Fall 11-1-2019

Satisfaction-Aware Data Offloading in Surveillance Systems

Marcos Paul Torres

Marcos Paul Torres

Candidate

Electrical and Computer Engineering

Department

This thesis is approved, and it is acceptable in quality and form for publication: *Approved by the Thesis Committee:*

10/18/2019

Dr. Eirini Eleni Tsiropoulou, Chair

10/24/19

Dr. Michael Devetsikiotis, Member

10/18/2019

Dr. Jim Plusquellic, Member

Satisfaction-Aware Data Offloading in Surveillance Systems

by

Marcos Torres

B.S, University of New Mexico, 2018

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Computer Engineering

The University of New Mexico

Albuquerque, New Mexico

December, 2019

Dedication

To my parents, Mark and Michele Torres, and fiancé Avery, who have constantly supported, loved, and encouraged me throughout my graduate degree.

Acknowledgments

I want to thank my advisor, Professor Eirini-Eleni Tsiropoulou, for the opportunity to pursue my interests in fully autonomous aerial systems and for providing me the room to make mistakes to learn, but being there for me when I needed guidance. I also want to acknowledge both Dr. Plusquellic and Dr. Devetsikiotis for being apart of my thesis committee.

I would also like to thank my friends from the PROTON lab for their help, especially when I was new to this type of research environment. I want to say thank you to Pavlos Athanasios Apostolopoulos for being my research partner and for the technical help during this thesis, Georgios Fragkos, for his expertise in software development and guidance in finding related works, and Nathan Patrizi for his technical pieces of advice and help.

Finally, I want to express my gratitude towards my parents, brother, and fiancé, for providing me with unfailing support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without them. Thank you.

Satisfaction-Aware Data Offloading in Surveillance Systems

by

Marcos Torres

B.S, University of New Mexico, 2018

M.S., Computer Engineering, University of New Mexico, 2019

Abstract

In this thesis, exploiting Fully Autonomous Aerial Systems' (FAAS) and Mobile Edge Computing (MEC) servers' computing capabilities to introduce a novel data offloading framework to support the energy and time-efficient video processing in surveillance systems based on satisfaction games. A surveillance system is introduced consisting of Areas of Interest (AoIs), where a MEC server is associated with each AoI, and a FAAS is flying above the AoIs to support the IP cameras' computing demands. Each IP camera adopts a utility function capturing its Quality of Service (QoS) considering the experienced time and energy overhead to offload and process remotely or locally the data. A non-cooperative game among the cameras is formulated to determine the amount of offloading data to the MEC server and/or the FAAS, and the novel concept of Satisfaction Equilibrium (SE) is introduced where the IP cameras satisfy their minimum QoS prerequisites instead of maximizing their performance by consuming additional system resources. A distributed learning algorithm determines the IP cameras' stable data offloading. Also, a reinforcement learning algorithm indicates the FAAS's movement among the AoIs exploiting the

accuracy, timeliness, and certainty of the collected data by the IP cameras per AoI. Detailed numerical and comparative results are presented to show the operation and efficiency of the proposed framework.

This work has been published in:

P.A. Apostolopoulos, M. Torres, and E.E. Tsiropoulou, ”Satisfaction-aware Data Offloading in Surveillance Systems,” in ACM MOBICOM WKSHPS: CHANTS2019: 14th Workshop on Challenged Networks, 2019

Contents

List of Figures

List of Tables

Glossary

T	Set of timeslots where T = $\{1, ..., t, ..., T\}$
A	Set of area of Interest where A = $\{1, ..., i, ..., A\}$
Z_i	Randomly placed coordinates for AoI, $Z_i = (X_i, Y_i)$
C_i	Cameras in area i where C$_i$ ={1,...,j,...,C$_i$}
M_i	Mobile Edge Computer in area i
$D_{ij}^{(t)}$	Set of data collected from camera j in area i
$B_{ij}^{(t)}$	Total collected bits from camera j in area i
$CP_{ij}^{(t)}$	Required CPOU cycles to process data of camera j in area i
$\phi_{ij}^{(t)}$	Intensity of video processing of camera j in area i
$dt_{ij}^{(t)}$	Time constraint of camera j in area i
$de_{ij}^{(t)}$	Energy constraint of camera j in area i
s_i	Set of vector strategies for the IP cameras
a_{ij}	Camera data offloading percentage
ch_{ij}	Camera offloading either to MEC or FAAS

Glossary

$_{ij}$	Uplink data rate
W_i	AoI i bandwidth
p_{ij}	Transmission Power of camera j in area i
g_{ij}	Channel gain between the camera and MEC server
$O_{ij}^{tr,t}$	Transmission time overhead
$O_{ij}^{tr,e}$	Transmission energy overhead
f_{M_i}	MEC computing capability
f_F	FAAS computing capability
$O_{ij}^{p,t}$	Processing time overhead
O_{ij}^{t}	Total time overhead
O_{ij}^{e}	Total energy overhead
u_{ij}	Utility function
G_i	Game satisfaction form
a_{ij}^{n}	The n^th available offloading percentage
h_{ij}	Satisfaction correspondence
AE_{ij}	Event that is detected
TE_{ij}	Total number of events that are detected
q_{ij}^{acc}	Quality - Accuracy parameter
q_{ij}^{t}	Quality - Timeliness parameter

q_{ij}^{cer}	Quality - Certainty parameter
w_i^{acc}	Weight - Accuracy parameter
w_i^t	Weight - Timeliness parameter
w_i^{cer}	Weight - Certainty parameter
E^p	FAAS processing energy of the received data
E^p	FAAS movement energy
rw_i	Reward function
QoI_i	Quality of Information from our accuracy, timeliness, and certainty parameters
E	FAAS total energy
P_i	Performance of satisfied cameras
π_{ij}	Probability of camera j to choose a certain action in area i
L_{ij}	Number of cameras offloading strategies
U_{ij}	Maximum utility that each camera j can perceive

Chapter 1

Overview

1.1 Introduction

Surveillance systems have recently gained considerable attention due to the increased number of terrorist attacks, natural disasters, and everyday security issues. All of which challenge our public safety and homeland security [1]. With the advent of the Internet of Things (IoT), the smart Internet Protocol (IP) cameras have enabled the surveillance systems to capture real-time video and process it locally [2] or remotely at the cloud computing environment. These cameras can observe and record certain behaviors or activities, making an area known to city officials that more or less help is needed in the area. For a system to make the right decisions, useful information must be provided, and the feedback has to be real-time. This allows for emergency responders to be more informed, and by this everyone involved will be safer. However, the surveillance systems confront the challenges of increased computing demand to process the recorded information and provide useful data from it. Even with the advances of camera systems, the cameras cannot offload all their data locally to be processed. The concepts of making use of Mobile Edge Computing Servers can help ease the increasing demand challenges, but even with increased capabilities, users may not be satisfied.

The problem of resource management in Public Safety Systems (PSS) and Public Safety Networks (PSNs) has gained great interest in the recent literature in order to safeguard citizens and improve rescue operations [3]. In [4], the problem of socio-physical and mobility-aware coalition formation among the trapped users and the first responders in public safety networks is addressed towards guaranteeing users' connectivity, stability, and energy-efficient communication. The energy-efficient operation of the PSNs assisted by UAVs has been further studied in [5] considering multiple types of devices communicating with the UAV. In [6], a novel evacuation-planning mechanism is introduced to support the distributed and autonomous evacuation process within the operation of a public safety system based on reinforcement learning and game theory. In [7, 8], the public safety network is assisted with Unmanned Aerial Vehicles (UAVs), and the problem of energy-efficient operation and communication of the IoT devices in the disaster area with the UAV is studied. Moreover, the battery life of the IoT devices in the disaster area is extended by adopting the wireless powered communication (WPC) techniques by the UAVs to charge the devices [9, 10, 11]. Towards providing resilient communications among the users and the first responders during a disaster, the concept of wireless protocol selection has also been proposed autonomously based on reinforcement learning techniques [12, 13].

However, all the approaches mentioned above provide efficient and reliable solutions during or after a disaster event in PSSs and PSNs without contributing to the process of predicting an upcoming disaster. Thus, the challenging problem of supporting PSSs, PSNs, and in general, smart cities environments, with solutions that can proactively safeguard them and support their smooth operation, while detecting potential threats becomes a real need rather than a desire. Towards this direction, the analysis of the content collected by surveillance systems can provide useful information to the Emergency Control Centers (ECCs) in smart city environments.

In recent years, mobile edge computing (MEC) has gained popularity in order to offload data real-time and analyze it for use. MEC is a network architecture that runs applications and processes tasks closer to the user so that congestion decreases, and thereby, performance increases [14, 15]. In our situation, cameras process data locally or offload to the MEC servers for data processing depending on if the need is there. After offloading, and looking at different parameters, for example, time or energy constraints, the cameras verify if they are ”satisfied” on the performance of offloading to the MEC servers or processing locally. The term ”satisfaction” is used to express if the cameras have satisfied their Quality of Service (QoS) requirements. The function to determine satisfaction will be discussed further in the system model. Even with assistance from the MEC servers, the cameras still may find themselves unsatisfied based on time or energy usage. This could have been because certain areas may have high critically, congestion or energy requirements, and needs more offloading power. Another tool that can be utilized is an unmanned aerial vehicle (UAV) or commonly known as drone with computing capabilities. Many communication providers are moving towards mounting UAVs with base stations or computing devices to offload data from users [16]. These drones provide a dynamic solution to providing coverage for areas [17].

Along with this has been a push to make these drones fully autonomous meaning there would be no human in the loop. Providers would set node destinations, and drones could get themselves around using their algorithms. These UAVs are known as fully autonomous aerial systems (FAAS). Utilizing this drone type, we can find different critical areas, and our drones can dynamically move to these spots to provide more assistance to satisfy more cameras. In order to dynamically move the FAAS our problem employs a reinforcement learning technique to learn the environment. Reinforcement learning is a machine learning technique in which rewards are given after an agent (in our situation, the FAAS) makes a decision and ends up in a new state [18]. All the state and actions will be tried out by the agent, and after it sufficiently learns the environment, it will maximize its reward by choosing the path that gave it the most substantial reward. After sufficient time has passed, the FAAS will proficiently know its environment and explore areas that in the past, have provided more critically in terms of data and need. The full discussion of the reinforcement learning algorithm and use is discussed fully in the system model.

1.2 Motivation

Public safety is a high priority in cities across the world, and every day new techniques are being created to analyze risks to the population more proficiently. Surveillance systems specifically, camera systems, have gained wide popularity for monitoring activities and behavior. Using all the information gathered, cities can better assess threats that may correspond to what these cameras are seeing. One crucial factor is processing this data, and in order for it to be useful, it must happen in real-time. To achieve this, authors in [19] have introduced an image uploading process from the IP cameras to the cloud, where the images captured by the cameras are stored and processed at cloud to decrease the cost of storing and processing the information locally. This allows the data to be processed and analyzed to become useful to the system. As mentioned above, the processing capabilities for these IP cameras need assistance to be useful. In [20], a drone-assisted surveillance system is studied, where the videos captured by the drone are forwarded to Fog Computing nodes through the drones' ground controller, in order to be processed and track vehicles' movement. Following a similar philosophy, in [21] an Unmanned Aerial Vehicle (UAV)-based crowd surveillance system is introduced where the UAVs capture videos that either offload to MEC servers for further processing or process them on board. The authors discuss the drawbacks and benefits of the two choices in terms of energy consumption and processing time. In [22], the drones' video capturing capability is exploited to track the moving target object, and the drones offload part of the computing tasks to a control center, while the rest are executed locally at the drone.

It is evident that the UAVs (the drones) and the remote computing capabilities (i.e., Cloud, MEC, and Fog Computing) have improved the performance of the surveillance systems. They allow for fast processing which makes this more useful to the user and environment. However, the UAVs and the drones still require human control from the ground to indicate the path that they follow during their flight. To address this issue, the FAAS has been recently introduced in the robotics and automation research field [23]. The FAAS is a flying robotic system equipped with sensors, surveillance systems, computing resources, wireless communication interfaces, or any combination of them and is able to operate fully autonomously with no human intervention. This increases the dynamism of the aerial system in which they are self proficient in covering an area. A user would drop specific nodes, and the system will create its flight plan.

In all the aforementioned approaches, each entity involved in the surveillance system (i.e., IP cameras) aims at maximizing its Quality of Satisfaction (QoS). The improvement of those entities' QoS is proportional to the consumption of communication and computing resources. However, a surveillance system is a resource-constrained setting; thus the maximization of each involved entity's QoS is a sub-optimal solution. Towards this direction, the games in satisfaction form have been introduced in the field of Game Theory [24], where the autonomous entities aim to "satisfy" their minimum QoS prerequisites in a distributed manner instead of targeting at maximizing their QoS, and have been applied in the uplink power control problem in wireless cellular networks [25, 26]. In this thesis, the FAAS's and the MEC servers' computing capabilities are exploited to introduce a novel data offloading framework based on the satisfaction games in order ultimately to support the energy and time-efficient video processing in surveillance systems consisting of IP cameras. Understanding this satisfaction amongst the system can be used in many real-world applications, specifically public safety. If these cameras can real-time map out where hot-spots are for illegal activity or unsafe areas, emergency responses can be more precise and accurate in these areas and deter individuals from performing these acts. With this increase in information, entire cities will be safer.

1.3 UAVs for Wireless Networks: Applications, Challenges, and Open Problems

As mentioned above, the use of flying systems (i.e. UAVs, FAASs) is gaining popularity amongst the communications community because of their attributes such as mobility, flexibility, and usability. Significant research has been performed in the area of aerial systems and their contribution to our wireless communication network architecture. A leading contributor on the push for UAV systems is the cost benefits, building and maintaining complete cellular infrastructure is expensive, whereas deploying UAVs become beneficial because it removes the expensive towers and deployment of infrastructure. Application for drone use include military, surveillance, telecommunication, delivery systems, and rescue operations [27]. This conventional research was typically focused on controls and navigation aspects of the UAV system. Communication challenges of these aerial devices have typically been ignored or part of the control and autonomy components, this is not always a fair assumption.

Proper use of UAVs for specific wireless networking application, several factors such as the capabilities and their flying altitudes must be taken into account. Different classifications exist and depending on the needs and goals, selecting the appropriate type of UAV that meets the various requirements set forth will be needed. UAVs are generally classified depending on the altitude they achieve. High altitude platforms (HAPs) have altitudes of 17km and generally quasi-stationary [28], meaning there is little movement when the drones altitude is achieved. Low altitude platforms (LAPs) are generally smaller and range from tens of meters to a few kilometers. LAPs can quickly move and are more flexible than HAPs. It is important to note that in the United States, federal aviation regulations exist restricting some attributes of the drones (i.e., maximum allowable altitude for a LAP without permit is 400 above ground level [29]). Deployment of LAP is more rapid and used in time-critical applications like data collection or emergency situations. HAPs are designed for long term operations, for example, providing wireless coverage for rural areas. UAVs are also classified based on type of wing, fixed-wing or rotary-wing UAVs. Fixed-wing UAVs generally weight more, but move faster whereas rotatory wing UAVs can hover and remain stationary. Authors in [16] provided an infographic in order to see the full overview of UAV classifications.

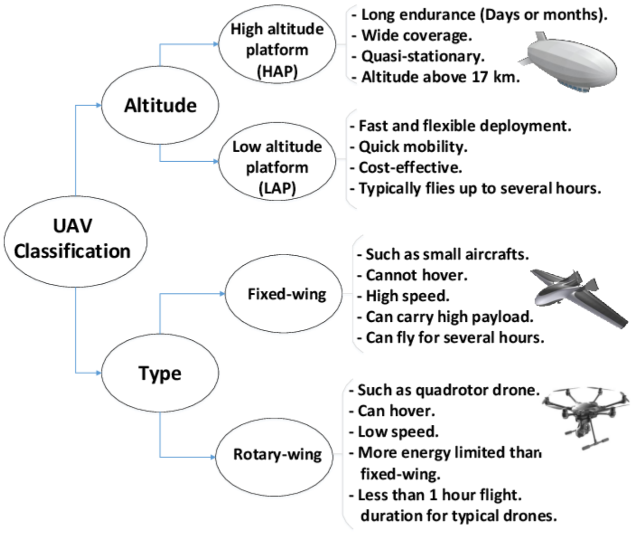
UAV Classification
Altitude
High altitude platform (HAP)
- Long endurance (Days or months).
- Wide coverage.
- Quasi-stationary.
- Altitude above 17 km.
Low altitude platform (LAP)
- Fast and flexible deployment.
- Quick mobility.
- Cost-effective.
- Typically flies up to several hours.
Type
Fixed-wing
- Such as small aircrafts.
- Cannot hover.
- High speed.
- Can carry high payload.
- Can fly for several hours.
Rotary-wing
- Such as quadrotor drone.
- Can hover.
- Low speed.
- More energy limited than fixed-wing.
- Less than 1 hour flight. duration for typical drones.

Figure 1.1: UAV Classification

If deployed correctly, UAVs can provide cheap and dependable solutions for wireless communications. In communications, UAVs generally fall into two application categories. The first application being the UAV acts as an aerial base station or the second, where it is a cellular-connected device. An aerial base station drone is equipped to provide dependent and cost-effective wireless service to users on demand. Cellular connected UAVs, on the other hand, coexist with users for application. These applications could be delivery systems, surveillance, or search and rescue. Both classes of UAV applications are exciting, but come with many challenges that must be researched. Open problems to this day exist, but with extended amounts of research, UAVs have a promising future in the field of wireless communications.

1.3.1 UAVs for Wireless Networks: Applications

The first class of applications for UAV wireless networks is aerial base station UAV, which is a system that is equipped with a small cell in order to facilitate mobile network providers to assist users. These UAVs are helpful because they can dynamically adjust their location to handle hotspots that pop up and to have a better line of sight for devices. Specifically, these drones are used in order to relay information from the BS to users and vice versa, as seen in Figure 1.2. The implementation of this base station architecture is becoming more popular amongst the communications community because of the unique advantages they hold, such as being faster and cheaper to deploy, having more flexibility in reconfiguration, and having better communication channels. The first application described in [16] for aerial base station UAV is coverage and capacity enhancement for wireless networks for the fifth generation cellular architectures. Over the last few years, there has been a huge growth in connected devices to our network, such as smart-phones, tablets, and IoT gadgets. By this increase, the capacity and coverage of existing networks are being strained. This led to the need for the fifth generation of wireless communications that plans to include device-to-device (D2D) communications, small cell networks, and millimeter-wave (mmW) communications [30]. Even with these new solutions, all aspects hold their own limitations. For example, small cell networks face challenges, including backhauling, interference amongst the network, and network modeling architectures. Challenges exist in D2D, as well as mmW. These challenges further call for the need for UAV base station scenarios. UAVs provide the on-demand connectivity, traffic offloading schemes, and high data rates that are essential for all the new devices that are connecting everyday [31]. Moreover, drone base stations are promising because of the potential of providing wireless connectivity in geographical areas with limited cellular infrastructure and providing additional coverage to temporary events (concerts or sporting events). The second application for aerial base stations is the use of public safety scenarios. Natural disasters such as floods, severe snowstorms, hurricanes are devastating to city or country infrastructures. Both hurricane Sandy and Irma showed evidence of existing terrestrial communication network damage, which led to communication overloading and loss of wireless service [32]. In these scenarios, there is a vital need for public safety communication between emergency responders and victims. The need for flexible, low-latency, and swift adapting systems is of great importance in these situations.

UAV based aerial networks can provide these services because they do not require highly constrained or expensive infrastructure, a drone connects to the users and the base station to provide services. These systems can easily fly and dynamically change positions to provide on-demand coverage for responders and victims. In 2017, AT&T's deployment of their "Cell on Wings" in the Puerto Rico natural disaster situation, drones were deployed to help the access node deliver network traffic to mobile users or act as a relay to download traffic from the access node to mobile users [33]. Many other applications exist for aerial base stations including UAV

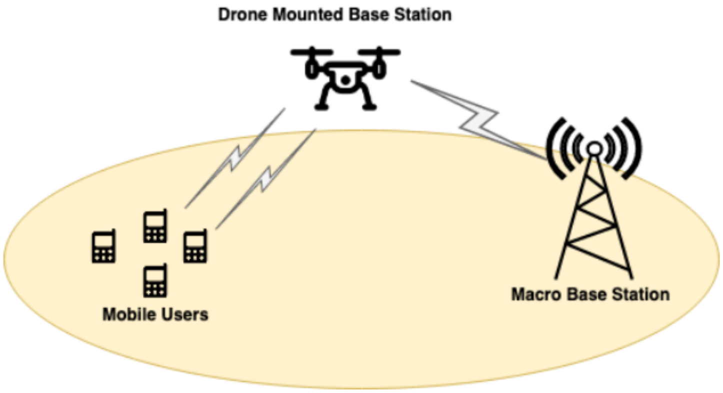

Figure 1.2: Example of UAV mounted base station

assisted terrestrial networks for information dissemination, 3D MIMO and Millimeter Wave Communications, UAVs for IoT Communications, and Cache-Enabled UAVs. These topics can be explored further and described in [16]. Many prototypes of these wireless systems have been developed and deployed. Sprint's "Magic Box" was initially launched in 2017 that could extend their 2.5 GHz data service up to 10 square miles. Facebook, Google, and more companies have released aerial base stations as well.

As mentioned above, the second class of applications for UAV wireless network deployments fall under cellular-connected devices as user equipment. Connecting and providing service is just one aspect of how UAVs can be helpful to users, another application of connected drone is using them as a tool for surveillance, sensing, virtual or augmented reality systems, or package delivery. Cellular connected drones will be a crucial enabler for connecting the trend of IoT devices, for example, Amazon is heavily considering the use of drones for delivering packages but must these systems connected to the network for notification and diagnostics [34]. Cellular connected drones potentially could also be used a surveillance systems for government entities. Equipping a UAV with a camera or sensor can be very beneficial. The flexibility of the system can provide surveillance and security from just about anywhere. Governments are also able to use these drones in search and rescue operations. In natural disasters or terrorist attacks, the deployment of drones with sensors will allow rescuers to find survivors more proficiently. As mentioned above, drones have the ability to move in all directions quickly and have algorithms to optimize their routes. The next application discussed in this thesis is the use of these types of drones in smart cities. In this situation, drones are not only used for surveillance or data collection, but can also be utilized as a mobile cloud computing system [35]. The UAV mounted cloudlet would be able to perform computationally intensive tasks pushed by the IoT city devices. Although we see many different application settings, drones still have many challenges that must be pursued and solved before drones are deployed permanently. Both cellular-connected drones and aerial base stations need reliable and low latency communications. Theses drones require the need for a sizeable cellular infrastructure in order to control the devices properly for them to meet their tasks. Drones also have energy limitations due to basic flight, adding in application devices deplete the battery even faster.

1.3.2 UAVs for Wireless Networks: Challenges

As briefly mentioned above, although there are many benefits to UAV drone deployment for wireless communications, there are some obstacles and challenges that are needed to be resolved before implementation. The first challenge discuss, is the wireless signal prorogation, which is affected by the medium between the transmitter and receiver. The Air-to-Ground (AtG) channel modeling characteristics for UAV-based communications differ significantly from classic approaches which affects both the coverage and capacity of the communication [36]. Any movement or vibration from the UAV can affect the channel. Height, angle, type of UAV, and environment are all conditions that must be considered when created the communication channel. If even one of these conditions is not met, the UAV will be susceptible to communication issues. On account of these conditions, it is essential to optimize the design and deployment of drone-based communication. This requires accurate AtG channel models, especially in UAVs applications such as coverage enhancement, or cellular-connected UAVs. Many different techniques have been proposed, but all lack in solving all issues described. An example of a solution is ray-tracing, which is an excellent approach for modeling the channels, but doesn't have precise accuracy and performs poorly at lower frequency operations [37].

Another critical challenge discussed is the three-dimensional deployment of UAVs. The complexity of deployment occurs from the fact that drones have many different factors in a deployment environment. Geographical area plays a role, location of users, and AtG channel characterization must be considered when deploying a drone. This topic has received significant attention from researchers in [5, 38, 39]. As a fact, deployment is a crucial design attribute while using a UAV for public safety, caching, smart cities, and coverage and capacity maximization. All the mentioned papers, are great starts to this research topic and try to optimize some sort of parameter, but still, with all this research, it is a challenge to deploy drones precisely.

One of the most worrisome challenges to researchers is UAV resource management and energy efficiency constraints. All applications depend on resource management or energy efficiency and sometimes even both. Resource management is an open problem in terrestrial networks, and UAVs provide unique and new challenges due to many factors: 1) UAV flight time, path plan, spectral efficiency, 2) flight limitations and stringent energy, 3) Line of sight interference from ground or air links, and 4) mobility of the UAV. Hence, there is a need for managing these resources optimally in order for UAV-assisted wireless networks to be effective [40]. In the work of [41], the proposed resource allocation scheme researched was a cache-enabled UAV that would effectively service users over licensed and unlicensed bands. In [42], the optimal resource allocation was researched for an energy harvesting flying access point. All of the researched schemes are great and should be utilized. Choosing the correct resource allocation scheme is very tricky and must be considered while developing research involving UAV wireless communication systems. Performance of a UAV communication system is affected by the length of the flight time (battery life) and is dependent on many factors such as energy source, weight, speed, and trajectory. In the table 1.1 below, examples of battery lifetime provided by [43], is presented.

Size	**Weight**	**Example**	**Battery Life**
Micro	$< 100g$	Kogan Nano Drone	6-8 min
Very small	100g - 2kg	Parrot Disco	45 min
Small	2kg - 25kg	DJI Spreading Wings	18 min
Medium	25kg - 150kg	Scout B-3330 UAV helicopter	180 min
Large	$> 150kg$	Predator B	1800 min

Table 1.1: Battery Lifetimes of UAVs

Energy consumption of wireless communication UAVs are generally categorized in two ways [43]. The first being communication-related energy and the second being propulsion energy. Communication energy is energy expended when the drone is transmitting a signal, computation of signals, or processing received signals. Propulsion energy is the energy exerted when the drone is moving or hovering. Generally, propulsion energy consumption is more significant than communication energy consumption. Without coming up with solutions, drones cannot be utilized efficiently if they always have to be recharged.

Other challenges exist in the field, but the most prominent ones are described above. If interested in the other problems wireless UAVs hold and the state of the art research being completed in order to overcome these challenges, please explore [16]. In the next section, open problems for the challenges described above are explored.

1.3.3 UAVs for Wireless Networks: Open Problems

State of the art research is being done revolving around the challenges that wireless communication drones face. Countless researchers in academia, commercially, and government agencies around the world are actively looking into optimizing drones to provide wireless connectivity solutions. Algorithms, models, and test-beds have been created to test these systems. Even with all this research, many open problems still exist for wireless signal propagation, UAV deployment, and resource management and energy efficiency constraints.

The challenge of channel modeling is still a huge research topic in not only UAV wireless communications, but also just in terrestrial networks. Tools and techniques to solve these mentioned issues are machine learning algorithms, extensive measurements, and ray tracing. The flexibility that a UAV provides through flight makes the problem of signal propagation more complex to solve. The foremost problem in air-to-ground channel modeling is the need for more realistic channels that come from real measurements, not simulated [44]. Efforts are currently being made, but mainly in single UAV situations wherein the long term, this is not effective. Furthermore, the environments that have been explored are very particular to the application. The need for the UAV to function whether it be an urban, suburban, rural, or indifferent weather conditions like wind, or rain must be explored, and these channels must be researched at length. These efforts would nicely complement the research already being conducted like ray tracing. Additionally, another interesting problem is that if drones are planned to be used as base stations, or user equipment, the need for air-to-air channeling modeling must be analyzed. Lastly, one must consider the UAV altitude or antenna's position while looking into multipath fading [45].

The second open problem that needs addressing is optimal drone deployment. Techniques for optimal deployment based on line of sight (LoS), spectral efficiency, others have actively been researched. The need for optimal deployment is still a hot research topic among the community. The need for new solutions that are unique features to drones such as 3-D position relative to terrestrial networks. There is still a need to study how UAVs must be deployed in existence with our current architecture in order to consider interference between the aerial system and ground systems. There are three primary deployment problems, the first being optimized deployment for bandwidth allocation for low latency communications. Finding the optimal location for each drone base station to allocate its bandwidth using the smallest latency is a must. The second deployment open problem is optimally placing the drone and cells for flight time minimization. The flight times can be affected significantly by the load of the data being transmitted and the number of users connected. Minimizing the number of users each drone handles should be optimally deployment in order to preserve the battery life to efficiency support users for more extended periods. The last open problem discussed related to deployment is obstacle aware UAV deployment to maximize wireless coverage. In high-frequency communication bands, this becomes a fundamental problem to handle. To maximize the total coverage area, one must consider not only the location of the users, but the obstacles as well. In particular, 3-D positions of drones base stations can be determined such that the max number of users is covered.

The most prominent open problem in wireless drone communication systems is resource allocation and energy usage of the UAV. The need for a dynamic architecture to handle drone positions, bandwidth, transmitter power and energy, and the amount of users each drone can handle are issues that must be solved by resource management. The resource management issue comes from how drones try to adaptively adjust their transmit power and trajectory that serves ground users. In this case, the problem is to optimize bandwidth allocation that capture the UAVs' location mobility, and line of sight interference. The need for designing efficient scheduling algorithms and mitigation of interference in the cellular network are pressing issues.

Novel research is currently being performed because in every situation, a drone is used to provide a wireless application, both scheduling and interference must be efficient. If not, the drone can not provide adequate service to the users. The last open problem discussed in this thesis is energy efficiency. The critical problem is that in order to be effective, we have to maximize the flight time so that UAV base stations can support users. As seen in table 1.1, small drones and below have less than 30 minutes of flight time. In order to maximize these flight times, research is being done in both resource allocation and deployment areas. It is known that the propulsion energy consumption is more costly to the drones battery life; optimal drone deployment is a critical open research topic as discussed above. Key developments in improving battery performance, making lighter more powerful drones, and more efficient drones are all important while solving the issues revolving around drone energy.

1.4 Contributions

Now that we have an understanding of UAV wireless communication systems work and applications/ challenges that still exist in the area, we discuss the novel research that has been conducted in this thesis. We first introduce a surveillance system consisting of areas of interest (AoI) with IP cameras; a few examples of these areas could be banks, schools, hospitals. In these areas, the cameras partially offload computing tasks related to the videos' processing to the MEC server that is associated with the AoI or to the FAAS, if the FAAS is flying above that specific AoI. The rest of the data the camera has will executed locally at the IP cameras. Each IP camera experiences a time and energy overhead in order to offload its data and process a part of the data locally. A holistic utility function is introduced representing the IP cameras' level of achieved QoS, while accounting for their time and energy constraints that they possess in the video processing procedure. A non-cooperative game among the IP cameras is formulated, and the concept of Satisfaction Equilibrium (SE) is adopted to determine a stable data offloading, where the IP cameras satisfy their minimum QoS prerequisites. A distributed learning algorithm determines the IP cameras' data offloading at the SE, if the latter exists. If the SE does not exist, the proposed Distributed Learning Satisfaction Equilibrium (DLSE) algorithm converges to the Generalized SE, where only a part of the cameras satisfy their QoS prerequisites. In order to efficiently move the FAAS around the different areas, we utilize a Reinforcement Learning (RL), known as the state-action-reward-state-action algorithm (SARSA). A reward function was created, which considers data criticality, congestion of the area, and the energy expended for the movement of the FAAS. With this information, the quality of information from the AoIs can be described, and we can move the FAAS accordingly. Finally, detailed numerical results are presented which evaluate the proposed framework's simple operation and its scalability performance, while a comparative evaluation is provided to show its drawbacks and benefits.

1.5 Outline

The rest of the thesis is organized as follows. Chapter 2 describes the data offloading scheme used in this project for surveillance systems. In this chapter, we explain and explore the system model that was created. We look at communication and computation model followed by game theory in satisfaction form. We then dive into reinforcement learning and how it was used for the FAAS movement scheme. We concluded Chapter 2 in describing distributed learning utilizing a satisfaction equilibrium algorithm. In Chapter 3, we divide our results into three main categories: pure, scalability, and comparative. In pure results, we look at the overall performance of our algorithms and see if the results are viable. We then introduce scalability results, which is a look at what occurs in the system if different attributes of the system increase or decrease. After that, we look at the implementation of different algorithms and compare to our algorithms results. Lastly, in Chapter 4, we present the conclusion and future works.

Chapter 2

Data Offloading in Surveillance Systems

2.1 System Model

As discussed, in this thesis, we model a surveillance system that is composed of multiple areas that all have either the same number of cameras or different. Over the areas consists of one FAAS system that chooses the area that it supports during the current timeslot. These cameras can either offload their data locally, offload remotely to a MEC server, or offload to the FAAS if the FAAS is over the area. Figure 2.1 represents a possible system architecture with six different areas of interest all with different numbers of cameras, each supported by one MEC and one area having the FAAS support. A surveillance area is modeled to be of size $L \times L$ consisting of AoIs (e.g., banks, airports), where the set of AoIs is $A = \{1, \ldots, i, \ldots, A\}$ and they are randomly placed with coordinates $Z_i = (X_i, Y_i)$, $X_i, Y_i \leq L$.

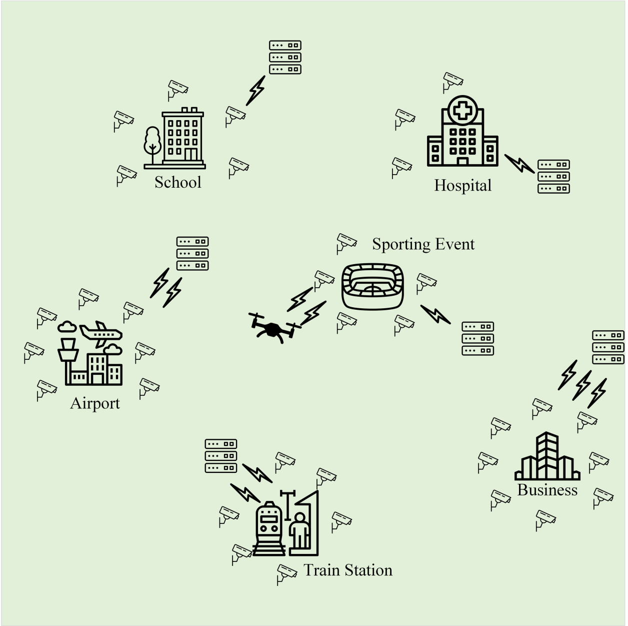

Figure 2.1: Possible System Architecture

We denote the set of areas of interest in the system, where each area of interest (AoI) $i, i \in A$ is located at a position in the city area and it has its own surveillance system that fully autonomous embedded wireless IP cameras $C_i = \{1, \ldots, j, \ldots, C_i\}$ that are capable of collecting and processing data [2] for surveillance purposes (i.e., crowd monitoring, face recognition, object detection). Moreover at each AoI $i \in A$ a mobile edge computing (MEC) server M_i, supports the cameras computing demands. A mobile edge computing (MEC) node (i.e., small data center at the edge of the network and possibly managed by a Wireless Internet Service Provider (WISP)), is considered. Furthermore, a fully autonomous aerial service vehicle (FAAS) is considered, which moves among the different AoIs with a velocity v and altitude d. At each timeslot t, the FAAS receives and processes data from the AoI's cameras of which the FAAS is located above. The whole operation time is divided in T timeslots, where set of timeslots is $T = \{1, \ldots, t, \ldots, T\}$ denotes the corresponding set, and at each timeslot, $t \in T$ the FAAS is located above only one AoI.

In addition, during each timeslot $t \in T$ at each AoI $i, i \in A$, the corresponding IP cameras C_i collect their data, which then are processed at the end of the timeslot. The set of collected data by each IP camera $j \in C_i$ belonging to the AoI i per timeslot t is denoted as $D_{ij}^{(t)} = (B_{ij}^{(t)}, CP_{ij}^{(t)}, \phi_{ij}^{(t)}, dt_{ij}^{(t)}, de_{ij}^{(t)})$, of the timeslot $t \in T$, which is characterized by specific features and requirements. In particular, let $B_{ij}^{(t)}[bits]$ be the total collected information, in terms of bits, and based on the surveillance task that the IP camera j at the AoI i performs (e.g., object detection, face recognition) the $CP_{ij}^{(t)} = \phi_{ij}^{(t)} B_{ij}^{(t)}$ is the number of required CPU cycles to process the data. We set $CP_{ij}^{(t)}$, where $\phi_{ij}^{(t)} > 0$ is the level of the video processing task's intensity of the $B_{ij}^{(t)}$ amount of data. Also, $dt_{ij}^{(t)}$ denotes the time constraint during which the data should be processed, and $de_{ij}^{(t)}$ is the IP camera's j of the AoI i energy availability for the timeslot t, since each IP camera $j \in C_i, \forall i \in A$ is fully autonomous and not connected to any energy resource. The amount of collected data $B_{ij}^{(t)}$ can be partitioned into subsets of specific size, which can be offloaded for remote processing to the MEC server M_i or the FAAS, assuming that the last one is located at the AoI i for the timeslot t. Therefore, each IP camera j by offloading a part of its collected $B_{ij}^{(t)}$ amount of data for remote processing, it keeps the rest to be processed locally.

2.2 Communication and Computation Model

Utilizing the system model described above, we denote $\mathbf{s}_i = (\mathbf{s_{i1}}, \ldots, \mathbf{s_{ij}}, \ldots, \mathbf{s_{iC_i}})$ the vector of strategies for the IP cameras residing in the AoI i, where $\mathbf{s_{ij}} = (ch_{ij}, a_{ij})$ and $a_{ij} \in [0, 1]$ is the IP camera's data offloading percentage, thus $a_{ij}, \forall i \in A, \forall j \in C_i$, and $ch_{ij} = 0$ if the IP camera offloads its $a_{ij} \cdot B_{ij}$ amount of data to the MEC server M_i, while $ch_{ij} = 1$ accordingly if it offloads to the FAAS. Therefore, considering that the FAAS is located at the AoI i, then for each other AoI $i' \in A, i' \neq i$ it holds true that $ch_{i'j} = 0$, $\forall j \in C_{i'}$. Thus, since the AoIs do not interfere with each other due to their distant locations, the IP camera's j in AoI i uplink data rate is [46]:

$$R_{ij} = W_i \cdot log(1 + \frac{p_{ij}g_{ij}}{\sigma_0^2 + \sum\limits_{k \in C_i \setminus \{j\}, ch_{ik}=ch_{ij}, a_{ik} \neq 0} p_{ik}g_{ik}}) \tag{2.1}$$

where W_i is the AoI's i bandwidth, R_{fix} the application's fixed transmission data rate, p_{ij} is the IP camera's j transmission power to offload part of its data, g_{ij} is the channel gain between the IP camera j and the MEC server M_i (if $ch_{ij} = 0$) or the FAAS (if $ch_{ij} = 1$), and σ_0^2 indicates the background noise power.

The IP camera j in the AoI i experiences the data transmission time overhead $O_{ij}^{tr,t} = \frac{a_{ij} \cdot B_{ij}}{R_{ij}}[sec]$ by offloading $a_{ij}B_{ij}$ amount of data and the data transmission energy consumption $O_{ij}^{tr,e} = p_{ij}\frac{a_{ij} \cdot B_{ij}}{R_{ij}}[Joules]$. Each MEC server M_i and the FAAS have the computing capability f_{M_i} and $f_F[Cycles/sec]$ respectively, which is shared among the IP cameras that are being served by them. The allocated computing capability to each IP camera j in order to remotely process its offloaded data is given as:

$$f_{ij} = \frac{a_{ij}B_{ij}\phi_{ij}}{\sum\limits_{k \in C_i \setminus \{j\}, ch_{ik}=ch_{ij}} a_{ik}B_{ik}\phi_{ik}} \cdot ((1 - ch_{ij})f_{M_i} + ch_{ij}f_F) \tag{2.2}$$

where the first factor of Eq. 2.2 reveals that an IP camera with a higher processing intensity (i.e, ϕ_{ij}) and greater amount of offloaded data acquires a higher computing capability, while the second one reveals that each IP camera j can offload a part of its data to only one computing resource (i.e., either the MEC server M_i or the FAAS). Based on the IP camera's j remote computing capability (Eq. 2.2), its offloaded data processing time overhead is $O_{ij}^{p,t} = \frac{a_{ij}B_{ij}\phi_{ij}}{f_{ij}}$. Moreover, the IP camera j has a local computing capability $f_{ij}^l[Cycles/sec]$ and processes the rest $(1 - a_{ij})B_{ij}$ data locally. Thus, its local processing time overhead is $\frac{(1-a_{ij})B_{ij}\phi_{ij}}{f_{ij}^l}$ and its local processing energy overhead is $(1 - a_{ij})B_{ij}\phi_{ij}e_{ij}$, where $e_{ij}[J/Cycle]$ is its local energy consumption to process the data. The IP camera's j overall time overhead is given as follows.

$$O_{ij}^{t} = \max\{\frac{a_{ij} \cdot B_{ij}}{R_{ij}} + \frac{a_{ij}B_{ij}\phi_{ij}}{f_{ij}}, \frac{(1-a_{ij})B_{ij}\phi_{ij}}{f_{ij}^{l}}\} \tag{2.3}$$

while its overall energy consumption is formulated as:

$$O_{ij}^{e} = p_{ij}\frac{a_{ij} \cdot B_{ij}}{R_{ij}} + (1-a_{ij})B_{ij}\phi_{ij}e_{ij} \tag{2.4}$$

2.3 Games in Satisfaction Form

In the last few years, game theory has played a central role in the analysis of communication type problems. Game theory is used in specific applications where a model has to be created to demonstrate different strategies that arise when players make specific actions in an environment. Some examples that utilize game theory is radio resource allocation or in quality of service (QoS) provisioning in wireless communication networks deal with: (i) single or multiple resources [47, 48], (ii) continuous or discrete resources [49, 50, 51], (iii) usage-based pricing mechanisms [52, 53], (iv) multicell interference mitigation [54, 55], (v) anomaly detection [56, 57] and others. There are different branches in game theory; some examples of these games could include: cooperative games, non-cooperative games, and zero-sum games. Cooperative games are ones in which the players adopt a strategy through agreements between them and other players. These games are seen in routing algorithms, sequencing algorithms, and much more. Non-cooperative games are such that players decide their own strategies to optimize their own returns. Applications for non-cooperative games are seen in wireless communications, economic market applications, and more. Zero-sum games are games in which if one player gains a different player or players lose. The applications for these games are generally found in economics, communications, and other fields. As discussed later, this project uses a non-cooperative game in order for cameras to make decisions. With game theory, the concept of equilibrium was introduced by Nash in [58], where these equilibrium points are where new choices by the players will not improve the environment anymore. The Nash equilibrium point comes from when the players or entities are interested in selfishly maximizing their own QoS. This isn't always the most optimal or practical QoS choice a system should converge. A more suitable choice may come from the entities tying only to guarantee a minimum QoS value, that way, more entities can be satisfied. This type of equilibrium point is classed a generalized Nash equilibrium (GNE). Depending on the system architecture or topology, though, a GNE may not be able to be found. In this case, a satisfaction equilibrium (SE) point is found. All of these equilibrium points call for some sort of function in order to tell whether or not an entity is to be satisfied. These functions are called utility functions and describe mathematically the welfare of choice being made. Creating utility functions are helpful in different scenarios because it indicates the patterns of change that occur in the system.

In this section, the IP cameras' utility function expressing their satisfaction from processing part of their collected data remotely and the rest locally, are formulated. Specifically, each IP camera j, $j \in C_i, \forall i \in A$ aims to satisfy its QoS prerequisites expressed in terms of time dt_{ij} and energy de_{ij} demands by offloading an amount of data and processing the rest locally. Thus, based on each IP camera's j overall time and energy overhead, (i.e., Eq. 2.3, 2.4), we formulate a generic utility function that represents each IP camera's QoS as follows.

$$u_{ij}(\mathbf{s_{ij}}, \mathbf{s_{-ij}}) = \begin{cases} -(\frac{dt_{ij}-O^t_{ij}}{dt_{ij}}) \cdot (\frac{de_{ij}-O^e_{ij}}{de_{ij}}) & \text{if } O^t_{ij} \geq dt_{ij}, O^e_{ij} \geq de_{ij} \\ (\frac{dt_{ij}-O^t_{ij}}{dt_{ij}}) \cdot (\frac{de_{ij}-O^e_{ij}}{de_{ij}}) & otherwise \end{cases} \tag{2.5}$$

where $\mathbf{s_{-ij}}$ is the strategy vector of all the IP cameras of the AoI i except the IP camera j. This utility function is used to indicate if a camera is satisfied or not. If the time constraint or the energy constraint is smaller than its related overhead, than the utility function will be negative indicating to the system that the camera system is unsatisfied with the amount of data offloading it performed in the previous iterations. When the camera overheads are less than their constraints than the utility will be positive. These utilities are generated in each iteration for each camera to observe the effects of how offloading in each iteration are affecting the system. Assuming that the FAAS is located at the AoI i, it is evident by Eq. 2.3, that when the IP camera's j chosen computing resource (i.e., the MEC server or the FAAS) is overloaded, then its perceived time and energy overhead increase, and its utility value u_{ij} is negative if the IP camera does not satisfy at least on of its QoS prerequisites (i.e., dt_{ij}, de_{ij}). Thus, each IP camera j aims to fulfill its time and energy demands, i.e., $u_{ij} \geq 0$, via autonomously determining its offloading strategy $\mathbf{s_{ij}} = (ch_{ij}, a_{ij})$.

A non-cooperative game is played among the IP cameras per AoI to determine a stable data offloading vector that fulfills the IP cameras' QoS prerequisites. The game is written in the satisfaction form $G_i = [C_i, \{S_{ij}\}_{j \in C_i}, \{u_{ij}\}_{j \in C_i}, \{h_{ij}\}_{j \in C_i}]$, where C_i is the set of the IP cameras in the AoI i, and considering that the FAAS is located in the AoI i, then $S_{ij} = \{(a^n_{ij}, 0), \cdots, (a^N_{ij}, 0), \cdots, (a^n_{ij}, 1), \cdots, (a^N_{ij}, 1)\}$, while $S_{ij} = \{(a^n_{ij}, 0), \cdots, (a^N_{ij}, 0)\}$ otherwise, and a^n_{ij} is the n^{th} available offloading percentage, thus $a^n_{ij} \in [0, 1]$, $\forall n \leq N, N \in N$. Moreover, u_{ij} is the AoI's i IP camera's j utility as expressed in Eq. 2.5, and h_{ij} is the satisfaction correspondence defined as follows [24].

$$h_{ij}(\mathbf{s_{-ij}}) = \{\mathbf{s_{ij}} \in S_{ij} | u_{ij}(\mathbf{s_{ij}}, \mathbf{s_{-ij}}) \geq 0\} \tag{2.6}$$

[**Satisfaction Equilibrium - SE**] A strategy vector $\mathbf{s_i^+} = (\mathbf{s_{i1}^+}, \cdots, \mathbf{s_{ij}^+}, \cdots, \mathbf{s_{iC_i}^+}) \in S_i = S_{1j} \times \cdots S_{iC_i}$ is an SE for the non-cooperative game G_i, if $\forall j \in C_i$, $\mathbf{s_{ij}^+} \in h_{ij}(\mathbf{s_{-ij}^+})$.

Essentially, following the above definition, an offloading strategy vector $\mathbf{s_i^+}$ is an SE point for the non-cooperative game G_i if and only if each IP camera $j \in C_i$ fulfills its time and energy demands (i.e., dt_{ij}, de_{ij}), thus $u_{ij} \geq 0$. At the SE, the IP cameras satisfy their minimum QoS prerequisites without overspending the system's resources, where the latter would occur if they were targeting at their QoS maximization. In our paper, we consider that the computing resources per AoI can support the IP cameras' minimum QoS prerequisites, i.e., an SE exists, and our goal is to achieve the better exploitation of the system's resources by allocating the cameras' computing tasks. For each non-cooperative game G_i, $i \in A$ there exists at least one strategy vector $\mathbf{s_i}^{+} \in S_i$ that guarantee the satisfaction (i.e., $u_{ij} \geq 0, \forall j \in C_i$) of the IP cameras of the AoI i, $\forall i \in A$.

2.4 FAAS Movement: Reinforcement Learning

The Quality of Information (QoI) of a surveillance system is an important and challenging task to capture, since misleading information may lead to undesired consequences, e.g., false alarm. Thus, in this thesis, deployment of three different quality factors that capture the QoI demonstrate the confidence in the information that each AoI's $i, i \in A$ surveillance system provides. Considering that several efficient algorithms (e.g., object detection, move detection) can be executed locally at the IP cameras' and remotely at the MEC servers' and FAAS's computing resources to assess each AoI's QoI. These algorithms assign values to the following quality factors at the end of each timeslot $t \in T$, based on each IP camera's captured data.

(a) **Accuracy** refers to how the observed information inside each AoI $i, i \in A$ conforms to the reality. Assuming that a local and remote processing of the corresponding IP cameras' collected amount of data, (i.e., the $(1 - a_{ij})B_{ij}$ amount of data is processed locally, while the rest $a_{ij}B_{ij}$ is processed remotely) the number of the correctly detected events AE_{ij} is evaluated, each IP camera's j of the AoI i and the IP camera's accuracy is $q_{ij}^{acc} = \frac{AE_{ij}}{TE_{ij}}$. Specifically, the AE_{ij} is the number of the correctly detected events, which is the output of the processing of the B_{ij} amount of data that was collected by the IP camera j, and TE_{ij} is the total number of events that were captured. Therefore, based on each IP camera's j accuracy quality factor, the overall accuracy of the AoI i can be evaluated as the average provided accuracy by the set C_i of IP cameras and thus is defined as $Q_i^{acc} = \frac{1}{C_i} \sum_{j \in C_i} q_{ij}^{acc}$.

(b) **Timeliness** refers to the availability of the information at the desired time. The value of this quality factor is a measure of information being available. Considering that each IP camera $j, j \in C_i$ inside the AoI $i, i \in A$ collects its B_{ij} amount of data, and then by offloading $a_{ij}B_{ij}$ amount of data for remote processing, it perceives an overall time overhead O_{ij}^t, its timeliness factor can be determined as $q_{ij}^{tls} = \frac{D_t}{D_t + O_{ij}^t}$, where D_t is the duration of each timeslot t and O_{ij}^t is the IP camera's overall time overhead to offload and process the data. Essentially, the faster the IP camera's j amount of data (i.e., B_{ij}) is processed, the higher is the IP camera's timeliness factor, and as a result t The AoI's overall timeliness factor is $Q_i^{tls} = \frac{1}{C_i} \sum_{j \in C_i} q_{ij}^{tls}$.

(c) **Certainty**: refers to the measurement of confirmation of the information and is strictly connected with to each IP camera's j hardware characteristics and capabilities (e.g., recording rate, sensor's pixels). In particular, this quality factor depicts the probability of error regarding the captured data of each IP camera j, and it is denoted as q_{ij}^{crt}. As a result, the overall certainty of the AoI i is evaluated as $Q_i^{crt} = \frac{1}{C_i} \sum_{j \in C_i} q_{ij}^{crt}$.

Finally, each AoI's $i, i \in A$ overall QoI for a specific timeslot is based on the past QoI values and is given as follows.

$$QoI_i = w_i^{acc} \frac{\sum_{t' \leq t} Q_i^{\prime acc}}{t} + w_i^{tls} \frac{\sum_{t' \leq t} Q_i^{\prime tls}}{t} + w_i^{crt} \frac{\sum_{t' \leq t} Q_i^{\prime crt}}{t} \tag{2.7}$$

where $w_i^{acc}, w_i^{tls}, w_i^{crt} \in [0,1]$ are the corresponding weights of each quality factor. The weights regarding accuracy, timeliness and certainty factors, respectively, can be tuned regarding the priorities and considerations of the overall city's surveillance system, e.g., a higher weight of the timeliness weight factor may be used in an AoI, such as airport, where a low real-time processing is needed.

In this section, the consideration of the FAAS as a sequential decision maker that aims to maximize a long-term objective is considered. Specifically, at each timeslot, $t, t \in T$, the FAAS is located at an AoI i, and acting as a computing resource, it provides a higher QoS to the corresponding IP cameras since the MEC server M_i is less overloaded. Therefore, the IP cameras' QoS prerequisites (i.e., time and energy demands) could be more easily met with the existence of the FAAS at the AoI i, and the overall AoI's timeliness quality factor to be increased. The existence of the FAAS at the AoIs with high QoI for example, a large number of actual detected events, high certainty of information, is important, since the overall city's surveillance system's performance and effectiveness could be increased by decreasing the delay between an event's detection and further actions (e.g., policy, ambulance). As a result, a high QoI indicates an AoI where the FAAS should be located for the effectiveness purposes of the overall city's surveillance system, while the FAAS's limited energy availability should be considered for both FAAS's flying movement and its role as a computing resource. Considering that the FAAS is located at the AoI i, and by denoting as $E_P[Joules/Cycle]$ the FAAS's energy consumption to process the received data, then its processing energy consumption is formulated as:

$$E^p = E_P \cdot \sum_{j \in C_i, ch_{ij}=1} a_{ij} B_{ij} \phi_{ij} \tag{2.8}$$

Furthermore, considering that the FAAS was located at the AoI $i', i' \neq i$ at the previous timeslot and the FAAS's velocity is v, then its movement energy consumption is given as:

$$E^m = E_M \cdot \frac{\sqrt{(X_i - X_{i'})^2 + (Y_i - Y_{i'})^2}}{v} \tag{2.9}$$

where $E_M[Watts]$ is the FAAS's constant consumed energy while moving with velocity v. All the aforementioned factors should be considered in the formulation of the FAAS's long-term objective that the FAAS aims to maximize, by optimizing its decision policy (i.e., the AoI i that visits at each timeslot $t, t \in T$), and this objective that essentially depicts the FAAS decision's (i.e., the of the AoI $i, i \in A$) reward is given as follows:

$$rw_i = -\frac{\epsilon_3 \cdot \frac{E^p + E^m}{E}}{\epsilon_1 \cdot QoI_i + \epsilon_2 \cdot P_i} \tag{2.10}$$

where $P_i = \frac{|C_i^s|}{C_i}$, $C_i^s = \{j \in C_i | u_{ij} \geq 0\}$ denotes the ratio of the IP cameras that meet their QoS prerequisites (i.e., dt_{ij}, de_{ij}), E is the FAAS's energy availability, and depicts the achieved relative performance of the AoI i with the existence of the FAAS. Moreover, the $\epsilon_1, \epsilon_2, \epsilon_3 \in [0, 1]$ denote the weights of the AoI's QoI, performance (i.e., P_i) and the FAAS's consumed normalized energy, respectively. The physical meaning of the negative reward value is that reward values closer to zero benefit the FAAS. Given that the overall city's surveillance system is essentially an uncertain environment, where each AoI's QoI and the corresponding AoI's IP cameras' QoS prerequisites vary with respect to the timeslots (e.g., timeslots where a large number of actual events take place at specific AoIs, AoIs that a low real-time processing of the captured data is needed during specific timeslots), the FAAS acting as a learning agent, it aims to learn an optimal decision policy (i.e., which AoI i to visit at each timeslot $t, t \in T$) in order to maximize its long-term objective (Eq. 2.10).

In the following, in order to capture the uncertainty regarding the FAAS's sequential decision making problem, and increase the effectiveness of the city's surveillance system, a Reinforcement Learning (RL) [59] approach is applied by the FAAS in order to maximize its long-term objective (Eq. 2.10). The RL algorithms demonstrate good results [60] - [61] in real-world sequential decision making problems, which are characterized by the environment's uncertainty. Two of the most widely used RL algorithms are the Q-learning [61] and SARSA [62] algorithms, which via stochastic approximation conditions lead the decision maker to converge to its optimal decision policy with high probability [63],[64]. The significant difference between Q-learning and SARSA is that Q-learning is an off-policy algorithm, whereas SARSA is considered an on-policy algorithm. The difference lies such that Q-learning regardless of what the agent does (or doesn't) an optimal policy will be learned as long as the agent explores enough. This type of algorithm can be dangerous to a system in which large bad rewards are currently learned. The opposite, on-policy algorithms, learn values from what the agent is currently carrying out and can be iteratively improved. SARSA is also known for its exploration parameter in which the agent can randomly choose an action regardless of what the maximum Q-value is. This parameter is typically found to be 5% through 10% and allows the system to update in case of changes within the environment. In our case, for the FAAS's sequential decision making problem (i.e., the AoI i that selects to be located at each timeslot t) we deploy the SARSA algorithm, which first examines the uncertain environment (i.e., the set of AoIs A), and then derives the optimal strategy based on the model knowledge that has already been constructed.

SARSA is an algorithm that learns through a Markov decision process policy. An agent (i.e., FAAS) interacts with the environment (i.e., surveillance system) in order to update its policy on the action it took (i.e., the AoI that is located). The experienced reward (Eg. 2.10) is known as the Q-value and is adjusted by a learning rate that weights new information higher than the previously gathered information. In order to do this, SARSA algorithm takes the agent's action in its current state and multiplies a specified discount future reward that the agent will receive from the next state action it observes. These Q-values represent the rewards that the agent is expected to receive in the next time step and will be considered by the agent when it is deciding which action to take when it is in a specific state. SARSA has a unique learning exploration factor, in this case, there is always a case the agent chooses randomly from the table instead of choosing a learned value. This allows for the agent to continually learn values that may have chanced in the enviroment. Instead of always chooseing what was good in the past, the agent has the ability to relearn to make sure nothing has changed.h

2.5 Distributed Learning Satisfaction Equilibrium Algorithm

Towards determining the SE for each non-cooperative game G_i, we propose the Distributed Learning Satisfaction Equilibrium Algorithm (DLSE) that allows the IP cameras C_i, $\forall i \in A$ to autonomously learn and converge to it. Based on Eq. 2.3, 2.4, each IP camera j needs to receive information which is related to the rest AoI's i IP cameras' offloading strategies in order to determine its own data offloading strategy $\mathbf{s_{ij}^{+}} \in S_{ij}$ that fulfills its time and energy demands. Each camera then evaluates its utility (Eq. 2.5) by receiving its allocated remote computing capability (Eq. 2.2) from the MEC server or FAAS and the interference factor (i.e., $\sum_{k \in C_i, ch_{ik}=ch_{ij}, a_{ik} \neq 0} p_{ik} g_{ik}$ in Eq. 2.1) and converges to the strategy $\mathbf{s_{ij}^{+}}$. For each non-cooperative game G_i, $\forall i \in A$ among the IP cameras, the following distributed algorithm that are located in the AoI i (i.e., C_i), converges to an SE in a fully distributed fashion. Assuming that the elements of the offloading strategy set S_{ij} are indexed with l_{ij}, thus $\mathbf{s_{ij}}^{(l_{ij})}$ is the $l_{ij}th$ offloading strategy, then $l_{ij} \leq L_{ij}$, and $L_{ij} = 2N$ if the FAAS is located in the AoI i, otherwise $L_{ij} = N$. Let us denote the IP camera's j offloading strategy at instant $r > 0$ as $\mathbf{s_{ij}}(r) \in S_{ij}$, where it is chosen following a discrete probability distribution $\pi_{ij}(r) = (\pi_{ij}^{(1)}(r), \cdots, \pi_{ij}^{(l_{ij})}(r), \cdots, \pi_{ij}^{(L_{ij})}(r))$, where $\pi_{ij}^{(l_{ij})}(r)$ is the probability with which the AoI's i IP camera j chooses its action $\mathbf{s_{ij}}^{(l_{ij})}$ at instant $r > 0$. Using this notation we present the Distributed Learning Satisfaction Equilibrium Algorithm (DLSE), which allows the convergence to an SE point in a distributed manner.

The initial (i.e., $r = 0$) probability distribution for each IP camera is $\pi_{ij}^{(l_{ij})}(r =$

Algorithm 1 DLSE Algorithm

```
Input/Initialization: AoI $i$, $C_i$,$D_{ij}$,$\forall j \in C_i$, $conv = 0$, $r = 1$
Output: $\mathbf{s_i^+} = (\mathbf{s_{i1}^+}, \cdots, \mathbf{s_{ij}^+}, \cdots, \mathbf{s_{iC_i}^+})$
Each $j \in C_i$ sets $S_{ij}$, $L_{ij}$ based on the FASS's existence
$\pi_{ij}^{(l_{ij})}(0) = \frac{1}{L_{ij}}, \forall j \in C_i, l_{ij} \leq L_{ij}$
Each $j \in C_i$ picks $\mathbf{s_{ij}}(0)$ based on $\pi_{ij}(0)$, and evaluates $U_{ij}$
while $conv == 0$ do
    for $j = 1$ to $C_i$ do
        $u_{ij} = u_{ij}(\mathbf{s_{ij}(r-1)}, \mathbf{s_{-ij}}(r-1))$, $b_{ij}(r) = \frac{U_{ij}+u_{ij}}{2U_{ij}}$
        if $u_{ij} \geq 0$, then $\mathbf{s_{ij}}(r) = \mathbf{s_{ij}}(r-1)$, $\pi_{ij}(r) = \pi_{ij}(r-1)$
        else $\forall l_{ij} \leq L_{ij}$
        $\pi_{ij}^{(l_{ij})}(r) = \pi_{ij}^{(l_{ij})}(r-1) + \lambda_{ij} b_{ij}(r)(1_{\{\mathbf{s_{ij}}^{(l_{ij})}=\mathbf{s_{ij}}(r-1)\}} - \pi_{ij}^{(l_{ij})}(r-1))$
    end for
    if SE or GSE point reached then $conv = 1$
    else $r = r + 1$
end while
```

$0) = 1/L_{ij}, \forall l_{ij} \leq L_{ij}$, where L_{ij} is the number of the IP camera's j offloading strategies. Let U_{ij} denote the maximum utility that each IP camera j perceives if it was the only one inside the AoI i. Each IP camera updates its probability distribution π_{ij} based on a learning parameter λ_{ij}, so that higher probabilities are allocated to offloading actions which lead the IP camera j to perceive a higher utility u_{ij}. Let us introduce the definition of a clipping action, which is considered for the study of the DLSE Algorithm's convergence to an SE point. [Clipping Action] At each non-cooperative game G_i, an IP camera j has a clipping action: $\forall \mathbf{s_{-ij}} \in S_{-ij}, \mathbf{s_{ij}^c} \in h_{ij}(\mathbf{s_{-ij}})$ where $S_{-ij} = S_{i1} \times \cdots \times S_{i(j-1)} \times S_{i(j+1)} \times \cdots \times S_{iC_i}$ [25]. Therefore, Definition 2.5 reveals that once an IP camera concludes to a clipping action $\mathbf{s_{ij}^c}$ at an instance r' of the DLSE Algorithm, then $\forall r \geq r'$ the IP camera keeps the same offloading strategy, i.e., $\mathbf{s_{ij}}(r) = \mathbf{s_{ij}^c}$. As a result, assuming that there exists an IP camera $j' \neq j$, such that its satisfaction correspondence $h_{ij}(\mathbf{s_{-ij'j}}) = \emptyset$, $\forall \mathbf{s_{-ij'j}} \in S_{-ij'j}$, where $\mathbf{s_{-ij'j}}$ is the offloading strategy vector of all the IP cameras except the camera j' and the IP camera j which plays its clipping action $\mathbf{s_{ij}^c}$, and $S_{-ij'j}$ is the corresponding set of vectors, then the DLSE Algorithm converges to a Generalized SE (GSE) point, whose definition is given as: [Generalized SE] A strategy profile is a GSE $\mathbf{s_i^{-}} = (\mathbf{s_{i1}^{-}}, \ldots, \mathbf{s_{iC_i}^{-}})$ of the non-cooperative game G_i, if there exists a partition of the C_i given by C_i^s and C_i^u, such that $\forall j \in C_i^s$, $\mathbf{s_{ij}} \in h_{ij}(\mathbf{s_{-ij}})$ and $\forall j' \in C_i^u$, $h_{ij'}(\mathbf{s_{-ij'}}) = \emptyset$. Given the DLSE algorithm and the existence of at least one SE point for each non-cooperative game G_i, and that there is no clipping action,then the DLSE Algorithm converges to the SE point for each game G_i. Otherwise, in the existence of a clipping action $\mathbf{s_{ij}^c}$ for at least one IP camera, the DLSE Algorithm converges to a GSE point.

Chapter 3

Experiments

3.1 Experiment Setup

In this chapter, a detailed numerical performance evaluation and comparative study of the proposed architecture is conducted through modeling and simulations. These simulations were generated utilizing the programming language Python 3.7. The python code generated .mat files from the lists collected in which all of the plots were then created using MATLAB. For the simulations we created a base template for what a city could possibly expect to see in a real work situation. The simulations used a surveillance system consisting of $A = 7$ AoIs with $C_i = 30$, $\forall i \in A$ cameras. The cameras are randomly distributed in an area with a radius less than $L = 500m$ from each MEC server M_i. The considered application characteristics are $B_{ij} \in [1000, 5000]KB$ and $CP_{ij} \in [1000, 5000]MCycles$. The IP cameras' strategy space consists of 11 data offloading strategies, where $\mathbf{a}_{i,j} \in [0, 1]$ with step 0.1. We used $e_{ij} = 10^{-9}J/Cycle$ as the energy constant for cameras offloading, $f^l_{ij} \in [10^{-2}, 10^{-1}]$, The bandwidth used was $W_i = 5MHz$, and the white noise $\sigma_0^2 = 10^{-13}$. For the power dissipated a value of $p_{i,j} \in [0, 1]W$ was used. Also, $dt_{i,j} \in [0, \frac{CP_{ij}}{f_j}]sec$, $de_{i,j} \in$

$[0, CP_{ij}e_{ij}]J$, $g_{i,j} = \frac{1}{d_{ij}^2}$, where d_{ij} is the IP camera's j distance from the MEC server M_i or FAAS. Finally, in our reinforcement learning algorithm the properties for the constants were set to the following: $w_i^{acc} = 0.333$, $w_i^{tls} = 0.333$, $w_i^{crt} = 0.333$, $Q_i^{\prime acc}, Q_i^{\prime tls}, Q_i^{\prime crt} \in [0,1]$, $E_P = 10^{-9}J$, $E_M = 0.0013W$, $E = 17.28 \cdot 10^6 J$, $\epsilon_1 = 0.35$, $\epsilon_2 = 0.55$, $\epsilon_3 = 0.10$, the duration of a timeslot is $1h$, and $v = 6.25m/s$ [65]. The following analysis below demonstrates: (i) the pure operation and characteristics of the proposed framework, (ii) its scalability performance, and (iii) a comparative evaluation.

3.2 Pure Operation of the Algorithm

For this first series of results, we look at the results purely at an operational level. Pure results were completed in order to validate the algorithm and take an in-depth look at its performance in the system environment. The results ensure that the algorithm performs as hypothesized and proves it's helpfulness in the application setting. In the first set of figures, a comprehensive look at four different IP cameras considering the amount of offloaded data, the time and the energy overhead and constraints for each camera is shown. Figure 3.1, displays the number of bits of data the camera chooses to offload and the number of iterations it took in order to converge. As seen in the image, cameras 1 and 28 seemed to converge rather quickly (less than 100 iterations) whereas camera 3 and 12 took around longer (400 iterations). In this figure, it is shown that the camera is trying to satisfy itself by choosing different numbers of offloading schemes, so more than likely cameras 1 and 28 were satisfied right away and didn't change their offloading scheme. Whereas in cameras 3 and 12 both continually tried to find an offloading number that satisfied their time and energy constraints. In order to not become infinitely stuck, as described above a generalized satisfaction equilibrium was used for cameras that were never going to be satisfied to still converge to a value.

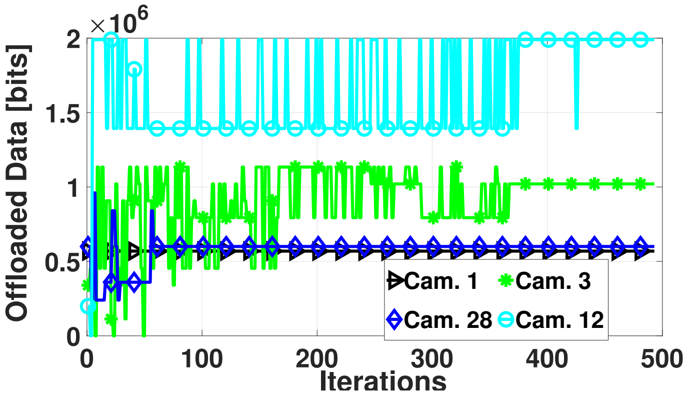

Figure 3.1: Data Offloading

In Figures 3.2 and 3.3 a look at the time and energy overheads respectively along with the constraints are provided. In these figures, we look to see if the time/energy overheads are less than their constraints. This look is essential because if one is not met, the camera system itself cannot be satisfied. This is because one or both constraints set forth by the system were not met. As we see, the overheads fluctuate with respect to 3.1 in which the camera is offloading more or less data to the MEC server in order to fall below its constraints. In Figure 3.2, we can see that cameras that had relatively low time constraints had more variation in their time overheads. We see that camera 28 started off by not meeting it's initial time constraint and had to move to a higher value of data offloading, this though corresponded to falling below the time constraint and now is likely to be a satisfied system. Whereas in camera 3 we observe a small time constraint and with variation of offloading it was never able to meet its time constraint. In Figure 3.3, an energy overhead and constraint image is presented. Similarly to the time overhead, if an energy constraint is not met, the cameras try to come to an offloading value that will satisfy their energy constraint.

Overall, from these results we can see that at a relatively fast rate (less than 400 iterations), the cameras converge to the amount of data they will offload to the

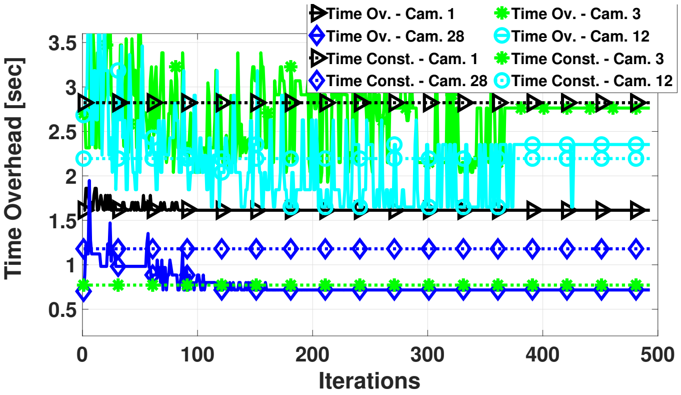

Figure 3.2: Time Overhead Overhead and Constraints

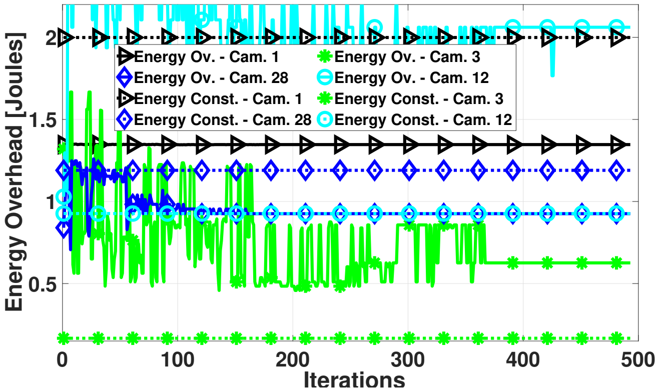

Figure 3.3: Energy Overhead and Constraints

MEC server. The IP cameras with ID 12 and 3 have strict time and energy constraints (Figures 3.2, 3.3). Thus they choose to offload a large amount of data to the MEC server in order to satisfy their QoS prerequisites. However, it is observed that even if they choose such a strategy, they cannot meet their QoS demands and the DLSE algorithm converges to a GSE point. On the other hand, the IP cameras with ID 1 and 28 have relaxed time and energy constraints, and they achieve to satisfy them, while the stricter the constraints are, the less time and energy overhead they experience, and the more data they offload. First, we evaluate a camera's performance in the environment while applying a game theory technique. The cameras can choose to offload a certain percentage of data to the MEC server trying to satisfy a time and energy constraint. The first three figures represent specific cameras data offloading, time constraints and overheads, and energy constraints and overheads. The first cluster represents that when a time and energy constraint are more strict for a camera, the camera should choose to offload more data to the MEC server in order to try to satisfy its constraints. The third cluster demonstrates that cameras are dependent on one another, so offloading large amounts of data to the MEC server does not necessarily guarantee satisfaction even if the time and energy constraints are relatively relaxed.

Several factors influence whether or not an IP camera meets its QoS prerequisites, such as the MEC server's computing capability, IP cameras' average distance from the MEC server, time and energy constraints, and number of cameras per AoI. In Table 3.1, the percentages of satisfied IP cameras per AoI are presented for different scenarios. For these results, only one influential factor was changed per each scenario. Each scenario consisted of seven different areas of interests, and by changing one variable, it enables us to see its impact on the whole environment. The variable of interest was adjusted varying from a scale $1x < 2x < ... < 7x$, where x is any of the aforementioned influential factors and the order follows the AoI's ID, while the values of the rest factors remain the same.

	Comp. Capab.	Distance	Time Const.	Percentages of Satisfied IP Cameras						
	Order	Order	Order	AoI1	AoI2	AoI3	AoI4	AoI5	AoI6	AoI7
Scenario1	X			0.22	0.31	0.33	0.94	0.366	0.39	0.43
Scenario2		X		0.82	0.82	0.81	1	0.75	0.49	0.14
Scenario3			X	0.04	0.30	0.59	1	0.76	0.88	0.95

Table 3.1: Percentage of Satisfied Cameras regarding different AoIs' characteristics and Cameras' QoS prerequisities

The results reveal that each one of these factors influence the number of satisfied cameras there are in the system. As the IP cameras' average distance from the MEC server increases, their communication channel conditions deteriorate; thus, a

	Energy Const.	Cameras	Percentages of Satisfied IP Cameras						
	Order	Order	AoI1	AoI2	AoI3	AoI4	AoI5	AoI6	AoI7
Scenario1	X		0.04	0.12	0.34	1	0.72	0.81	0.92
Scenario2		X	1	0.91	0.72	0.92	0.29	0.02	0

Table 3.2: Percentage of Satisfied Cameras regarding different AoIs' characteristics and Cameras' QoS prerequisities

smaller number of IP cameras meet its QoS prerequisites. Also, as the computing capability of the MEC server per AoI becomes stronger, the MEC server can more efficiently serve the cameras' computing demands; thus, a more significant number of them fulfill its QoS prerequisites. As the cameras' time and energy constraints become stricter, a smaller number of IP cameras get satisfied, and again, we see a shift in less cameras being satisfied. As the number of cameras per AoI increases, the environment becomes more congested in terms of the communication and computing aspects. Therefore a smaller number of cameras meet its QoS demands. Finally, it is essential to note what is occurring in AoI 4. It is observed that AoI 4 has superior performance in all scenarios as the FAAS resides in that area. From the table, we see that the variables either influence an area positively or negatively. For example, when the average distance increases from the camera to the MEC, we see that the average ratio of satisfied cameras drastically changes from a high percentage to a low percentage whereas increasing the time or energy constraint will actually help get higher percentages of satisfaction.

Now that we have an understanding of what it takes for a camera to be satisfied or not and what factors affect this, we verify the use of a FAAS. We also look at the performance of the SARSA algorithm that was used for deciding the FAAS placement. Figure 3.4. presents the FAAS's average reward versus the timeslots. After almost 50 timeslots, the FAAS learns its environment, and then it can choose the path that provides the maximum reward (Eq. 2.10). At the start of the timeslots, we have negative reward of about -.22, but as time increases we see that the system starts to learn the environment and flattens out at roughly -.11 in value. It is

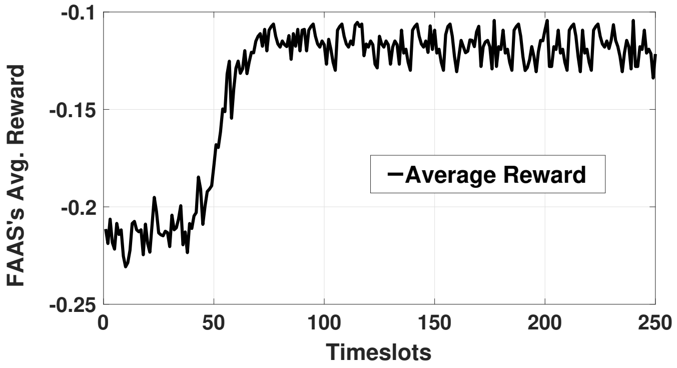

Figure 3.4: FAAS' Average Reward

important to note that our reward will never be positive, and the system is trying to find the less harmful value it can find. The reason for convergence after about 50 iterations comes from the fact that the SARSA algorithm will visit all different combinations of paths before choosing the path that gave it the highest rewards. The Q-table is populated at first with all zeros, so by the algorithm choosing negative values will force the FAAS to explore every option, then choose what the best path was. As mentioned above, SARSA has an exploration parameter in which sometimes the algorithm will choose a different state randomly instead of choosing the highest reward. This verifies that if something in the environment changes, it is possible to find this a better path than the FAAS was on previously.

In Figure 3.5 it is shown if the FAAS is over a specific area, that the percentage of satisfied cameras is significantly higher and the corresponding IP cameras' time and energy overheads are lower as the time evolves, thus showing the great benefits of adopting the FAAS in the considered overall architecture.

Figure 3.6 presents that as the time evolves if the FAAS is over an area on average, the time overhead for that area is lower than if the FAAS is not over the area. Similar behavior is seen in figure Figure 3.7 where the energy in the area the FAAS supports is less than just supported by the MEC. Both these phenomenons can be explained

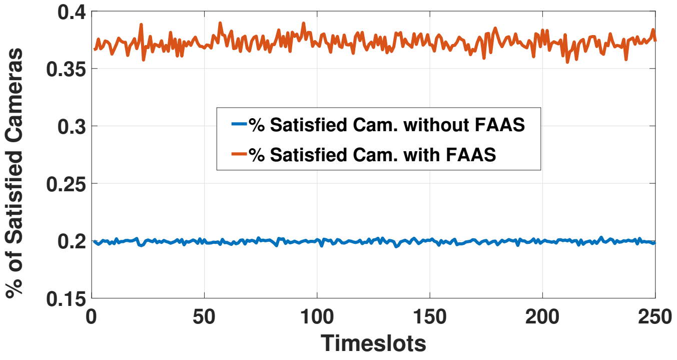

Figure 3.5: Percent of Satisfied Cameras

because the FAAS acts as an additional resource that the area can utilize. Having now three options of data execution (locally, offload to MEC, and offload to FAAS) leads to both smaller energy and time overheads. Therefore, one can assume that this means that more cameras will be satisfied in these scenarios because the time and energy overheads are smaller and their respective constraints will be met. This can be verified in Figure 3.5, by the fact that there is an increased in satisfied ratios, where almost 40 percent of cameras are satisfied whereas in the areas that don't have fast are nearly divided by two being closer to 20 percent. These images prove that by the FAAS being over the area, we see significant improvement for the corresponding area.

Finally, in Figures 3.8 and 3.9, the average number of FAAS' visits and the average Quality of Information per AoI in a time frame of 250 timeslots is presented. It is observed that if an AoI has high QoI, the SARSA algorithm will efficiently consider the FAAS's perceived reward and enable the FAAS to visit more often the critical AoIs, i.e., the ones having high value of QoI. AoI has a relative high average QoI value being closer to 1, and therefore a spike in the visits is shown being closer to 50 visits by the FAAS. On the other hand, AoI 5 is shown to be slightly smaller than the other areas, and the FAAS visited that area the least amount of times. We see this

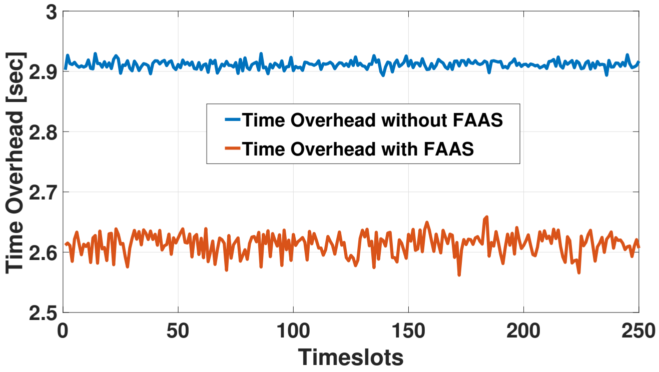

Figure 3.6: Time Overhead without or without the FAAS

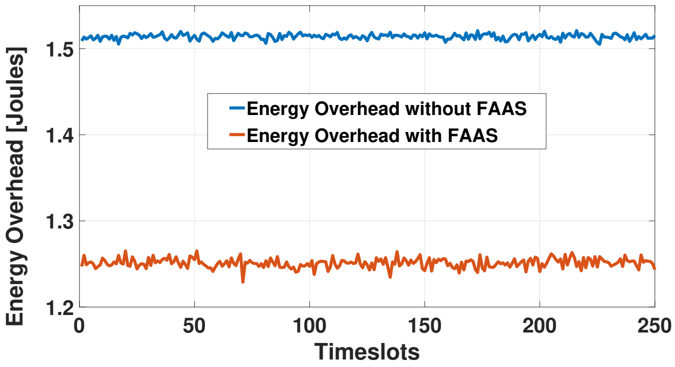

Figure 3.7: Energy Overhead without or without the FAAS

occur because as mentioned in section 2.4, data criticality is a factor taken account by the reward while choosing the next FAAS area. Information criticality could be something like an area with a high crime rate with activity in evening hours where the system considers it useful.

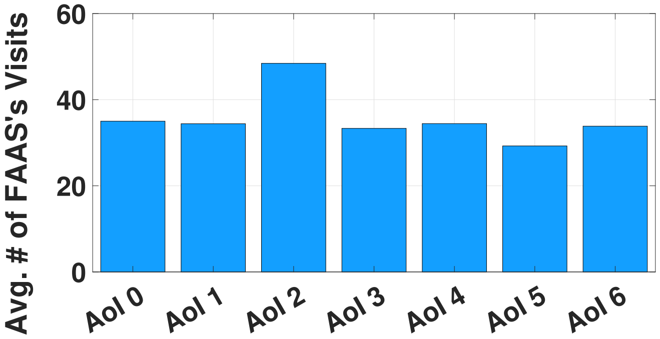

Figure 3.8: Average visits per 250 timeslots

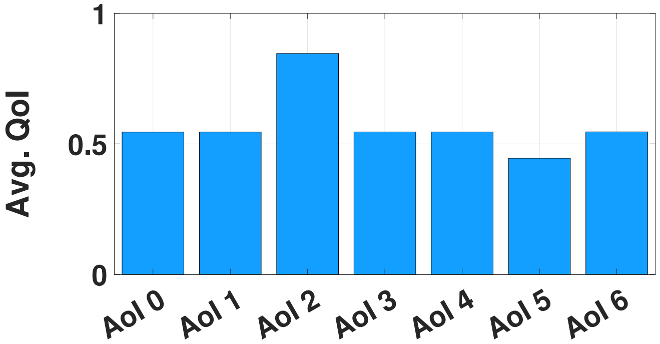

Figure 3.9: Average quality of information per 250 timeslots

3.3 Scalability Results

With the verification of the pure operation of the algorithm, next created test cases checking the scalability of our results is presented. For these results, a specific parameter in the system is scaled to see how it affects the percentage of satisfied cameras. For the first two scalability scenarios, the time and energy overheads as well as the average ratio of camera satisfaction were extracted. The first scenario shown is an analysis of the increasing number of cameras. The second analysis is looking at the available strategies. Looking at the scalability of these different parameters, we can show the performance of the proposed framework in order to validate our proposed algorithm further.

Increasing the number of cameras influences both time and energy overheads negatively by adding more congestion into the network. Figure 3.10, represents the time and energy overhead with the increase in cameras. As we see, there are increases in both these overheads that will affect the system such that fewer cameras will meet their constraints and therefore be unsatisfied. Figure 3.11 shows the percentage of satisfied IP cameras for an increasing number of cameras per AoI. The results reveal that as the number of cameras increases, the AoIs become more congested in terms of their communication and computing environment, thus the IP cameras' time and energy overhead increases, while the percentage of the cameras that meet their QoS prerequisites decreases. As we see in the figures, when the cameras are at 5 cameras per area, the satisfaction is above 60% whereas it's about 10% when there are more 50 cameras. This corresponds to that fewer cameras had values of around 2 seconds and 1 Joules overheads less cameras, but closer to 3.5 seconds and 1.6 Joules for a larger number of cameras in the system. This conclusion can be explained such that in a more significant number of cameras, more offloading would occur at the MEC server. The MEC's computation capability doesn't increase, and therefore, we see less cameras being satisfied.

In the second scenario, a look at the varying number of data offloading strategies is shown. These strategies are percentages that the camera chooses to offload to the MEC. Having 11 strategies corresponds to offloading 0%, 10%, , 90%, and 100% of the data to the MEC server. At the number of strategies increases, the IP cameras have greater flexibility of choices. Looking at the images, when the strategies where at 11 the time and energy overheads were relatively low close to 3.1 seconds and 1.6 Joules which corresponded to .15 camera satisfaction. Satisfaction continually rises as the number of strategies increases where when doubled the number at 22

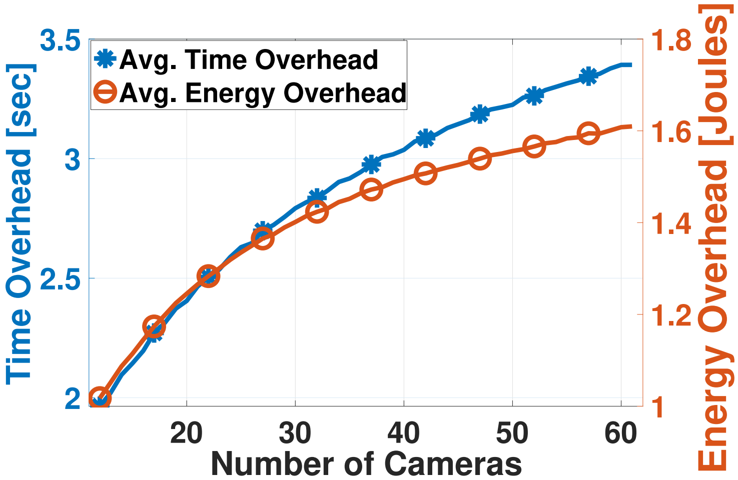

Figure 3.10: Time/Energy vs. Number of Cameras

strategies, the time overhead was 2.8 seconds, and energy overhead was 1.35, which had a 25% satisfaction. As shown in Figures **??**, allowing the cameras to choose a more precise amount to offload allows for better decisions to be made in order for a camera to be satisfied. With the increase in choices the IP cameras can chose from, we see that the time and energy overhead decrease and the corresponding percentage of satisfied cameras increases. As the Figure shows at smaller numbers of strategies, the time and energy overheads are larger and decrease as the number of strategies increase. At the same time, the percent of satisfied cameras hold the inverse effect such that less cameras are satisfied when there are less strategies and increase when more strategies are available.

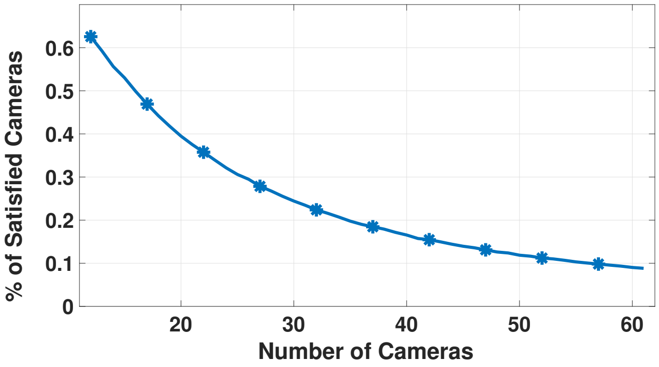

Figure 3.11: % of Cameras Satisfied vs. Number of Cameras

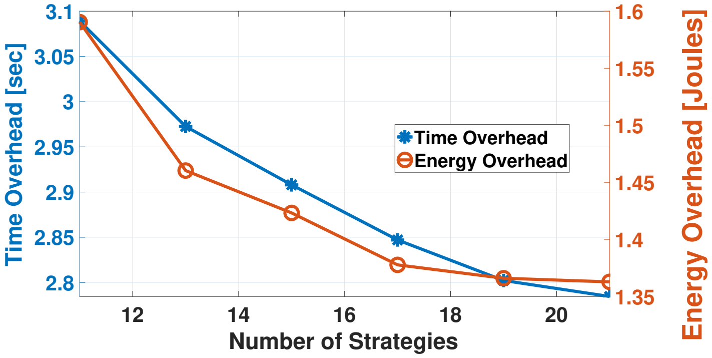

Figure 3.12: Time/Energy vs. Number of Strategies

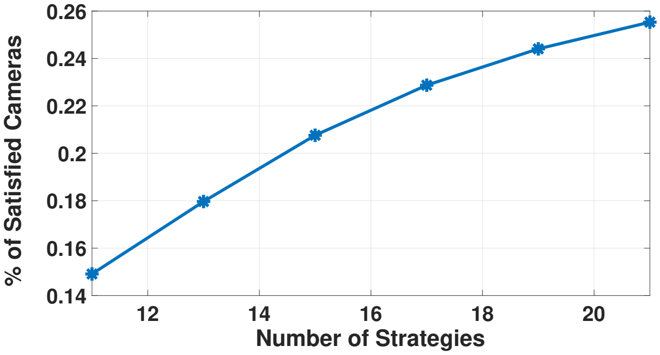

Figure 3.13: % of Cameras Satisfied vs. Number of Strategies

In the last Figure for scalability section, a look at what happens when increases the number of Areas of Interest are presented. Using a timeslot of 250, the average reward of each of these scenarios is taken. As seen in the Figure, the more areas meant that there are more state action possibilities for the FAAS to choose from. For example in a situation where 4 AoI are present in the environment then after 16 timeslot iterations the system is known, whereas in an environment with 10 AoI at least 100 timeslots must be analyzed before all states are known. By this, it takes a longer time in order for the FAAS to learn its environment as the number of AoI increases and will not have an opportunity to choose from this list of learned rewards for longer times. This meant on average for 100 timeslots the reward that the FAAS was able to choose from were more negative, indicating not as good paths were chosen.

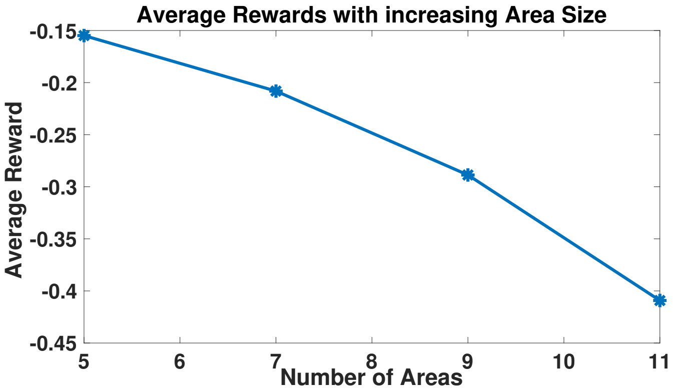

Figure 3.14: Increases the number of AoI

3.4 Comparative Results

The final results section is the comparative results that are broken into two different layers. These comparative results look at other techniques for the camera satisfaction selection and choosing what area the FAAS visited in the next timeslot. These comparative scenarios are presented to confirm the benefits of the proposed algorithmic approach presented in this thesis. The comparative scenarios are broken into the two sets, examining: (i) the Satisfaction Equilibrium's benefits, and (ii) the benefits of the adoption of reinforcement learning.

Regarding the first set of comparative scenarios, five different approaches are presented: i) minimizing the energy overhead (MEO) and ii) the time overhead (MTO), iii) determining the Nash Equilibrium (NE), vi) offloading the entirety of the data (OE), and v) random amount of the data to the MEC server. As shown, the novel concept of SE resulted in the highest percentage of satisfied cameras (Fig. 3.15, whereas the Nash equilibrium value followed closely. Minimizing the time and energy overheads both have around .15 satisfaction. Logically thinking, this makes sense to not perform as well as the satisfaction or Nash equilibrium algorithms. Minimizing one overhead more than likely other constraints is not met, and therefore we lose satisfaction. The worse performers were found to be randomly choosing the data to offload and the worse being offloading the entirety of the data set. Offloading the entirety of the data set makes sense in being the worse performer; by not using the cameras locally executing capability, it limits itself to how many cameras can offload anything at all. All of these runs were performed under Monte Carlo simulations in which the results were averaged over 250 different runs. This explains why the random simulations were better than offloading the entirety. In some cases, randomly actually outperformed the satisfaction equilibrium algorithm, but other times it was worse than offloading all the data.

In Figure 3.16, different scenarios of FAAS's navigation among the AoI are presented. In the examined scenarios, the FAAS visits the area: i) closest to the current area, ii) with the largest average energy constraint, iii) sequentially, vi) maximizing its reward, v) randomly, and vi) with the largest average time constraint. The results reveal that the SARSA algorithm produced an average FAAS's reward closer to zero compared to the other scenarios, thus indicating a better FAAS's path in terms of collecting valuable information from the surveillance system. The next best algorithm followed was the closet area. As shown above in chapter 2, the reward function depends heavily on the movement and energy expended from the FAAS itself. Therefore for large movements, the reward will be more negative, so choosing the closet area will yield more favorable reward values. After moving in order yielded close to -0.2 reward value. Choosing the max reward from previous runs but not utilizing the best path from SARSA yielded values about -0.21. This can be explained because choosing the highest reward function every time will continually make the decision worse if it wasn't the best path. The worse algorithmic performances occurred when choosing the area that either had the highest time or energy constraint in the last timeslot. Based on the above analysis, we conclude that the proposed approach demonstrates superior performance among all the scenarios, achieving both the highest percentage of satisfied cameras and the largest FAAS's reward.

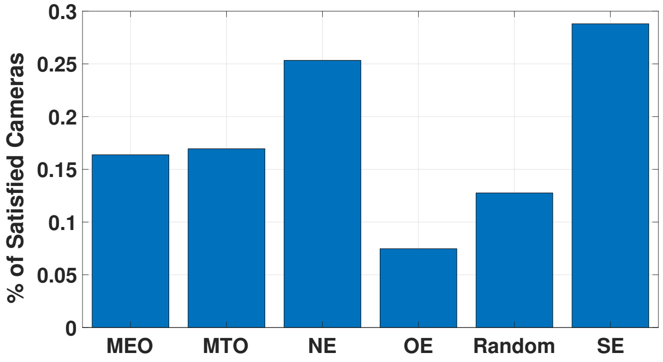

Figure 3.15: Percentage of Satisfied Cameras

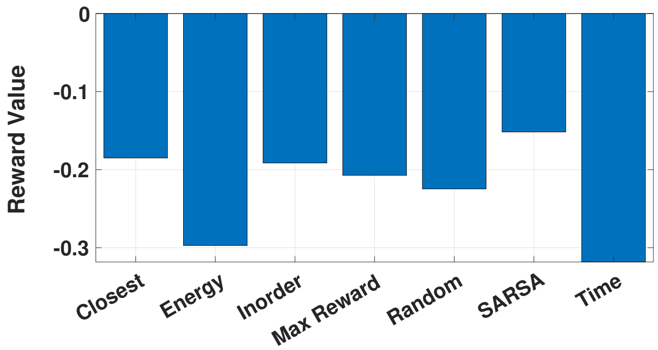

Figure 3.16: FAAS's Avg. Reward with respect to different offloading and FAAS's policy approaches

Chapter 4

Conclusion and Future Works

In this thesis, the problem of the IP cameras' data offloading strategies' determination in a surveillance system consisting of AoIs and assisted by MEC servers and a FAAS, in order the IP cameras to fulfill their energy and time QoS prerequisites, is studied. At first, a brief introduction of the thesis is presented, and explained surveillance systems, mobile edge computing servers, and drones are presented. Motivations followed, which included better public safety, utilizing resources more proficiently, and having dynamic means of data offloading. A more extensive description of drones and their applications is presented with the challenges and open problems that are associated with the research area. After, began the look into the system model which is composed of a surveillance area with different areas of interest each of which can have the same or different numbers of cameras and are supported by a MEC server and the FAAS if over the certain area. The communication and computation model is described, explaining the process for which the cameras will offload and determine their time and energy constraints. This is followed by explaining the game theory involved in the thesis and the satisfaction equilibrium explored. An in-depth look at the FAAS movement strategy created by SARSA the reinforcement learning algorithm is discussed. Then, the thesis introduced a low complexity and distributed learning approach that leads the IP cameras to converge to an SE or a GSE point based only on local information. The SARSA algorithm based on the FAAS's long-term objective, which is constructed by several factors (i.e., QoI, AoI's performance, FAAS's energy consumption) determines its optimal movement policy.

Results were then created in order to evaluate the proposed architecture. Three different types of results were created, which consisted of pure, scalability, and comparative. Pure results looked at the performance of first the non-cooperative game theory portion and how cameras choose to either offload or execute data locally. A table is then presented in which the parameters that either positively or negatively influence cameras satisfaction. This is followed by a look at convergence of the SARSA algorithm and its reward it chooses after learning the system. Finally, the pure results are wrapped up in showing that areas that have relatively high quality of information are visited more often than areas with competing quality. Scalability results are then analyzed in which we consider three different cases, increasing cameras in the area, allowing more strategies to be chosen by the camera system, and increasing areas of interest. It is shown that as the number of cameras increases the overall ratio of satisfied cameras is decreased. This is because of more competition amongst the system and overall increase in congestion. We do see that with increasing the number of strategies, the camera systems are more satisfied, this makes logical sense because better offloading is being performed. With the increase of areas of interest, we see that the average reward becomes more negative if the same number of timeslots is used, this comes from the fact that more exploring by the FAAS must first be performed. Finally, comparative results are presented, in which we compare first the techniques for the camera satisfaction selection and then compare choosing what area the FAAS visited in the next timeslot. Our satisfaction equilibrium method proved to be the most proficient in camera satisfaction and our SARSA algorithm outperformed the others when it came to choosing the best rewards.

The applications for this type of problem are endless. With the advancement in battery life and UAV performance, the idea of seeing these systems is not far out of real-world application. Especially, with the increase in IoT systems being connected every day, the need for better and more dynamic data offloading schemes is necessary. Part of our current and future work includes the testing of the proposed framework in a real city environment, as the outcomes will help extend and tune our theoretical model. Next, a look into what happens if multiple MECs or FAASs exist in the environment is needed.

References

[1] P. Natarajan, P. K. Atrey, and M. Kankanhalli, "Multi-camera coordination and control in surveillance systems: A survey," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 11, no. 4, p. 57, 2015.

[2] N. Kahar, R. Ahmad, Z. Hussin, and A. Rosli, "Embedded smart camera performance analysis," in *2009 International Conference on Computer Engineering and Technology*, vol. 2, pp. 79–83, IEEE, 2009.

[3] A. Kumbhar, F. Koohifar, I. Güvenç, and B. Mueller, "A survey on legacy and emerging technologies for public safety communications," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 97–124, 2016.

[4] E. Tsiropoulou, K. Koukas, and S. Papavassiliou, "A socio-physical and mobility-aware coalition formation mechanism in public safety networks," *EAI Endorsed Trans. Future Internet*, vol. 4, p. 154176, 2018.

[5] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (uavs) for energy-efficient internet of things communications," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7574–7589, 2017.

[6] G. Fragkos, P. A. Apostolopoulos, and E. E. Tsiropoulou, "Escape: Evacuation strategy through clustering and autonomous operation in public safety systems," *Future Internet*, vol. 11, no. 1, p. 20, 2019.

[7] D. Sikeridis, E. EleniTsiropoulou, M. Devetsikiotis, and S. Papavassiliou, "Self-adaptive energy efficient operation in uav-assisted public safety networks," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, IEEE, 2018.

[8] S. A. R. Naqvi, S. A. Hassan, H. Pervaiz, and Q. Ni, "Drone-aided communication as a key enabler for 5g and resilient public safety networks," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 36–42, 2018.

[9] D. Sikeridis, E. E. Tsiropoulou, M. Devetsikiotis, and S. Papavassiliou, "Wireless powered public safety iot: A uav-assisted adaptive-learning approach towards energy efficiency," *Journal of Network and Computer Applications*, vol. 123, pp. 69–79, 2018.

[10] G. Baldini, S. Karanasios, D. Allen, and F. Vergari, "Survey of wireless communication technologies for public safety," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 619–641, 2013.

[11] D. Sikeridis, E. E. Tsiropoulou, M. Devetsikiotis, and S. Papavassiliou, "Sociospatial resource management in wireless powered public safety networks," in *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pp. 810–815, IEEE, 2018.

[12] K. Muraoka, J. Shikida, and H. Sugahara, "Feasibility of capacity enhancement of public safety lte using device-to-device communication," in *2015 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 350–355, IEEE, 2015.

[13] D. Sikeridis, E. E. Tsiropoulou, M. Devetsikiotis, and S. Papavassiliou, "Context-aware wireless-protocol selection in heterogeneous public safety networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 2009–2013, 2018.

[14] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys Tutorials*, vol. 19, pp. 2322–2358, Fourthquarter 2017.

[15] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Game-theoretic learning-based qos satisfaction in autonomous mobile edge computing," in *2018 Global Information Infrastructure and Networking Symposium (GIIS)*, pp. 1–5, IEEE, 2018.

[16] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, and M. Debbah, "A tutorial on uavs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2019.

[17] P. Vamvakas, E. E. Tsiropoulou, and S. Papavassiliou, "On the prospect of uav-assisted communications paradigm in public safety networks," in *IEEE INFOCOM WKSHPS: WCNEE 2019: Wireless Communications and Networking in Extreme Environments*, pp. 1–6, IEEE, 2019.

[18] P. Vamvakas, E. E. Tsiropoulou, and S. Papavassiliou, "Dynamic provider selection & power resource management in competitive wireless communication markets," *Mobile Networks and Applications*, vol. 23, no. 1, pp. 86–99, 2018.

[19] S. Purbaya, E. Ariyanto, D. W. Sudiharto, and C. W. Wijiutomo, "Improved image quality on surveillance embedded ip camera by reducing noises," in *3rd Intern. Conf. on Science in Inform. Techn.*, pp. 156–159, IEEE, 2017.

[20] N. Chen, Y. Chen, Y. You, H. Ling, P. Liang, and R. Zimmermann, "Dynamic urban surveillance video stream processing using fog computing," in *IEEE 2nd Intern. Conf. on Multimedia Big Data*, pp. 105–112, IEEE, 2016.

[21] N. H. Motlagh, M. Bagaa, and T. Taleb, "Uav-based iot platform: A crowd surveillance use case," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128–134, 2017.

[22] B. Kim, H. Min, J. Heo, and J. Jung, "Dynamic computation offloading scheme for drone-based surveillance systems," *Sensors*, vol. 18, no. 9, p. 2982, 2018.

[23] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue," *IEEE robotics & automation magazine*, vol. 19, no. 3, pp. 46–56, 2012.

[24] M. Fasoulakis, E. E. Tsiropoulou, and S. Papavassiliou, "Satisfy instead of maximize: Improving operation efficiency in wireless communication networks," *Computer Networks*, 2019.

[25] S. M. Perlaza, H. Tembine, S. Lasaulce, and M. Debbah, "Satisfaction equilibrium: A general framework for qos provisioning in self-configuring networks," in *2010 IEEE GLOBECOM 2010*, pp. 1–5, IEEE, 2010.

[26] P. Promponas, P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Redesigning resource management in wireless networks based on games in satisfaction form," in *12th IFIP Wireless and Mobile Networking Conference*, pp. 1–6, IEEE, 2019.

[27] K. P. Valavanis and G. J. Vachtsevanos, *Handbook of unmanned aerial vehicles*. Springer, 2015.

[28] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, 2016.

[29] D. Gettinger and A. H. Michel, "Drone sightings and close encounters: An analysis," *Center for the Study of the Drone, Bard College*, 2015.

[30] S. Samarakoon, M. Bennis, W. Saad, M. Debbah, and M. Latva-Aho, "Ultra dense small cell networks: Turning density into energy efficiency," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1267–1280, 2016.

[31] I. Bor-Yaliniz and H. Yanikomeroglu, "The new frontier in ran heterogeneity: Multi-tier drone-cells," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 48–55, 2016.

[32] K. Gomez, A. Hourani, L. Goratti, R. Riggio, S. Kandeepan, and I. Bucaille, "Capacity evaluation of aerial lte base-stations for public safety communications," in *2015 European conference on networks and communications (EuCNC)*, pp. 133–138, IEEE, 2015.

[33] Q. Wu, J. Xu, and R. Zhang, "Capacity characterization of uav-enabled two-user broadcast channel," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1955–1971, 2018.

[34] D. Bamburry, "Drones: Designed for product delivery," *Design Management Review*, vol. 26, no. 1, pp. 40–48, 2015.

[35] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a uav-mounted cloudlet: Optimal bit allocation and path planning," *to appear*, 2016.

[36] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *2014 IEEE global communications conference*, pp. 2898–2904, IEEE, 2014.

[37] Z. Yun and M. F. Iskander, "Ray tracing for radio propagation modeling: Principles and applications," *IEEE Access*, vol. 3, pp. 1089–1100, 2015.

[38] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, "Efficient 3-d placement of an aerial base station in next generation cellular networks," in *2016 IEEE international conference on communications (ICC)*, pp. 1–5, IEEE, 2016.

[39] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal lap altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.

[40] S. Mumtaz, K. M. S. Huq, A. Radwan, J. Rodriguez, and R. L. Aguiar, "Energy efficient interference-aware resource allocation in lte-d2d communication," in *2014 IEEE International Conference on Communications (ICC)*, pp. 282–287, IEEE, 2014.

[41] M. Chen, W. Saad, and C. Yin, "Liquid state machine learning for resource allocation in a network of cache-enabled lte-u uavs," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pp. 1–6, IEEE, 2017.

[42] E. T. Ceran, T. Erkilic, E. Uysal-Biyikoglu, T. Girici, and K. Leblebicioglu, "Optimal energy allocation policies for a high altitude flying wireless access point," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 4, p. e3034, 2017.

[43] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. G. Giordano, A. Garcia-Rodriguez, and J. Yuan, "Survey on uav cellular communications: Practical aspects, standardization advancements, regulation, and security challenges," *IEEE Communications Surveys & Tutorials*, 2019.

[44] W. Khawaja, I. Guvenc, D. W. Matolak, U.-C. Fiebig, and N. Schneckenberger, "A survey of air-to-ground propagation channel modeling for unmanned aerial vehicles," *IEEE Communications Surveys & Tutorials*, 2019.

[45] M. M. Azari, F. Rosas, K.-C. Chen, and S. Pollin, "Optimal uav positioning for terrestrial-aerial communication in presence of fading," in *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, IEEE, 2016.

[46] E. E. Tsiropoulou, P. Vamvakas, and S. Papavassiliou, "Joint utility-based uplink power and rate allocation in wireless networks: A non-cooperative game theoretic framework," *Physical Communication*, vol. 9, pp. 299–307, 2013.

[47] E. E. Tsiropoulou, P. Vamvakas, G. K. Katsinis, and S. Papavassiliou, "Combined power and rate allocation in self-optimized multi-service two-tier femtocell networks," *Computer Communications*, vol. 72, pp. 38–48, 2015.

[48] M. R. Musku, A. T. Chronopoulos, D. C. Popescu, and A. Stefanescu, "A game-theoretic approach to joint rate and power control for uplink cdma communications," *IEEE Transactions on Communications*, vol. 58, no. 3, pp. 923–932, 2010.

[49] E. E. Tsiropoulou, A. Kapoukakis, and S. Papavassiliou, "Energy-efficient subcarrier allocation in sc-fdma wireless networks based on multilateral model of bargaining," in *2013 IFIP Networking Conference*, pp. 1–9, IEEE, 2013.

[50] E. Yaacoub and Z. Dawy, "A survey on uplink resource allocation in ofdma wireless networks," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 322–337, 2011.

[51] E. E. Tsiropoulou, A. Kapoukakis, and S. Papavassiliou, "Uplink resource allocation in sc-fdma wireless networks: A survey and taxonomy," *Computer Networks*, vol. 96, pp. 1–28, 2016.

[52] C. U. Saraydar, N. B. Mandayam, D. J. Goodman, *et al.*, "Efficient power control via pricing in wireless data networks," *IEEE transactions on Communications*, vol. 50, no. 2, pp. 291–303, 2002.

[53] E. E. Tsiropoulou, P. Vamvakas, and S. Papavassiliou, "Energy efficient uplink joint resource allocation non-cooperative game with pricing," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2352–2356, IEEE, 2012.

[54] J. Zheng, Y. Cai, Y. Liu, Y. Xu, B. Duan, and X. S. Shen, "Optimal power allocation and user scheduling in multicell networks: Base station cooperation using a game-theoretic approach," *IEEE Transactions on Wireless Communications*, vol. 13, no. 12, pp. 6928–6942, 2014.

[55] G. Katsinis, E. Tsiropoulou, and S. Papavassiliou, "Multicell interference management in device to device underlay cellular networks," *Future Internet*, vol. 9, no. 3, p. 44, 2017.

[56] T. Alpcan and T. Basar, "A game theoretic approach to decision and analysis in network intrusion detection," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, vol. 3, pp. 2595–2600, IEEE, 2003.

[57] E. E. Tsiropoulou, J. S. Baras, S. Papavassiliou, and G. Qu, "On the mitigation of interference imposed by intruders in passive rfid networks," in *International Conference on Decision and Game Theory for Security*, pp. 62–80, Springer, 2016.

[58] J. F. Nash *et al.*, "Equilibrium points in n-person games," *Proceedings of the national academy of sciences*, vol. 36, no. 1, pp. 48–49, 1950.

[59] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of AI Research*, vol. 4, pp. 237–285, 1996.

[60] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.

[61] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[62] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press, 2018.

[63] T. Jaakkola, M. I. Jordan, and S. P. Singh, "Convergence of stochastic iterative dynamic programming algorithms," in *Advances in neural information processing systems*, pp. 703–710, 1994.

[64] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári, "Convergence results for single-step on-policy reinforcement-learning algorithms," *Machine learning*, vol. 38, no. 3, pp. 287–308, 2000.

[65] E. Bertran and A. Sànchez-Cerdà, "On the tradeoff between electrical power consumption and flight performance in fixed-wing uav autopilots," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 11, pp. 8832–8840, 2016.