Satisfaction-aware Data Offloading in Surveillance Systems

Pavlos Athanasios Apostolopoulos, Marcos Torres, Eirini Eleni Tsiropoulou
pavlosapost,mtorres22,eirini@unm.edu
Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131, USA

ABSTRACT

In this paper, we exploit the capabilities of Fully Autonomous Aerial Systems' (FAAS) and the Mobile Edge Computing (MEC) to introduce a novel data offloading framework and support the energy and time efficient video processing in surveillance systems based on game theory in satisfaction form. 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 collectively 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 its data either remotely or locally. A non-cooperative game among the cameras is formulated to determine the amount of offloading data to the MEC server and/or the FAAS. The novel concept of Satisfaction Equilibrium (SE) is introduced where the IP cameras satisfy their minimum QoS prerequisites instead of maximizing their performance by wasting additional system resources. A distributed learning algorithm determines the IP cameras' stable data offloading, while a reinforcement learning algorithm determines the FAAS's movement among the AoIs exploiting the accuracy, timeliness, and certainty of the collected data by the IP cameras per AoI. The performance evaluation of the proposed framework is achieved via modeling and simulation.

CCS CONCEPTS

• **Networks** → Network resources allocation;

KEYWORDS

Surveillance Systems; Mobile Edge Computing; Reinforcement Learning; Satisfaction Games.

ACM Reference Format:
Pavlos Athanasios Apostolopoulos, Marcos Torres, Eirini Eleni Tsiropoulou. 2019. Satisfaction-aware Data Offloading in Surveillance Systems. In *14th Workshop on Challenged Networks (CHANTS '19), October 25, 2019, Los Cabos, Mexico.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3349625.3355437

1 INTRODUCTION

Surveillance systems have recently gained great attention due to the increased number of terrorist attacks, which challenge the public safety and homeland security [10]. With the advent of Internet of Things (IoT), the smart Internet Protocol (IP) cameras have enabled the surveillance systems to capture real-time video and process it either locally [8], or remotely at the cloud computing environment. However, the surveillance systems confront the challenges of increased computing demand and time criticality in processing the recorded information. The use of Unmanned Aerial Vehicles (UAVs) and remote computing capabilities has been shown to improve the performance of the surveillance systems.

1.1 Related Work and Motivation

The authors in [12] have introduced an image uploading process from the IP cameras to the cloud, where the images captured by the IP cameras are processed at the cloud to decrease the local processing cost. In [3], 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 track vehicles' movement. In [9] an UAV-based crowd surveillance system is introduced where the UAVs capture videos that either process them on board or offload them to MEC servers.

Despite the advantages introduced, the use of UAVs still requires human control. To address this issue, the Fully Autonomous Aerial Systems (FAAS) have been recently introduced. The FAAS is a flying robotic system equipped with sensors, computing resources, wireless communication interfaces or any combination of them and is able to operate fully autonomously with no human intervention.

ACM ISBN 978-1-4503-6933-6/19/10…$15.00
https://doi.org/10.1145/3349625.3355437

One common characteristic of the aforementioned approaches is their assumption that each entity involved in the surveillance system aims at maximizing its Quality of Service (QoS). However, the maximization of each involved entity's QoS is a sub-optimal solution, which under several circumstances lead to energy inefficiencies. Towards this direction, the games in satisfaction form have been introduced in Game Theory [4], where the autonomous entities aim to "satisfy" their minimum QoS prerequisites in a distributed manner instead of targeting at maximizing their QoS. In this paper, 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 to ultimately support the energy and time efficient video processing in surveillance systems consisting of several IP cameras.

1.2 Contributions and Outline

Specifically, we introduce a surveillance system paradigm consisting of areas of interest (AoI) with IP cameras. The cameras partially offload the computing tasks related to the videos' processing to the MEC server that is associated with the AoI or to the FAAS, if the latter is flying above the AoI, while the rest are executed locally at the IP cameras. Each utilized IP camera experiences a time and energy overhead in order to offload part of its data and process the remaining part of the data locally (Section 2). A holistic utility function is introduced representing the IP cameras' level of achieved QoS, while accounting for their time and energy constraints associated with the video processing procedure. To realize an autonomous system operation, 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 (Section 3). A Reinforcement Learning (RL) algorithm is adopted that determines the FAAS's movement by considering several factors associated with the quality of information from the AoIs (Section 4). Detailed numerical results are presented to evaluate the proposed framework's operation (Section 5.1), while a comparative evaluation is provided to reveal its drawbacks and benefits (Section 5.2).

2 SYSTEM MODEL

A surveillance area of size $L \times L$ consisting of several AoIs (e.g., banks, airports) randomly distributed with coordinates $Z_i = (X_i, Y_i), X_i, Y_i \leq L$ is considered. The set of AoIs is denoted by $\mathbb{A} = \{1, \dots, i, \dots, A\}$. Each AoI has a set of IP cameras that collect and process data [8] $\mathbb{C}_i = \{1, \dots, j, \dots, C_i\}$ and a MEC server M_i, which supports their computing demands. A FAAS flies over the AoIs with a velocity v and altitude d. At each timeslot, the FAAS receives and processes data from the AoI's cameras of which the FAAS is located above. The set of timeslots is $\mathbb{T} = \{1, \dots, t, \dots, T\}$ and at each timeslot, the FAAS is assumed to cover and support only one AoI. The set of collected data by each IP camera $j \in \mathbb{C}_i$ belonging to the AoI i per timeslot t is denoted as $D_{ij}^{(t)} = (B_{ij}^{(t)}, CP_{ij}^{(t)}, \phi_{ij}^{(t)}, dt_{ij}^{(t)}, de_{ij}^{(t)})$, where $B_{ij}^{(t)}[bits]$ is the total collected information, $CP_{ij}^{(t)} = \phi_{ij}^{(t)} B_{ij}^{(t)}$ is the number of required CPU cycles to process the data, where $\phi_{ij}^{(t)} > 0$ is the level of the video processing task's intensity, $dt_{ij}^{(t)}$ denotes the time constraint during which the data should be processed, and $de_{ij}^{(t)}$ is the IP camera's energy availability for the timeslot t. The amount of collected data can be partitioned into subsets of specific size, which can be offloaded to the MEC server or the FAAS, assuming that the last one is located at the AoI i for the timeslot t. For the rest of the analysis, we drop the (t) for notational convenience.

We denote $\mathbf{s}_i = (\mathbf{s_{i1}}, \dots, \mathbf{s_{ij}}, \dots, \mathbf{s_{iC_i}})$ the vector of strategies for the cameras residing in the AoI i, where $\mathbf{s_{ij}} = (ch_{ij}, a_{ij})$ and $a_{ij} \in [0, 1]$ is the camera's data offloading percentage, and $ch_{ij} = 0$ if the camera offloads its $a_{ij} \cdot B_{ij}$ data to the MEC server M_i, while $ch_{ij} = 1$ if it offloads to the FAAS. Considering that the FAAS is located at the AoI i, then for each other AoI $i' \in \mathbb{A}, i' \neq i$ it holds true that $ch_{i'j} = 0$, $\forall j \in \mathbb{C}_{i'}$. Thus, assuming that the AoIs do not interfere with each other, the IP camera's j in AoI i uplink data rate is:

$$R_{ij} = W_i \cdot log\left(1 + \frac{p_{ij} g_{ij}}{\sigma_0^2 + \sum\limits_{k \in \mathbb{C}_i \setminus \{j\}, ch_{ik} = ch_{ij}, a_{ik} \neq 0} p_{ik} g_{ik}}\right) \quad (1)$$

where W_i is the AoI's i bandwidth, p_{ij} is the camera's j transmission power, g_{ij} is the channel gain between the 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 $a_{ij} B_{ij}$ 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. The allocated computing capability to each IP camera j in order to remotely process its offloaded data is given as:

$$f_{ij} = \frac{a_{ij} B_{ij} \phi_{ij}}{\sum\limits_{k \in \mathbb{C}_i \setminus \{j\}, ch_{ik} = ch_{ij}} a_{ik} B_{ik} \phi_{ik}} \cdot ((1 - ch_{ij}) f_{M_i} + ch_{ij} f_F) \quad (2)$$

where the first factor of Eq. 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), 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.

$$O_{ij}^{t} = \max\{\frac{a_{ij}\cdot B_{ij}}{R_{ij}} + \frac{a_{ij}B_{ij}\phi_{ij}}{f_{ij}}, \frac{(1-a_{ij})B_{ij}\phi_{ij}}{f_{ij}^{l}}\} \quad (3)$$

while its overall energy consumption is formulated as:

$$O_{ij}^{e} = p_{ij}\frac{a_{ij}\cdot B_{ij}}{R_{ij}} + (1-a_{ij})B_{ij}\phi_{ij}e_{ij} \quad (4)$$

3 SATISFACTION EQUILIBRIUM OPERATION

Each IP camera 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, we formulate a generic utility function that represents each IP camera's QoS as follows.

$$u_{ij}(\mathbf{s_{ij}}, \mathbf{s_{-ij}}) = \begin{cases} -(\frac{dt_{ij}-O_{ij}^{t}}{dt_{ij}})\cdot(\frac{de_{ij}-O_{ij}^{e}}{de_{ij}}) & \text{if } O_{ij}^{t} \geq dt_{ij}, O_{ij}^{e} \geq de_{ij} \\ (\frac{dt_{ij}-O_{ij}^{t}}{dt_{ij}})\cdot(\frac{de_{ij}-O_{ij}^{e}}{de_{ij}}) & otherwise \end{cases} \quad (5)$$

where $\mathbf{s_{-ij}}$ is the strategy vector of all the IP cameras of the AoI i except the IP camera j. Assuming that the FAAS is located at the AoI i, it is evident by Eq. 5, 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 one 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 $G_i = [\mathbb{C}_i, \{S_{ij}\}_{j\in\mathbb{C}_i}, \{u_{ij}\}_{j\in\mathbb{C}_i}, \{h_{ij}\}_{j\in\mathbb{C}_i}]$, where $\mathbb{C}_i$ is the set of the IP cameras in the AoI i, and considering that the FAAS is located in the AoI i, then $S_{ij} = \{(a_{ij}^{n}, 0), \cdots, (a_{ij}^{N}, 0), \cdots, (a_{ij}^{n}, 1), \cdots, (a_{ij}^{N}, 1)\}$, while $S_{ij} = \{(a_{ij}^{n}, 0), \cdots, (a_{ij}^{N}, 0)\}$ otherwise, and a_{ij}^{n} is the n^{th} available offloading percentage, thus $a_{ij}^{n} \in [0,1]$, $\forall n \leq N, N \in \mathbb{N}$. Moreover, u_{ij} is the AoI's i IP camera's j utility as expressed in Eq. 5, and h_{ij} is the satisfaction correspondence defined as follows [4].

$$h_{ij}(\mathbf{s_{-ij}}) = \{\mathbf{s_{ij}} \in S_{ij} | u_{ij}(\mathbf{s_{ij}}, \mathbf{s_{-ij}}) \geq 0\} \quad (6)$$

Definition 1 (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 game G_i, if $\forall j \in \mathbb{C}_i$, $\mathbf{s_{ij}^+} \in h_{ij}(\mathbf{s_{-ij}^+})$.*

At the SE, the IP cameras satisfy their minimum QoS prerequisites without overspending the system's resources.

Towards determining the SE for each non-cooperative game G_i, we propose the Distributed Learning Satisfaction Equilibrium Algorithm (DLSE). Each camera evaluates its utility (Eq. 5) by receiving its allocated remote computing capability (Eq. 2) from the MEC server or FAAS and the interference factor (i.e., $\sum_{k\in\mathbb{C}_i, ch_{ik}=ch_{ij}, a_{ik}\neq 0} p_{ik}g_{ik}$ in Eq. 1) and converges to the strategy $\mathbf{s_{ij}^+}$. 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 $\boldsymbol{\pi_{ij}}(r) = (\pi_{ij}^{(1)}(r), \cdots, \pi_{ij}^{(l_{ij})}(r), \cdots, \pi_{ij}^{(L_{ij})}(r))$, 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 $r > 0$. The initial probability distribution for each IP camera is $\pi_{ij}^{(l_{ij})}(r = 0) = 1/L_{ij}, \forall l_{ij} \leq L_{ij}$, 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 $\boldsymbol{\pi_{ij}}$ 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.

Definition 2 (Clipping Action). *At each non-cooperative game G_i, an IP camera j has a clipping action $\mathbf{s_{ij}^c} \in S_{ij}$ iff $\forall \mathbf{s_{-ij}} \in S_{-ij}, \mathbf{s_{ij}^c} \in h_{ij}(\mathbf{s_{-ij}})$, where $S_{-ij} = S_{i1} \times \cdots \times S_{i(j-1)} \times S_{i(j+1)} \times \cdots \times S_{iC_i}$ [11].*

Therefore, Definition 2 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., $\mathbf{s_{ij}}(r) = \mathbf{s_{ij}^c}$. Thus, 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, which is defined as follows.

Definition 3 (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* $\mathbb{C}_i$ *given by* $\mathbb{C}_i^s$ *and* $\mathbb{C}_i^u$*, such that* $\forall j \in \mathbb{C}_i^s$*,* $\mathbf{s_{ij}} \in h_{ij}(\mathbf{s_{-ij}})$ *and* $\forall j' \in \mathbb{C}_i^u$*,* $h_{ij'}(\mathbf{s_{-ij'}}) = \emptyset$*.*

In a nutshell, given the existence of at least one SE point for each game G_i, if 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.

4 FAAS MOVEMENT BASED ON REINFORCEMENT LEARNING

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, 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 corresponding MEC server M_i is less overloaded. Specifically, a Reinforcement Learning approach is adopted that enables the FAAS to autonomously decide the most appropriate AoI to visit and support at each timeslot, trying to optimize a specific objective that accounts for performance in terms of satisfied cameras, FAAS energy consumption and Quality of Information (QoI), formally defined below.

Three different quality factors are introduced that collectively capture the QoI that each AoI's surveillance system provides. We consider that several processes (e.g., object detection, movement detection) can be executed locally at the cameras' and remotely at the MEC servers' and FAAS's computing resources to assess each AoI's QoI. Such processes assign values to the following quality factors at each timeslot based on each camera's captured data.

(a) **Accuracy** refers to how the observed information inside each AoI conforms to the reality. After the processing of the IP cameras' collected data, the number of the correctly detected events AE_{ij} is evaluated and the IP camera's accuracy is $q_{ij}^{acc} = \frac{AE_{ij}}{TE_{ij}}$, where TE_{ij} is the total number of events that were captured. The overall accuracy of the AoI i is defined as $Q_i^{acc} \triangleq \frac{1}{C_i} \sum_{j \in \mathbb{C}_i} q_{ij}$.

(b) **Timeliness** refers to the availability of the information at the desired time. The timeliness factor is defined as $q_{ij}^{tls} \triangleq \frac{D_t}{D_t + O_{ij}^t}$, 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. The AoI's overall timeliness factor is $Q_i^{tls} \triangleq \frac{1}{C_i} \sum_{j \in \mathbb{C}_i} q_{ij}^{tls}$.

(c) **Certainty**: refers to the measurement of confirmation of the information and is strictly related to each IP camera's hardware characteristics (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 and it is denoted as q_{ij}^{crt}. The overall certainty of the AoI i is evaluated as $Q_i^{crt} \triangleq \frac{1}{C_i} \sum_{j \in \mathbb{C}_i} q_{ij}^{crt}$.

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

$$QoI_i = w_i^{acc} \frac{\sum_{t' \leq t} Q_i^{\prime acc}}{t} + w_i^{tls} \frac{\sum_{t' \leq t} Q_i^{\prime tls}}{t} + w_i^{crt} \frac{\sum_{t' \leq t} Q_i^{\prime crt}}{t} \tag{7}$$

where $w_i^{acc}, w_i^{tls}, w_i^{crt} \in [0, 1], w_i^{acc} + w_i^{tls} + w_i^{crt} = 1$ are the corresponding weights of each quality factor.

The support provided by the FAAS to the AoI with high QoI is important, since the overall surveillance system's performance and effectiveness could be increased by decreasing the delay between an event's detection and further reactive actions. Also, 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 $E^p = E_P \cdot \sum_{j \in \mathbb{C}_i, ch_{ij}=1} a_{ij} B_{ij} \phi_{ij}$. 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 $E^m = E_M \cdot \frac{\sqrt{(X_i - X_{i'})^2 + (Y_i - Y_{i'})^2}}{v}$, where $E_M[Watts]$ is the FAAS's constant consumed energy while moving with velocity v. Based on the above discussion, we formulate the reward that the FAAS experiences while visiting an AoI i as follows.

$$rw_i = -\frac{\epsilon_3 \cdot \frac{E^p + E^m}{E}}{\epsilon_1 \cdot QoI_i + \epsilon_2 \cdot P_i} \tag{8}$$

where $P_i = \frac{|\mathbb{C}_i^s|}{C_i}$, $\mathbb{C}_i^s = \{j \in \mathbb{C}_i | u_{ij} \geq 0\}$ 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 $\epsilon_1, \epsilon_2, \epsilon_3 \in [0, 1]$ denote the weights of the AoI's QoI, the 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.

In the following, we adopt a Reinforcement Learning (RL) approach that enables the FAAS to autonomously learn its dynamic environment and decide which AoI to visit per timeslot towards maximizing its long-term objective (Eq. 8) [7] .

The RL algorithms demonstrate good results [1, 5, 13] 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 and SARSA [13] algorithms, which via stochastic approximation conditions lead the decision maker to converge to its optimal decision policy with high probability [6]. In our case, for

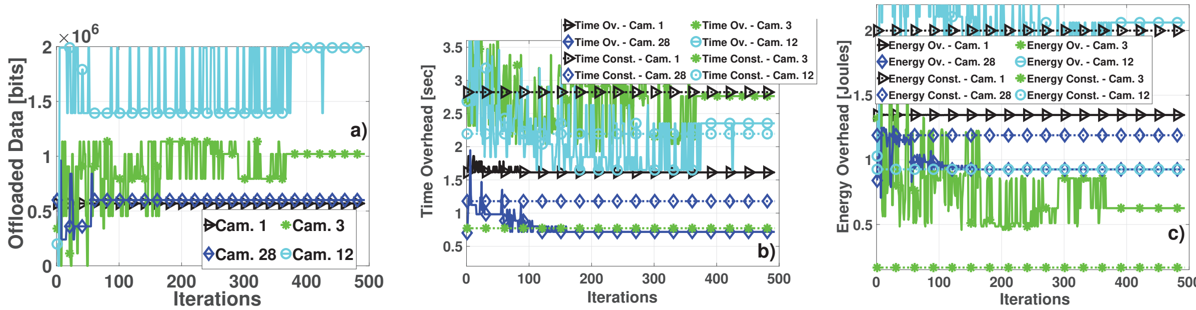

Figure 1: (a)Data Offloading (b)Time and (c)Energy Overhead versus the DLSE Algorithm's iterations for convergence

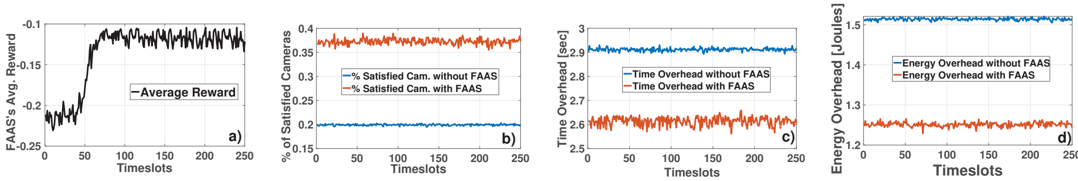

Figure 2: (a) FAAS's Avg. Reward (b)Percentage of Satisfied Cameras, (c)Time, and (d)Energy Overhead versus timeslots

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 $\mathbb{A}$), 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. 8) 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 at a specific state.

5 NUMERICAL RESULTS

In this section, a detailed numerical performance evaluation and comparative study of the proposed architecture is conducted through modeling and simulations. We consider a surveillance system consisting of $A = 7$ AoIs with $C_i = 30, \forall i \in \mathbb{A}$ cameras and size $500m \times 500m$. We have $B_{ij} \in [1000, 5000]KB$ 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. Also, we have $e_{ij} = 10^{-9} J/Cycle$, $W_i = 5MHz$, $f^l_{ij} \in [10^{-2}, 10^{-1}]$, $\sigma_0^2 = 10^{-13}$, $p_{i,j} \in [0, 1]W$, $dt_{i,j} \in [0, \frac{CP_{ij}}{f_j}]sec$, $de_{i,j} \in [0, CP_{ij}e_{ij}]J$, $g_{i,j} = \frac{1}{d^2_{ij}}$, where d_{ij} is the IP camera's j distance from the MEC server M_i or FAAS, $w^{acc}_i = 0.333$, $w^{tls}_i = 0.333$, $w^{crt}_i = 0.333$, $Q'^{acc}_i, Q'^{tls}_i, Q'^{crt}_i \in [0, 1]$, $E_P = 10^{-9}J$, $E_M = 0.0013W$, $E = 17.28 \cdot 10^6 J$, $\epsilon_1 = 0.35$, $\epsilon_2 = 0.55$, $\epsilon_3 = 0.10$, the duration of a timeslot is $1h$, and $v = 6.25m/s$ [2]. The evaluation focuses on: (i) pure operation of the proposed framework, (ii) scalability performance, and (iii) comparative analysis.

5.1 Pure Operation of the Algorithm

In Fig. 1.a-c, the amount of offloaded data, along with the corresponding time and energy overhead are presented for four different IP cameras. The IP cameras with ID 12 and 3 have strict time and energy constraints (Fig. 1.b,c), thus they choose to offload large amount of data to the MEC server in order to satisfy their QoS prerequisites. However, even if they choose such a strategy, they still cannot meet their QoS demands and the DLSE algorithm converges to an GSE point. 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.

Fig. 2.a 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. 8). In Fig. 2.b-c, it is shown that the percentage of satisfied cameras is significantly higher and the corresponding IP cameras' time and energy overheads are lower as the time evolves, when the FAAS visits those areas, thus showing the great benefits of adopting the FAAS in the overall considered architecture. Fig. 3.a, b depict the average number of FAAS's visits and the average Quality of Information per AoI in a time frame of 250 timeslots. It is observed that if an AoI has high QoI, the SARSA algorithm

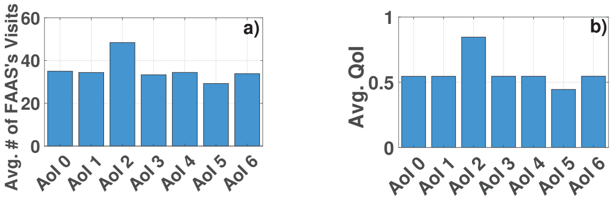

Figure 3: (a)Avg. no of FAAS's visits/AoI (b)AoIs' Avg. QoI

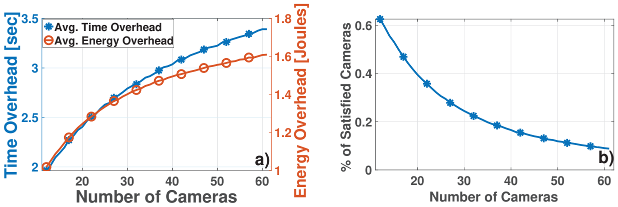

Figure 4: Scalability Analysis

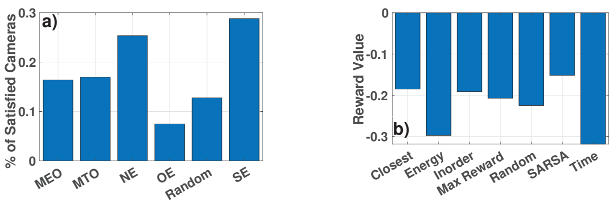

Figure 5: (a)% of Satisfied Cameras (b)FAAS's Avg. Reward w.r.t. different offloading and FAAS's policy approaches

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.

Fig. 4a,b show the time and energy overhead and the percentage of satisfied IP cameras for 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. thus the IP cameras' time and energy overhead increases, while the percentage of the cameras that meet their QoS prerequisites decreases.

5.2 Comparative Analysis

Finally, comparative scenarios are presented to confirm the advantages of our proposed approach in terms of: (i) Satisfaction Equilibrium's benefits, and (ii) the benefits of the adoption of reinforcement learning. Regarding the former, five different comparative approaches are presented: i) minimizing the energy (MEO), ii) minimizing the time overhead (MTO), iii) determining the Nash Equilibrium (NE), vi) offloading the data entirety (OE), and v) random amount of data is offloaded to the MEC server. As clearly shown in Fig. 5.a, the novel concept of SE resulted to the highest percent of satisfied cameras. Similarly, in Fig. 5.b, the proposed reinforcement learning SARSA algorithm is compared against several other different scenarios regarding the FAAS's navigation among the AoIs. 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 clearly 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.

ACKNOWLEDGMENT

The research of Dr. Eirini Eleni Tsiropoulou was conducted as part of the NSF CRII-1849739.

REFERENCES

[1] Pavlos Athanasios Apostolopoulos, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. 2018. Demand response management in smart grid networks: A two-stage game-theoretic learning-based approach. *Mobile Networks and Applications* (2018), 1–14.

[2] Eduard Bertran and Alex Sànchez-Cerdà. 2016. On the tradeoff between electrical power consumption and flight performance in fixed-wing UAV autopilots. *IEEE Trans. on Veh. Tech.* 65, 11 (2016), 8832–8840.

[3] Ning Chen, Yu Chen, Yang You, Haibin Ling, Pengpeng Liang, and Roger Zimmermann. 2016. Dynamic urban surveillance video stream processing using fog computing. In *IEEE 2nd Intern. Conf. on Multimedia Big Data*. IEEE, 105–112.

[4] Michail Fasoulakis, Eirini Eleni Tsiropoulou, and Symeon Papavassiliou. 2019. Satisfy Instead of Maximize: Improving Operation Efficiency in Wireless Communication Networks. *Computer Networks* (2019).

[5] Georgios Fragkos, Pavlos Athanasios Apostolopoulos, and Eirini Eleni Tsiropoulou. 2019. ESCAPE: Evacuation strategy through clustering and autonomous operation in public safety systems. *Future Internet* 11, 1 (2019), 20.

[6] Tommi Jaakkola, Michael I Jordan, and Satinder P Singh. 1994. Convergence of stochastic iterative dynamic programming algorithms. In *Advances in neural information processing systems*. 703–710.

[7] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. *Journal of AI Research* 4 (1996), 237–285.

[8] NF Kahar, RB Ahmad, Z Hussin, and ANC Rosli. 2009. Embedded smart camera performance analysis. In *2009 International Conference on Computer Engineering and Technology*, Vol. 2. IEEE, 79–83.

[9] Naser Hossein Motlagh, Miloud Bagaa, and Tarik Taleb. 2017. UAV-based IoT platform: A crowd surveillance use case. *IEEE Com. Mag.* 55, 2 (2017), 128–134.

[10] Prabhu Natarajan, Pradeep K Atrey, and Mohan Kankanhalli. 2015. Multi-camera coordination and control in surveillance systems: A survey. *ACM Trans. on Mult. Comp., Com., and Appl.* 11, 4 (2015), 57.

[11] Samir M Perlaza, Hamidou Tembine, Samson Lasaulce, and Mérouane Debbah. 2010. Satisfaction equilibrium: A general framework for QoS provisioning in self-configuring networks. In *2010 IEEE GLOBECOM 2010*. IEEE, 1–5.

[12] Setiya Purbaya, Endro Ariyanto, Dodi Wisaksono Sudiharto, and Catur Wirawan Wijiutomo. 2017. Improved image quality on surveillance embedded IP camera by reducing noises. In *3rd Intern. Conf. on Science in Inform. Techn.* IEEE, 156–159.

[13] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction.* MIT press.