UL/DL slot assignment in 5G systems," in *Proc. of IEEE 80th Vehicular Technology Conference*, Vancouver, Canada, Sep. 2014.

[74] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[75] M. Hassanalieragh, A. Page, T. Soyata, G. Sharma, M. Aktas, G. Mateos, B. Kantarci, and S. Andreescu, "Health monitoring and management using internet of things (IoT) sensing with cloud-based processing: Opportunities and challenges," in *Proc of IEEE International Conference on Services Computing*, New York, USA, Jun. 2015, pp. 285–292.

[76] M. Mozaffari, A. T. Z. Kasgari, W. Saad, M. Bennis, and M. Debbah, "Beyond 5G with UAVs: Foundations of a 3D wireless cellular network," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 357–372, Jan. 2019.

[77] M. Chen, W. Saad, and C. Yin, "Echo-liquid state deep learning for 360° content transmission and caching in wireless VR networks with cellular-connected UAVs," *IEEE Transactions on Communications*, vol. 67, no. 9, pp. 6386–6400, Sep. 2019.

[78] Y. Zhang, M. Chen, N. Guizani, D. Wu, and V. C. M. Leung, "SOVCAN: Safety-oriented vehicular controller area network," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 94–99, Aug. 2017.

[79] S. F. Abedin, M. G. R. Alam, S. M. A. Kazmi, N. H. Tran, D. Niyato, and C. S. Hong, "Resource allocation for ultra-reliable and enhanced mobile broadband iot applications in fog network," *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 489–502, Jan. 2019.

[80] T. Park and W. Saad, "Distributed learning for low latency machine type communication in a massive internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5562–5576, Jun. 2019.

[81] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proc. of IEEE International Conference on Computer Communications*, Florida, USA, Mar. 2012, pp. 2731–2735.

[82] B. Zhou and W. Saad, "Joint status sampling and updating for minimizing age of information in the internet of things," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7468–7482, Nov. 2019.

[83] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proc. of IEEE International Conference on Computer Communications*, Florida, USA, Mar. 2012, pp. 2731–2735.

[84] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Scheduling policies for minimizing age of information in broadcast wireless networks," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2637–2650, Dec. 2018.

[85] I. Kadota, A. Sinha, and E. Modiano, "Scheduling algorithms for optimizing age of information in wireless networks with throughput constraints," *IEEE/ACM Transactions on Networking*, vol. 27, no. 4, pp. 1359–1372, Aug. 2019.

[86] R. Talak, S. Karaman, and E. Modiano, "Minimizing age-of-information in multi-hop wireless networks," in *Proc. of 55th Annual Allerton Conference on Communication, Control, and Computing*, Illinois, USA, Oct. 2017, pp. 486–493.

[87] S. Leng and A. Yener, "Age of information minimization for wireless ad hoc networks: A deep reinforcement learning approach," in *Proc. of IEEE Global Communications Conference*, Hawaii, USA, Dec. 2019, pp. 1–6.

[88] R. Talak, I. Kadota, S. Karaman, and E. Modiano, "Scheduling policies for age minimization in wireless networks with unknown channel state," in *Proc. of IEEE International Symposium on Information Theory*, Colorado, USA, Jun. 2018, pp. 2564–2568.

[89] R. Talak, S. Karaman, and E. Modiano, "Optimizing age of information in wireless networks with perfect channel state information," in *Proc. of 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, Shanghai, China, May 2018, pp. 1–8.

[90] B. Zhou and W. Saad, "Minimum age of information in the internet of things with non-uniform status packet sizes," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 1933–1947, Mar. 2020.

[91] R. Talak, S. Karaman, and E. Modiano, "Distributed scheduling algorithms for optimizing information freshness in wireless networks," in *Proc. of IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications*, Kalamata, Greece, Jun. 2018, pp. 1–5.

[92] Y. Sun, E. Uysal-Biyikoglu, and S. Kompella, "Age-optimal updates of multiple information flows," in *Proc. of IEEE Conference on Computer Communications Workshops*, Hawaii, USA, Apr. 2018, pp. 136–141.

[93] R. D. Yates and S. K. Kaul, "The age of information: Real-time status updating by multiple sources," *IEEE Transactions on Information Theory*, vol. 65, no. 3, pp. 1807–1827, Mar. 2019.

[94] L. Huang and E. Modiano, "Optimizing age-of-information in a multi-class queueing system," in *Proc. of IEEE International Symposium on Information Theory*, Hong Kong, China, Jun. 2015, pp. 1681–1685.

[95] M. Costa, M. Codreanu, and A. Ephremides, "On the age of information in status update systems with packet management," *IEEE Transactions on Information Theory*, vol. 62, no. 4, pp. 1897–1910, Apr. 2016.

[96] Y. Inoue, H. Masuyama, T. Takine, and T. Tanaka, "The stationary distribution of the age of information in FCFS single-server queues," in *Proc. of IEEE International Symposium on Information Theory*, Aachen, Germany, Jun. 2017, pp. 571–575.

[97] E. Najm, R. Yates, and E. Soljanin, "Status updates through m/g/1/1 queues with harq," in *Proc. of IEEE International Symposium on Information Theory*, Aachen, Germany, Jun. 2017, pp. 131–135.

[98] M. K. Abdel-Aziz, C. Liu, S. Samarakoon, M. Bennis, and W. Saad, "Ultra-reliable low-latency vehicular networks: Taming the age of information tail," in *Proc. of IEEE Global Communications Conference*, Abu Dhabi, UAE, Dec. 2018, pp. 1–7.

[99] K. Chen and L. Huang, "Age-of-information in the presence of error," in *Proc. of IEEE International Symposium on Information Theory*, Barcelona, Spain, Jul. 2016, pp. 2579–2583.

[100] A. Maatouk, M. Assaad, and A. Ephremides, "Minimizing the age of information: Noma or oma?" in *Proc. of IEEE Conference on Computer Communications Workshops*, Paris, France, Apr. 2019, pp. 102–108.

[101] Y. Sun and B. Cyr, "Sampling for data freshness optimization: Non-linear age functions," *Journal of Communications and Networks*, vol. 21, no. 3, pp. 204–219, Jun. 2019.

[102] T. Park, W. Saad, and B. Zhou, "Centralized non-linear age of information minimization in the internet of things," in *submitted to IEEE Global Communications Conference*, 2020.

[103] A. Even and G. Shankaranarayanan, "Utility-driven assessment of data quality," *SIGMIS Database*, vol. 38, no. 2, p. 75–93, May 2007.

[104] S. Ioannidis, A. Chaintreau, and L. Massoulie, "Optimal and scalable distribution of content updates over a mobile social network," in *Proc. of IEEE International Conference on Computer Communications*, Rio de Janeiro, Brazil, Apr. 2009, pp. 1422–1430.

[105] A. S. Chakrabarti, C. B. K, A. Chatterjee, and M. Mitra, "The Kolkata Paise Restaurant Problem and Resource Utilization," *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 12, pp. 2420–2426, Jun. 2009.

[106] 3GPP, "Evolved universal terrestrial radio access (E-UTRA) and evolved universal terrestrial radio access network (E-UTRAN) (release 15)," TS 36.300 v15.8.0, Jan. 2020.

[107] M. A. Abd-Elmagid, N. Pappas, and H. S. Dhillon, "On the role of age of information in the internet of things," *IEEE Communications Magazine*, vol. 57, no. 12, pp. 72–77, Dec. 2019.

[108] R. D. Yates, E. Najm, E. Soljanin, and J. Zhong, "Timely updates over an erasure channel," in *Proc. of IEEE International Symposium on Information Theory*, Aachen, Germany, Jun. 2017, pp. 316–320.

[109] A. Mazin, M. Elkourdi, and R. D. Gitlin, "Comparison of slotted aloha-NOMA and CSMA/CA for M2M communications in iot networks," in *Proc. of IEEE 88th Vehicular Technology Conference*, Illinois, USA, Aug. 2018, pp. 1–5.

[110] W. Ejaz, M. Naeem, A. Shahid, A. Anpalagan, and M. Jo, "Efficient energy management for the internet of things in smart cities," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 84–91, Jan. 2017.

[111] T. Park and W. Saad, "Kolkata paise restaurant game for resource allocation in the internet of things," in *Proc. of 51st Asilomar Conference on Signals, Systems, and Computers*, California, USA, Nov. 2017, pp. 1774–1778.

[112] K. L. V. S. P. Sakuru and M. Visali, "Power control based resource allocation in LTE uplinks," in *Proc. of International Conference on Communications and Signal Processing*, Melmaruvathur, India, Apr. 2015, pp. 579–582.

[113] G. Hampel, C. Li, and J. Li, "5G ultra-reliable low-latency communications in factory automation leveraging licensed and unlicensed bands," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 117–123, May 2019.

[114] A. Koike, T. Ohba, and R. Ishibashi, "IoT Network Architecture Using Packet Aggregation and Disaggregation," in *Proc. of 5th International Congress on Advanced Applied Informatics*, Kumamoto, Japan, July 2016.

[115] Y. Chen and W. Wang, "Machine-to-Machine Communication in LTE-A," in *Proc. of IEEE Vehicular Technology Conference*, Ottawa, Canada, Sept. 2010.

[116] E. A. Jorswieck, "Stable Matchings for Resource Allocation in Wireless Networks," in *Proc. of 17th International Conference on Digital Signal Processing*, Corfu, Greece, Jul. 2011.

[117] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game Theory in Wireless and Communication Network: Theory, Models, and Applications*. Cambridge University Press, 2011.

[118] G. Bacci, S. Lasaulce, W. Saad, and L. Sanguinetti, "Game theory for networks: A tutorial on game-theoretic tools for emerging signal processing applications," *IEEE Signal Processing Magazine*, vol. 33, no. 1, pp. 94–119, Jan. 2016.

[119] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5G be?" *IEEE J. Sel. Areas in Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

[120] N. Panwar, S. Sharma, and A. K. Singh, "A survey on 5G: The next generation of mobile communication," *Physical Communication*, vol. 18, pp. 64 – 84, 2016.

[121] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 3949–3963, Jun, 2016.

[122] Z. Ding, Y. Liu, J. Choi, Q. Sun, M. Elkashlan, C. L. I, and H. V. Poor, "Application of non-orthogonal multiple access in LTE and 5G networks," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 185–191, Feb. 2017.

[123] A. Benjebbour, Y. Saito, Y. Kishiyama, A. Li, A. Harada, and T. Nakamura, "Concept and practical considerations of non-orthogonal multiple access (NOMA) for future radio access," in *Proc. Int. Symposium on Intelligent Signal Processing and Commun. Systems*, Naha, Japan, Nov. 2013, pp. 770–774.

[124] T. Park, N. Abuzainab, and W. Saad, "Learning how to communicate in the internet of things: Finite resources and heterogeneity," *IEEE Access*, vol. 4, pp. 7063–7073, Nov. 2016.

[125] S. R. Islam, N. Avazov, O. A. Dobre, and K.-S. Kwak, "Power-domain non-orthogonal multiple access (NOMA) in 5G systems: Potentials and challenges," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 721–742, Secondquarter 2017.

[126] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *arXiv preprint arXiv:1902.10265*, 2019.

[127] M. Shirvanimoghaddam, M. Dohler, and S. J. Johnson, "Massive non-orthogonal multiple access for cellular iot: Potentials and limitations," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 55–61, Sep. 2017.

[128] X. Zhang and M. Haenggi, "The performance of successive interference cancellation in random wireless networks," *IEEE Transactions on Information Theory*, vol. 60, no. 10, pp. 6368–6388, Oct 2014.

[129] A. E. Mostafa, Y. Zhou, and V. W. Wong, "Connectivity maximization for narrowband iot systems with NOMA," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, Paris, France, May 2017, pp. 1–6.

[130] M. Baghani, S. Parsaeefard, M. Derakhshani, and W. Saad, "Dynamic non-orthogonal multiple access (noma) and orthogonal multiple access (oma) in 5g wireless networks," *arXiv preprint arXiv:1811.11090*, 2018.

[131] Z. Yang, Z. Ding, P. Fan, and N. Al-Dhahir, "A general power allocation scheme to guarantee quality of service in downlink and uplink noma systems," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7244–7257, Aug. 2016.

[132] S. Chen, K. Peng, and H. Jin, "A suboptimal scheme for uplink NOMA in 5G systems," in *Proc. of Int. Wireless Commun. and Mobile Computing Conf.*, Dubrovnik, Croatia, Aug. 2015, pp. 1429–1434.

[133] S. A. R. Naqvi and S. A. Hassan, "Combining NOMA and mmWave technology for cellular communication," in *Proc. of IEEE 84th Vehicular Technology Conference*, Montreal, QC, Canada, Sep. 2016, pp. 1–5.

[134] J. N. Chattha and M. Uppal, "Relay-aided non-orthogonal multiple access with noisy network coding," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, Paris, France, May 2017, pp. 1–6.

[135] L. Salaün, C. S. Chen, and M. Coupechoux, "Optimal joint subcarrier and power allocation in NOMA is strongly NP-hard," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, Kansas City, USA, May 2018, pp. 1–7.

[136] C. Wang, Y. Chen, Y. Wu, and L. Zhang, "Performance evaluation of grant-free transmission for uplink URLLC services," in *Proc. IEEE Veh. Technol. Conf.*, Sydney, Australia, June 2017, pp. 1–6.

[137] T. Park, G. Lee, and W. Saad, "Message-aware uplink transmit power level partitioning for non-orthogonal multiple access (NOMA)," in *Proc. of IEEE Global Commun. Conf.*, Abu Dhabi, UAE, Dec. 2018, pp. 1–6.

[138] E. Balevi, F. T. A. Rabee, and R. D. Gitlin, "ALOHA-NOMA for massive machine-to-machine IoT communication," in *Proc. of IEEE International Conference on Communications*, Missouri, USA, May 2018, pp. 1–5.

[139] M. Basharat, W. Ejaz, M. Naeem, A. M. Khattak, A. Anpalagan, and O. Alfandi, "Energy efficient resource allocation for NOMA in cellular iot with energy harvesting," in *Proc. of 13th International Conference on Emerging Technologies*, Islamabad, Pakistan, Dec. 2017, pp. 1–6.

[140] D. Zhai, R. Zhang, L. Cai, B. Li, and Y. Jiang, "Energy-efficient user scheduling and power allocation for NOMA-based wireless networks with massive IoT devices," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1857–1868, Jun. 2018.

[141] R. Abozariba, M. K. Naeem, M. Patwary, M. Seyedebrahimi, P. Bull, and A. Aneiba, "NOMA-based resource allocation and mobility enhancement framework for IoT in next generation cellular networks," *IEEE Access*, vol. 7, pp. 29 158–29 172, Jan. 2019.

[142] M. Shirvanimoghaddam, M. Condoluci, M. Dohler, and S. J. Johnson, "On the fundamental limits of random non-orthogonal multiple access in cellular massive IoT," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 10, pp. 2238–2252, Oct. 2017.

[143] D. Zwillinger, *Table of integrals, series, and products*. Elsevier, 2014.