Research, Society and Development, v. 11, n. 8, e8811830543, 2022
(CC BY 4.0) | ISSN 2525-3409 | DOI: http://dx.doi.org/10.33448/rsd-v11i8.30543

# Experimental network performance evaluation for human-robot interaction collision detection using cameras

## Avaliação experimental de desempenho de rede para detecção de colisão de interação humano-robô usando câmeras

## Evaluación experimental del rendimiento de la red para la detección de colisiones entre humanos y robots mediante cámaras

Received: 05/18/2022 | Reviewed: 06/05/2022 | Accept: 06/16/2022 | Published: 06/17/2022

**Assis T. de Oliveira Filho**
ORCID: https://orcid.org/0000-0001-9873-6929
Universidade Federal de Pernambuco, Brazil
E-mail: assis.tiago@gprt.ufpe.br
**Gibson Barbosa**
ORCID: https://orcid.org/0000-0001-9023-4019
Universidade Federal de Pernambuco, Brazil
E-mail: gibson.nunes@gprt.ufpe.br
**Iago Richard Rodrigues**
ORCID: https://orcid.org/0000-0002-8242-9059
Universidade Federal de Pernambuco, Brazil
E-mail: iago.silva@gprt.ufpe.br
**Carolina Cani**
ORCID: https://orcid.org/0000-0001-7986-4554
Universidade Federal de Pernambuco, Brazil
E-mail: carolina.cani@gprt.ufpe.br
**Judith Kelner**
ORCID: https://orcid.org/0000-0002-2673-5887
Universidade Federal de Pernambuco, Brazil
E-mail: jk@gprt.ufpe.br
**Djamel Sadok**
ORCID: https://orcid.org/0000-0001-5378-4732
Universidade Federal de Pernambuco, Brazil
E-mail: jamel@gprt.ufpe.br
**Ricardo Souza**
ORCID: https://orcid.org/0000-0001-5378-4732
Ericsson Research, Brazil
E-mail: ricardo.s.souza@ericsson.com

**Abstract**

A practical solution to human-robot collision detection using devices commonly found in workplaces, such as 2D cameras, requires thorough planning and evaluation of network restrictions that may deny timely access to and process important context data collected by IoT devices. In this study, we evaluate the behavior of the AMQP and MQTT application protocols for camera image transmission. We examine the packet overhead for each protocol when streaming video signals. Also, we evaluate the impact of transmission delay on the total decision time starting from the moment the camera captures an image to the moment it is decided if there is a robot-human collision or not. Finally, we also evaluated an open-source platform to emulate wireless Mininet-wifi, seeking to understand the level of influence it would have on the results. The results show that transmission overhead represents as much as 80% of the total decision time and that the AMQP protocol takes around 5% less transmission time than MQTT. The results also show that the use of hardware accelerators such as a GPU increases by 37 times the number of detections. We found that the size of the image to be transmitted and wireless communications did not influence the results for our scenario. In addition, we also noticed that the use of emulation through Mininet-wifi does not negatively influence the behavior of the experiments.

**Keywords:** Human Robot Interaction; Network communication; AMQP; MQTT; Teaching.

**Resumo**

Uma solução prática para detecção de colisão humano-robô usando dispositivos comumente encontrados em locais de trabalho, como câmeras 2D, requer planejamento e avaliação completos das restrições de rede que podem negar acesso oportuno e processar dados de contexto importantes coletados por dispositivos IoT. Neste estudo, avaliamos o comportamento dos protocolos de aplicação AMQP e MQTT para transmissão de imagens de câmeras. Examinamos a sobrecarga de pacotes para cada protocolo ao transmitir sinais de vídeo. Além disso, avaliamos o impacto do atraso de transmissão no tempo total de decisão desde o momento em que a câmera captura uma imagem até o momento em que é decidido se há ou não uma colisão robô-humano. Por fim, também avaliamos uma plataforma de código aberto para emular Mininet-wifi sem fio, buscando entender o nível de influência que ela teria nos resultados. Os resultados mostram que a sobrecarga de transmissão representa até 80% do tempo total de decisão e que o protocolo AMQP leva cerca de 5% menos tempo de transmissão que o MQTT. Os resultados também mostram que o uso de aceleradores de hardware, como uma GPU, aumenta em 37 vezes o número de detecções. Descobrimos que o tamanho da imagem a ser transmitida e as comunicações sem fio não influenciaram os resultados para nosso cenário. Além disso, notamos também que o uso de emulação através do Mininet-wifi não influencia negativamente no comportamento dos experimentos.

**Palavras-chave:** Interação Humano Robô; Comunicação de Rede; AMQP; MQTT; Ensino.

**Resumen**

Una solución práctica para la detección de colisiones entre humanos y robots utilizando dispositivos que se encuentran comúnmente en los lugares de trabajo, como cámaras 2D, requiere una planificación y evaluación exhaustivas de las restricciones de la red que pueden negar el acceso oportuno y procesar datos de contexto importantes recopilados por dispositivos IoT. En este estudio, evaluamos el comportamiento de los protocolos de aplicación AMQP y MQTT para transmitir imágenes de cámara. Examinamos la sobrecarga del paquete para cada protocolo al transmitir señales de video. Además, evaluamos el impacto del retraso de la transmisión en el tiempo de decisión total desde el momento en que la cámara captura una imagen hasta el momento en que decide si hay o no una colisión robot-humano. Finalmente, también evaluamos una plataforma de código abierto para emular Mininet-wifi inalámbrico, buscando comprender el nivel de influencia que tendría en los resultados. Los resultados muestran que la sobrecarga de transmisión representa hasta el 80 % del tiempo total de decisión y que el protocolo AMQP requiere aproximadamente un 5 % menos de tiempo de transmisión que MQTT. Los resultados también muestran que el uso de aceleradores de hardware, como una GPU, aumenta 37 veces el número de detecciones. Descubrimos que el tamaño de la imagen a transmitir y las comunicaciones inalámbricas no influyeron en los resultados de nuestro escenario. Además, también notamos que el uso de la emulación a través de Mininet-wifi no influye negativamente en el comportamiento de los experimentos.

**Palabras clave:** Interacción entre humanos y robots; Comunicación de Red; AMQP; MQTT; Enseñanza.

## 1. Introduction

The increasing use of collaborative robots is a current trend, mainly due to the benefits of productivity improvement and production cost reduction. Research predicts that by 2021, there will be a 60% increase in the use of robots in industries. However, the interaction between humans and robots may introduce new safety concern factors in the workplace in the form of accidents (Microsoft, 2019). Considering the demonstrated data from industrial accidents reported by the United States Bureau of Labor Statistics (of Labor Statistics, 2021), although the number of accidents has declined, about 2.8 million incidents have been reported recently.

It is essential to reduce this number even (Jovanović & Jovanović, 2004) (Rajak et al., 2021) (Jeong, 2021). Hence, there is a need to dedicate new resources to deal with the problems brought by these new technologies. Also, it is necessary to adopt simple, achievable solutions that mitigate risk without the need for extensive investments or significant structural changes. There are indeed many recent studies that address these challenges with different approaches. Didactic suggestions with more preventive characteristics can and should be used (Matsas & Vosniakos, 2017). Still, recently more effective reactive and proactive strategies employing technologies such as computer vision and machine learning have emerged and stood out (Mohammed et al., 2017) (Wang et al., 2017).

Since we are dealing with a safety application that requires timely access to transmitted data, the availability of adequate network resources becomes a relevant concern. Information regarding a possible or existing collision(s) must be agile to obtain and process. Besides, the network infrastructure itself cannot represent a working hazard. For example, in a factory environment, one should avoid extending communication cables around spaces where robots and humans move. Other risks such as fire and the presence of heavy machines, for example, could accidentally cut or damage communication lines and disrupt their connectivity. As a result, several prominent wireless technologies, such as WirelessHART and ISA100.11a. Both utilize the IEEE 802.15.4 Direct Sequence Spread Spectrum (DSSS) radios and operate in the 2.4GHz ISM radio band.

This work aims to evaluate the performance and impact of a collision detector that uses a deep learning model running across a structured network that transfers images before processing. This model is trained using a transfer learning approach with data generated and transmitted by a 2D surveillance camera. We evaluate this scenario for different image sizes, with and without GPU utilization, and an emulated network. Furthermore, we seek to understand the impact of video transmission on a wireless network using the AMQP and MQTT protocols for collision detection.

The results show that GPU inclusion increases by 37 times the detection number. Moreover, the image size to be transmitted does not influence the evaluation of the performance metrics in this study. By analyzing the impact of the user application-level protocol for image transmission and transmission time on the overall detection time, we found that the AMQP protocol has a 5% advantage over the MQTT transmission time. However, the overhead and number of packets are pretty much the same. For both protocols, the transmission time represents 80% of the total decision time separating the instant an image is taken from the camera until the detection module offering feedback on whether a collision occurs.

The remainder of this work is structured as follows. Section 2 presents the related works to the present study. Section 3 described the structured system and presented the scenario, the network configuration, and the learning model used. Section 4 describes the experiments performed for system evaluation. Section 5 presents and discusses the results of the described experiments. Finally, section 6 reports our conclusions.

## 2. Related Studies

Several works provide a comparative analysis of messaging protocols related to the application layer of the traditional TCP/IP model. Initially, we highlight the works where there is a concise conceptual and comparative analysis of some well-accepted messaging protocols, including AMQP and MQTT, in the context of IoT and industry. In n (Breivold & Sandström, 2015), see an explanation of the challenges in using IoT in industry. In this work, relevant peculiarities of this scenario are also addressed, and possible solutions to be used are suggested. In (Moraes et al., 2019), the study also takes another more detailed approach to evaluate these protocols and the CoAP. The authors highlight some aspects, such as throughput, message size, and packet loss, are the adopted metrics, and experiments indicate that the CoAP protocol provides the best results. Halder et al (Halder et al., 2018) compare various technologies used for smart device communication. They investigate issues such as power consumption and average packet time. Although the results using the CoAP protocol are superficially better in some aspects of performance, they suggest that the final choice of the protocol used is, in fact, highly dependent on the nature and specificity of the application.

The research (Jaloudi, 2019) e evaluated the use of MQTT in conjunction with MODBUS TCP, a standard used for automation and user control in the industry. The goal is to create an IoT interoperability environment for monitoring, capturing data, and to monitor the Internet. The authors based their proposal on the transfer of MODBUS messages as an MQTT payload.

Thangavel et al. (Thangavel et al., 2014) analyzed the MQTT and CoAP protocols through a standard and flexible middleware. A gateway was modeled that enabled a homogeneous and more simplified environment for protocol performance evaluation. In this exercise, they highlighted particularities in the network conditions. The final results indicated that MQTT performs better in a low packet loss environment and offers less overhead with small messages. They also noted that CoAP causes less overhead in high packet loss scenarios.

In (Sultana & Wahid, 2019), application-layer protocols are observed in the context of video Internet of Things (IoT), more specifically, following generation video surveillance systems (VSS). The MQTT, AMQP, HTTP, XMPP, CoAP, and

DDS protocols were included. The application took place by constructing a test scenario and development developed by the authors themselves. Measurements were used, such as Latency, CPU Throughput, Memory Usage, and Energy Consumer. Other opportunities for using AMQP and the possibility of using AMQP and the possibility of using it in real-time, due to the low latency of the protocols, especially AMQP being efficient, and MQTT choice in restricted means of communication. In addition, the authors also note the high overhead and noncompliance of the HTTP and XMP protocols due to their complexity. In this study, it was also possible to observe the significant delay through CoAP, which makes it incompatible with real-time applications.

In (Naik, 2017), a qualitative evaluation of four messaging protocols for IoT systems is demonstrated: MQTT, CoAP, AMQP, and HTTP. The relative assessment is based on several studies that include these protocols. Several metrics were considered for evaluating the mentioned protocols, such as Message Size, Message Overload, Power Consumption, Resource Requirement, Bandwidth, Latency, Reliability, and Interoperability, among others. Among the main results that the authors observed, the high consumption, overhead, energy consumption, latency, and resource requirement stand out, contrasting with the bandwidth and interoperability related to the HTTP protocol in the messaging. The excellent performance of the CoAP protocol regarding latency and power consumption. In addition to the average performance, in general, of the AMQP and MQTT protocols. However, this study did not consider the dynamic network conditions and the overheads incurred in the retransmission of packets, which may produce different results from the comparisons performed.

The work (Uy & Nam, 2019) stands out for the observance of the behavior of the AMQP and MQTT protocols, taking into account the network aspects, wherein, in this case, the delay and the loss were varied. The authors mainly use the metric, the transmission time, to estimate. The experiments were transmitted through the times of a test, through the continuous wireless network, or through predefined times. The authors found that the MQTT protocol offers benefits such as low energy use and more efficiency in the use of the transmission line, being a good choice, where there is a loss rate between five to ten percent and the data flow Couples are not transmitted. The AMQP protocol stood out regarding reliability and data integrity, resulting in mechanisms mainly for continuous and durable queues, clustering, and high availability queues.

Finally, in (Gemirter et al., 2021) an experimental evaluation of the MQTT, AMQP, and HTTP protocols is presented using Azure IoT SDKs, where a public smart city dataset in real-time is used for data transmission. Mean message latencies and percentage of CPU usage were measured by varying different loads and message sizes. Through this study, it was found by the authors that in all test scenarios, the MQTT and AMQP protocols are four times faster for message latencies and use four times less CPU than the HTTP protocol. Furthermore, it was also found that MQTT and AMQP provide similar results overall. Finally, it was also identified that AMQP offers more stable latencies, and MQTT are the lightest protocol.

It is essential to highlight that our work differs from those mentioned mainly, as it seeks to evaluate in a context of a natural and specific use case, where we perform a representation of a concrete use case of an application that seeks to offer more security in an industrial environment. Thus, our research developed and explained in the present work offers results that are more consistent with the peculiarities of a natural network environment. In addition, we also seek to evaluate an application that serves to emulate WiFi environments, seeking to highlight the level of influence of this application on the results. This application can facilitate the conduction and development of future studies. Another point is that in our work, we seek to use a range of different data received from other sensors to represent a more heterogeneous IoT network. So simultaneously, we mainly assessed the transmission network impact of broadcast video streaming caused by the two IoT transport protocols, MQTT and AMQP.

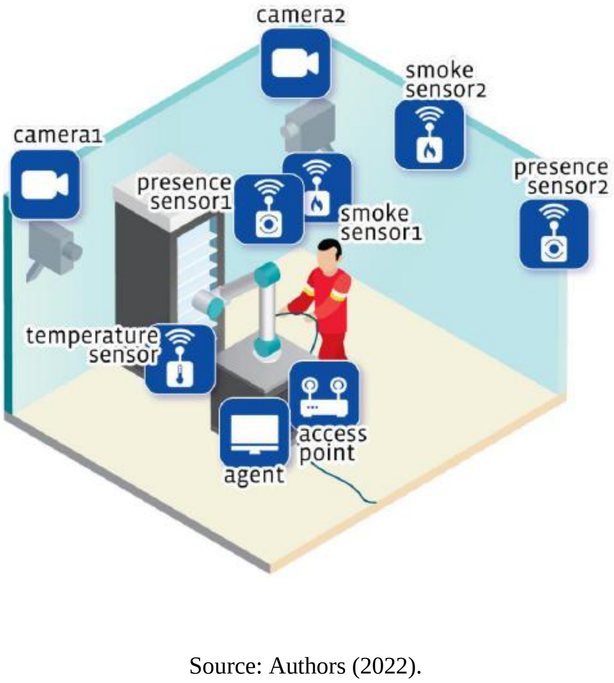
## 3. System Description

A collision detection system must enjoy a fast reaction time as we deal with factors that can cause physical injuries, fatal accidents, and financial loss. Thus, the network should not be, in any way, a limiting factor or bottleneck in this process. It is, therefore, necessary that its topology be configured to provide high availability and low response time. For example, in the vent of equipment damage or a fire, the system should be planned to remain operational and withstand these problems to a certain degree. We avoid using a cabled network infrastructure in an environment such as the industrial floor where robots, machines, and humans operate closely. Wired network connections can be unplugged, cut, or degraded by the environment, devices, or moving robots and the presence of chemical substances; this is the reason for considering a scenario with a wireless network in the first place. To achieve our requirements, we present in section 3.1 the components and their arrangement in the scenario and describe in section 3.2 the adopted network structure.

### 3.1 Scenario

The overall scenario represents an industry floor with sensors, machines, and humans interacting with robots. For this purpose, some of the most common sensors applied in the industry are considered, as pointed out by Embitel (Embitel, 2018). Thus, our scenario has two smoke sensors, two presence sensors (infrared), a temperature sensor, and two image sensors (cameras). A fixed robotic arm is employed. We are using a UR5 robotic arm from Universal Robots. It is used to insert and remove cables in a network patch panel of a communication rack. The robot also interacts with human operators to pick up or deliver cables. A computer processes the data received by the sensors while running a module called Agent. The access point (AP) connects all of these devices. The arrangement of the components in the scenario is described in Figure 1.

**Figure 1.** Industry Scenario with sensors and human-robot interaction.



Source: Authors (2022).

### 3.2 Network Configuration

We chose to deploy an emulated wireless network for connecting the devices described in section 3.1. It gives us more control and allows more flexibility in varying network factors for testing (bandwidth, packet loss, device positioning, number of components, etc.). It prevents real network unpredictability, such as network congestion, disconnects, and improper connections.

We built the emulated wireless environment using the Mininet-wifi tool. Like Mininet network emulator software, Mininet-wifi is a CBE (Container Based Emulation) emulator. In this virtualization model at the process level, containerization shares many system resources. Hence it requires less natural machine resources, making it lighter than other virtualization models (Fontes et al., 2015).

Figure 2 illustrates a three-dimensional vision of the placement of the emulated components in Mininet-wifi. The computer used to emulate the network environment and devices has the specifications reported in Table 1.

**Table 1.** Specifications of the computer used to emulate the scenario.

| | |
|---|---|
| Processor | Intel Core I7 3770 |
| RAM | 16GB |
| GPU | GeForce GTX 1060 6GB |
| HD | WDC WD10EZEX00R 1TB |
| OS | Ubuntu 18.04.3 LTS |
| Kernel | 5.0.032generic x86_64 |

Source: Authors (2022).

### 3.3 Collision Detector

We chose to We will take into account an industrial scenario where it is necessary to make the risk identification for the employees, in this case, we can see that there are response time restrictions because The system needs to respond quickly when it identifies that someone is at risk to have a quick reaction time. We use a collision detection system represented with characteristics similar to that described in (Silva et al., 2020).

## 4. Experiments

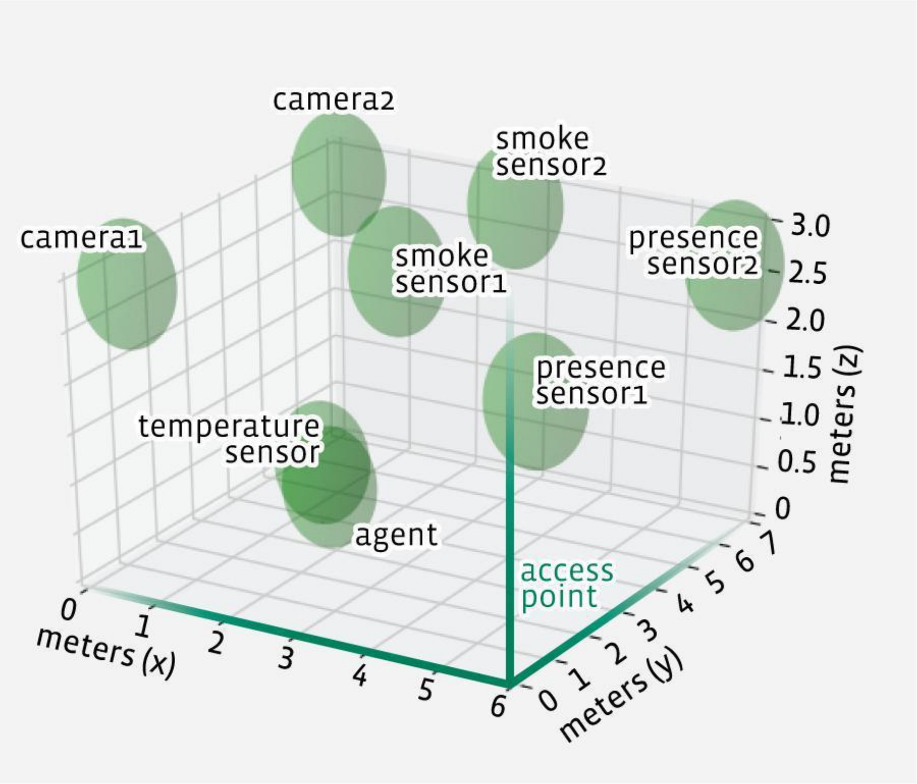
This section describes the developed experiments. In the first one (Section 4.1), the detection program performance is evaluated by varying the image size to be detected, whether a GPU is used or not, and if the processing is on a Mininet-wifi emulated machine. Next, the experiment in section 4.2 aims to evaluate the impact of varying the application protocol used. Both experiments use the Collision Detector reported in section 3.3, and they are executed on the machine described in Table 1.

### 4.1 Experiment 1

The first experiment´s objective is to evaluate the maximum number of detections (decisions) per second. As a result, we first implement a solution for detecting collision events based on the machine learning model without considering any network connection. Therefore, this allows us to have a performance parameter without the constraints of the data reception rate and latency possibly imposed by any used network. This enables the isolation of learning from network overheads. We will use these results as a baseline to compare with other obtained results obtained when using wireless transmission between devices in the following experiment.

The varied factors or factor levels for this experiment are listed in Table 2. The goal is to examine the benefit of using a GPU and evaluate whether or not the machine learning model is optimized for processing with this type of device and capture the improvement level brought by its use. We also run the detection program in the emulated environment, regardless of network characteristics, to evaluate if running in a Wifi environment, such as the emulated Mininet-wifi process, forces any processing restrictions compared to running it on a separate machine without Mininet-wifi. Additionally, we vary the image size to measure if it impacts the algorithm detection time, regardless of accuracy, to assess whether it is worth resizing the image for later detection. To keep our experiments manageable, we considered only two image sizes: 286KB (1920x1080 pixels), which is the raw image size returned by the camera used, and 114KB (550x630 pixels), which represents the image cropped in the area of interest, which, in this case, is the robotic arm's workspace.

**Figure 2.** Emulated Scenario.



Source: Authors (2022).

A single computer core was dedicated exclusively to process the tests to reduce the interference from other processes executing on the same host, such as the Mininet-wifi emulator itself, and to reduce the number of process state changes between cores.

**Table 2.** Factors used in the first experiment.

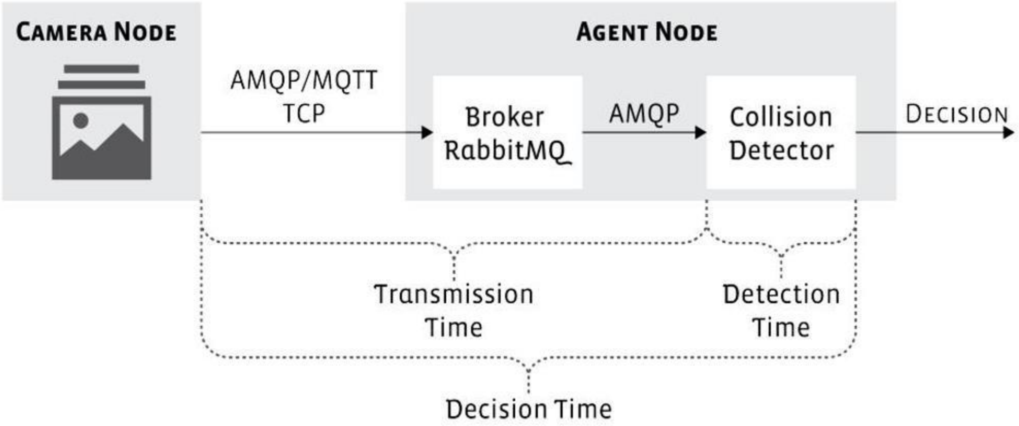
| Factors | Value |
|---|---|
| GPU | Use / Do not use |
| Mininet-wifi | Use / Do not use |
| Image Size | 114KB / 286KB |

Source: Authors (2022).

### 4.2 Experiment 2

This second experiment evaluates the impact of AMQP and MQTT protocols on decision-making concerning collision detection. Such protocols are widely used for message exchange in conjunction with the IoT context. Hence it is essential to evaluate which one behaves best in the presented scenario, especially considering image transmission. This experiment focuses on video stream transmission since the camera is the component that generates the highest throughput in our context. The network topology and elements are structured using the Mininet-wifi emulation software, as shown in Figure 2. Without loss of generality, we assume that packets are not lost during transmission. The transmission flow is depicted in Figure 3. The modules that makeup such a flow are described below:

**Figure 3.** Transmission Flow.



Source: Authors (2022).

• **Camera Node:** This node uses an MQTT or AMQP connection to transmit video frames to the Broker in the Agent node. It simulates a video stream sending collision and non-collision images previously loaded into memory at a user-defined frequency. We used the Python Pika library to publish AMQP images. The publication of MQTT messages is achieved by using the Paho Python library.

• **Agent Node:** This node defines where message reception and processing for decision-making occur. Thus, it reads as input AMQP or MQTT messages and returns an indication of whether a collision has occurred.

Below we describe the modules that make up the agent node:

- **Broker RabbitMQ:** It is a message exchange service that serves as an intermediary between a network component that needs to transmit a message (publisher) and another component that wants to read it (or subscriber). We choose RabbitMQ because of its flexibility in supporting different communication protocols such as AMQT and MQTT.

- **Collision Detector:** This module processes frames and decides whether the present image depicts a possible collision. The input of this module is the frame to be processed, and the output is the response (1collision or 0 noncollision).

Therefore, we performed several tests considering the variation of parameter levels or experimental factors. Initially, the number of sent Frames Per Second (FPS) by the camera is varied from 1 to 30, performing for each FPS rate 30 tests with each transmission lasting one minute. This allows evaluating the transmission limit for each protocol, AMQP or MQTT. The image has a 114KB size. This information is summarized in Table 3.

**Table 3.** Factors used in the second experiment.

| **Factors** | **Value** |
|---|---|
| FPS | 1 to 30 |
| Protocol | AMQP / MQTT |
| Image Size | 114KB |

Source: Authors (2022).

Next, as one of the protocols cannot reach a 30 FPS rate, and for a fair comparison, we use such a maximum as a limit for both protocols. So, we tested just for a 22FPS image rate and repeated 30 executions for each experiment. Each execution lasted 30 minutes. The number of transmitted packets and protocol overhead are evaluated in this case. Moreover, we assessed the transmission, detection, and decision times, which are represented in Figure 3 and will be described below:

• **Transmission Time:** This is the period from the moment a Camera Node sends a message until it arrives at the Collision Detector module.

• **Detection Time:** Represents the time that the Collision Detector takes to identify whether or not there is a collision in the received frame.

• **Decision Time:** It counts the time since a Camera Node has sent a message until the Collision Detector reaches a decision. In other words, it is the sum of the Transmission Time and the Detection Time.

The following formula relates the two factors (Equation 1):

$$Decision(T) = Transmission(T) + Decision(T) \qquad (1)$$

## 5. Results

In this section we present and carry out a brief discussion on the results collected by the experiments.

### 5.1 Experiment 1

The experiment results described in section 4.1 can be seen in Table 4. When we compare the use of a smaller (114KB) and a larger (286KB) image size, it is possible to realize that there is no considerable variation in the number of images detected. So, according to this criterion, the image can be resized or cropped in the interest area to reduce the data exchanged over on the network without a considerable impact on detection.

**Table 4.** Number of images detected per second for different configurations (standard deviation is in parentheses).

| Image | GPU Used | Mininet Used | |
|---|---|---|---|
| | | Yes | No |
| 114KB | No | 2.61 (0.010) | 2.61 (0.010) |
| | Yes | 96.33 (1.442) | 96.24 (1.351) |
| 286KB | No | 2.57 (0.018) | 2.59 (0.011) |
| | Yes | 96.50 (1.417) | 96.10 (1.341) |

Fonte: Authors.

When we do not use the GPU, we manage approximately 2.5 detections (meaning we examine only 2.5 images) per second, making it impractical for real scenarios where there could be multiple cameras. This scenario suffers from a significant information loss, for example, in the event of a fast collision, which can occur in a time shorter than the detection period. When using a GPU, there was an improvement of approximately 37 times more detections than when using a computer without a GPU. Thus, it is possible to indicate that GPU use is required in a solution that must process images of a real scenario. Regarding the use of the wireless network emulator Mininet-wifi, the performance did not decrease, guaranteeing the reliability and correctness of the results obtained in the emulated tests in describing (emulating) real environments.

Considering the 96.24 FPS processing rate and cameras sending 30FPS, the system could support the processing of signals from three cameras. Also, the processing rate per camera can be reduced to fit a more significant number of cameras. For example, assuming that a collision lasts 100 milliseconds, our application would need an accuracy of one detection every 0.1 seconds, which corresponds to the processing of 10 images per second. In this case, our detection system could support up to 9 cameras considering that it can process 96.24 FPS using the GPU. In general, each scenario must first analyze its requirements to apply sufficient cameras and detection accuracy.

### 5.2 Experiment 2

The AMQP protocol did not support transmissions with a frame rate higher than 22 FPS, while MQTT reached 30 FPS. This occurred because of AMQP protocol message publishing time restrictions for the image size used. As a result, we limited our comparison to rates below or equal to 22 FPS for the other variables evaluated in this experiment.

The AMQP protocol has a slight advantage in the number of transmitted packets and overhead for the type of messages sent in our test, as shown in Table 5. Nonetheless, there is no significant difference in justifying one of the protocols according to these parameters.

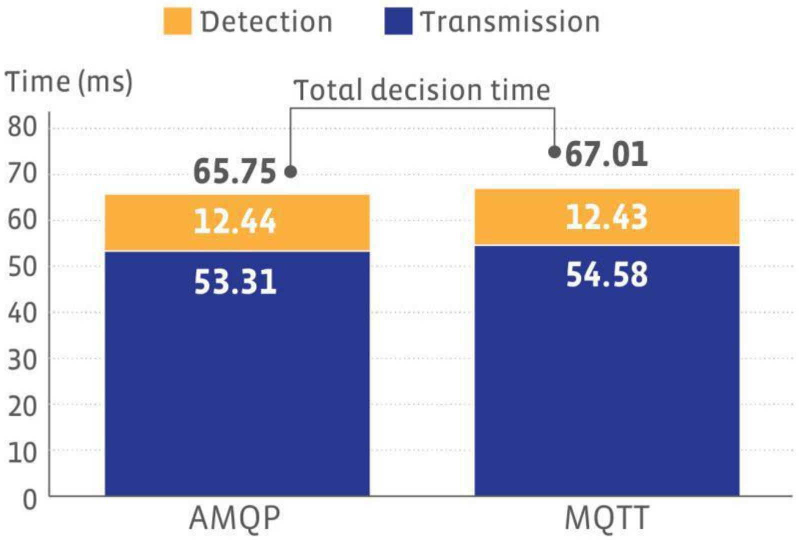
**Table 5.** Number of packets and overhead per frame for each protocol, considering 22 FPS transmission.

| Protocol | Packages/Frame | Overhead/Frame |
|---|---|---|
| AMQP | 103.55 | 5.70 KB |
| MQTT | 104.49 | 5.76 KB |

Fonte: Authors.

Figure 4 shows that the transmission time represents approximately 80% of the total time for decision-making in both protocols considering a network without delays. Considering that the detection time is stable for computers with the same configuration, it is essential to focus on network optimization so that there is no delay in helping potential collision victims. Furthermore, the AMQP protocol exhibits a 5% lower transmission time than MQTT, which can be significant in networks with many cameras and routing layers, such as when using multiple access points.

**Figure 4.** Transmission, detection and decision Times in milliseconds for each protocol, considering a 22 FPS transmission.



Fonte: Authors.

## 6. Conclusion and Future Researches

This paper analyzes several network factors for a collision detector fed with 2D camera images. Firstly, we examined system behavior by observing factors such as the image size, GPU usage, and the Mininet-wifi tool. The collision detector's performance was first evaluated without network influence to serve as a baseline for the rest of our evaluations. Next, we compare the use of AMQP and MQTT message protocols, looking for their strong points and limitations.

The results stress the need for GPU usage, especially when more cameras are planned, which is usually the case when one wants to expand the area of monitoring and improve accuracy. GPU usage made it possible to speed up collision detection by as much as 37 times, regardless of image size. Also, we observed that using the network emulation tool Mininet-wifi and running this jointly with our detection module had no negative impact.

Finally, we identified a limitation of the AMQP protocol in its capacity to transfer images. AMQP supported up to a transfer rate of only 22FPS, whereas MQTT achieved more. However, AMQP reduces transmission time by 5% compared to MQTT. Also, we can see that transmission time represents 80% of total detection time, showing the importance of applying strategies to make image flow more efficient on the network.

As part of future work, we will introduce CoAP and HTTP protocols. We also plan to evaluate the use of more cameras to improve detection accuracy and response time. New models based on other learning architectures will be examined. We will seek to establish a proper balance or middle ground between the accuracy and decision time for each analyzed model.

## Acknowledgments

This work was supported by the Research, Development and Innovation Center, Ericsson Telecommunications Inc., Brazil, the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), and the Fundação de Amparo à Ciência e Tecnologia de Pernambuco (FACEPE).

## References

Breivold, H. P., & Sandstrom, K. (2015). Internet of Things for Industrial Automation -- Challenges and Technical Solutions. 2015 IEEE International Conference on Data Science and Data Intensive Systems. https://doi.org/10.1109/dsdis.2015.11

Embitel. (2018). *7 Most Commonly Used Sensors for Developing Industrial IoT Solutions.* https://www.embitel.com/blog/embedded-blog/7-most-commonly-used-sensors-for-developing-industrial\-iot-solutions.

Fontes, R. R., Afzal, S., Brito, S. H. B., Santos, M. A. S., & Rothenberg, C. E. (2015). Mininet-WiFi: Emulating software-defined wireless networks. 2015 11th International Conference on Network and Service Management (CNSM). https://doi.org/10.1109/cnsm.2015.7367387

Gemirter, C. B., Senturca, C., & Baydere, S. (2021). A Comparative Evaluation of AMQP, MQTT and HTTP Protocols Using Real-Time Public Smart City Data. 2021 6th International Conference on Computer Science and Engineering (UBMK). https://doi.org/10.1109/ubmk52708.2021.9559032

Halder, M., Sheikh, M. N., Rahman, M., Rahman, M. (2018, 07). Performance analysis of coap, 6lowpan and rpl routing protocols of iot using cooja simulator. *International Journal of Scientific and Engineering Research*, *9* , 1671-1677.

Jaloudi, S. (2019). Communication Protocols of an Industrial Internet of Things Environment: A Comparative Study. *Future Internet*, *11*(3). doi:10.3390/fi11030066

Jeong B. Y. (2021). Prevalence of occupational accidents and factors associated with deaths and disabilities in the shipbuilding industry: Comparisons of novice and skilled workers. *Work (Reading, Mass.)*, *69*(3), 997–1005. https://doi.org/10.3233/WOR-213530

Jovanović, J. i Jovanović, M. (2004). Occupational Accidents and Injuries: Results of a Safety Preventive Programme. *Arhiv za higijenu rada i toksikologiju, 55* (4), 261-268. Preuzeto s https://hrcak.srce.hr/254

Matsas, E., Vosniakos, G.-C. (2017). Design of a virtual reality training system for human–robot collaboration in manufacturing tasks. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, *11* (2), 139– 153.

Microsoft, . D. (2019). *2019 Manufacturing Trends Report. Avail- able in:.* https://info.microsoft.com/rs/157-GQE-382/images/EN-US-CNTNT-Report-2019-Manufacturing-Trends.pdf.

Moraes, T., Nogueira, B., Lira, V., & Tavares, E. (2019). Performance Comparison of IoT Communication Protocols. 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC). https://doi.org/10.1109/smc.2019.8914552

Mohammed, A., Schmidt, B., Wang, L. (2017). Active collision avoidance for human–robot collaboration driven by vision sensors. *International Journal of Computer Integrated Manufacturing, 30* (9), 970–980.

Naik, N. (2017). Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. 2017 IEEE International Systems Engineering Symposium (ISSE). https://doi.org/10.1109/syseng.2017.8088251

of Labor Statistics, B. (2021). *Employer-reported Workplace Injuries and Ill- nesses - 2020.* https://www.bls.gov/news.release/pdf/osh.pdf.

Rajak, R., Chattopadhyay, A., Maurya, P. (2021). Accidents and injuries in workers of iron and steel industry in west bengal, india: Prevalence and associated risk factors. *International Journal of Occupational Safety and Ergonomics*, *0* (0), 1-8. https://doi.org/10.1080/ 10803548.2021.2012021 (PMID: 34842067) doi: 10.1080/10803548.2021

.2012021

Silva, I. R. R., Barbosa, G. B. N., Ledebour, C. C. D., Filho, A. T. O., Kelner, J., Sadok, D., & Souza, R. (2020). Assessing Deep Learning Models for Human-Robot Collaboration Collision Detection in Industrial Environments. *Intelligent Systems*, 240–255. 10.1007/978-3-030-61377-8_17

T. Sultana, & K. A. Wahid. (2019). Choice of Application Layer Protocols for Next Generation Video Surveillance Using Internet of Video Things. *IEEE Access*, *7*, 41607–41624. 10.1109/ACCESS.2019.2907525

Thangavel, D., Ma, X., Valera, A.C., Tan, H., & Tan, C.K. (2014). Performance evaluation of MQTT and CoAP via a common middleware. *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 1-6.

Uy, N.Q., & Nam, V.H. (2019). A comparison of AMQP and MQTT protocols for Internet of Things. *2019 6th NAFOSTED Conference on Information and Computer Science (NICS)*, 292-297.

Y. Wang, X. Ye, Y. Yang, & W. Zhang. (2017). *Hand Movement Prediction Based Collision-Free Human-Robot Interaction*. 10.1109/CVPRW.2017.72