A Simulation of Energy Optimized Distributed Video Processing on 28 GHz Network

Nattaon Techasarntikul
Osaka University
Osaka, Japan
techa.nattaon.cmc@osaka-u.ac.jp

Hideyuki Shimonishi
Osaka University
Osaka, Japan
shimonishi.cmc@osaka-u.ac.jp

Masayuki Murata
Osaka University
Osaka, Japan
murata@ist.osaka-u.ac.jp

***Abstract*—Transmitting a high data rate video to the cloud for real-time processing purpose requires minimizing the latency, maximizing the application requirements, and optimizing power consumption for the entire system. In this study, we employed a distributed video processing model for an object detection task, assuming that video streams are captured by robots operating in the licensed 28 GHz Milliwave network, ensuring the stability of video uploads. Through the optimization of power consumption, the system efficiently allocated video analysis frames to appropriate devices, resulting in an 18% decrease in overall power usage.**

***Index Terms*—distributed video processing, 28 GHz Millimeter-wave, object detection**

I. Introduction

A digital twin is a virtual representation of a physical object. It can be used to depict the current state or to predict the future state of a machine, location, or an entire city. To create a virtual representation of the real world, cameras and other sensors are employed to collect data, in which a substantial amount of data will be needed to be processed in order to generate a real-time digital twin. Recently, AI and deep learning (DL) techniques, such as image recognition, have been developed to assist in analyzing sensor data to comprehend real-world environmental contexts. These DL algorithms typically run on a cloud server because of their need for significant memory, CPU, and GPU resources. Due to the current trend of using DL for data training and prediction, the volume of data traffic sent to cloud servers is increasing excessively. Consequently, electricity consumption for both data transfer and cloud processing is expected to surge in the near future [1]. Moreover, real-time systems, such as IoT or object detection, typically have latency requirements. To address the challenge of minimizing the latency caused by transmitting data to a distant cloud, distributed edge computing has been introduced [2].

In this paper, we 1) utilize and refine the distributed edge computing model [2] to help minimize the power consumption of the entire system, and 2) run simulations of distributed video processing based on the millimeter-wave (mmWave) throughput map.

The next section describes the use case applications conducted under the mmWave network and discusses the work of distributed edge cloud systems. Section 3 details the optimized distributed video analysis system, including model definition, power consumption optimization algorithm, and related parameters. In Section 4, the 28 GHz mmWave testbed environment and its characteristic is introduced. Section 5 outlines the evaluation of the simulated distributed video analysis system. The results are described in Section 5, and Section 6 concludes the paper.

II. Related Work

A. Millimeter-wave Network Use Case

The mmWave band is the radio frequency range of 3–300 GHz (1–100 mm wavelength) in the 5G spectrum [3]. Although frequencies under 6 GHz have already been exploited, frequencies above 6 GHz are still largely available and have been extensively researched in recent years to understand their propagation mechanisms [4]. As the 1 GHz-wide channels can offer a data rate of several Gbps, the use of higher frequency ranges is promising; however, it comes with a trade-off in signal losses due to penetration, reflection, or diffraction.

Recently, with the availability of 60 GHz WiGig (IEEE 802.11ad) routers, many use case applications have been developed and tested, such as Mobile VR Streaming [5] and 360° video streaming [6], [7]. Additionally, for real-time VR 360° video streaming, edge computing played a role in facilitates video processing offloading and foveated rendering [8], [9].

In contrast to the more prevalent unlicensed 60 GHz bands, systems operating in the licensed 28 GHz band are less common but offer increased reliability because of the absence of interference from the same radio frequency. Notably, a 4K UHD video streaming system utilizing the 28 GHz frequency achieved successful implementation, covering a distance of 400 meters. This system was specifically designed for areas where the fiber network is inaccessible [10]. Moreover, teleoperation of an industrial robot arm using the 28 GHz frequency was successfully demonstrated as well [11].

While many application studies have predominantly focused on video streaming tasks with users in static positions, we simulated tests on a distributed video analysis system specifically designed for object detection purposes on the 28

GHz band. In our scenario setting, the video transmission device was assumed to be a robot that moves around the environment while capturing the video streams and sending them to edge or cloud servers to analyze the frame images.

B. Distributed Edge Cloud Computing

Cloud computing is a powerful centralized computing paradigm with abundant CPU, GPU, and storage resources. However, due to the limited number of data stations and the distance to the end device, transmission latency can occur depending on the network traffic. Therefore, edge computing has been introduced to enable data processing in close proximity to end devices, thereby reducing latency in real-time systems [12].

Nevertheless, in video analysis applications, determining the distribution of the processing load poses challenges because there are network congestion, variations in computational resources, and diverse application requirements. Simply splitting the video processing pipeline between available edge and cloud servers [13] may not satisfy the latency and accuracy requirements of applications because edge and cloud servers have different computational capabilities and distances from terminal devices. To address this, if an edge server is incapable of handling a large DL model, splitting the deep neural network pipeline can leverage edge resources without relying solely on the cloud server [14]. However, employing all servers leads to increased power consumption.

On the other hand, our approach involves splitting video streams into frames and distributing them among terminals, edges, or a cloud. This concept is illustrated in Fig. 1. Initially, a terminal device may analyzes video frame images for a designated portion. The analysis results, along with the remaining unprocessed frames, are then transmitted to the edge. The same process is repeated at the edge, and the results, along with the remaining unprocessed frames, are forwarded to the cloud. The cloud analyzes the final set of frames required for video analysis accuracy. The results from these three locations are integrated to obtain the final analysis results. This distributed processing allows for flexible distribution of processing across various devices.

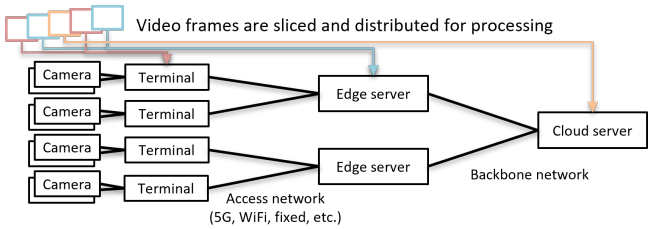

Fig. 1. A distributed video analysis system.

III. Distributed Video Analysis Optimization

For a real-time application with timing constraints or a video analysis application that requires high detection accuracy, meeting these requirements is essential. Additionally, the construction of an edge-cloud system involving multiple devices can lead to increased power usage. Therefore, optimizing the entire system is beneficial for reducing the energy consumption, which aligns with the global focus on sustainable energy.

The system optimization model used in this study is shown in Fig. 2. It encompasses the target system (i.e., video analysis application) in which each video session entails an application request detailed requirements such as recognition accuracy and processing latency. The video stream originating from the terminal device's camera undergoes segmentation into frames and is subsequently distributed across different devices (terminals, edges, or cloud server). The optimizer actively monitors the network's environment changes. It employs a Genetic Algorithm to find the optimized control results, including processing portions and the selection of AI models (in this study, Yolov3-tiny, Yolov3, Yolov3-spp) for the system control module. This control aims to satisfy predefined application latency and object detection accuracy constraints while simultaneously optimizing the overall power consumption of the entire system.

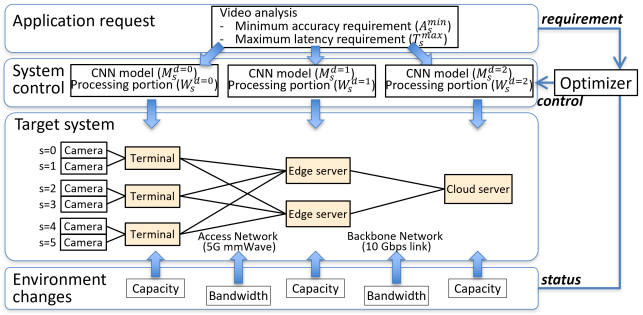

Fig. 2. System optimization model.

This distributed video analysis system is considered a constrained combinatorial optimization problem that aims to satisfy (1) the end-to-end processing time constraint (T_s), (2) average video analysis accuracy constraint (A_s), and (3) minimization of overall power consumption (P).

$$\underset{W_s^d, M_s^d}{\text{argmin}}\ P \ \textbf{s.t.}\ T_s \leq T_s^{max}, A_s \geq A_s^{min}, \forall s \in S$$

Through this section, S is the set of video sessions captured by the cameras. Each session (s) is processed by a set of devices (D_s) and transmitted through a set of networks (N_s). The optimization problem is defined as follows:

$$\begin{aligned} F = -P &+ \sum_{s \in S} min(T_s^{max} - T_s, 0)\alpha^f \\ &+ \sum_{s \in S} min(A_s - A_s^{min}, 0)\beta^f, \end{aligned} \tag{1}$$

where are α^f and β^f hyperparameters representing the penalty factors for latency and accuracy violations, respectively, and are set to be 10^6.

A. Total Power Consumption

$$P = \sum_{d \in D_s} P^d + \sum_{n \in N_s} P^n. \tag{2}$$

P^d is the power consumption of device d and P^n is that of network n.

We estimate the P^d by combining the device's idle power (P_{Idle}), CPU power, GPU power, and base power adjustment ($\gamma_{\text{Power-base}}$) as follows:

$$\begin{aligned} P^d = {} & P_{\text{Idle}} + P_{\text{CPU-TDP}} L^d_{\text{CPU}} \gamma_{\text{CPU-TDP}} \\ & + P_{\text{GPU-TDP}} L^d_{\text{GPU}} \gamma_{\text{GPU-TDP}} + \gamma_{\text{Power-base}}. \end{aligned} \tag{3}$$

The P_{Idle} is determined by measuring the device's power when turned on and not engaged in any tasks other than running the Ubuntu 20.0 operating system. The values for $P_{\text{CPU-TDP}}$ and $P_{\text{GPU-TDP}}$ are the maximum power consumption of the CPU and GPU, respectively. These values are sourced from the hardware catalogue. The coefficients $\gamma_{\text{CPU-TDP}}$ and $\gamma_{\text{GPU-TDP}}$ are determined through regression analysis conducted during the system measurement process.

The estimated load of the CPU (L^d_{CPU}) is defined as follows:

$$L^d_{\text{CPU}} = \sum_{s \in S^d} (\alpha_{\text{CPU}} W^d_s + \beta_{\text{CPU}} W'^d_s) + \gamma_{\text{CPU-base}}, \tag{4}$$

where S^d is the set of sessions running on the device d. W^d_s denotes the number of frames per second analyzed on the current device. W'^d_s denotes the number of frames per second received from previous device but unprocessed and sent to the next device.

The coefficients α_{CPU} and β_{CPU} correspond to analyzed frames and bypassed frames. The $\gamma_{\text{CPU-base}}$ represents the base CPU load primarily utilized for running the operating system. These value are also determined through regression analysis.

Next, the estimated load of the GPU (L^d_{GPU}) is defined as follows:

$$L^d_{\text{GPU}} = \frac{\sum_{s \in S^d} F^d_s}{C^d E^d}, \tag{5}$$

where F^d_s represents the FLOPS necessary for processing a session s. C^d reflects the GPU's capability in 32-bit floating-point (FP32) calculations on the device d, obtained from the graphics card specifications. E^d is the estimated efficiency of FP32 calculations for the GPU on the device d, determined through regression analysis of the measured data.

The F^d_s is modeled as follows:

$$F^d_s = (O_M + O^d_A) W^d_s + O^d_B W'^d_s, \tag{6}$$

where O_M represents the number of FLOPS for the model-dependent part of video processing, as obtained from the AI model specification (Table I). O^d_A is the FLOPS in model-independent part of video processing, and O^d_B is the FLOPS used in the fixed processing part for each device d. These parameters were determined through regression analysis based on the measured data and are depicted in Table II and Table III.

TABLE I
THE MAP AND FLOATING OPERATIONS OF THE AI MODELS

Model	mAP	FP32 operations (O_M)
Yolov3-tiny	33.1%	5.6B / frame
Yolov3	55.3%	65.9B / frame
Yolov3-spp	60.6%	141.5B / frame

TABLE II
DEVICE SPECIFICATION

	Terminal	Edge	Cloud
CPU Type	i7 8700T	i9 10940X	Xeon Gold 6330 x2
$P_{\text{CPU-TDP}}$	35 Watt	165 Watt	205x2 = 410 Watt
Nvidia GPU Type	GTX 1070	RTX A5000	Tesla A100
FP32 [TFLOPS] (C^d)	6.463	27.77	156
$P_{\text{GPU-TDP}}$	150 Watt	230 Watt	400 Watt
Idle power (P_{Idle})	21.3 Watt	98.3 Watt	212.8 Watt

TABLE III
SYSTEM PARAMETER
(FROM REGRESSION ANALYSIS OF THE MEASUREMENT DATA)

	Terminal	Edge	Cloud
GPU Load			
FP operation efficiency (E^d)	0.417	0.495	0.187
FLOPs A per frame [BFLOPs] (O^d_A)	2.05	14.60	22.12
FLOPs B per frame [B FLOPs] (O^d_B)	0.189	0	0
CPU Load			
Analyzed frame coefficient (α_{CPU})	0.186	0.145	0.139
Bypassed frame coefficient (β_{CPU})	0.265	0.025	0
Base load [Watt] ($\gamma_{\text{CPU-base}}$)	15	1.85	2.20
Power Consumption			
CPU TDP coefficient ($\gamma_{\text{CPU-TDP}}$)	1.98	0	0
GPU TDP coefficient ($\gamma_{\text{GPU-TDP}}$)	0.7	1.53	0.95
Base power [Watt] ($\gamma_{\text{Power-base}}$)	11.6	30.2	22.1

B. Session Latency

$$T_s = \sum_{d \in D_s} T^d_s + \sum_{n \in N_s} T^n_s \leq T^{max}_s, \tag{7}$$

T^d_s is the video analysis time for one frame on device $d \in D_s$. T^n_s is the transmission delay for one frame on network $n \in N_s$. T^{max}_s is the maximum time constraint for the session.

The T^d_s is calculated as follows:

$$T^d_s = \frac{O_M + O^d_A + O^d_B}{C^d E^d L^d_{\text{GPU}}}, \tag{8}$$

where each component is explained in Equation (5) and (6).

C. Session Accuracy

$$A_s = \frac{\Sigma_{d \in D_s} A^{M^d_s} W^d_s}{|D_s|} \geq A^{min}_s, \tag{9}$$

where, A_s^{min} is a session minimum accuracy constraint. M_s^d is the AI model that is selected to process the video session s on the device d. Its accuracy $A^{M_s^d}$ is given by the value of the mean average precision (mAP) of the AI model's recognition accuracy. W_s^d represent the processing portion of the video frames.

D. Network Model Equations

The transmission delay and power consumption in a network node were not measured in this study, as we considered them to be part of the public infrastructure. Therefore, we assume that the available bandwidth in the network cannot be predetermined but is measured in real time when the video stream is sent through network. The computational model for the power consumption P^n and network latency T_s^n are assumed as follows:

$$P^n = \sum_{s \in S} M\Big(1 - \sum_{d \in D'_s} (W_s^d + W_s'^d)\Big)p^n, \qquad (10)$$

$$T_s^n = M\Big(1 - \sum_{d \in D'_s} (W_s^d + W_s'^d)\Big)/B^n, \qquad (11)$$

where M is the number of bits per frame. p^n is the power consumption for one bit transmission on the network n. B^n is the bandwidth available at time t on the network n given as input in the simulation. D'_s is the set of devices that session s has passed. This means that the network bandwidth will not be consumed if there is no data transmit to the next device.

IV. Experimental mmWave Environment

We have a 28 GHz testbed network installed on a campus building. The mmWave base stations are constructed at a ceiling height of 2.7 meters from the floor. The building has floor dimensions of 20x40 meters. The mmWave specifications are presented in Table IV. Due to the nature of mmWave, it is susceptible to deterioration from shielding and inference caused by obstacles. Therefore, three base stations were installed to guarantee optimal communication quality.

The Reference Signal Received Power (RSRP) from each base station (BS) over the area is shown in Fig. 3. The circles represent the measured position (typically every 2x2 grid meters), and their color depicts the value measured at a height of 1 meter from the floor. The value less than -100 dBm is considered a bad signal. The red square with a number on the map shows each base station position. The arrow shows beamforming direction, set to an elevation angle in the range of -15° to +15°, and an azimuth angle in the range of -45° to + 45°.

BS#1 was installed in a student room, which contains only tables and chairs (no high obstacles in the area). The RSRP signal was in good condition for the area in front of the base station. BS#2 was installed in the reception area, where the RSRP signal propagated along the horizontal pathway inside the building. The measurements show that the signal degrades approximately 15 meters away from the base station in the orthogonal direction along the wall. BS#3 was installed on one side of the pathway. The RSRP signal is in good condition through the path in the front direction over 34 meters, but degrades in both the left and right directions, which are the non line of sight areas.

We also measured the upload bandwidth across the floor through a connection with each base station. The maximum upload bandwidth among the base stations, ranging from 18 to 48 Mbps (mean 39.5, SD 8.7), is shown in Fig. 4.

TABLE IV
Millimeter-wave specifications of RV1302-00M1 5G NSA base station

Components	Details
Radio frequency	Licensed 28 GHz (Band n257)
Center frequency	DL: 28.75008 GHz, 28.85004 GHz, 28.95000 GHz, 29.04996 GHz
	UL: 28.75008 GHz
Channel bandwidth	DL: 400 MHz (100 MHz x 4CC)
	UL: 100 MHz
Antenna	Directional antenna × 64 (8×8)
	Polarization (Vertical/Horizontal)
Transmission power (TRP)	-10 – +25 dBm
Antenna gain	5 dBi/antenna + 18 dB
Maximum EIRP	63 W/+48 dBm
Primary modulation method	QPSK, 16QAM, 64QAM
Modulation method	DL: OFDM, UL: OFDM
Duplex method	TDD (Synchronous TDD)
Radio type	X7W

V. Evaluation of distributed video analysis

We assumed a scenario where robots operate in a designated area, such as in a campus or a warehouse, with the purpose of finding objects. In this context, the robot functions as a terminal device responsible to capture the environment and uploads video streams to edge and cloud servers for object detection analysis. The network throughput conditions were subject to changes depending on the current position of the robot. In particular areas, where the radio signal is low can lead to fluctuations in network bandwidth. Moreover, in scenarios where multiple robots are co-operating in the same area, network congestion would occur, thereby, affecting the transmission delay.

In this simulation, the paths of the robots are established according to Fig. 5, which shows the variation of the network upload throughput along each path. Each circle mark on the map represents the measured throughput position. We determined the robot to take 60 seconds to move to each circle mark, and it required an additional 60 seconds for rotation. In total, it takes 1,440 seconds (24 minutes) for the robot to complete one path.

By following the green path, the robot traversed a low uplink bandwidth area of 28 Mbps for two minutes. On the yellow path, the robot will encounter 21-27 Mbps for two minutes. The blue path entails the robot passing

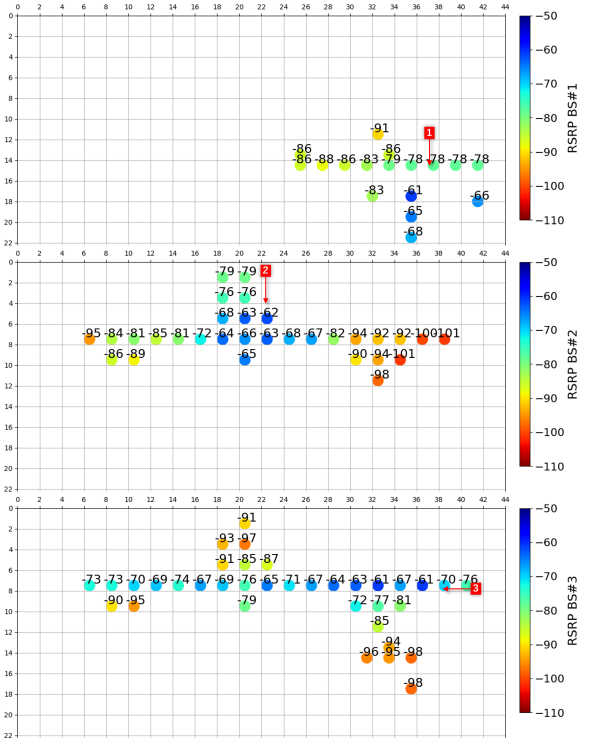

Fig. 3. RSRP value map for each base station

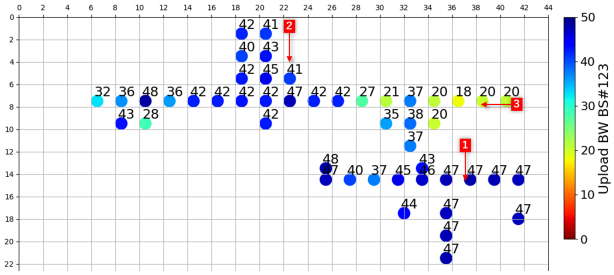

Fig. 4. Maximum Upload bandwidth map of all base stations

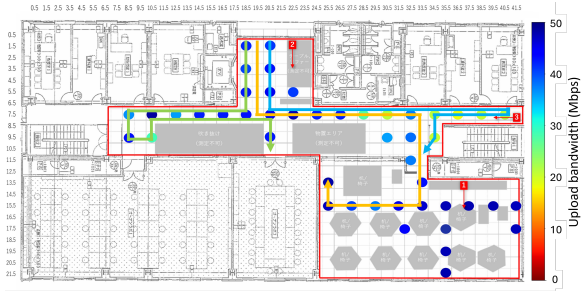

Fig. 5. Robot paths

through a low uplink bandwidth range of 18-27 Mbps for 14 minutes, which is almost half of its total runtime. When there are more than two robots operate in the same network environment, the upload throughput is divided by the number of robots. Each robot is assumed to equip with two cameras, each capturing a video stream at a 1920×1080 pixel resolution, with an average data rate of 6 Mbps (12 Mbps for two video sessions) sent to the edge or cloud servers to process the videos. The results of the video analysis are the bounding box area of the object detected, object name, and confidence value. Fig. 6 show the video analysis results overlaid on input video frame image.

Based on the observation, the cloud server we used in this study is the most power consumption efficient in processing video analysis compared to terminal and edge servers. Therefore, when there is no network congestion, the optimization algorithm typically selects a cloud server to handle a video session.

Fig. 6. Video analysis results when overlaid on input video frame image.

VI. Results

A. Simulation results when running one robot

First, we conducted simulations with one robot at a time on each path (green, yellow, and blue). The application's maximum latency requirement (T_s^{max}) was 0.1 ms for both video sessions, and the minimum object detection accuracy (A_s^{min}) requirements were 0.6 and 0.5 for each session. These constraints are consistent for the robots on different paths.

Fig. 7 (a) shows results of total power consumption, latency, and accuracy when one robot is operated in the environment. The total power consumption is less than 400 watts for all three paths, as the throughput of mmWave network is enough for transmitting two video sessions to process at the cloud server. For the blue path which including low bandwidth areas, the sessions latency is slightly higher than other paths, however not exceeded the requirement of 0.1 ms, therefore no latency violation occurs. No violations in object detection accuracy have been observed.

B. Simulation results when running two robots

Next, three simulations involving the operation of the two robots were conducted with combinations of each of

the following two paths: green+blue, green+yellow, and yellow+blue. The application's maximum latency requirements (T_s^{max}) were 0.1, 0.1, 0.2, and 0.05 ms for each video session, and the minimum object detection accuracy requirements (A_s^{min}) were 0.6, 0.5, 0.6 and 0.3 for each session.

The simulation results are shown in Fig. 7 (b). The total power consumption when operating the two robots was slightly higher than that when operating one robot. This is because one more terminal device (robot) is increased. In the green+yellow path, which has a relatively high bandwidth, all computations are sent to the cloud, resulting in a stable and the lowest power consumption. However, for the green+blue and yellow+blue paths, where the robots traverse areas of low bandwidth, leading to session latency exceeded the 0.2 ms requirement; therefore, a portion of the video frames is allocated the terminal device to help process them. This results in an increase of the total power consumption, as the terminal device need to run GPU to conduct a video analysis. Despite this, there are also no violations of object detection accuracy observed in the simulation with the two robots.

C. Simulation results when running three robots

Finally, simulations were performed using three robots in the environment. The application's maximum latency requirements (T_s^{max}) were set to 0.05, 0.05, 0.1, 0.1, 0.2 and 0.2 ms for each video session, and the minimum object detection accuracy requirements (A_s^{min}) were 0.3, 0.3, 0.5, 0.5, 0.6 and 0.6 for each session.

We compared two scenarios: one with power optimization (the default condition for previous simulations), and another without power optimization. In the absence of power optimization settings in the algorithm, the video analysis tasks were allocated to all available devices (terminal, edge, and cloud servers); hence, the total power consumption exceeded 1,000 watts. However, when running the system with power optimization, the video analysis tasks were allocated only to terminals and the cloud server, resulting in a decrease in the total power usage to approximately 800 watts.

Latency violation also occurred for both simulations due to the robots operating in low-bandwidth areas (blue path).

In this simulation, although latency violations were observed, there were no instances of object detection accuracy violations. This was attributed to the combined utilization of distributed terminals, edges, and cloud server, which proved capable of sustaining the analysis.

VII. Conclusion

In this study, we applied the optimized distributed video analysis model to a scenario involving robots detecting objects. We took the real measured of 28 GHz mmWave network uplink bandwidth in the environment, ensuring a realistic representation of environmental conditions for the simulation. Our evaluation involved a comprehensive set of simulations, assessing the performance of the distributed video processing model in scenarios with one, two, and three robots navigating various paths within the environment, taking and sending the video stream to the edge-cloud server for object detection processing. With the power consumption optimization, the system efficiently allocated video analysis frames to suitable devices. Resulting in 18% reduction in power consumption on a scenario where three robots capturing 1080p video of the environment on the mmWave network over a 24 minutes duration.

Acknowledgment

This work was supported by MIC under a grant entitled "R&D of ICT Priority Technology (JPMI00316)".

References

[1] A. S. Andrae, "Prediction studies of electricity use of global computing in 2030," *International Journal of Science and Engineering Investigations*, vol. 8, no. 86, pp. 27–33, 2019.

[2] H. Shimonishi, M. Murata, G. Hasegawa, and N. Techasarntikul, "Energy optimization of distributed video processing system using genetic algorithm with bayesian attractor model," in *2023 IEEE 9th International Conference on Network Softwarization (NetSoft)*. IEEE, 2023, pp. 35–43.

[3] Z. Pi and F. Khan, "An introduction to millimeter-wave mobile broadband systems," *IEEE communications magazine*, vol. 49, no. 6, pp. 101–107, 2011.

[4] T. S. Rappaport, Y. Xing, G. R. MacCartney, A. F. Molisch, E. Mellios, and J. Zhang, "Overview of millimeter wave communications for fifth-generation (5g) wireless networks—with a focus on propagation models," *IEEE Transactions on antennas and propagation*, vol. 65, no. 12, pp. 6213–6230, 2017.

[5] J. Chakareski, M. Khan, T. Ropitault, and S. Blandino, "6dof virtual reality dataset and performance evaluation of millimeter wave vs. free-space-optical indoor communications systems for lifelike mobile vr streaming," in *2020 54th Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2020, pp. 1051–1058.

[6] T. T. Le, D. Van Nguyen, and E.-S. Ryu, "Computing offloading over mmwave for mobile vr: Make 360 video streaming alive," *IEEE Access*, vol. 6, pp. 66 576–66 589, 2018.

[7] H.-W. Kim, T. T. Le, and E.-S. Ryu, "360-degree video offloading using millimeter-wave communication for cyberphysical system," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 4, p. e3506, 2019.

[8] X. Huang, J. Riddell, and R. Xiao, "Virtual reality telepresence: 360-degree video streaming with edge-compute assisted static foveated compression," *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[9] Y. Liu, J. Liu, A. Argyriou, and S. Ci, "Mec-assisted panoramic vr video streaming over millimeter wave mobile networks," *IEEE Transactions on Multimedia*, vol. 21, no. 5, pp. 1302–1316, 2018.

[10] A. S. Biswas, S. N. Sur, R. Bera, and M. Mitra, "4k uhd video streaming and gigabit data to an unreachable smart home using millimeter wave radio," in *2020 URSI Regional Conference on Radio Science (URSI-RCRS)*. IEEE, 2020, pp. 1–4.

[11] K. Yaovaja and J. Klunngien, "Teleoperation of an industrial robot using a non-standalone 5g mobile network," in *2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*. IEEE, 2019, pp. 320–323.

[12] Y. Chen, Y. Sun, C. Wang, and T. Taleb, "Dynamic task allocation and service migration in edge-cloud iot system based on deep reinforcement learning," *IEEE Internet of Things Journal*, 2022.

[13] K. Rao, G. Coviello, W.-P. Hsiung, and S. Chakradhar, "Eco: Edge-cloud optimization of 5g applications," in *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. IEEE, 2021, pp. 649–659.

[14] R. Mehta and R. Shore, "Deepsplit: Dynamic splitting of collaborative edge-cloud convolutional neural networks," in *2020 International Conference on COMmunication Systems and NETworkS (COMSNETS)*, 2020, pp. 720–725.

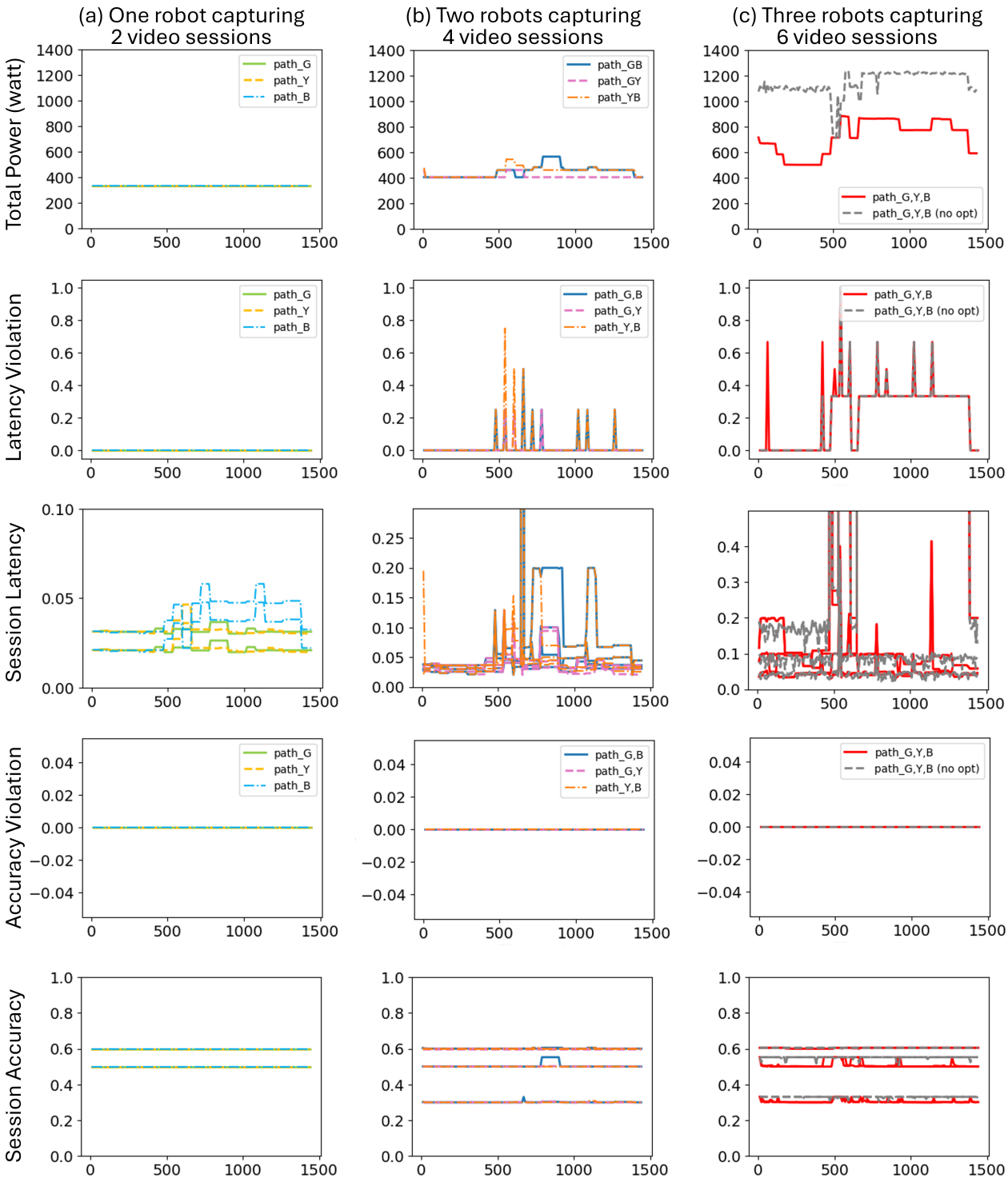

Fig. 7. Simulation results of distributed video analysis when operating robots in mmWave environment for 24 minutes (1440 seconds).