Availability Modeling for Drone Image Processing Systems with Adaptive Offloading

Fumio Machida
Department of Computer Science
University of Tsukuba
Tsukuba, Japan
machida@cs.tsukuba.ac.jp

Ermeson Andrade
Department of Computing
Federal Rural University of Pernambuco
Recife, Brazil
ermeson.andrade@ufrpe.br

Abstract— **Availability of a computing process running on a flying drone is an essential quality aspect for mission-critical drone systems. Computing tasks such as image processing tasks can be lost when the process encounters a failure. Since the failure probability of the process depends on workload intensities, reducing drone workloads by computation offloading or load-balancing must have impacts on the system availability. While many existing studies discuss the performance-cost trade-off associated with computation offloading, potential impacts on the system availability have not been deeply investigated yet. In this paper, we propose stochastic models for estimating the availability, the performance, and the energy consumption of a drone system with image processing tasks that can be either offloaded to a fog node or distributed to a collaborative drone. Our comprehensive numerical analysis with the proposed model clarifies the trade-offs among the availability, the throughputs, and the energy consumption under different computation modes. Furthermore, we propose an adaptive offloading scheme that can change the computation modes dynamically according to workload intensities and network conditions. A simulation study with a phased mission scenario shows that the proposed adaptive scheme can achieve high availability with 26% of energy reduction and less than 4% of throughput losses.**

Keywords—availability, energy consumption, drone, modeling, offloading

I. Introduction

Recent smart drone systems can adopt Artificial Intelligence (AI) processing on chips to recognize the environment around the drone from real-world images captured by cameras. Image processing tasks such as real-time object recognition and classification are typically computation-intensive and energy-consuming tasks. Thus, as an alternative computing mode, computation offloading to any available node in a networked computing infrastructure becomes a promising solution for the drone in order to reduce the workload and save battery life [1]. While performance-energy trade-offs by computation offloading have been studied in the literature [2][3][25], little is known about the availability impacts. The availability of computing processes for image processing is an important quality aspect that is also affected by the types of computing nodes and communication links. Availability analysis and design are fundamental challenges in the Quality of Service (QoS) management for smart drone systems.

The availability of image processing tasks on a drone is affected by several probabilistic and controlled factors. One of the significant probabilistic factors is the failure of a computing process on the drone that may be caused by software faults or bugs [4]. The failure rate of the drone's application process may depend on the execution state, as the process is more failure-prone under high workload conditions than in an idle state [5]. The amount of workload is not stable during the mission period, and thereby the process failure rate is a probabilistic factor that can influence the availability. When tasks are offloaded to another computing node, the failure rate of the process at the offloaded node also needs to be considered. Besides, the reliability of communication links for offloading also affects the system availability. On the other hand, the choice of the computation mode, whether using computation offloading or not, is a controlled factor that impacts availability. The decision of computation mode needs to be carefully made in consideration with other probabilistic factors involved in the drone mission.

In this paper, we conducted a comprehensive trade-off analysis among the availability, the performance, and the energy consumption of a drone-based image processing system considering three different computation modes. The image processing tasks are assumed to run on either a primary drone, a computation node in a fog computing infrastructure, or a nearby collaborative drone. To quantitatively analyze the availability, the performance, and the energy consumption under various probabilistic factors, we use stochastic reward nets (SRNs) [11] to model the state transitions of system components and network links. Based on the analysis of availability-performance-energy trade-offs among different computation modes, we propose a model-based adaptive offloading scheme named *Adapt-off* (***Adapt**ive **off**loading*) that selects the computation modes according to the workload states and communication link conditions. *Adapt-off* determines computation modes that can keep high availability under given conditions and save energy consumption by offloading or distributing tasks to other nodes when the expected performance loss is acceptable. Our simulated experiment shows that *Adapt-off* can save up to 26% of the energy consumption with less than 4% of throughput losses to keep high availability by changing computation modes according to different working conditions.

The contribution of the work is summarized as follows.

1. We analytically investigate the availability impact of a drone image processing system in three different computation modes: i) single drone processing, ii) fog offloading, and iii) drone load-balancing.
2. Through the numerical experiments on the proposed SRN model with realistic parameter values, we show the trade-offs to be considered among the availability, the performance, and the energy consumption of the drone's image processing system.
3. We proposed a model-based adaptive offloading scheme, called *Adapt-off*, and evaluate its effectiveness through simulation experiments.

The rest of the paper is organized as follows. In Section II, we explain the system architecture and the different computation modes of a drone image processing system. In Section III, we introduce the availability measure and its trade-offs to the performance and the energy consumption. In Section IV, we present SRNs for representing the system behavior considering the three computation modes. Section V details the numerical results and shows the effectiveness of *Adapt-off*. Section VI discusses the related work. Finally, Section VII presents our conclusion and briefly introduces future works.

II. System architecture

We consider a smart drone system using AI to recognize the real-world environment and decide actions for an assigned mission such as disaster rescue, urban surveillance, and remote monitoring. Image processing is an essential computational task that needs to be processed for AI functions. Since image processing is a computation-intensive and energy-consuming task, it is not always effective to execute the tasks on the drone. Offloading tasks to other computation nodes through available communication networks can be considered as an alternative computation mode [1]. Alternative computation nodes may be a fog node in a fog computing infrastructure or another drone flying nearby [25]. Besides the single drone mode without relying on computation offloading, we can consider a fog offloading mode and a drone load-balancing mode shown in Figure 1. The architectures of the different computation modes are detailed below.

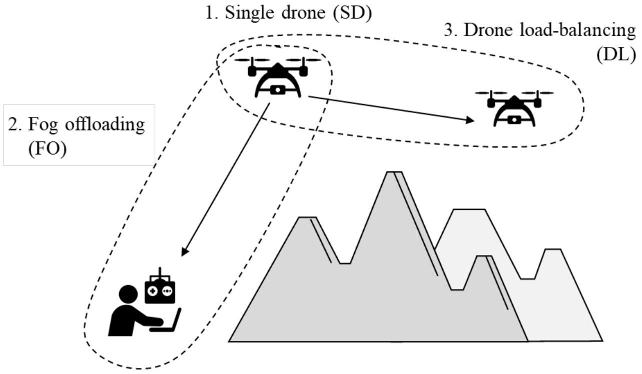


Figure 1. Three computation modes for drone image processing

A. Single drone mode

As a baseline computation mode, images captured by cameras on a drone are directly processed on a chip on the drone. The images are taken by a specific sampling rate that can be adjusted according to the status of the mission. For example, in a mission-critical condition such as finding survivors in a disaster-affected area, the processing rate should be higher to enhance the sensitivity. Object detection or human recognition is a typical example of image processing tasks that can be effectively handled by machine learning models [6][7]. We assume machine learning models are trained off-line and deployed to the drone in advance. The processing rate of the input images is determined by the deployed model as well as the performance of the software platform and the chip. The results of image processing are used in subsequent tasks such as issuing alerts and updating the trajectory.

B. Fog offloading mode

Computation tasks on the drone can be offloaded to a fog node in a fog computing infrastructure to reduce the computation costs and save battery life [1]. Fog computing is a virtualized platform that provides computing resources between end devices and traditional cloud data centers [8]. Any computing device accessible from the drone through the network can be considered a fog node. For commercial drones, they are typically controlled by users' mobile devices. A drone establishes a secure network connection with a fog node over wireless communication links such Wi-Fi, 4G LTE, or 5G. In fog offloading mode, all images taken on the drone are sent to the fog node. The fog node can deploy machine learning models to process the images, while its processing rate depends on the computational resources used in the fog node.

C. Drone load-balancing mode

In a severe condition where the drone has no stable connection to the base station on the ground, collaboration with other drones flying in the same area can be considered another alternative computation mode. Two drones can communicate to each other via the local wireless network if they are flying at a sufficiently close distance. A primary drone may ask a helper drone to share image processing workloads via the wireless network connection. If the helper drone deploys the same machine learning model and has available computing resources, it can serve as a collaborative drone to load-balance the image processing tasks for the primary drone. Since the process on the helper drone is also constrained by the battery life, workloads may need to split between the drones by a certain ratio.

III. Availability and performance

In this section, we discuss the QoS aspect of different computation modes for a drone image processing system. The quality measures considered in our trade-off analysis are system availability, throughput performance, and drone energy consumption. While this section gives the definitions of the measures and clarifies their trade-offs concerned in the design, further analysis is conducted on our comprehensive models proposed in the subsequent sections.

A. Availability

The system is said to be available when the function of the system works properly at a given time instant. The system can become unavailable when any critical components in the system fail. The failed components need to be repaired during the operation to make the system available again. As component failures, in this study, we focus on computational process and network link failures. The process failure may be caused by software defects or any configuration errors. On the other hand, network link failures are mainly caused by environmental factors that deteriorate the quality of wireless communication links. The availability can be quantified by the probability that the system is in an available state. Let $\pi_{m,s}$ be the probability that the system in state s of the computation mode m. The computation mode m is either the single drone (S), the fog offloading (F), or the drone load-balancing (D). For the single drone mode, the computation state s can be idle (i), processing (p), or down (d). The availability of the system in the drone mode A_S can be expressed as

$$A_S = \pi_{S,i} + \pi_{S,p}, \tag{1}$$

where the total probability satisfies

$$\sum_{s\in\{i,p,d\}} \pi_{S,s} = 1. \tag{2}$$

For the fog offloading and load-balancing modes, we should consider more states considering the availability of a fog node or a helper drone.

B. *Performance*

While several performance measures can be considered, we focus on the throughput performance of image processing tasks in this study. The throughput represents how many requests are processed in the system for a unit time. High throughput performance is encouraged for a smart drone system to improve the recognition capability. The maximum throughput is limited by the resource capacity, which is determined by the performance of the chip. Furthermore, system dynamics also affect the effective throughput that can be quantified by multiplying the processing state probability with processing rate. For the single drone mode, the throughput performance P_S can be given by

$$P_S = \pi_{S,p} \cdot \mu_d, \tag{3}$$

where μ_d represents the service rate for image processing on the drone. For the fog offloading and load-balancing modes, the throughput is affected by the service rate of the fog node or the helper drone as well as their state probabilities.

C. *Energy consumption*

Energy efficiency is a crucial factor in mission-critical drone systems, as they need to fly and complete the assigned tasks under the constraints of the limited battery life. The energy consumption of the drone depends on the computation workloads, and hence there is a trade-off between the computation performance and energy conservation. The more the drone processes captured images, the more it consumes the energy, while the throughput also increases. For the drone processing mode, the expected energy consumption E_S can be estimated by

$$E_S = e_i\pi_{S,i} + e_p\pi_{S,p} \tag{4}$$

where e_i and e_p represent the average energy consumption in idle and processing states, respectively. It is natural to assume that $e_i < e_p$. When we change the mode to the fog offloading or the drone load-balancing, the state probability $\pi_{S,p}$ decreases, and hence the energy consumption can be reduced.

D. *Trade-off analysis*

The measures defined above represent different quality aspects, while they are correlated to each other in terms of the state probabilities. From the definitions, we can easily derive the trade-off relations among them. First, we can observe that $\pi_{S,p}$ affects all the measures as given in (1), (3), and (4). Increasing $\pi_{S,p}$ results in higher availability and higher throughput, and an increase in energy consumption. There is a clear trade-off between availability/performance and energy. Next, by comparing A_S and P_S, it is noted that $\pi_{S,i}$ only contributes to availability. It means that the throughput is not improved even when the system is highly available due to a higher idle state probability. In other words, high availability is considered a necessary condition for high throughput performance. On the other hand, by comparing A_S and E_S, we observe that the higher value of $\pi_{S,i}$ is preferable both in terms of the availability and energy efficiency because e_i is smaller than e_p. In short, the balance between $\pi_{S,i}$ and $\pi_{S,p}$ is a matter to consider for making a good trade-off. Since $\pi_{S,i}$ and $\pi_{S,p}$ are varied under the constraint of total probability (2), component failure probabilities do affect the balance. Although we only discuss the trade-off in the single drone mode, the relations among the measures and the constraint are generally applicable to the fog offloading and the drone load-balancing modes.

E. *Limitation of the simple model*

The abovementioned formulation is helpful to understand the general relations among the quality measures, but some essential dynamics that affect these measures are dropped from such a simple analysis. Specifically, the following probabilistic factors are not adequately addressed and need more comprehension.

1) Varying workload condition: A drone working on a real mission observes the varying demand for image processing according to the states of the mission. High throughput performance is achievable only under an acceptable level of high workload conditions. When the workload intensity is not so significant, the process mostly stays in the idle state, and the throughput is low. The balance between $\pi_{S,i}$ and $\pi_{S,p}$ is dominated by the workload state, while such a workload factor is not incorporated in the simple formula discussed above.

2) State-dependent failure probability: The process failure probability in a processing state is not equal to that in an idle state. When the process is in an idle state, it has less opportunity to encounter a failure. Since the process failure rate is state-dependent, the balance between $\pi_{D,i}$ and $\pi_{D,p}$ also affects the total process failure probability that in turn affects availability under the constraint (1). It means that the state probabilities cannot be computed without knowing different failure rates in different states.

3) Dependence on the network link availability: When the drone chooses the fog offloading mode or the drone load-balancing mode, the availability of communication links is another probabilistic factor that needs to be taken into account. In these modes, the system is not available if the communication links are not established. The throughput on the fog node or the helper drone cannot be counted when the links are not available even though the fog node or the helper drone works properly. Similar to the workload variation, the availability of communication links may change depending on the environment where the drone is flying. The variation of the link availability is not also incorporated in the simple formula discussed above.

To overcome those limitations and to understand the availability trade-offs more comprehensively, we introduce the availability model for the smart drone system based on stochastic Petri nets that is presented in the next section.

IV. Availability modeling

We use SRNs for analyzing the availability of a smart drone system for the following reasons. First, SRNs provide a compact and relatively intuitive representation of stochastic behaviors of target systems by Petri net notations. Second, reward functions defined on SRNs allow us to estimate various performance measures such as steady-state availability and service throughput. In this section, we first briefly introduce SRNs and then present availability models for a drone system considering three computing modes.

A. *SRNs*

SRN is a variant of stochastic Petri nets (SPN) that supports reward functions to evaluate performance measures associated with the stochastic dynamics of the systems. An SPN is a

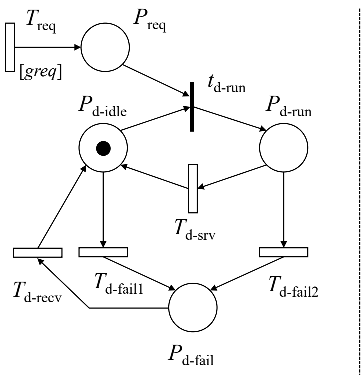
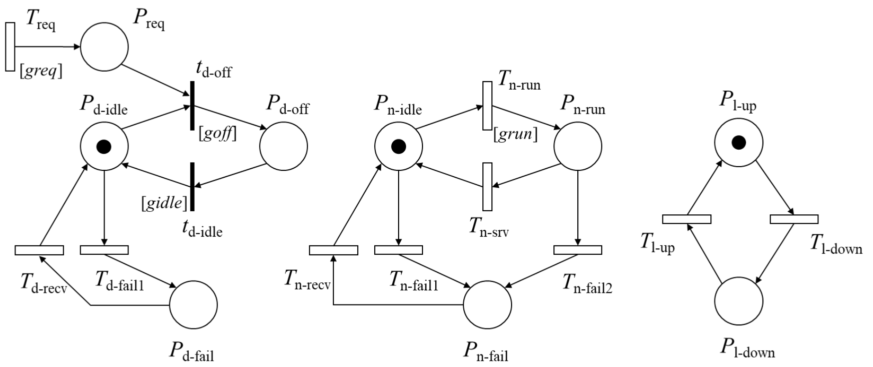


Figure 2. SRN for the drone processing mode



(a) Drone model

(b) Fog node model

(c) Wireless link model

Figure 3. SRN for the fog offloading mode

directed bipartite graph that consists of two kinds of nodes called *place* and *transition*, represented by a circle and a rectangle, respectively. Places may have *tokens*, which are represented as dots in the graphical notation. Tokens in the places constitute a *marking* that corresponds to a specific system state. A transition is *enabling* when a specified number of tokens are deposited in the places connected by incoming arcs (i.e., a directed arc from a place to a transition). When an enabling transition *fires*, all the tokens are removed from the input places, and new tokens are deposited to the places connected by outgoing arcs (i.e., a directed arc from a transition to a place), resulting in a new making. In this manner, system state transitions can be modeled as the marking transitions in SPN. By assigning probability distributions representing the firing times for individual transitions, an SPN can be transformed into a state-space model. When we define reward functions on the set of markings of interests, we can compute any transient and steady-state rewards resulting from the stochastic behaviors of the systems. For further details on SRN, readers can refer to [10][11].

B. Drone processing model

Figure 2 shows the SRN that models the state transitions of a drone image processing system in the drone processing mode. Repeated requests for image processing are represented by timed transition T_{req} that can fire by the request arrival rate γ. The firing of T_{req} is constrained by guard [$greq$], which enables T_{req} when there is no token in P_{req}. In other words, we assume that one request can be buffered in the queue. A token is deposited in P_{req} by firing T_{req}, then subsequently immediate transition $t_{\text{d-run}}$ fires if there is a token in $P_{\text{d-idle}}$. An immediate transition is a special type of transition that has zero transition time. The firing of $t_{\text{d-run}}$ represents that the image processing starts on the drone. When a token is deposited in $P_{\text{d-run}}$, transitions $T_{\text{d-srv}}$ and $T_{\text{d-fail2}}$ are enabled. If $T_{\text{d-srv}}$ fires first, a token is deposited in $P_{\text{d-idle}}$ again, which represents the drone is ready for processing the next request. On the other hand, if $T_{\text{d-fail2}}$ fires first, a token is deposited in $P_{\text{d-fail}}$, which corresponds to the failure state. The recovery of a failed process is represented by $T_{\text{d-recv}}$. Meanwhile, $T_{\text{d-fail}}$ represents the process failure in the idle state. By assigning different failure rates λ_{d1} and λ_{d2} for $T_{\text{d-fail}}$ and $T_{\text{d-fail2}}$, respectively, we can incorporate the state-dependent failure probabilities discussed in Section III-E. To compute the service availability, the throughputs, and the energy consumption, the reward functions are defined as shown in Table I. The definitions of the functions follow (1), (3), and (4). μ_d represents the service rate on the drone, and e_i and e_p represent the average energy consumption in idle and processing states, respectively. The state probability is computed by prob(#P_x==1), representing the probability that a token is deposited in P_x.

TABLE I. REWARD FUNCTIONS FOR THE DRONE PROCESSING MODE

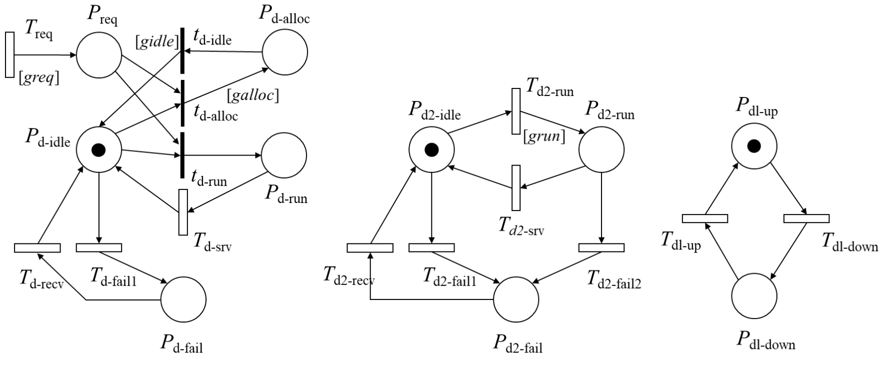
Name	Measure	Function
avail	Service availability	prob(#$P_{\text{d-idle}}$==1) + prob(#$P_{\text{d-run}}$==1)
thru	Throughput performance	prob(#$P_{\text{d-run}}$==1) * μ_d
energy	Energy consumption	prob(#$P_{\text{d-idle}}$==1) * e_i + prob(#$P_{\text{d-run}}$==1) * e_p

C. Fog offloading model

Figure 3 shows the SRNs for the drone image processing system in the fog offloading mode. The model consists of three SRN subnets corresponding to a drone, a fog node, and a wireless link between them. The drone model is updated from the one presented in Section IV-B. It models that a request is offloaded to the fog node instead of being processed on the drone. To represent the system behavior, these subnets interact in the following way. When immediate transition $t_{\text{d-off}}$ fires, a token is deposited in $P_{\text{d-off}}$, which represents the drone starts offloading a request. Due to guard function [$goff$], $t_{\text{d-off}}$ is enabled only when $P_{\text{l-up}}$ has a token, meaning that the communication link is available. When a token is deposited in $P_{\text{d-off}}$, by guard function [$grun$], $T_{\text{n-run}}$ in the fog node model is enabled. The communication delay for request offloading is considered as the rate ω, which is assigned to $T_{\text{n-run}}$. By firing $T_{\text{n-run}}$, it deposits a token in $P_{\text{n-run}}$, which subsequently enables $t_{\text{d-idle}}$ by guard function [$gidle$] in the drone model. It represents that the fog node starts processing the offloaded request, and hence the drone can return to the idle state. The next request does not arrive unless the fog node becomes idle (i.e., a token is deposited in $P_{\text{n-idle}}$). Failure and recovery transitions are considered in all the subnets. In the fog node model, the different failure rates λ_{n1} and λ_{n2} are assigned to $T_{\text{n-fail}}$ and $T_{\text{n-fail2}}$, respectively. The reward functions defined for computing the quality measures are shown in Table II.

TABLE II. REWARD FUNCTIONS FOR THE FOG OFFLOADING MODE

Name	Measure	Function
avail	Service availability	(prob(#$P_{\text{d-idle}}$==1) + prob(#$P_{\text{d-off}}$==1))*(prob(#$P_{\text{n-idle}}$==1) + prob(#$P_{\text{n-run}}$==1))* prob(#$P_{\text{l-up}}$==1)
thru	Throughput performance	prob(#$P_{\text{n-run}}$==1) * μ_n
energy	Energy consumption	prob(#$P_{\text{d-idle}}$==1) * e_i + prob(#$P_{\text{d-off}}$==1) * e_o



(a) Primary drone model (b) Helper drone model (c) Wireless link model

Figure 4. SRN for the drone load-balancing mode

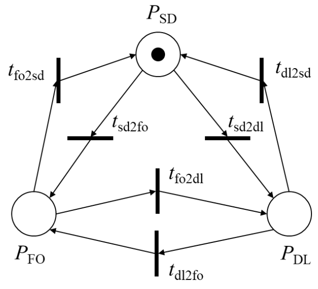


Figure 5. SRN for mode transitions

The service is available when all of the components are available. Since requests are processed in the fog node, throughput performance is affected by the service rate μ_n on the fog node. Meanwhile, the energy consumption is affected by both the average energy consumptions in the idle and offloading states that are denoted as e_i and e_o, respectively.

D. Drone load-balancing model

Figure 4 shows the SRNs for the drone image processing system in the drone load-balancing mode. The model consists of three SRN subnets corresponding to the primary drone, the helper drone, and the wireless communication link between the drones. This model uses a helper drone instead of a fog node for processing requests. The places and transitions of the helper drone model are analogous to the ones in the fog node model in Section IV-C. Since the requests are load-balanced between two drones in this mode, there are two immediate transitions $t_{\text{d-alloc}}$ and $t_{\text{d-run}}$ for triggering the request allocation to the helper node or the execution on the drone, respectively. While each immediate transition can fire equally with the probability of 0.5, the firing of $t_{\text{d-alloc}}$ is restricted by guard function [$galloc$], which is enabled when a token is deposited in $P_{\text{dl-up}}$ (i.e., when the communication link is available). Most of the other transition rules are the same as presented in the fog offloading model. Table III shows the reward functions for computing the quality measures. It should be noted that the service is available even if the helper drone or the wireless link is not available since requests can be processed on the drone. We assume that the image processing rate on the helper drone is the same as the one for the primary drone. For energy consumption, e_o represents the average energy consumption in the request allocation state on the drone.

TABLE III. REWARD FUNCTIONS FOR THE DRONE LOAD-BALANCING MODE

Name	Measure	Function
avail	Service availability	prob(#$P_{\text{d-idle}}$==1) + prob(#$P_{\text{d-run}}$==1) + prob(#$P_{\text{d-alloc}}$==1)
thru	Throughput performance	(prob(#$P_{\text{d-run}}$==1) + prob(#$P_{\text{d2-run}}$==1) * μ_n
energy	Energy consumption	prob(#$P_{\text{d-idle}}$==1) * e_i + prob(#$P_{\text{d-run}}$==1) * e_p + prob(#$P_{\text{d-alloc}}$==1) * e_o

E. Adaptive offloading model

The three computation modes discussed above are not necessarily mutually exclusive during a mission period. We can consider distinct adaptive control of computation modes that are dynamically selected depending on workload conditions and accessibilities to a fog node or a collaborative drone. To determine the condition where computation mode is switched from one to another, we can exploit the model presented above to analyze the trade-offs among the expected quality measures. We call this approach *Adapt-off* and investigate its effectiveness compared with non-adaptive computation modes in phased mission scenarios detailed in Section V.

The model for *Adapt-off* can be simply constructed by a combination of three SRNs for individual computation modes. On top of that, it requires a subnet for representing the selected mode, as shown in Figure 5. At each instant, the drone image processing system operates in either single drone (SD), fog offloading (FO), or drone load-balancing (DL) mode, which is represented by a token deposited in either P_{SD}, P_{FO}, or P_{DL}, respectively. We assume that the mode transition times are negligible, and guard functions are assigned to all the immediate transitions. The guard functions represent the external rules to control the mode changes.

In order to determine the most appropriate mode under a given condition, we define a utility function that specifies the preference of the mode in consideration of the availability, throughput, and energy consumption measures. Utility functions are often used for evaluating the aggregated quality of a service [35][36]. Let $A_m, P_m,$ and E_m be the expected availability, throughput, and energy consumption under the computation mode m for a given environmental state. We define the utility function as

$$U_m = (c_A A_m)^{\alpha_A} \cdot (c_P P_m)^{\alpha_P} \cdot [c_E(E_{\max} - E_m)]^{\alpha_E}\,, \qquad (5)$$

where $c_A, c_P, c_E, \alpha_A, \alpha_P,$ and α_E are the parameters to specify the preference of the performance, and $E_{\max}$ is the maximum power consumption of the drone in the processing state. The value of $E_{\max} - E_m$ represents the energy efficiency and thus is preferred to be higher. As a result, the mode that can expect a higher utility value should be selected as the appropriate mode for the given condition. Note that the different parameter value assignments represent different preferences of the performance measures. For instance, if the user prioritizes the energy efficiency over the availability and throughput, a higher value of c_E and/or α_E should be used. We assume the parameter values are determined by users in advance in consideration of the mission and the expected environment.

V. NUMERICAL EXPERIMENTS

To compare the QoS of a drone image processing system in different computation modes, we conduct numerical experiments on the SRNs proposed in Section IV. In the first

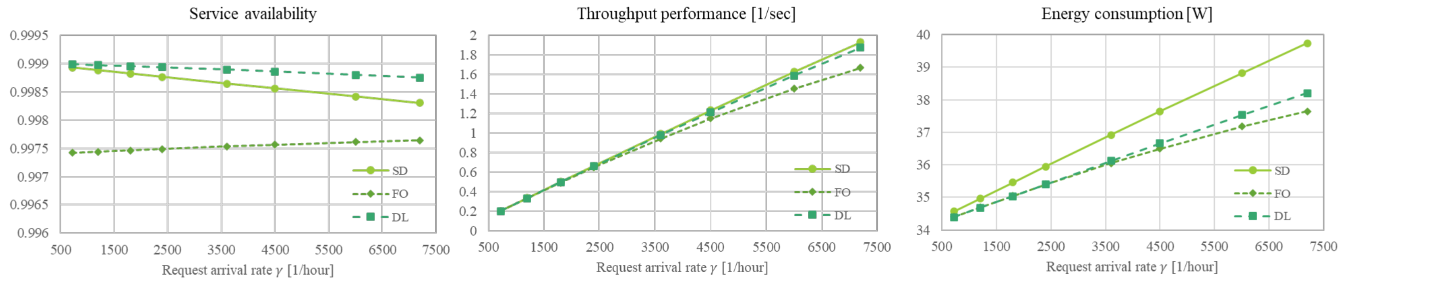


Figure 6. Comparison of the single drone (SD), the fog offloading (FO), and the drone load-balancing (DL) mode.

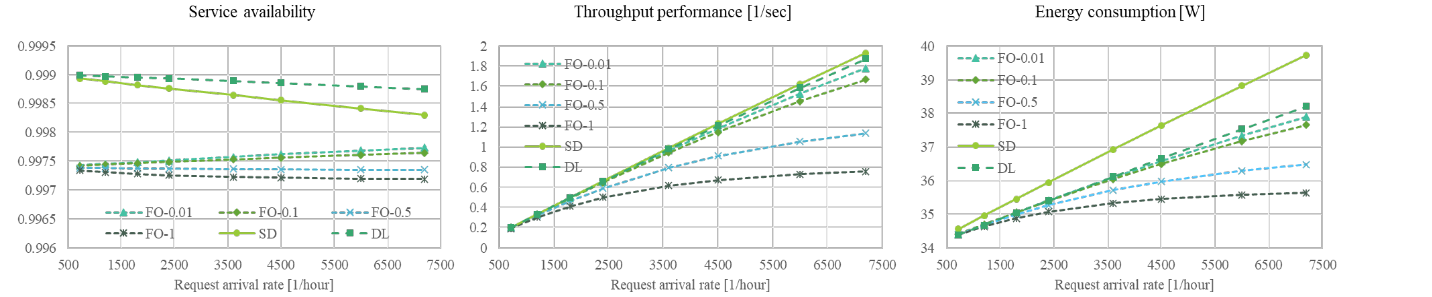


Figure 7. Sensitivity analysis of the service time on the fog node {0.01, 0.1, 0.5, 1 [sec]}.

experiments, we show general properties of individual modes where we assume a fog node and a helper drone are accessible with stable network links. In the second experiment, we analyze the impacts of the network reliability by varying the failure rates of communication links. Finally, we conduct simulated experiments to show the effectiveness of *Adapt-off* for a synthetic phased mission scenario.

A. Experimental setting

For the numerical experiments, we choose the default parameter values shown in TABLE IV.

TABLE IV. PARAMETER VALUES USED FOR THE EVALUATION

Variable	Description	Value [1/hour]
γ	Request arrival rate	720
ν_d	Service rate on a drone	36000
ν_n	Service rate on a fog node	36000
λ_d	Process failure rate on a drone in an idle state	0.002976190
λ_{d2}	Process failure rate on a drone in a processing state	0.013888889
μ_d	Process recovery rate on a drone	3
λ_n	Process failure rate on a fog node in an idle state	0.000462963
λ_{n2}	Process failure rate on a fog node in a processing state	0.001388889
μ_n	Process recovery rate on a fog node	2
ω	Communication rate between a drone and a fog node or another drone	18000
λ_l	Communication link disconnection rate	0.5
μ_l	Communication link connection rate	360
e_i	Energy consumption of a drone in an idle state [W]	34
e_p	Energy consumption of a drone in a processing state [W]	64
e_o	Energy consumption of a drone in an offloading state [W]	45

Note that some values cannot be deterministically given or known, and thus, we vary the values in our sensitivity analysis. While the default request arrival rate is set to 720 (per hour), representing that a request arrives every five seconds, the value is varied up to 7200 in our analysis. Since cameras on smart drones can shot images at more than one frame per second, the range of requests rate is in a feasible and reasonable range. We fix the service rate for image processing on drones to 36000 (i.e., 0.1 second per request), while we vary the service rate of the fog node in the range of [360, 360000] in a sensitivity analysis. Similarly, the link disconnection rates are varied in the range [0.5, 120] for analyzing the sensitivity to link reliability in Section V-D. Other parameters are chosen arbitrarily for experiments based on some prior knowledge in the domain [12][13][28], which can be adjusted when any specific system configurations are given.

We use SPNP [10] to implement the proposed SRNs and solve the models to compute the expected reward values. The Gauss-Seidel method is used for numerically computing the steady-state solution. We can obtain an exact solution for each model with given parameter values and hence do not need repeated experiments with random samples like simulation.

B. Architecture comparison

By setting the default parameter values to the proposed SRNs, we compute the service availabilities, the throughputs, and the expected energy consumptions for the three computation modes. Figure 6 shows the results of the computed values under different workloads (i.e., γ is varied from 720 to 7200) and computation modes.

Interestingly, each computation mode has its own advantage in a specific quality measure. DL achieves the highest availability regardless of the workloads, SD yields the highest throughput performance, while FO is the best in terms of reduced energy consumption on the drone. It is reasonable that DL achieves the highest availability because the primary drone can continue the service even when the helper drone fails. In terms of the throughput, SD is preferable because it is not affected by the latency and bandwidth of communication channels. The energy consumption on the drone in FO is the smallest as it minimizes the workloads on the drone. The results clearly show that a desirable computation mode can vary depending on what quality measure is more prioritized during a drone mission.

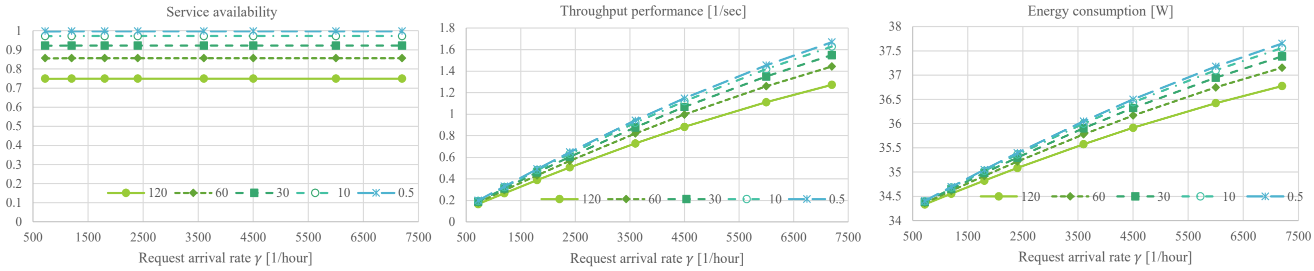


Figure 8. Sensitivity analysis of the network disconnection rate {120, 60, 30, 10, 0.5 [1/hour]} for FO

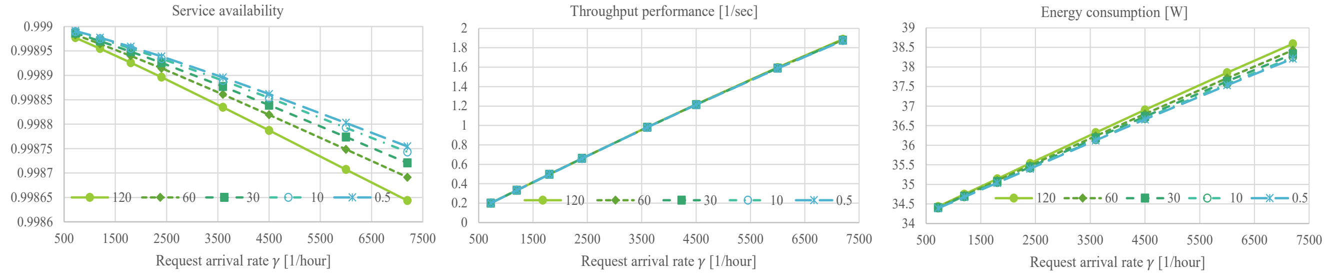


Figure 9. Sensitivity analysis of the communication disconnection failure rate {120, 60, 30, 10, 0.5 [1/hour]} for DL

C. *Impacts of fog node's service rate*

In the above comparison, we assume that the service rate on the fog node is equal to the drone. This assumption may not hold in real system configurations. The processing power of a fog node depends on the node type and available resources. In order to analyze the impacts of the fog node's service rate on the availability, performance, and energy consumption, we conduct a sensitivity analysis by varying the value of $1/\nu_n$ in $\{0.01, 0.1, 0.5$ and $1\}$. The results are shown in Figure 7. In addition to SD and DL, the results of FO-*x* are plotted, where *x* corresponds to the value of $1/\nu_n$. While the service availability and the throughput are improved by using the fog node with higher processing rates, the energy consumption of the drone can also increase. Nevertheless, the advantage of FO against SD and DL in terms of low energy consumption still holds in all the range of the service rates. Moreover, for service availability and throughput, FO is the worst choice regardless of the service rates chosen. Therefore, we use the default value of ν_n in the following comparative analysis.

D. *Impacts of the link reliabilities*

Since FO and DL can be adopted when a fog node or another collaborative drone is accessible, the quality measures for these modes must be affected by the communication link reliability. To investigate this, we conduct sensitivity analyses by varying the disconnection rate of the communication links in [0.5, 120]. Figure 8 and Figure 9 show the results of the sensitivity analysis on FO and DL models, respectively.

For FO, the availability and the throughput performance decrease by increasing the disconnection rate of the communication link. Meanwhile, the energy consumption is reduced by increasing the link failure rate. This is because the drone tends to be in an idle state to wait for the communication links to become available for the offloading. Nevertheless, choosing FO with an unreliable communication link is not encouraged, considering a significant loss in availability and throughput performance.

For DL, the availability decreases due to the increased link disconnection rate, while the throughput is not much affected because the primary drone can process the requests even when the helper drone is not accessible. The difference in the energy consumption is also marginal between the cases with different link reliabilities, but lower link reliability contributes to smaller energy consumption, as expected. Considering the availability decrease due to the deterioration of the link reliability, choosing DL is not preferable, especially when request arrival rates are high.

E. *Evaluation of Adapt-Off*

The sensitivity analysis results above lead us to consider potential requirements for adaptively changing computation mode in accordance with communication link states and request demands. As introduced in Section IV.E, *Adapt-Off* is a model-based adaptive offloading scheme that can address such requirements. We evaluate the effectiveness of *Adapt-Off* through simulated phased mission scenarios.

We consider two missions, both of which consist of four phases, as shown in Figure 10. Mission A consists of prefixed phases, while Mission B contains some randomized factors on the phases that also include the branches. Each phase of the mission has a specific request arrival rate γ, a fog node link disconnection rate λ_{ln}, and a drone link disconnection rate λ_{ld}. For Mission A, the drone is supposed to sojourn in each phase equally for ten minutes. For mission B, the sojourn time in each phase is varied in the range of [8, 12] minutes. The value of λ_{ln} is also randomly chosen from {0.5, 60, 120}, representing reliable, unstable, and unreliable communication links, respectively. The accessibility to a helper drone is also considered a probabilistic factor represented by split phases for phase 2 and phase 4. We assume that either phase 2a or phase 2b occurs at communication links probability 0.5, and similarly, either phase 4a or phase 4b occurs at probability 0.5.

To determine the computation mode from the conditions of each phase, we use the utility function as defined by (5) in Section IV-E. We set the parameter values of the function as shown in TABLE V.

TABLE V. PARAMETER VALUES FOR THE UTILITY FUNCTION

Parameter	c_A	c_P	c_E	α_A	α_P	α_E	$E_{\max}$
Value	10	10	1	0.1	0.1	0.8	64

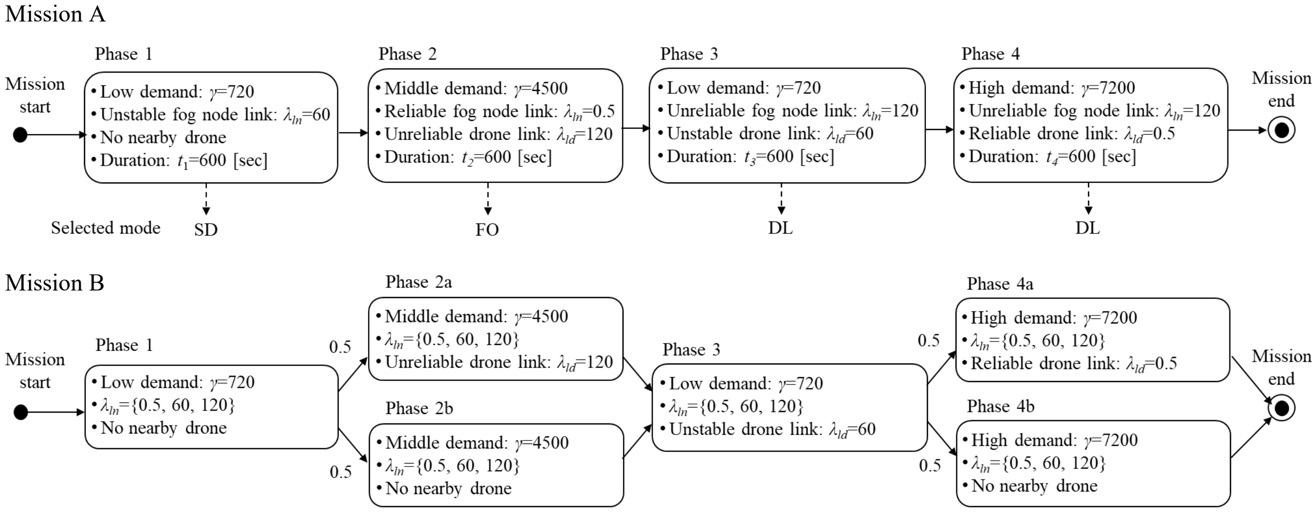


Figure 10. Phased mission scenarios for evaluating *Adapt-Off*: Mission A consists of the phases with fixed parameter values, while Mission B contains randomized factors such as the duration of each phase, the fog node link failure rate, and the existence or absence of a helper drone.

As the values of the availability and the throughput can be smaller than one, we multiply these values by ten ($c_A, c_P = 10$). The defined utility function assigns a relatively higher weight for energy efficiency ($\alpha_E = 0.8$), but it depends on the preference of a user. We use these values just for comparative evaluation of *Adapt-Off* against non-adaptive methods.

The utility function determines the decision rule of the mode. The utility values vary depending on the workload states, the availability of another drone, and the communication link qualities. Figure 11 shows the decision rule obtained from the defined utility function by varying the request arrival rates, the communication link failure rates for a fog node and a helper drone. The three matrices correspond to different workload conditions ($\gamma = 720, 4500, 7200$). In each matrix, each column corresponds to a link disconnection rate of the drone-drone communication, while each row corresponds to a link disconnection rate of the drone-fog-node communication. In each cell, the utility values of SD, FO, and DL are displayed vertically in this order. The node color represents the preferable mode which provides the highest utility value.

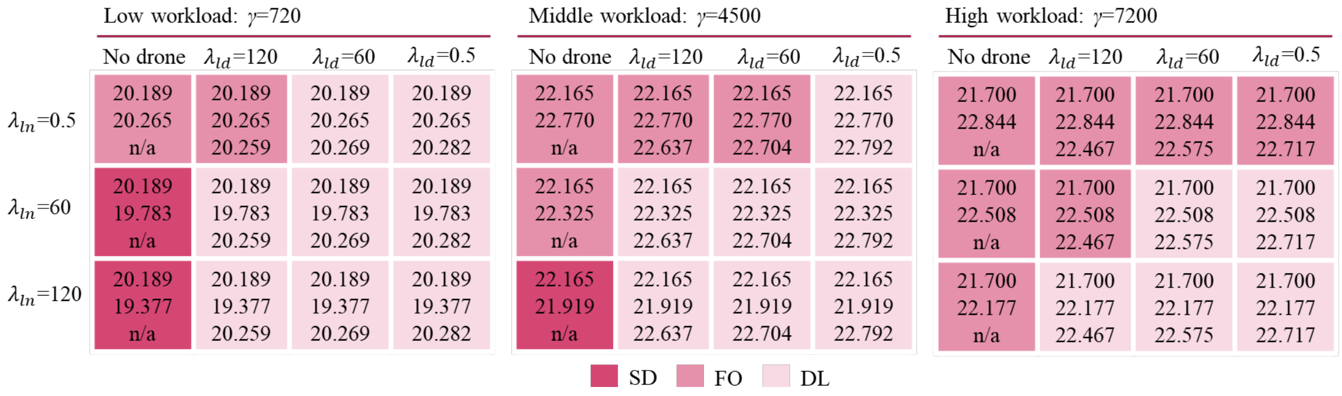
For low workload condition ($\gamma = 720$), if the reliability of the communication link to the fog node is not stable ($\lambda_{ln} = 60, 120$) and there is no nearby helper drone, SD is the best computation mode. If the fog node link is stable ($\lambda_{ln} = 0.5$) and the communication link to the other drone is not stable ($\lambda_{ld} = 120$), FO is chosen as the best mode. For other cases under the low workload condition, it is better to employ a helper drone to change the mode to DL. For middle and high workload conditions ($\gamma = 4500, 7200$), the boundaries among SD, FO, and DL change. The relative advantage of FO increases when workloads become higher, as FO is the most energy-efficient mode among others in such conditions. Based on the obtained decision rule, the desirable mode in each phase of Mission A is determined as shown in Figure 10. For Mission B, the mode is determined dynamically in the simulation experiments explained later in this section.

To evaluate the mission performance, we define the following aggregated performance measures corresponding to availability, performance, and energy consumption.

1) Average availability: Expected service availability varies across the phases, and hence we compute the average availability of *n* phases in the mission for mode *m* by

$$\mathcal{A}_m = \frac{1}{T}\sum_{j=1}^{n} A_{m,j} \cdot t_j, \tag{6}$$

where T is the total mission time, $A_{m,j}$ represents the expected availability in computation mode *m* and phase *j*.



	Low workload: γ=720				Middle workload: γ=4500				High workload: γ=7200			
	No drone	λ_{ld}=120	λ_{ld}=60	λ_{ld}=0.5	No drone	λ_{ld}=120	λ_{ld}=60	λ_{ld}=0.5	No drone	λ_{ld}=120	λ_{ld}=60	λ_{ld}=0.5
λ_{ln}=0.5	20.189 20.265 n/a	20.189 20.265 20.259	20.189 20.265 20.269	20.189 20.265 20.282	22.165 22.770 n/a	22.165 22.770 22.637	22.165 22.770 22.704	22.165 22.770 22.792	21.700 22.844 n/a	21.700 22.844 22.467	21.700 22.844 22.575	21.700 22.844 22.717
λ_{ln}=60	20.189 19.783 n/a	20.189 19.783 20.259	20.189 19.783 20.269	20.189 19.783 20.282	22.165 22.325 n/a	22.165 22.325 22.637	22.165 22.325 22.704	22.165 22.325 22.792	21.700 22.508 n/a	21.700 22.508 22.467	21.700 22.508 22.575	21.700 22.508 22.717
λ_{ln}=120	20.189 19.377 n/a	20.189 19.377 20.259	20.189 19.377 20.269	20.189 19.377 20.282	22.165 21.919 n/a	22.165 21.919 22.637	22.165 21.919 22.704	22.165 21.919 22.792	21.700 22.177 n/a	21.700 22.177 22.467	21.700 22.177 22.575	21.700 22.177 22.717

SD FO DL

Figure 11. A mode decision rule of *Adapt-Off*: Three 3x4 matrices correspond to the decision rules for different workload conditions ($\gamma = 720, 4500, 7200$). In each matrix, each row represents a link disconnection rate of drone-fog-node communication ($\lambda_{ln} = 0.5, 60, 120$), while each column represents a link disconnection rate of drone-drone communication ($\lambda_{ld} = 0.5, 60, 120$) or the state of no collaborative drone. In each cell, the expected utility values of SD, FO, and DL are displayed vertically in the order. The color of the cell represents the mode that maximize the utility value.

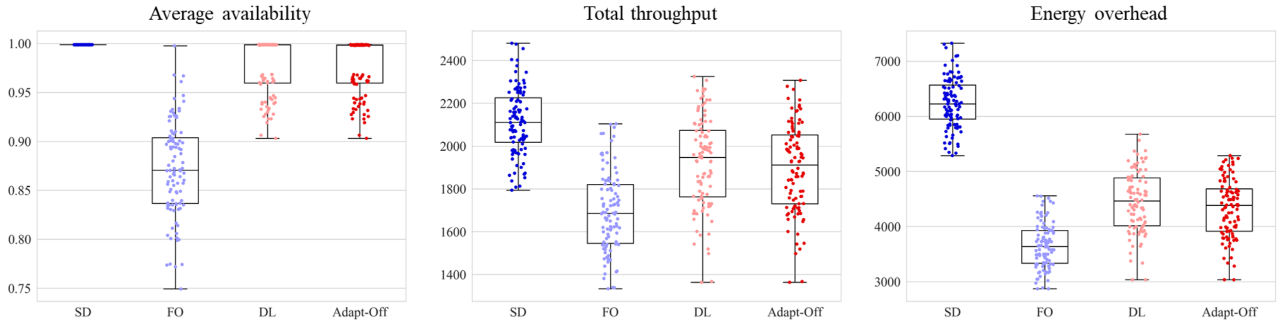


Figure 12. Performance evaluation results of Mission B with different computation modes under varying phase sojourn times in [8, 12] minutes, fog link failure rates {0.5, 60, 120}, and accessibility to a helper drone. Each point corresponds to a simulation result. Box charts show the quantiles.

2) Total throughput: The aggregated throughput can be summarized as the total number of requests processed during the mission, which can be expressed by

$$\mathcal{P}_m = \sum_{j=1}^{n} P_{m,j} \cdot t_j, \tag{7}$$

where $P_{m,j}$ represents the expected service throughput in phase j (one request per second) and computation mode m.

3) Energy overhead: Since the energy consumption is the cost factor, we compute the energy overhead due to the image processing. The total energy overhead during the mission can be given by

$$\mathcal{E}_m = \sum_{j=1}^{n} \left(E_{m,j} - E_B\right) \cdot t_j, \tag{8}$$

where $E_{m,j}$ represents the expected energy consumption [Ws] in computation mode m under the condition of phase j, and E_B is the baseline energy consumption that is the expected energy consumption in the idle state.

For mission A, the mission performance is evaluated simply by numerical analysis of the SRNs in consideration of the decision rule. The solution of SRNs yields the expected values of $A_{m,j}, P_{m,j}$ and $E_{m,j}$ for $m \in \{S, F, D\},\ j \in \{1,2,3,4\}$, and subsequently the mission performances are computed by (6), (7), and (8). The results of the computed mission performances are summarized in Table VI.

TABLE VI. EVALUATION RESULTS OF ADAPT-OFF WITH MISSION A

Computation mode	SD	FO	DL	*Adapt-Off*
Average availability	0.998685	0.838004	**0.998865**	0.998560
Total throughput	**2137.6**	1660.3	2094.8	2052.7
Total energy overhead	6305.6	**3583.1**	4855.7	4613.7

As can be seen, there is no perfect solution that can achieve the best score in all the criteria. However, we can observe that *Adapt-Off* always avoids the worst choice of the computation mode in each phase by adaptively changing the computation mode. DL is preferable in terms of availability, but it requires a collaborative drone, which may not always be available through a reliable link. For the performance, SD is the best choice with the highest cost of energy consumption. FO can reduce the energy overhead significantly but may face poor availability and performance. *Adapt-Off* maintains a good availability (>0.9985) with 26% of reduced energy overhead and less than 4% of throughput loss compared with SD. *Adapt-Off* can effectively choose the alternative computation modes (FO and DL) depending on the phase of the mission.

For Mission B, the computation modes of *Adapt-Off* are not deterministically given because the reliability of the link to the fog node and the accessibility to a nearby drone are not fixed. To analyze the impacts of such uncertainties, we conduct a simulation experiment based on the specification of Mission B as presented in Figure 10. In each iteration of the simulation, we change the phase sojourn times, the link failure rates to a fog node, and the accessibility to a helper drone. For each phase's condition, *Adapt-Off* determines a desirable computation mode according to the decision rule shown in Figure 11. We compare the performance of *Adapt-Off* with other computation modes (i.e., SD, FO, and DL). For DL mode, we assume that the computation mode is set to SD if there is no nearby drone in phase 2 and phase 4. Figure 12 shows the results of a hundred times of simulation. Each dot corresponds to one simulation result, and the underlying box charts show the quantiles. The average values are also summarized in Table VII.

TABLE VII. EVALUATION RESULTS OF ADAPT-OFF WITH MISSION B

Computation mode	SD	FO	DL	*Adapt-Off*
Average availability	**0.998686**	0.872481	0.976664	0.976428
Total throughput	**2114.3**	1692.8	1921.2	1895.8
Total energy overhead	6236.7	**3655.5**	4452.4	4329.2

The results are generally consistent with our previous observations. SD achieves the highest availability and the high throughput while it incurs the largest energy overhead. In terms of energy overhead, FO is the best choice, but the low availability and poor throughput may be unacceptable. The reason for the poor performance of FO is the high variability of the fog link failure rates in Mission B. It is not desirable to keep FO during the phases with unreliable communication links to the fog node. On the other hand, DL achieves a comparable performance to *Adapt-Off*. Both DL and *Adapt-Off* can avoid the worst choice and achieve balanced performances. The good performance of DL in this scenario is partly caused by the phases without a helper drone. In phase 2b and phase 4b, the mode is changed to SD because of inaccessibility to any helper drone. In a sense, the computation mode is adaptively changed in accordance with phase conditions similar to *Adapt-Off*. In summary, the results of experiments with two mission scenarios show that *Adapt-Off* can save the energy consumption of the drone without significantly sacrificing availability and throughput.

F. *Threats to validity*

An internal threat to the validity of the work is the granularity of the modeling we consider in our SRNs. The main objective of our study is to analyze high-level trade-off relations among the availability, performance, and power consumption in different computation modes. To achieve this goal and make the models tractable for analysis, we did not consider the details of communication channels, computation processes, and policies for offloading or load-balancing in the proposed model. The model can be improved by incorporating fine-grained models for communication channels [16][17], computation processes [18][19], and various offloading models for edge computing [34]. Despite the high expressive power of SRN, we may encounter a scalability issue of modeling and solution in such cases. An external threat of the validity is the ranges of the parameter values used in our experiments. Our findings are based on the parameter values determined by reasonable guesstimate for ordinal system configurations and use cases. However, parameter values potentially highly deviate from our guesstimates in specific systems and mission conditions. It is an important future work to validate the model with statistically estimated values from a specific real-world system.

Verification of the constructed model is also an important aspect of model-based analysis. Verification refers to the process of determining that a model implementation correctly represents the conceptual model and is solved correctly [14]. In order to avoid potential errors in model implementation, we implemented the proposed SRNs independently with different tools. While the results of SPNP are presented in our experiments, the same SRNs are also implemented and evaluated in Mercury [15] by a different team. We confirmed the correctness of the implementation by comparing the availability values as shown in TABLE VIII. Note that both the availability computations are conducted using numerical solution methods. The tools also support simulation, but we do not use that function because estimating such high availability values by simulation is another challenge.

TABLE VIII. VERIFICATION OF THE MODEL WITH MERCURY

	SD		FO		DL	
γ	SPNP	Mercury	SPNP	Mercury	SPNP	Mercury
720	0.9989363	0.9989363	0.9974227	0.9974226	0.9965814	0.9965813
1800	0.9988279	0.9988279	0.9974670	0.9974669	0.9965003	0.9964998
3600	0.9986493	0.9986493	0.9975356	0.9975355	0.9963649	0.9963638
7200	0.9983068	0.9983068	0.9976458	0.9976457	0.9961054	0.9961062

VI. RELATED WORK

The performance of image processing and video analytics on smart drone systems are becoming a hot and challenging issue. Computation offloading is a commonly adopted strategy to make a better trade-off between performance and power consumption [2][3][25]. The response time for a drone surveillance application and its energy efficiency can be improved by an adaptive task-offloading using Multipath TCP (MPTCP) [9]. For real-time video analytics, offloading tasks to the cloud or edge computing nodes can benefit through the reduction of cost [20], network bandwidths [21][22], and energy consumption on drones [2]. Instead of offloading tasks to any ground computation nodes, distributing the tasks to other nearby drones can also be considered [23][24][25]. Considering the trade-off between the performance and the energy consumption under geometrical constraints for drone trajectories, optimal task allocations for clusters and swarms of drones have been investigated [26][27]. However, none of the above studies incorporate service availability as a quality measure that must be affected by computation offloading.

Several studies address the importance of reliability and availability of drone systems [12][28][29][30][31]. Every component on a drone, such as a mainframe, power plant, navigation system, electric components, and other sensors, can fail anytime during the flight. Reliability assessment and corrective maintenance procedures for drone systems are essential [12]. Reliability block diagrams were used to evaluate the reliability of a flight control system [28] and the survivability of a drone-based post-emergency monitoring system [29]. Fault trees and Markov chains can also be used to analyze the reliability of drone systems [31]. Reliability of tasks is considered as the constraints of fog-based computation offloading into a swarm of drones [32]. In contrast to these works, we presented comprehensive SRNs that can model dynamic behaviors of system components and analyze the trade-off among the availability, performance, and energy consumption. SRN is also used in a recent study [33], while the work only considers the trade-off between performance and availability. The trade-off in terms of energy-consumption was not discussed before. Moreover, our model covers both the fog-offloading mode and the drone load-balancing mode.

VII. CONCLUSION

This paper proposed availability models for quantitatively analyzing QoS trade-offs in a drone-based image processing system. We consider both the fog offloading mode and the drone load-balancing mode, in addition to the baseline single drone processing mode. Through comprehensive numerical experiments on our proposed model, we show that each computation mode has a distinct advantage in a specific quality measure: SD achieves the highest throughput performance, FO can minimize the energy consumption by offloading, and DL can maintain the highest availability. Based on these findings, we proposed *Adapt-off* that can change the computation mode to either FO or DL adaptively considering workload intensities and communication link reliabilities. The results of our simulation experiments with two mission scenarios show that *Adapt-off* can save the energy consumption of the drone without significantly sacrificing the availability and the throughput.

There are several possible extensions of this work. In this paper, we consider the three computation modes in the basic configuration. If we have multiple drones and multiple fog nodes, the design spaces and offloading options are further enlarged. The availability model needs to be extended for incorporating more complex dependencies and dynamic behaviors among multiple system components. There is also room for improvement in the offload decision scheme of *Adapt-off*. In this paper, we only consider the utility-based decision rule, while there could be more advanced decision methods based on the expected quality values using our models. The validation of the model with a real system or other simulation tools is also an important future work.

ACKNOWLEDGMENT

This work was supported in part by the grant of University of Tsukuba Basic Research Support Program Type S. This research was partially funded by CNPq - Brazil, grant 406263/2018-3.

REFERENCES

[1] J. Chen, S. Chen, S. Luo, Q. Wang, B. Cao, and X. Li, An intelligent task offloading algorithm (iTOA) for UAV edge computing network, Digital Communications and Networks, vol. 6, no. 4, pp. 433-443, 2020.

[2] J. Yu, A. Vandanapu, C. Qu, S. Wang, and P. Calyam, Energy-aware dynamic computation offloading for video analytics in multi-UAV systems, in 2020 International Conference on Computing, Networking and Communications (ICNC), pp. 641-647, 2020.

[3] M. A. Messous, S. M. Senouci, H. Sedjelmaci, and S. Cherkaoui, A game theory based efficient computation offloading in an UAV network, IEEE Transactions on Vehicular Technology, vol. 68, no. 5, pp. 4964-4974, 2019.

[4] X. Zhu, C. Liang, Z. Yin, Z. Shao, M. Liu, and H. Chen, A new hierarchical software architecture towards safety-critical aspects of a drone system, Frontiers of Information Technology & Electronic Engineering, vol. 20, no. 3, pp. 353-362, 2019.

[5] R. K. Iyer and I. Lee, Measurement-based analysis of software reliability, Handbook of Software Reliability Engineering, pp. 303-358, 1996.

[6] C. Aker and S. Kalkan, Using deep networks for drone detection, in 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1-6, 2017.

[7] B. Mishra, D. Garg, P. Narang, and V. Mishra, Drone-surveillance for search and rescue in natural disaster, Computer Communications, vol. 156, pp. 1-10, 2020.

[8] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, Fog computing and its role in the internet of things, in Proceedings of the first edition of the MCC workshop on Mobile cloud computing, pp. 13-16, 2012.

[9] W. S. Jung, J. Yim, and Y. B. Ko, Adaptive offloading with MPTCP for unmanned aerial vehicle surveillance system, Annals of Telecommunications, vol. 73, no. 9, pp. 613-626, 2018.

[10] G. Ciardo, J. K. Muppala, K. S. Trivedi et al., Spnp: Stochastic petri net package. In Int'l Workshop on Petri Nets and Performance Models, vol. 89, 1989, pp. 142-151.

[11] K. S. Trivedi, Probability and statistics with reliability, queuing, and computer science applications, John Wiley, New York, 2001.

[12] E. Petritoli, F. Leccese, and L. Ciani, Reliability and maintenance analysis of unmanned aerial vehicles, Sensors, vol. 18, no. 9, pp. 3171, 2018.

[13] C. E. Lin and P. C. Shao, Failure analysis for an unmanned aerial vehicle using safe path planning, Journal of Aerospace Information Systems, vol. 17, no. 7, pp. 358-369, 2020.

[14] K. S. Trivedi and A. Bobbio, Reliability and availability engineering : modeling, analysis and applications, Cambridge University Press, 2017.

[15] B. Silva, et al., Mercury: An integrated environment for performance and dependability evaluation of general systems, In Proceedings of Industrial Track at 45th Dependable Systems and Networks Conference, 2015.

[16] A. A. Khuwaja, Y. Chen, N. Zhao, M.-S. Alouini, and P. Dobbins, A survey of channel modeling for UAV communications, IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 2804-2821, 2018.

[17] M. M. Azari, F. Rosas, A. Chiumento, A. Ligata, and S. Pollin, Uplink performance analysis of a drone cell in a random field of ground interferers, in IEEE Wireless Communications and Networking Conference (WCNC), pp. 1-6, 2018.

[18] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. Leung, An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing, IEEE/ACM Transactions on Networking, vol. 26, no. 6, pp. 2651-2664, 2018.

[19] M. Chen and Y. Hao, Task offloading for mobile edge computing in software defined ultra-dense network, IEEE Journal on Selected Areas in Communications, vol. 36, no. 3, pp. 587-597, 2018.

[20] D. Chemodanov, C. Qu, O. Opeoluwa, S. Wang, and P. Calyam, Policy-based function-centric computation offloading for real-time drone video analytics, in IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), pp. 1-6, 2019.

[21] X. Wang, A. Chowdhery, and M. Chiang, Skyeyes: adaptive video streaming from UAVs, in Proceedings of the 3rd Workshop on Hot Topics in Wireless, pp. 2-6, 2016.

[22] J. Wang, Z. Feng, Z. Chen, S. George, M. Bala, P. Pillai, S.-W. Yang, and M. Satyanarayanan, Bandwidth-efficient live video analytics for drones via edge computing, in IEEE/ACM Symposium on Edge Computing (SEC), pp. 159-173. 2018.

[23] R. Valentino, W. S. Jung, and Y. B. Ko, A design and simulation of the opportunistic computation offloading with learning-based prediction for unmanned aerial vehicle (UAV) clustering networks, Sensors, vol. 18, no. 11, p. 3751, 2018.

[24] C. Grasso and G. Schembra, A fleet of MEC UAVs to extend a 5G network slice for video monitoring with low-latency constraints, Journal of Sensor and Actuator Networks, vol. 8, no. 1, p. 3, 2019.

[25] A. A. A. Ateya, A. Muthanna, R. Kirichek, M. Hammoudeh, and A. Koucheryavy, Energy-and latency-aware hybrid offloading algorithm for UAVs, IEEE Access, vol. 7, pp. 37587-37600, 2019.

[26] J. Yao and N. Ansari, Online task allocation and flying control in fog-aided internet of drones, IEEE Transactions on Vehicular Technology, vol. 69, no. 5, pp. 5562-5569, 2020.

[27] X. Hou, Z. Ren, J. Wang, S. Zheng, W. Cheng, and H. Zhang, Distributed fog computing for latency and reliability guaranteed swarm of drones, IEEE Access, vol. 8, pp. 7117-7130, 2020.

[28] Y. Pashchuk, Y. Salnyk, and S. Volochiy, Reliability synthesis for UAV flight control system, in International Conference on ICT in Education, Research, and Industrial Applications, pp. 569-582, 2017.

[29] V. Kharchenko, A. Sachenko, V. Kochan, and H. Fesenko, Reliability and survivability models of integrated drone-based systems for post emergency monitoring of NPPs, in International Conference on Information and Digital Technologies (IDT). pp. 127-132, 2016.

[30] L. Gupta, R. Jain, and G. Vaszkun, Survey of important issues in UAV communication networks, IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1123-1152, 2015.

[31] R. Abdallah, Reliability approaches in networked systems: Application on unmanned aerial vehicles, Ph.D. dissertation, Universite Bourgogne, Franche-Comte, 2019.

[32] X. Hou, Z. Ren, W. Cheng, C. Chen, and H. Zhang, Fog based computation offloading for swarm of drones, in ICC 2019-2019 IEEE International Conference on Communications (ICC), pp. 1-7, 2019.

[33] F. Machida and E. Andrade, PA-Offload: performability-aware adaptive fog offloading for drone image processing, in 5th IEEE International Conference on Fog and Edge Computing (ICFEC), pp. 66-73, 2021.

[34] H. Lin, S. Zeadally, Z. Chen, H. Labiod, and L. Wang, A survey on computation offloading modeling for edge computing, Journal of Network and Computer Applications, Vol. 169, pp. 102781, 2020.

[35] Q. Yang and S. J. Yoo, Optimal uav path planning: Sensing data acquisition over iot sensor networks using multi-objective bio-inspired algorithms, IEEE Access, vol. 6, pp. 13671-13684, 2018.

[36] O. Alsaryrah, I. Mashal, and T. Y. Chung, Bi-objective optimization for energy aware internet of things service composition, IEEE Access, vol. 6, pp. 26809-26819, 2018.