Analysis of an Edge Computing Enabled Real-Time Object Detection Platform Using Network Calculus

Boyang Zhou
Dept. of Electrical and Computer Engg.
Lehigh University
Bethlehem, PA 18015
boz319@lehigh.edu

Ryan Cheng
Choate Rosemary Hall
333 Christian St
Wallingford, CT 06492
rcheng24@choate.edu

Unmesh Khanolkar and Liang Cheng
Dept. of Electrical Engg. and Computer Science
University of Toledo
Toledo, OH 43606
unmesh.khanolkar@rockets.utoledo.edu
liang.cheng@utoledo.edu

***Abstract*—Numerous Drone as a Service (DaaS) applications, such as surveillance, search and rescue, and infrastructure inspection, may employ real-time object detection to achieve computer vision-based autonomous functions. However, running object detection algorithms, e.g., YOLO, locally on a drone requires extensive computational power, which is expensive in terms of cost and energy consumption. Conversely, edge computing facilitates the implementation of an affordable and efficient platform where drones compress and transmit images to an edge server for real-time object detection. Nevertheless, DaaS designers applying edge computing for real-time object detection must be cognizant of the network design of an Edge Computing Enabled Real-Time Object Detection (ECOD) platform to consistently realize object detection in real-time. In our research, we propose utilizing network calculus to analyze the delay performance of an ECOD platform, which provides a principled approach for the platform design. A testbed was implemented to evaluate the accuracy of this approach, which was then used to analyze DaaS design scenarios with different numbers of drones and various network capacities.**

***Index Terms*—Drone as a Service, Edge computing, Delay analysis**

I. Introduction

Drone as a Service (DaaS) is a growing industry to provide customers with the hardware, software, and human resources necessary to accomplish certain tasks using unmanned aerial vehicles (UAVs), i.e., drones [1]. Drones can provide functions such as asset inspection, crop monitoring, and live streaming events, which can be useful in the energy, agriculture, and entertainment industries [2]. Many of these tasks might require real-time object detection for autonomous functions [20].

Object detection algorithms such as YOLO (You Only Look Once), R-CNN (Region-based Convolutional Neural Networks), and SSD (Single Shot Detector) require GPUs to accelerate the inference process. However, the price of microcomputers with GPUs, e.g., NVIDIA Jetson products, is high. Moreover, these microcomputers weigh more and consume more power than those without GPUs. It is better to avoid using large and powerful microcomputers as onboard computers for drones because the battery capacity is limited.

On the other hand, edge computing is desirable as processing data at the edge of a network rather than uploading it to the cloud reduces latency, cost, and energy consumption [3]. Thus, edge computing could be a suitable candidate for real-time object detection on drones. Edge-assisted object detection is not a new thing. A lot of work has been done to realize real-time object detection using edge servers [4], [5], [22]–[24]. However, all previous work does not pay attention to the network analysis in their design. Considering specific application scenarios using a swarm of drones, DaaS designers applying edge computing must be cognizant of the network design in terms of the traffic profile, the network device capacity, and the delay requirements, to consistently realize object detection in real-time.

In our research, we propose an Edge Computing Enabled Real-Time Object Detection (ECOD) platform for DaaS, as shown in Figure 1. We apply network calculus to analyze the delay performance of the communication network in the ECOD platform, which might be 4G, 5G, or WiFi. Network calculus is a prevalent framework that provides performance analysis of networks to model the delay performance of networks. Based on the analysis result, we provide a principle guideline to help designers design DaaS platforms. A testbed has also been implemented to evaluate the accuracy of this approach.

The major contributions of this paper include: (i) We provide a network-calculus-based approach to analyzing the delay performance of an ECOD for DaaS, which furthermore offers a principle guideline to design the ECOD platform based on 4G, 5G, and WiFi networks, ii) We propose a measurement-based method to identify the parameters of arrival curves and stochastic service curves needed by network calculus, and (iii) The network-calculus-based approach has been validated by experiments in a testbed and then used to analyze DaaS design scenarios with different requirements.

The rest of the paper is organized as follows. Section II discusses background knowledge of the ECOD platform and network calculus. Section III describes the research problem of the paper and our solution. Section IV presents the network delay model of the ECOD platform. Section V proposes a measurement-based approach to identify the essential parameters for network calculus. Section VI verifies our model by experiments. Section VII apply our model for the design of the ECOD platform. Finally, Section VIII concludes this paper.

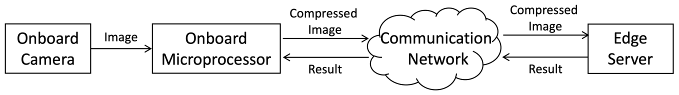


Fig. 1. An Edge Computing Enabled Real-Time Object Detection Platform

II. Background Knowledge

A. *Edge Computing Enabled Real-Time Object Detection*

In our research, we designed an Edge Computing Enabled Real-Time Object Detection (ECOD) platform for Drone as a Service (DaaS) using a F450 Quadcopter running ArduCopter firmware, a Raspberry Pi serving as the onboard microcomputer, a Raspberry Pi Camera Module, and a Dell Precision computer serving as the edge server.

Figure 1 shows how the ECOD platform for DaaS works. The Raspberry Pi Camera Module captures an image, to which the Raspberry Pi applies JPEG compression [8], reducing transmission size by 90%. Next, the Raspberry Pi transmits the compressed image to the Dell Precision computer using the TCP protocol to guarantee data transmission reliability through a wireless communication network. Then the Dell Precision computer decompresses the compressed image and runs the object detection algorithm YOLOv5 [9] on it. After completing object detection, the result is sent to the Raspberry Pi through the network. The Raspberry Pi decides what to do next, given the received information.

In our research, we focus on the delay from the time when an image is captured by the drone-onboard camera to the time when the drone gets the object detection result, which is called $feedback\ time$ in this paper. Figure 2 shows the feedback time in two scenarios, namely using the Raspberry Pi only versus the ECOD platform with a TL-WR840N wireless router. In this example, 100 images with a 1280*720 resolution have been used. The x-axis shows the image index, and the y-axis shows the feedback time. The average feedback time using Raspberry Pi for object detection is 3.723 s, which is not suitable for real-time functions or applications of drones, e.g., obstacle avoidance. Using the ECOD platform is much faster than using the Raspberry Pi only, achieving an average feedback time of 0.258 s. Figure 2 also shows the processing time needed by the edge server to run YOLO on an image for object detection without receiving nor transmitting any data through a network, which has a mean value of 34 ms.

B. *Network Calculus*

Network calculus is a mathematical framework for analyses of delay bounds and backlog bounds experienced by traffic flows passing through networked nodes (i.e., wired switches, wireless routers, or base stations). It is widely used in the delay analysis of wired and wireless networks [6], [14]–[16]. As network calculus uses per-flow per-node modeling, the following information is needed to apply NC for analysis:

1) Network topology and traffic flow paths

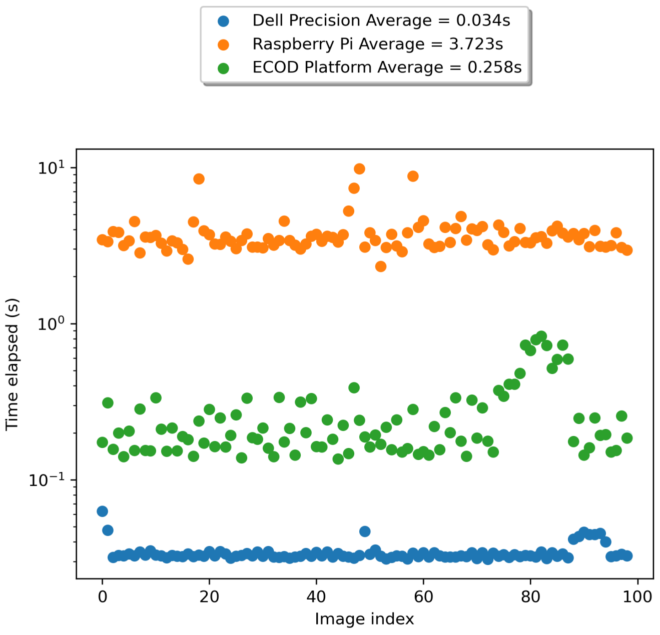


Fig. 2. The time elapsed for object detection in three scenarios

2) Arrival curves of all flows where a flow's arrival curve describes the arrival process of the flow (details in subsection II-B1)
3) Service curves of networked nodes where a service curve describes the minimum service provided by a network device, e.g., an access point, a switch, or a router (details in subsection II-B2)

Knowing the arrival curve of a flow $\alpha(t)$ and the service curve of a networked node $\beta(t)$, we can derive the maximum delay bound and backlog bound for each flow experienced at each node using network calculus as follows. How to obtain $\alpha(t)$ and $\beta(t)$ is explained in subsections II-B1 and II-B2.

Equation 1 shows how to calculate the maximum delay bound where $\oslash$ represents a deconvolution operator.

$$delay : \forall t \geq 0 : D(t) \leq \inf\{d \geq 0 | (\alpha \oslash \beta)(-d)\} \quad (1)$$

The deconvolution $\oslash$ of functions f_1 and f_2 is defined as

$$deconvolution\ (f_1 \oslash f_2)(d) = \sup_{u \geq 0}\{f_1(d+u) - f_2(u)\}. \quad (2)$$

The backlog bound of the node can be calculated using

$$backlog : \forall t \geq 0 : B(t) \leq (\alpha \oslash \beta) \quad (3)$$

1) Arrival process and arrival curve: Obtaining the arrival curve of a flow uses the concept of arrival process. When a flow passes through a networked node, there is an incoming arrival process $F_{in}(t)$ of the flow representing the cumulative amount of data arrived at the node between time 0 to t. $F_{in}(t)$ is a non-negative and non-decreasing function with $F_{in}(0) = 0$. There is also an outgoing process $F_{out}(t)$ of the flow after passing through the node, which is the incoming process of the flow arriving at the next node of the flow path.

Definition 1. Given an arrival process $F_{in}(t)$, a real-valued, non-negative, nondecreasing function $\alpha(t)$ defined for any $t \geq 0$ is an arrival curve of $F_{in}(t)$ if and only if

$$\forall t \geq s \geq 0 : F_{in}(t) - F_{in}(s) \leq \alpha(t - s) \tag{4}$$

In other words, the arrival curve $\alpha(t)$ is an upper bound of its arrival process during any backlogged period $[t, s]$.

2) Service curve: The service curve of a networked node can be defined using a convolution operator. The following equation shows how to calculate the convolution of f_1 and f_2.

$$convolution\ (f_1 \otimes f_2)(d) = \inf_{0 \leq s \leq d} f_1(d - s) + f_2(s) \tag{5}$$

Definition 2. When a flow passes through a networked node, there is an incoming arrival process $F_{in}(t)$ and an outgoing process $F_{out}(t)$. A real-valued, non-negative, non-decreasing function $\beta(t)$ is the service curve of the node if and only if

$$F_{out}(t) \geq F_{in}(t) \otimes \beta,\ where\ \beta(0) = 0 \tag{6}$$

Theorem 1. *Considering the flow passing through a series of networked nodes $S_i, i = 1, ..., n$, where Node S_i has a service curve β_i, we can treat all nodes as one virtual node with a service curve $\beta = \beta_1 \otimes \beta_2 \otimes ...\beta_n$.*

In some network calculus methods, such as Separate Flow Analysis (SFA) used in [13], leftover service curve calculations are necessary when there is more than one flow passing through the same networked node. We provide the definition of leftover service curves as follows.

Definition 3. Suppose two flows f_1 and f_2 pass through a lossless node with arbitrary multiplexing. f_1 and f_2 have arrival curves α_1 and α_2, respectively. If the node has a service curve $\beta(t)$, then the leftover service curve $\beta^1(t)$ for f_1 can be calculated as follows where $[x]^+$ equals to $max(x, 0)$:

$$\beta^1(t) = [\beta(t) - \alpha_2(t)]^+ \tag{7}$$

III. Research Problem and Solution

Edge-assisted real-time object detection has been well investigated in recent years [4], [5], [22]–[24]. Although some work takes the network capacity and the bandwidth usage of the network into consideration [4], [22], none of them provides a formal method to theoretically model the network performance in their design. Moreover, their designs are mostly tested in the case that an edge server is only used to do the object detection for one device. However, in the ECOD platform, it is common that each edge server might be responsible for the tasks coming from more than one drone. Thus, the main research problem becomes how to design the ECOD platform in terms of the traffic profile, the network device capacity, and the delay requirements.

In order to solve this problem, we apply network calculus to analyze the delay performance of the 4G, 5G, or WiFi network in the ECOD platform, and thus use the analysis result to provide a guideline principle for the design of the ECOD platform. In this paper, our solution provides i) network calculus modeling of the ECOD platform, ii) parameters identification for arrival curves and service curves using a measurement-based approach, iii) verify the network calculus result using the testbed, and iv) ECOD platform design based on the network calculus result.

IV. Modelling Network Delay

Since we target to provide a principle guideline for the design of the ECOD platform, modeling the network delay of the platform using network calculus is necessary.

As discussed in Subsection II-A, we focus on the feedback time, which consists of several components, including the processing time by the Raspberry Pi for image compression, the network delay for wireless data communication, and the processing time by the edge server for image decompression and object detection. As the network delay is the most significant part of the feedback time and the rest of the feedback components are constant, assuming that the edge server has enough computational capacity, we focus on modeling network delay since the other parts of the feedback time are mostly constants under our assumption.

In our research, we apply network calculus to model and analyze the network delay in an ECOD platform for DaaS. It is normal that the network topology, along with its traffic pattern, is a single-hop network with bidirectional flows between drones and the edge server in the ECOD platform.

In this paper, we use the leaky-bucket arrival curve and the rate-latency service curve, which comprise two linear piecewise components. The leaky-bucket arrival curve can be defined by using rate ρ and burst b as expressed by Equation 8. The rate-latency service curve can be described using rate r and latency T, which is defined in Equation 9.

$$\alpha_{\rho,b}(t) = \begin{cases} \rho t + b & t > 0 \\ 0 & Otherwise \end{cases} \tag{8}$$

$$\beta_{r,T}(t) = [r(t - T)]^+ \tag{9}$$

Knowing the arrival curves and service curves, we can provide a closed-form mathematical expression for the delay bounds derived using network calculus in the ECOD platform. If there is more than one drone in the system, we calculate the leftover service curves for each drone using Equation 7. Note that wireless networks are usually half-duplex, meaning that bidirectional flows need to compete for the same service while they do not in full-duplex networks.

For each drone i, there are flow i_1 sourcing from drone i to the edge server and flow i_2 sourcing from the edge server to drone i, respectively. These two flows have arrival curves $\alpha_{i_j}(t) = \rho_{i_j} t + b_{i_j}$ where j is 1 or 2. Then the leftover service curve for flow i_j can be calculated as follows:

$$\beta_{i_j} = [\beta(t) - \sum_{m \neq i} (\alpha_{m_1} + \alpha_{m_2}) - \alpha_{i_{3-j}}]^+ \tag{10}$$

After calculating the leftover service curve of flow i_j, we can use Equation 1 to derive the worst-case delay bound. The delay bound D_{i_j} for flow i_j can be calculated as follows:

$$D_{i_j} = \frac{r * T + \sum b_{m_n}}{r - \sum_{m \neq i \& n \neq j} \rho_{m_n}} \tag{11}$$

V. Model Parameter Identification

In Section IV, we provide a closed-form expression for calculating the network delay in the ECOD platform. In order to apply Equation 11, the arrival curves and the service curve must be known. So as to make the theoretical results conform to reality, we propose a measurement-based approach to identify the parameters of arrival curves and the service curve.

A. Arrival curve parameters

There are two parameters needed to be identified, which are rate ρ and burst b, in the arrival curve. It is straightforward to calculate these two parameters when dealing with periodic UDP flows using Equation 12 [10], where σ is the frame size and p is the period. However, in this ECOD platform, TCP is used for reliable data transmission, which makes the arrival curve parameter identification hard to evaluate due to TCP's retransmission mechanism.

$$\alpha_{\rho,b}(t) = \frac{\sigma}{p} t + \sigma \tag{12}$$

We employ a measurement-based approach to identify the arrival curve parameters. Each image is treated as one virtual packet, although a (compressed) image needs to be split into a group of packets for TCP transmission. This treatment conforms with the model because the edge server will not start processing a (compressed) image until all packets associated with it are fully received. The Raspberry Pi runs a packet sniffer [7] to record $F_{in}(t)$. The average transmission rate of the packet flow is used as rate ρ of the arrival curve. Then $F_{in}(t)$ and ρ are used to find b that meets Definition 1 using binary search.

1) Service curve parameters: In order to measure the service curve of a wireless router, we need to make several assumptions. First, the drone only moves within a specific area away from the access point (a base station or a router). This is due to the channel fading influenced by the distance between the drone and the access point [18]. If the distance is too large, the signal attenuation will be significant and lead to limited network performance [22]. Second, without the loss of generality, the bandwidth measured at the farthest point from the access point can reflect the minimum service provided by the access point. Meanwhile, instead of using a deterministic service curve, we use a stochastic service curve to reflect the service provided by the wireless network.

Figure 3 shows the distribution of the bandwidth measurement results of a TL-WR940N router at the farthest point within a specific range (about 10 m in our lab) 1100 times. Note that the bandwidth in this paper refers to the data rate of the network. The measured bandwidth fluctuates significantly

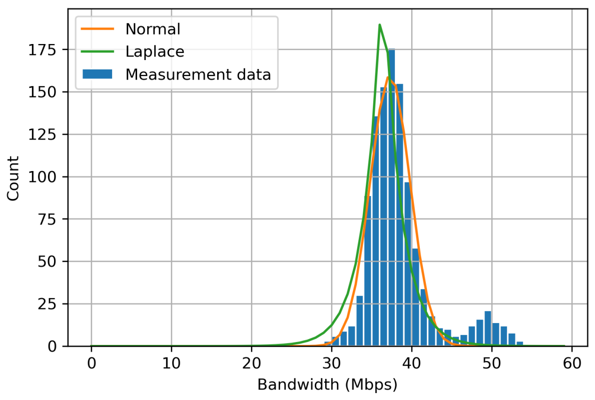


Fig. 3. Network bandwidth distribution measurement of the wireless router.

due to the channel fading of the wireless network [17] ranging from 16 $Mbps$ to 55 $Mbps$. In order to evaluate the network bandwidth under the impact of channel fading, we use a stochastic service curve to describe the service provided by the router. According to [19], the rate of a stochastic service curve can be described using the rate r with its Cumulative Distribution Function (CDF). Based on Figure 3, we consider using the normal distribution and the Laplace distribution to model the Probability Density Function (PDF) of the router bandwidth. Thus, we calculate the overlapping area of the histogram and the PDFs of the two distributions with different parameters to determine the better PDF. The parameters of the normal distribution are the mean and the standard deviation, while those of the Laplace distribution are the location and the scale. The maximum overlapping area between the histogram and the normal distribution is 982.6 out of 1100, and the value is 961.6 using the Laplace distribution. The two distributions with the maximum overlapping areas are also shown in Figure 3. Therefore, we decide to use the normal distribution with the mean being 37.3 and the standard deviation being 2.5 to model the PDF of the rate r of the router because the overlapping area between the histogram and this normal distribution is the maximum. Thus, $P(R = r) = \frac{1}{2.5\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-37.3}{2.5})^2}$. Then, the CDF of the rate is also known after determining the PDF. We can search for a T to meet the service curve meets Definition 2. In order to search for T, we use two machines with PTP (Precise Time Protocol) time-synchronization to record $F_{in}(t)$ and $F_{out}(t)$. Then, after fixing the rate r, we use binary search to find the T corresponding to r with measured $F_{in}(t)$ and $F_{out}(t)$.

VI. Model Verification

In this section, we verify the effectiveness of the network-calculus-based model for the delay performance of the ECOD platform using experimental results of the feedback time measured in a single-hop single-drone scenario. Since network calculus is used to analyze the network delay, we subtract the processing time components of the edge server and the onboard computer from the feedback time to get the network delay experienced by the traffic.

In order to verify the network calculus result, we need to regulate the traffic from the Raspberry Pi to the edge server. We send a 1280*720 image from the Raspberry Pi to the edge server every 0.1 s 100 times. After the edge server receives each image and uses YOLO to process the image, it will send the detection result back to the Raspberry Pi. Since the network contains only one router, the topology and the traffic pattern are fixed. All flows will pass through the same wireless router to their destinations.

In order to create different scenarios for the single-router single-drone network, we control the largest bandwidth of the wireless network from 4 Mbps to 16 Mbps. However, the router itself does not have an accurate bandwidth restriction function. Thus, instead of restricting the bandwidth of network devices, we restrict the incoming and outgoing transmission rates of the NIC (Network Interface Card). This can be done using the $wondershaper$ on Raspbian [11]. In this way, the NIC of the Raspberry Pi can be abstracted as a node whose bandwidth is the restricted value. According to Theorem 1, the NIC and the router can be treated as one virtual node whose service curve is the convolution of their service curves, meaning that the rate of the virtual node is the minimum rate of the NIC and the network device. We have mentioned in Section V-A1 that the router we use has a stochastic service curve whose rate conforms to a normal distribution. A specific rate should be chosen for the service curve to provide a probabilistic delay bound. We decide to use 16 $Mbps$, which is the smallest bandwidth in the bandwidth measurement of the router, as the rate of the service curve. According to the CDF of the normal distribution, the wireless router has about 92.0% probability of providing a higher rate than 16 $Mbps$. The latency T of the service curve is 0.44 s under this scenario. Thus, the service curve of the wireless router will be $\beta(t) = 16(t - 0.44)$. The probabilistic delay bounds derived using this service curve should be able to model the delay in approximately 92.0% cases.

In order to analyze the delay performance of network calculus, we use SFA (Separate Flow Analysis) to derive the worst-case delay bounds [13]. Table I shows the rates and bursts of outgoing and incoming arrival curves under different bandwidth restrictions in the wireless network. The incoming and outgoing arrival curves are all captured on the onboard computer. The outgoing flow is the flow from the onboard computer to the edge server, and the incoming flow is the flow from the edge server to the onboard computer. Since we use leaky bucket arrival curves, we only need to know the rate and the burst to determine an arrival curve.

The outgoing rate and outgoing burst in Table I defines the arrival curves of flows from the Raspberry Pi to the edge server. Although we restrict the bandwidth of the NIC to a certain value, it is not guaranteed that this certain value is the exact bandwidth that the NIC reaches. However, the bandwidth of the NIC is described by the rate of the outgoing flow. Thus, the outgoing rate is the real bandwidth of the NIC.

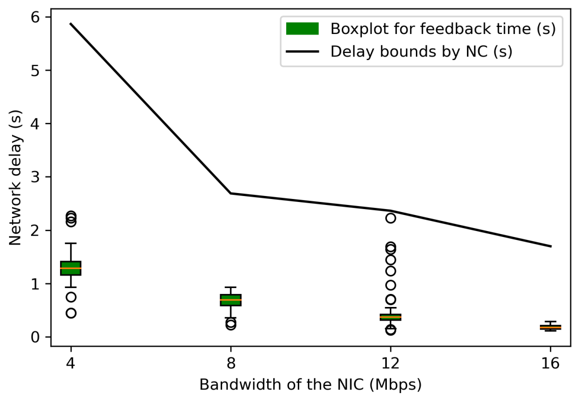
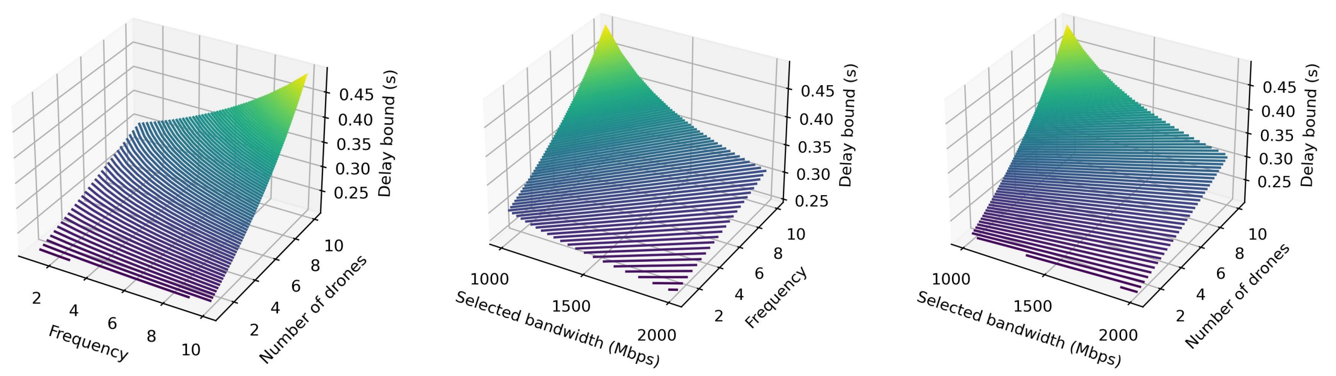


Fig. 4. Boxplots for network delays and delay bounds derived from network calculus under different bandwidths in the wireless network

When calculating the worst-case delay bound for the traffic from the edge server to the Raspberry Pi, its arrival curve should be known. The arrival curve of the traffic at its destination, which is the incoming arrival curve at the Raspberry Pi, can be used to model the arrival curve at its source, which is the edge server. This works because the rate of arrival curves in the network will not change, and the bursts of the arrival curves are non-decreasing if the network is globally stable [12]. Thus, the arrival curve of a flow at any node on its path can bound its arrival process at the source. In Table I, the rates and bursts of the incoming arrival curves are much lower than those of outgoing arrival curves. This is because the incoming flow mostly contains acknowledgments and detection results, which consume fewer network resources than transmitting images. Knowing all the necessary information for network calculus, the worst-case delay bounds can be calculated for flows under different bandwidths. We can use Equation 11 to derive the worst-case delay bounds for different scenarios. Figure 4 shows the boxplots for network delays and the theoretical results of delay bounds under different bandwidths in the wired network. The x-axis shows the bandwidth of the NIC, and the y-axis represents the network delay. The delay bound shown in the figure is derived using the service curve $\beta(t) = 16(r - 0.44)$. Since we use 16 $Mbps$ as the rate of the service curve and the probability of the route providing a larger rate is about 92.0%, the frames in the network should encounter a delay less than the derived delay bound in the figure in about 92.0% cases [17]. Note that the network delay equals the total feedback time minus the processing time in both the edge server and the onboard computer. All measurements in the experiment are done using local timers. As we can see in Figure 4, delay bounds derived by network calculus can bound the network delays in all scenarios in the experiment.

In this section, we have done experiments in the single-drone scenario to verify the effectiveness of delay bounds derived by network calculus. From the result, we can conclude



(a) Selected bandwidth = 1000 Mbps

(b) Number of drones= 10

(c) Frequency= 10

Fig. 5. How the number of drones, the frequency of taking photos, and the selected bandwidth influence the delay bounds

TABLE I
RATES AND BURSTS OF THE OUTGOING ARRIVAL CURVE UNDER DIFFERENT BANDWIDTH RESTRICTIONS USING THE WIRELESS NETWORK

Bandwidth restriction	4 Mbps	8 Mbps	12 Mbps	16 Mbps
Outgoing rate (Mbps)	4.12	9.36	10.40	13.60
Outgoing burst (Mb)	3.52	4.54	4.13	2.70
Incoming rate (Mbps)	0.04	0.06	0.06	0.07
Incoming burst (Mb)	0.03	0.03	0.03	0.02

network calculus can provide probabilistic delay bounds based on the CDF of the stochastic service curve that can model the delay performance in the ECOD using wireless networks.

VII. Model Application for the Design of the ECOD Platform

In this section, we apply the model described in Section IV to study the relationships between the traffic profile, the capacity of the network device, and the delay bound of a hypothetical ECOD platform that is designed for DaaS.

The traffic profile of the network depends on two parameters, which are the arrival curve for each drone and the number of drones in the platform. Thus, the design space of the ECOD platform contains four components i) traffic for each drone, ii) the number of drones, iii) the service curve provided by the access point, and iv) delay requirements by applications.

We assume that drones transmit images with a 1280*720 resolution. Then, the maximum burst is taken as the frame size for incoming and outgoing flows in Table I. The arrival curve can be calculated using Equation 12 after figuring out period p. The arrival curves of the bidirectional flows of drone i are $\alpha_{i_1} = 4.34/p + 4.34$ and $\alpha_{i_2} = 0.03/p + 0.03$. p decides the frequency of taking a photo by each drone. We assume that all drones have the same p for the same application scenario.

For the service curve, we assume that the all service curves have a fixed 0.1 s latency in this scenario. Then, a rate/bandwidth should be selected from the stochastic service curve based on the model requirement (percentage of scenarios required to be modeled by the network calculus result). Note that this percentage is about 92% in Section VI.

Figure 5 shows how the selected bandwidth, the number of drones, and the frequency of taking photos influence the delay bound in the ECOD platform. Note that we only choose one selected bandwidth for each router based on the requirements and the CDF of the bandwidth. Figure 5 (a) shows how the frequency/($\frac{1}{p}$ and the number of drones influence the delay bound when the selected bandwidth of the access point is fixed. Figure 5 (b) depicts how the selected bandwidth and the frequency influence delay bound when the number of drones is fixed. Figure 5 (c) illustrates the influence of the selected bandwidth and the frequency on the delay bound when the number of drone is fixed. Based on the contour figure, we can conclude that when the frequency and the number of drones increase, the delay bound also increases. When the selected bandwidth increases, the delay bound decreases.

Figure 5 provides a principle guideline for the design of the ECOD platform. If we know any three components of the four in the design space, we can easily calculate the fourth one. For example, if we know the bandwidth is 1000 $Mbps$, the frequency is 2, and the delay requirement is 0.3 s, it is easy to find that the maximum number of drones can be accommodated by the ECOD platform meeting all the requirements is 8. This number can be directly read from Figure 5 (a). If we know two components, we can find a set of combinations of the other components to meet the requirement using our approach. For example, if we know the selected bandwidth is 1000 $Mbps$, and the delay requirements is 0.3 s, any combination of the number of drones and the frequency on the contour below the blue plane, which shows the plane with delay bound equaling 0.3 s, in Figure 6 meets the delay and bandwidth requirements. If there is only one component having requirements, the design will be more flexible. We can draw the figure with the known component like Figure 5, and

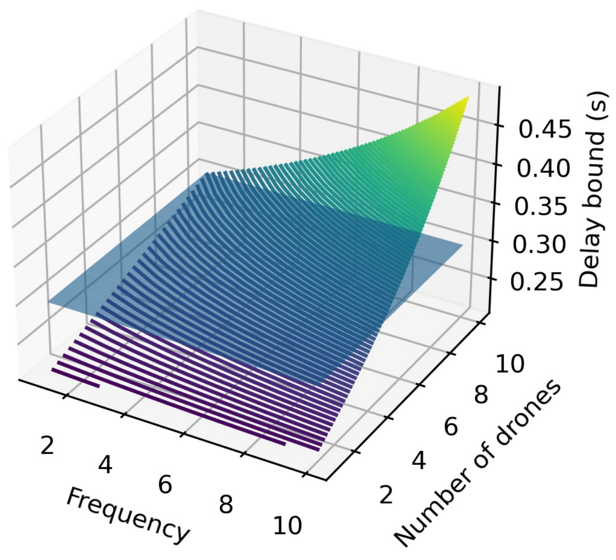


Fig. 6. Candidates of combinations of the number of drones and the frequency when the bandwidth is 1000 $Mbps$ and the delay requirement is 0.3 s

any point on the contour can be a feasible design.

In this section, we have provided a principled approach to the ECOD platform design based on wireless networks (4G, 5G, and WiFi). There are totally four components in the design space, namely the delay requirements, the number of drones, the frequency, and the selected bandwidth of the access point. We propose a method of designing the ECOD platform based on the knowledge of any component in the design space.

VIII. Conclusion

In this paper, we have modeled and analyzed the delay performance of an ECOD platform for DaaS using network calculus. Moreover, we propose a measurement-based approach to identify the parameters for network calculus based on the real testbed. The effectiveness of the network-calculus-based model and analysis is verified using our testbed. Network calculus can also provide a principled approach to designing an ECOD platform based on its requirements. Knowing the requirements of any design component, we can propose a set of combinations of the other design components in the ECOD platform to meet all the requirements based on network calculus.

Acknowledgment

This work is supported by NSF Award No. 1646458 and No. 2146968. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the sponsors of the research.

References

[1] C. Bernstein. “What is drone services (UAV services)?” Tech Target. https://www.techtarget.com/iotagenda/definition/drone-services-UAV-services (accessed Jun. 23, 2022).

[2] ”Applications and Uses for Multirotor Drones.” Rise Above. https://riseabove.com.au/pages/uav-applications-and-uses (accessed Jun. 23, 2022).

[3] W. Yu *et al.*, ”A Survey on the Edge Computing for the Internet of Things,” in *IEEE Access*, vol. 6, pp. 6900-6919, 2018, doi: 10.1109/ACCESS.2017.2778504.

[4] Liu, L., Li, H. & Gruteser, M. Edge Assisted Real-Time Object Detection for Mobile Augmented Reality. *The 25th Annual International Conference On Mobile Computing And Networking*. (2019), https://doi.org/10.1145/3300061.3300116

[5] J. Wang *et al.*, ”Edge-Based Live Video Analytics for Drones,” in *IEEE Internet Computing*, vol. 23, no. 4, pp. 27-34, 1 July-Aug. 2019, doi: 10.1109/MIC.2019.2909713.

[6] B. Zhou, I. Howenstine, S. Limprapaipong and L. Cheng, ”A Survey on Network Calculus Tools for Network Infrastructure in Real-Time Systems,” in *IEEE Access*, vol. 8, pp. 223588-223605, 2020, doi: 10.1109/ACCESS.2020.3043600.

[7] *Python 3 Network Packet Sniffer*. (2021). Accessed: Jun. 23, 2022. [Online]. Available: https://github.com/EONRaider/Packet-Sniffer

[8] G. K. Wallace, ”The JPEG still picture compression standard,” in *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii-xxxiv, Feb. 1992, doi: 10.1109/30.125072.

[9] *YOLOv5*. (2022). Ultralytics. Accessed: Jun. 23, 2022. [Online]. Available: https://github.com/ultralytics/yolov5

[10] B. Zhou and L. Cheng, ”A Reality-Conforming Approach for QoS Performance Analysis of AFDX in Cyber-Physical Avionics Systems,” *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, 2021, pp. 1-6, doi: 10.1109/IWQOS52092.2021.9521335.

[11] *Wonder Shaper*. (2021). [Online]. Available: https://github.com/magnific0/wondershaper

[12] Bouillard, A. Stability and performance guarantees in networks with cyclic dependencies. *ArXiv*. abs/1810.02623 (2018)

[13] Le Boudec, J. & Thiran, P. Network Calculus: A Theory of Deterministic Queuing Systems for the Internet. (2004, 6)

[14] Yang, Huan, and Liang Cheng. ”Bounding network-induced delays of wireless prp infrastructure for industrial control systems.” ICC 2019-2019 IEEE International Conference on Communications (ICC). IEEE, 2019.

[15] Zhang, Lianming, Jianping Yu, and Xiaoheng Deng. ”Modelling the guaranteed QoS for wireless sensor networks: a network calculus approach.” EURASIP Journal on Wireless Communications and Networking 2011.1 (2011): 1-14.

[16] Zhou, Boyang, and Liang Cheng. ”A Reality-Conforming Approach for QoS Performance Analysis of AFDX in Cyber-Physical Avionics Systems.” 2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS). IEEE, 2021.

[17] Goldsmith, Andrea J., and Pravin P. Varaiya. ”Capacity of fading channels with channel side information.” IEEE transactions on information theory 43.6 (1997): 1986-1992.

[18] Zhao, Ming, Yujin Li, and Wenye Wang. ”Modeling and analytical study of link properties in multihop wireless networks.” IEEE Transactions on Communications 60.2 (2012): 445-455.

[19] She, Huimin, et al. ”Modeling and analysis of Rayleigh fading channels using stochastic network calculus.” 2011 IEEE Wireless Communications and Networking Conference. IEEE, 2011.

[20] Martins, Wander Mendes, et al. ”A computer vision based algorithm for obstacle avoidance.” Information Technology-New Generations. Springer, Cham, 2018. 569-575.

[21] Tanghe, Emmeric, et al. ”The industrial indoor channel: large-scale and temporal fading at 900, 2400, and 5200 MHz.” IEEE Transactions on Wireless Communications 7.7 (2008): 2740-2751.

[22] Matsubara, Yoshitomo, and Marco Levorato. ”Neural compression and filtering for edge-assisted real-time object detection in challenged networks.” 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021.

[23] Kim, Seung-Wook, et al. ”Edge-network-assisted real-time object detection framework for autonomous driving.” IEEE Network 35.1 (2021): 177-183.

[24] Wang, Xu, et al. ”Edgeduet: Tiling small object detection for edge assisted autonomous mobile vision.” IEEE INFOCOM 2021-IEEE Conference on Computer Communications. IEEE, 2021.