

## Research Article

# Real-Time Video Latency Measurement between a Robot and Its Remote Control Station: Causes and Mitigation

**Admir Kaknjo** , **Muzaffar Rao** , **Edin Omerdic** , **Luke Robinson** ,  
**Daniel Toal** , and **Thomas Newe** 

*Centre for Robotics and Intelligent Systems, Department of Electronic and Computer Eng., University of Limerick, Limerick V94T9PX, Ireland*

Correspondence should be addressed to Daniel Toal; [daniel.toal@ul.ie](mailto:daniel.toal@ul.ie) and Thomas Newe; [thomas.newe@ul.ie](mailto:thomas.newe@ul.ie)

Received 27 June 2018; Revised 3 October 2018; Accepted 4 November 2018; Published 2 December 2018

Academic Editor: Pierre-Martin Tardif

Copyright © 2018 Admir Kaknjo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work presents a detailed study, characterization, and measurement of video latency in a real-time video streaming application. The target application consists of an automatic control system in the form of a control station and the mini Remotely Operated Vehicle (ROV) equipped with a camera, which is controllable over local area network (LAN) and the Internet. Control signal transmission and feedback measurements to the operator usually impose real-time constraints on the network channel. Similarly, the video stream, which is required for the normal system control and maneuvering, imposes further strict requirements on the network in terms of bandwidth and latency. Based on these requirements, controlling the system in real time through a standard Internet connection is a challenging task. The measurement of important network parameters like availability, bandwidth, and latency has become mandatory for remotely controlling the system in real time. It is necessary to establish a methodology for the measurement of video and network latency to improve the real-time controllability and safety of the system as such measurement is not possible using existing solutions due to the following reasons: insufficient accuracy, relying on the Internet resources such as generic Network Time Protocol (NTP) servers, inability to obtain one-way delay measurement, and many solutions only having support for web cameras. Here, an efficient, reliable, and cost-effective methodology for the measurement of latency of a video stream over a LAN and the Internet is proposed. A dedicated stratum-1 NTP server is used and the necessary software needed for acquiring and measuring the latency of a video stream from a generic IP camera as well as integration into the existing ROV control software was developed. Here, by using the software and dedicated clock synchronization equipment (NTP server), it was found that normal video latencies in a LAN were in the range of 488ms – 850ms, while latencies over the Internet were measured to be in the range of 558ms – 1211ms. It is important to note that the values were obtained by using a generic (off-the-shelf) IP camera and they represent the actual latencies which might be experienced during control over long range and across international territory borders.

## 1. Introduction

Low latency demanding applications [1–4] have always been an important consideration in telecom networks for voice, video, and data communication. Latency is an inherent property of communication channels regardless of the type, medium, and protocols used for the data transmission. Therefore, it can only be minimized with effort and ingenuity. In recent years, the topics of network latency, its causes, characterization, and reduction methodologies have been investigated by many researchers. Clear distinctions can be made between different types of services based on the requirements

imposed on the network connection. For example, some services demand high bandwidth and high throughput, but latency is not an issue (e.g., browsing the web, downloading files). Other services require little bandwidth but demand a low-latency connection (e.g., VoIP) [5]. On the other hand, there are services which demand high reliability and low delivery latency (e.g., safety applications in vehicular ad hoc networks (VANETs)) [6]. Increased network latency can cause unpredictable behaviour in applications which require real-time network constraints. The number of applications which involve real-time video acquisition and broadcasting over the Internet has increased in recent years [7] and the

specific nature and often low-latency requirements of these applications has led to the establishment of a special branch of the Internet of Things (IoT) called the Internet of Video Things (IoVT) [8, 9].

This work presents the experimental setup and real-time testing results from which extensive measurements of network latency have been acquired with a specific focus on real-time video stream applications. In terms of video, this work focuses on capture-to-display (glass-to-glass) latency, which represents the time that passes from the moment when an event occurs in front of a camera lens to the moment when that event is displayed on a monitor. The monitor can be connected directly to the camera (monitor is in close proximity to the camera and connected with single cable to it), or the monitor can be located anywhere in the world (and connected to the camera over the Internet, not in close proximity). Transfer of the video over the Internet in real time imposes strict demands on the network in terms of latency. The importance of low latency depends on the application in which the video is used. A category of applications where video feedback has great significance in stable functioning of the system is remote presence applications. Such systems often comprise of a human operator located at a control centre and equipment or robots deployed at a remote location. The locations are connected by a dedicated or generic network link over which all information necessary for normal system operation is transmitted. While remote presence systems differ in various purposes and the fields they are deployed in, one common thing they all share is their dependence on the video feedback from the remote plant to the operator. The effect of video latency in remote presence systems has been studied in [11], with telerobotic surgery system as the target application.

In this work, the target application used for the proposed video latency measurement methodology is the remote marine presence system developed by the Centre for Robotics and Intelligent Systems (CRIS) at the University of Limerick [12]. The requirements, which are imposed on the network link, must comply with the constraints for near-real-time control of a marine ROV system used as a remote tool for Inspection, Repair, and Maintenance (IRM) operations on offshore oil and gas subsea structures. The system comprises of the mini ROV and a control station that can be linked together through a local area network (LAN) when they are in close proximity to one another, or the link can be established over the Internet in cases when the operator and the mini ROV are at remote locations to each other. The system in general was designed with the purpose of permanent deployment to offshore oil and gas platforms or energy farms. Such platforms by default have an energy and communications connection to the shore through subsea cables, so a high-bandwidth connection between control stations located onshore and an offshore platform usually exists. The overall idea is to have a remote robotic system permanently deployed offshore, capable of delivering IRM, while the operator/pilot can control the system over the Internet from anywhere in the world. The cost of having such a system permanently integrated to an offshore platform is far less than the losses which can occur due to

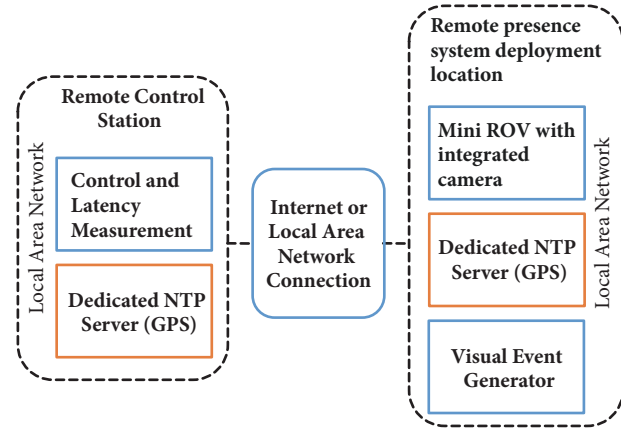


FIGURE 1: Remote presence system architecture with integrated latency measurement system.

production termination in the case of serious equipment failures. The quality of the control over the ROV depends largely on the communication link between the operator and the remote site, which makes it necessary to monitor the network parameters at all times during control. When controlling the mini ROV (locally or remotely), the operator has the ability to input control commands while relying on feedback information from on-board sensors. One of the most valuable feedback sensors for the operator is the video stream, which is normally displayed on one of the operator's screens when operating the ROV. The importance of a high-quality and low-latency video stream is more evident when one knows that almost all pilot decisions, when controlling the remote ROV, are based on video feedback. This fact is one of the major decisions for implementing the video and the network latency methodology discussed here. The overall system high-level architecture with the latency measurement setup included is shown in Figure 1. For the system to operate successfully, it is assumed that the network channel used is a stable communications link, with sufficient bandwidth and minimum latency capabilities to support video streaming.

The system under test (a ROV controlled over the Internet) is asymmetric in terms of requirements, which are imposed on the network. The network traffic needed for successful control contains command signals which go from the operator to the ROV and the feedback signals which are sent from the ROV and presented to the operator. While the command and feedback signals both contain numerical values (sensor measurements, command sequence, etc.), the feedback signal also contains a video stream which is sent from the ROV to the operator. The existence of video in the control loop imposes strict requirements in terms of bandwidth and network latency. By measuring round-trip time (RTT) only, it is not possible to identify the direction of the path which is causing major delays (if such paths are in the system). This is important information in the case where the application depends on performance in one direction. Due to these reasons, it is important to have information about one-way delays (OWD) in the control system. In addition, [13]

points out the importance of OWD compared to round-trip delay. The authors in [13] state the following:

- (i) The paths from source to receiver and from receiver to source may be asymmetric due to different network equipment used on those paths. Independent measurement of each path highlights the difference in performance of the paths which can be caused by different Internet service providers or different types of network.
- (ii) Asymmetric queuing can cause major difference in the performance of otherwise similar paths.
- (iii) Application performance can depend mostly on performance in one direction (TCP-based communication will experience reduced throughput if congestion occurs in one direction).

The remainder of the paper is organised as follows: the related works section details the previously published literature in the field of video latency measurement. The proposed video latency methodology section provides a detailed description of the method used. Procedures used during experiments are presented in three subsections: (a) Web Camera Latency Measurement, (b) Web Camera Latency Measurements by Using a PPS (Pulse Per Second) Signal, and (c) Latency in Network Video Streaming Applications. Results are presented, starting with latency graphs and ending with a discussion of all considered cases. The paper ends with a conclusion in which the results and recommendations are discussed.

## 2. Related Works

There are several articles published on video latency measurements, but to the best of the authors' knowledge, no one has used dedicated NTP servers. For example, in [14], a tool called AvCloak was presented. This tool is intended to be used for various video and audio stream measurements, mainly in conference call applications. This methodology for video delay measurements relies on embedding timestamp information directly into the video stream, in the form of a barcode. This work used a webcam as video source, while the two ends were synchronized to generic NTP servers available on the Internet.

Another tool was presented for video latency measurement, called vDelay [15, 16]. This work introduces the capture-to-display latency (CDL) measurement methodology for use in real-time video chat applications. The CDL measurement is based on embedding the timestamp information into the source video stream and decoding the timestamp on the receiving end. Like in [14], the source and receiving end are assumed to be synchronized to a generic NTP server and as such its accuracy is limited.

A tool named VideoLat was presented in [17, 18]. The main purpose of this tool is to provide latency measurements for video conferencing applications. It provides a way for acquiring glass-to-glass video delays and speaker-to-microphone audio delays. Since this tool is intended for measurement of RTT for video, it does not require any form

of time synchronization. The method of measuring video delay relies on generating a series of QR codes which are displayed in front of the sender's web camera. This image is then transferred to the remote computer (over the Internet or LAN), where it is shown on display. The remote web camera, which is pointed at the remote display, captures the image which is then transferred back to the sender's computer. The sender's computer has the VideoLat web camera pointed to its display. This web camera is used to detect the QR code and notify the VideoLat software that the QR code was received back to the sender's PC and that it should calculate round-trip time (RTT) for that image.

Extensive measurements of latencies induced by various equipment used in Augmented Reality systems are presented in [20]. In terms of video, measurement of end-to-end camera latency was conducted using an LED as a source of visual event and a photo transistor to detect the event. The author stated that measured end-to-end latency was 40ms when using an analog video camera for the test.

In [21], point-to-point latencies are thoroughly analysed and a novel latency reduction algorithm was used. Also, glass-to-glass and glass-to-algorithm latencies are examined in detail. The authors employed a method of using an LED as a source for the visual event and a photo-transistor as the light detector. This research, however, focuses on the sources of delay in the camera itself and in the related equipment (display, PC, etc.), without actual measurement of the delays introduced by the network. The delay measurement methodology was utilized with the purpose of research, so that the author's proposed algorithm for reducing the latency of the video could be tested. The proposed latency reduction methodology resulted in a latency of 21.2ms.

Authors in [22] provide an extensive overview of the video delays in various applications such as video conferencing, video transmission for teleoperation, smartphones, and low-delay video communication prototypes. Two different teleoperation systems in the form of off-the-shelf drones were used, among other applications. While one system demonstrated a mean latency value of 254ms, the latencies in another system ranged from 28.33ms (for an analog camera) to 57.35ms (for a digital camera) using an uncompressed video stream. It should be noted that measurements on these systems were done locally, not over the Internet.

Results presented in this work represent a logical extension to the results which were published previously in [23]. Here, the results dealt only with network latencies in the control and feedback data, while this paper provides an extensive overview of video delays in video feedback used for marine remote presence applications.

To the best of our knowledge, none of work published in previous articles utilised dedicated time synchronization equipment for their tests in a way that would enable independent measurement of one-way delay (OWD) for both directions of data flow. Also, applying the video latency measurement methodology to the remote presence marine application with a mini ROV and a remote-control station is unique. In addition, our methodology enables us to acquire measurements of one-way time delays between network nodes (PCs and laptops used for testing will be referred to as

nodes in further text, since all pieces of equipment represent nodes on the network) whose clocks are synchronized with an accuracy of up to 1ms [24]. The presented methodology can be utilised in order to obtain measurements for video streaming applications that are used in other domains in addition to marine. Moreover, availability of latency measurements in real time is a feature that can be used as an additional metric for network Quality of Service measurements. A few interesting approaches for improvement network resilience and decreasing delays are presented below.

Resilience represents the ability to recover or adopt to a change or disruptive event, or the capability to maintain functionality in case of failure of some of the components [25, 26]. In [27], a video architecture was proposed which would improve both error resilience and video delay. The technique implied introducing proxy servers near the locations of the end nodes on the network. The proxy would serve as a buffer for the video content, so that, in case of an error, the receiving node would not have to send retransmission requests to the remote source node, but instead to a proxy server which is located much closer. This technique is most effective in wireless networks used for video communications. In [28], the service resilience is improved by utilizing redundant paths between two nodes on the network which are found in advance. In this case, the time needed to reestablish a connection between two nodes depends on the length of the backup paths. Two resilience mechanisms were also presented in [29]. These are called resilient routing layers and multiple routing configuration and are based on multi-topology routing and stub routers. In [30], the fog computing architecture is presented as a solution for decreasing network traffic and delay and for increasing network resilience in the Internet of Things. This fog is defined as a horizontal virtualized layer located between the edge networks and the cloud. It is seen as provider of computing, storage, and networking services to edge devices. The proximity of the fog architecture to end users decreases latency and response time. It can be seen that improving the resilience of the network is accomplished by increasing redundancy in the network architecture (backup paths, servers, routers, etc.). Ways to decrease delays mostly rely on introducing additional layers in the network architecture (e.g., fog architecture in IoT).

### 3. Proposed Video Latency Measurement Methodology

One definition of video latency states that it is the period of time needed for events that occur in front of a camera sensor to appear on the operator's monitor. This delay is also denoted as glass-to-glass delay (G2G) [31]. Since the remote controlled mini ROV system involves a human operator in the loop, this type of latency has a huge impact on system operation. In addition, the total amount of video latency can be divided into several components that contribute to latency such as video camera encoding, video decoding, network transmission, and screen delay.

The latency in a targeted application is measured by implementing a controlled scene which will be observed by the camera. Most of the time, the controlled scene consists

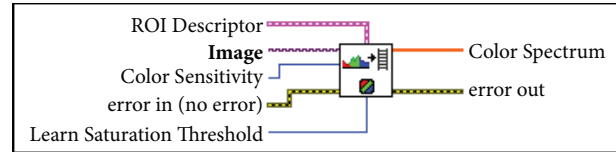


FIGURE 2: IMAQ ColorLearn VI [10].

of an application that runs on a computer and is shown on a screen. The application normally shows a sequence of events, which are programmed to occur at precisely defined times (e.g., an image container periodically changing colour). At the same time, another application accepts a stream from a camera pointed at the screen with the timed change events and detects the exact time of every change of visual event. The latency is calculated as the time difference between the application changing the scene (i.e., the colour of the image container) and the time when that change was detected in a video streamed from the camera.

All applications used for the tests were developed in LabVIEW (Laboratory Virtual Instrument Engineering Workbench). All third-party libraries were also utilized from LabVIEW. There are two main applications (Figure 3) which run in all tests. The first one is called Timestamp Sender; this application is responsible for generating visual events and sending initial timestamps. The timestamps are taken as windows system time with a resolution of 1ms. The second application is called Latency Measurement; this application can connect to an IP/web camera, either directly or over the Video Server. This application will accept and decode the stream, perform image processing to detect the visual event, display the video to the user, and calculate latency time. Here, all the steps in the application's loop will be briefly explained.

For manipulations with the video stream, an open-source library called FFmpeg was used. FFmpeg is a leading framework generally used for the various multimedia operations including but not limited to decoding, encoding, transcoding, filtering, and streaming both video and audio. Among all the operations, which the library is capable of performing, the Latency Measurement application uses only a subset of functions for decoding the video stream which are contained in specific library called libavcodec [32]. For the purpose of the tests, FFmpeg version 4.0 built with gcc 7.3.0 (GCC) was used.

Once the video frames are decoded, they are fed to the software component, which is responsible for handling image processing and detection of colour features of an image. A part of the colour processing palette of NI Vision Development Module (IMAQ ColorLearn VI) was used for the processing. The abbreviation "VI" stands for "Virtual Instrument," which is at the same time one of the extensions used by LabVIEW. The output of this VI is a colour spectrum which contains the colour features found in the image, as shown in Figure 2.

The input parameters to the VI are described in Table 1. Parameters "error in" and "error out" are used for error handling and they are not stated in the table.

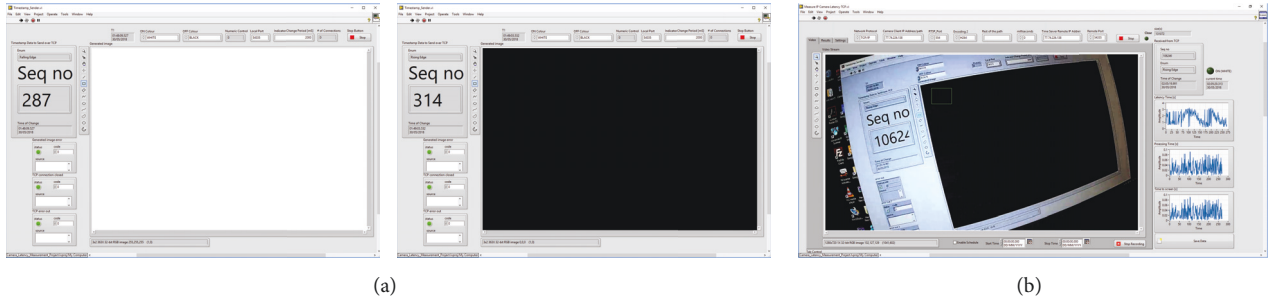


FIGURE 3: Applications used for detecting video latency: (a) Timestamp Sender with white and black indicator and (b) Latency Measurement application.

TABLE 1: Image processing VI input parameters description and values used during the tests [19].

| Parameter Name             | Description  | Value during experiments  |
|----------------------------|--|---|
| ROI Descriptor             | Region in the image which contains colour for inspection           | Region of the image defined as rectangle which contains colours of interest.  |
| Image                      | Reference to the image which was decoded and needs processing      | Every decoded frame   |
| Colour Sensitivity         | Sensitivity of the colour information in the image                 | This input determines a number of bins in hue colour space, so that “low” gives 16 bins and “medium” provides 30 bins, while “high” gives 58 bins [19].<br>Value was set to “low” in experiments. |
| Learn Saturation Threshold | Threshold value to distinguish two colours with the same hue value | The default value of 80 was used.   |

Colour spectrum is an array of  $(n+2)$  elements, where  $n$  is the number of bins in the colour space (hue colour space is used) and it is denoted by the “colour sensitivity” input. The last two elements in colour spectrum represent black and white colour, respectively, and these two elements of the array were used in the LabVIEW applications.

The Video Server is a software component, which consists of two applications called Camera Server and Camera Client. These two applications are used for acquiring information about the latency introduced by the network transmission. The flow of data in the case when Video Server is used is as follows:

- (1) Camera Server opens a connection to the IP camera.
- (2) Camera Client opens a connection to the Camera Server.
- (3) Latency Measurement application opens a connection to the Camera Client in order to start video streaming and start the test.
- (4) Camera Server generates a timestamp and appends it to each TCP/UDP frame.
- (5) Camera Client unpacks each TCP/UDP frame and calculates the time needed for the frame to travel from Camera Server to Camera Client.

The experiment setup, together with data flow from the camera to the Internet Control Station, is depicted in Figure 4: The procedure which was followed when executing the tests is shown as step 1 to step 10 in Figure 4.

TABLE 2: Configuration of tests with web camera.

| Test No. | Event Source | FPS | Screen Refresh Frequency [Hz] |
|----------|--------------|-----|-------------------------------|
| Test 1   | Screen       | 5   | 60                            |
| Test 2   | Screen       | 25  | 60                            |
| Test 3   | PPS LED      | 25  | N/A                           |

The capture-to-display latency represents the overall latency value of the video streaming application. If the application is running over the Internet, then this latency value represents the sum of all delay times which occurred between the time when the scene was observed by a camera and the time when that exact scene appeared on the remote screen, including the network transmission delay times. To measure typical capture-to-display latency of a camera, two configurations (with node and integrated web camera) were tested (Tests 1 and 2 in a configuration with a screen and Test 3 in a configuration with a PPS LED diode; see Table 2). The web camera used in the tests is an integrated webcam of a Dell Inspiron 7720, with 0.92M (megapixel) and maximum resolution of 1280x720 pixels. By using an integrated web camera, it is possible to eliminate the effects of network transmission on the latency of the video, since the video stream is not transferred over the network to a remote device but is instead handled locally on a PC.

**3.1. Web Camera Latency Measurements.** First measurements were captured using the setup shown in Figure 5 ( $T_2$  in the image represents the time of the visual event while  $T_1$  is the

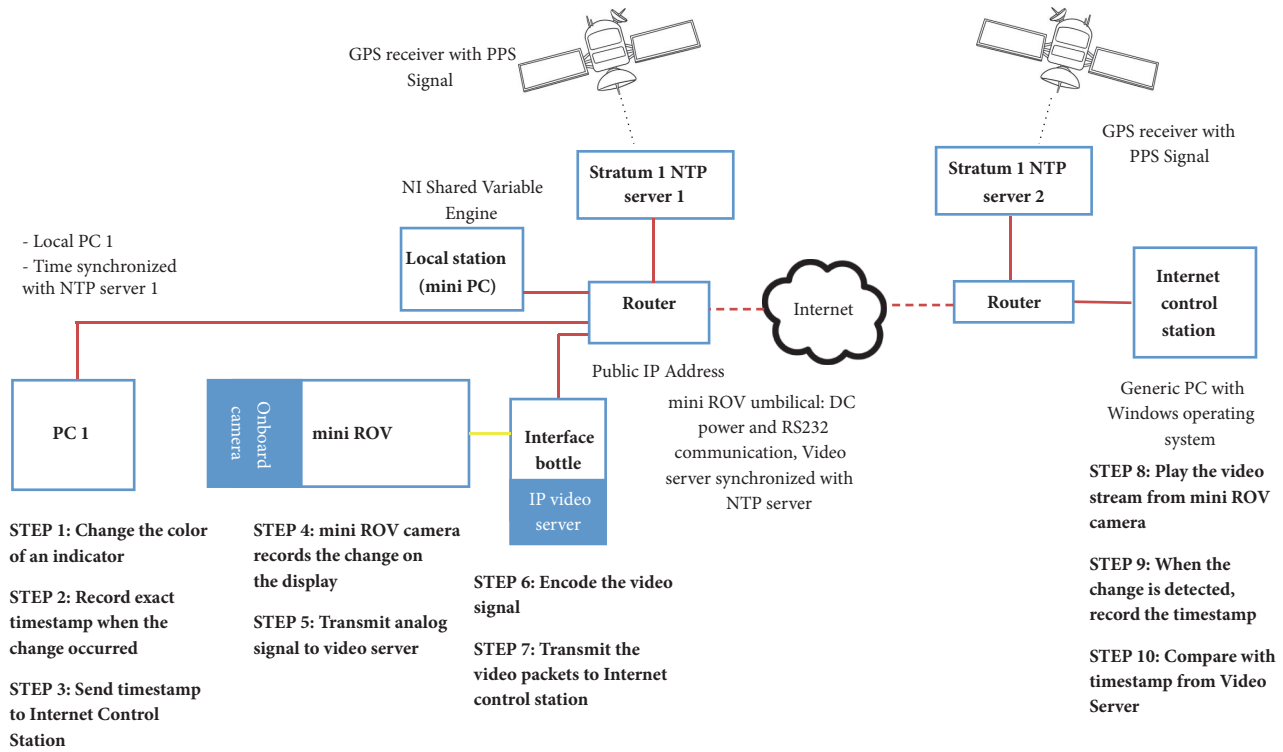


FIGURE 4: Target system hardware architecture.

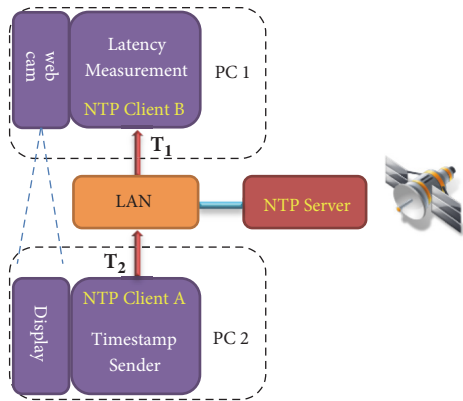


FIGURE 5: Capture-to-display video latency measurement in configuration with web cam and additional screen.

time when the event was detected by the Latency Measurement application). This method of capture-to-display delay is affected by parameters such as screen refresh rate and web camera fps (frames per second). The nodes which were used for tests both have a screen refresh rate of 60Hz. On the other hand, the integrated web cameras have a frames-per-second value that is configurable within the range of 1 – 25fps.

The overall accuracy can be derived as follows.

Display refresh period is as follows:

$$f_D = 60\text{Hz},$$

$$T_D = \frac{1}{f_D} = \frac{1}{60\text{Hz}} = 16.667\text{ms}. \quad (1)$$

Camera fps period is as follows:

$$f_{C25} = 25\text{Hz},$$

$$T_{C25} = \frac{1}{f_{C25}} = \frac{1}{25\text{Hz}} = 40.00\text{ms}. \quad (2)$$

Measurement error is as follows:

$$\Delta T_{25} = T_{C25} + T_D = 40.00\text{ms} + 16.667\text{ms} = 56.667\text{ms}. \quad (3)$$

Since the camera acquisition time and event generation are not synchronized to the screen refresh rate, the value from (3) is the measurement error which can occur in the worst-case scenario. The error value when the camera's frames-per-second value is set too low, i.e., 5fps, has an even higher error value of  $\Delta T_5 = 216.67\text{ms}$ .

Values  $\Delta T_{25}$  and  $\Delta T_{10}$  are worst-case scenario errors and represent the impact of equipment delays on measurement values. In addition to these values, there are also image processing and image display times on the second node. The remainder of the time is taken up by buffering and memory management routines. However, since both the screen and IP camera are part of the system under test, reducing the variance caused by them is not included in this work.

### 3.2. Web Camera Latency Measurements by Using PPS Signal.

To eliminate the effect of screen refresh rate on the measurements, an alternative approach was used for measuring capture-to-display latency. In this case, an NTP server was used for generating a visual event which was detectable by a web camera. The event is generated by an LED which is

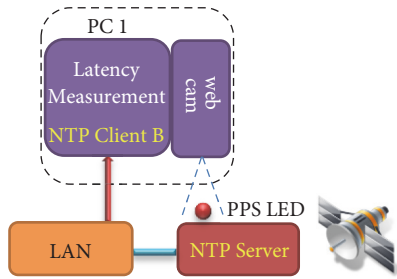


FIGURE 6: Test configuration with web camera and PPS LED from NTP server.

connected to the source of a PPS (pulse per second) signal from a GPS receiver (Figure 6). The GPS receiver used during the test is a Quectel L80 GPS module with an external antenna. The PPS signal is a 100ms pulse and occurs every second with an accuracy in the range of 10ns [33] and is visible as a short blink of the LED. The PPS signal accuracy was not measured since such measurement is not part of the scope of this paper and is taken from the product's datasheet. This pulse effectively signals the NTP server that the most recent GPS second has become effective.

By using the LED as a visual event source, the effect of screen refresh rate is removed from the measurements and the maximum error now depends solely on the camera frames-per-second value. For a frame per second value of  $f_{C25} = 25\text{Hz}$ , the maximum possible error will be  $\Delta T_{25} = T_{C30} = 40\text{ms}$ .

This approach with the PPS LED is not used in the rest of experiments, despite the fact that it represents a way to eliminate inaccuracy caused by the screen refresh rate. One reason for this is that using the PPS LED requires a more complicated setup than the one in which the display is used. Not all NTP servers have a PPS output or LED connected. The second reason was that the sole purpose of the tests is to measure latency which can be experienced by an operator who is looking at the display and controlling the remote robot mostly based on the video feedback, i.e., user-perceived latency. Previous studies [11] showed that just-noticeable difference (JND, defined as the change in the intensity of a stimulus needed for humans to perceive a difference) for video latency applications was measured to be approximately 15ms.

**3.3. Latency in Network Video Streaming Applications.** For video latency measurement over the Internet and in a LAN environment, a generic IP camera was used alongside the NTP servers and nodes. To get a thorough insight into the main causes of video latency, several tests were conducted with different camera settings, connection types, and underlying transmission protocols as well as different network types (LAN or Internet).

The tests were conducted using a Hikvision DS-2CD2010F-I IP camera. Encoding parameters set in the camera are as follows: resolution (640x6480), bitrate type (variable), video quality (lowest), frame rate (25), and max bitrate (2048). The decoding of the video was conducted

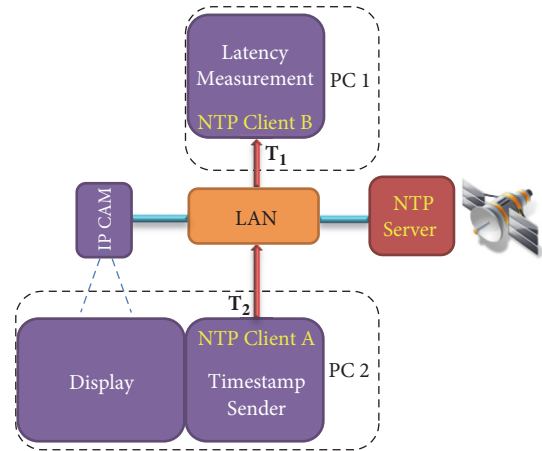


FIGURE 7: Equipment configuration in IP camera latency test in a LAN.

on a laptop PC using the FFmpeg library. The PC is a Dell Inspiron 7720 with an Intel Core i7-3630QM @ 2.4GHz Processor and an NVIDIA GeForce GT 650M graphics card.

In the IP camera configuration, the encoding algorithm used was investigated. Two encoding algorithms were used during the tests: H.264 and MJPEG [34, 35]. In all tests, the Real Time Streaming Protocol (RTSP) [36] was used for control and transmission of the stream. This protocol can be used with two types of network protocols, UDP and TCP. The connection type differs in terms of the existence of intermediary nodes between the IP camera and the application used for receiving the stream, which was introduced for the purpose of the tests. If there are no intermediary network nodes between the application which receives the stream and the IP camera, the connection type is called direct, while in another case additional Video Server applications were used for testing. In terms of network type, tests were made with all devices in a LAN as well as over the Internet.

The equipment configurations which were used during the tests are depicted in Figures 7 and 8. In these configurations, the IP camera is pointed to a PC screen which is running an application that generates a periodic change in the camera's field of view with a fixed frequency.

When the application connects directly to the camera, it can only measure the total delay, i.e., capture-to-screen delay. However, to distinguish between the delays caused by network transmission and delays caused by other contributors, the latency measurement was implemented using a Video Server connection type. The main purpose of the Video Server connection type was to measure transport delay. This involves the time that passes from the point when a frame transfer is started over a packet switching network to the point when the frame arrives at the receiving end. It is mandatory to have a single reference clock for all equipment which is involved in the experiment. The Video Server test equipment and the software configuration used are shown in Figure 9.

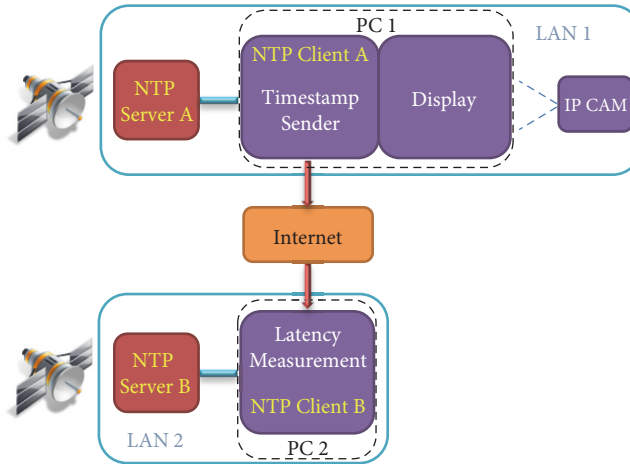


FIGURE 8: Equipment configuration in IP camera latency test over the Internet.

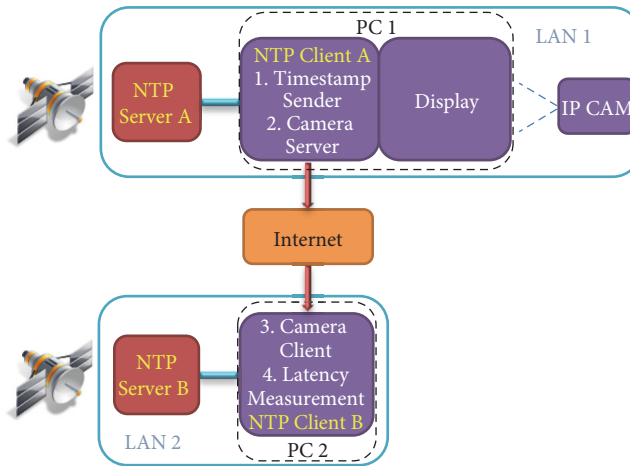


FIGURE 9: Equipment and software configuration in IP camera latency test with Video Server over the Internet.

To measure the transport delay of the stream, it was necessary to implement intermediate nodes between the camera and the receiving node. These additional nodes are represented in the form of two applications, which are acting as video stream server/client. The server application opens a connection to the camera and initiates a video stream (Camera Server in Figure 9). This application parses the stream, generates the timestamp at this moment of time, injects this timestamp information into ethernet packets containing the key frames, and forwards the stream to the client application. The timestamp value is taken as the windows system time with millisecond resolution. The client application (Camera Client in Figure 9) which is normally running on a separate node reads the stream sent by the Server application. The stream is accepted and timestamp information extracted from the data. The transport delay is then calculated as the difference between the two timestamps. As the timestamps are both synchronized to the same reference clock by the NTP, the accuracy of one compared to the other should be less than 1ms. This means that the accuracy of the measurement itself is also less than 1ms.

## 4. Results and Discussion

**4.1. Video Latency Measurement Overhead.** The following sections deal with latency measurement which includes the processing time of the video stream in order to detect a visual event in front of the camera. To measure video latency which is not influenced by video processing time, it is necessary to measure the time the application needs to process the frames and output the result. These values are then subtracted from the total video latency time in order to get accurate measurements. Here, the time of the visual event in front of the camera is denoted as  $T_1$  and the time when the event was detected on the receiving end as  $T_2$ . The start of frame processing is denoted as  $T_3$ . The overall video latency is then calculated as follows:

$$T_{VL} = (T_2 - T_1) - (T_2 - T_3) = T_3 - T_1. \quad (4)$$

Video processing times were logged for every test and the overall latency was then calculated using (4). Figures 10 and 11 show video processing times for test T2\_GI. Video processing measurements in milliseconds are as follows: arithmetic



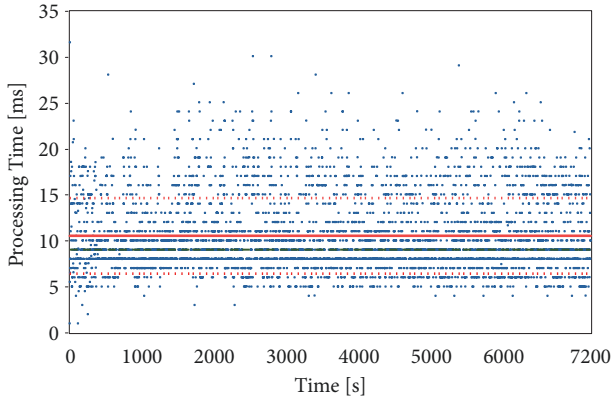


FIGURE 10: Video processing time.

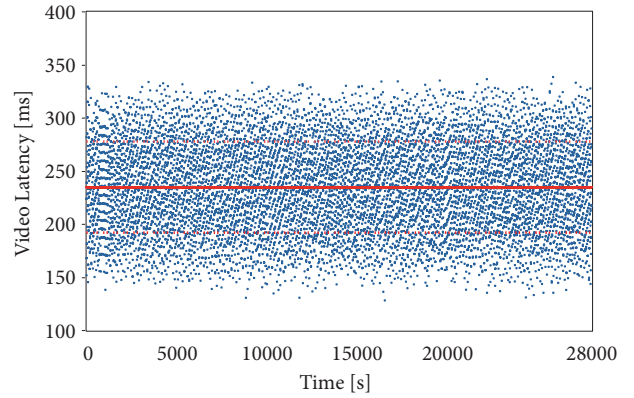


FIGURE 12: Web camera latency (Test 1).

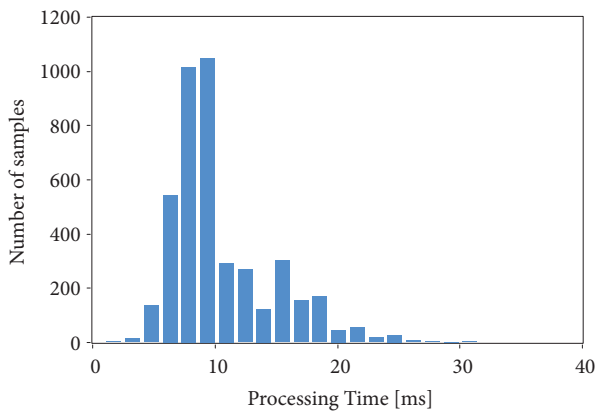


FIGURE 11: Processing times histogram.

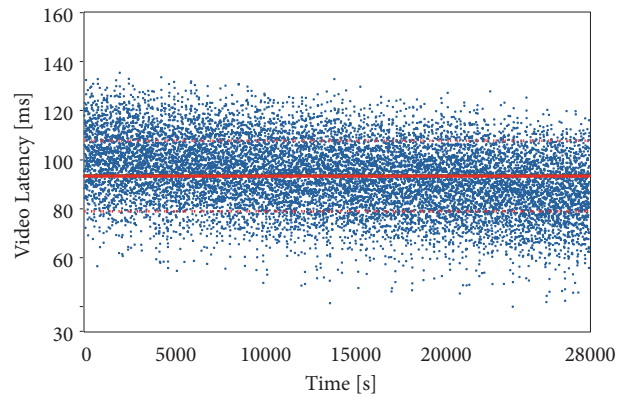


FIGURE 13: Web camera latency (Test 2).

mean (10.52), median (4.11), standard deviation (9.03), min value (1.00), and max value (31.59).

**4.2. Capture-to-Display Latency Using a Web Camera.** To acquire capture-to-display latencies, several tests were conducted using an integrated web camera as the capturing device. In terms of a visual event source, there were two possible configurations: (a) when another PC screen was used for generating an event detectable by the camera and (b) when the NTP server’s PPS signal LED was used for the same purpose.

In the configuration of (a), one of the nodes served as a source of the visual event and timestamp, while the other node was running an application for detecting the visual event and calculating the total latency. The node clocks were synchronized to the stratum-1 NTP server and connected in the LAN. The setup is depicted in Figure 5 to show the effects of parameters such as screen refresh period or camera frames per second on acquired measurements. Two tests were conducted with different fps settings on the camera (Tests 1 and 2). Results of the tests using the web camera are shown in Table 3.

As expected, the lowest latencies are recorded in Test 3, with the web camera and PPS LED signal as the source of visual event. This test represents the case where only the fps

parameter of the web camera is affecting the measurement. Test 2 gives midrange latencies where the screen refresh frequency of the visual event generation node influences the measurement, in addition to the camera’s fps parameter. Test 1 represents the worst-case scenario, where the camera’s fps value was set to 10 and the visual event was generated on a screen with a refresh rate of 60Hz. The results of Table 3 are depicted in Figures 12–15.

Each latency plot shows calculated mean value (solid red line) together with mean + standard deviation, mean – standard deviation (dashed red lines), and median value (dashed green line). Due to the fact that the median and mean values are close to equal in most of the tests, the green line is not visible in all graphs.

The results of test 3 plot resembles a reversed sawtooth pattern; i.e., latency slowly decreases to some value and then there is a sharp increase after which the signal decreases again. The cause of the sawtooth pattern lies in camera’s clock low accuracy when compared to the accuracy of the PPS signal. Since the PPS is a pulse signal with a high-level duration of 100ms, it is clear that the camera with 25 fps will capture at most 3 frames with the PPS LED in the ON state. The number of frames in which the PPS LED is captured in the ON state depends on synchronization of the PPS pulse signal and the camera’s frame capture period. If the camera’s capture timer was an accurate 25fps, the latency graph would

TABLE 3: Web camera video latencies.

| Test No. | Arithmetic Mean [ms] | Median [ms] | Standard Deviation [ms] | Min Value [ms] | Max Value [ms] |
|----------|----------------------|-------------|-------------------------|----------------|----------------|
| Test 1   | 234.71               | 234.73      | 42.89                   | 128.3          | 338.32         |
| Test 2   | 93.24                | 93.58       | 14.39                   | 39.91          | 135.35         |
| Test 3   | 34.77                | 34.78       | 10.67                   | 10.14          | 59.54          |

TABLE 4: Overview of existing video latency measurement tools.

| Tool/Author.       | Industry Field/Purpose | OWD/RTT    | Synchronization      | Minimum video latency |
|--------------------|------------------------|------------|----------------------|-----------------------|
| Jansen [20]        | Augmented Reality      | OWD        | N/A                  | 40ms                  |
| vDelay [15, 16]    | Video Conference       | OWD        | Generic NTP          | 69ms-343ms*           |
| AvCloak [14]       | Video Conference       | OWD        | Generic NTP          | 50ms-120ms*           |
| VideoLat [17, 18]  | Video Conference       | RTT        | N/A                  | N/A                   |
| Bachhuber [21, 22] | Research               | OWD        | N/A                  | 19.18ms               |
| <b>This work</b>   | <b>Control Systems</b> | <b>OWD</b> | <b>Dedicated NTP</b> | <b>34.77ms</b>        |

\*Tests are not directly comparable to other results. In the minimum video latency column, minimum refers to the best achieved result in a group of tests. The values presented here are effectively mean values but are the lowest compared to the mean values obtained in other tests.

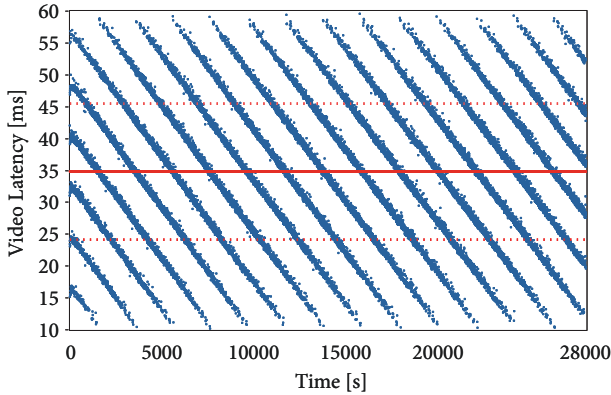


FIGURE 14: Web camera latency (Test 3).

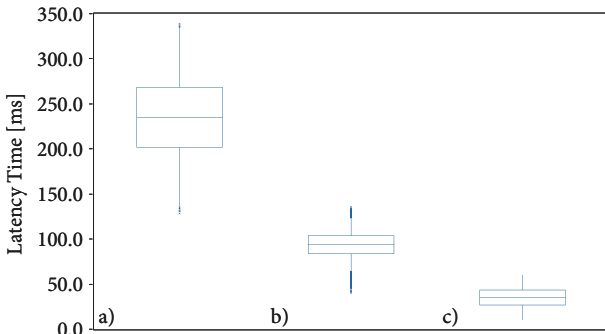


FIGURE 15: Box plots for webcam measurements.

show straight line since the camera would always capture the LED in the ON state at the same time relative to the PPS signal edge. However, if the fps is less than 25, the exact time at which camera captures the PPS LED in the ON state will move relative to the PPS signal, which will manifest in a sawtooth signal shown in Figure 14. Similar pattern was presented in [37].

To show the importance of precise clock synchronization when measuring the latencies, a simple test was made when the applications for latency measurement were started before the nodes were precisely synchronized to NTP source clock. This case is shown in Figure 16.

At the 55<sup>th</sup> second in Figure 15, time synchronization was performed which is shown as a falling edge with an amplitude of 350ms in the video latency measurement. This shows us that the error in latency prior to clock synchronization was 350ms.

**4.3. Comparison with Previous Results.** Table 4 provides an overview of some existing latency measurement tools intended for measuring video latency, together with results that were obtained using the tools. To compare the results from previous work to the results in this work, minimum video latency was measured using a web camera (digital output). It can be seen that our result is closest to [20], where the achieved video latency is 40ms, although a camera with analog output was used in this work. Bachhuber [21], on the other hand, presented a new methodology for video latency reduction and his result is approximately two times lower. vDelay [15, 16] published results of tests conducted over the Internet and are not directly comparable to the rest of the listed results. The same applies for the tool Avcloak [14]. Results for Avcloak, listed in Table 1, are approximate and are taken from a graph, since the author pointed out that an exact latency value is not as important for his research as confirming that the tool is working properly.

**4.4. Latency Testing Results in Network Streaming Application.** Each test has four settings: transport protocol (TCP/UDP), encoding algorithm (H.264/MJPEG), connection type (Direct/Video Server), and network type (LAN/Internet). Tests are divided into four groups (I-IV) based on the encoding algorithm and transport layer protocol used during the test. The test names are formed by joining the test number and group number (e.g., T1\_GI marks the first test in group I). Numeric values for mean and median latencies, standard

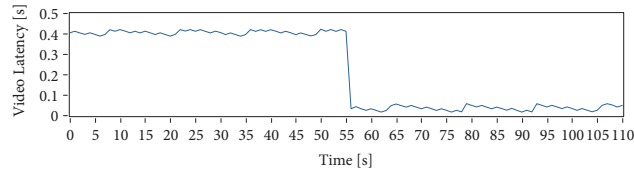


FIGURE 16: Computer clock synchronization.

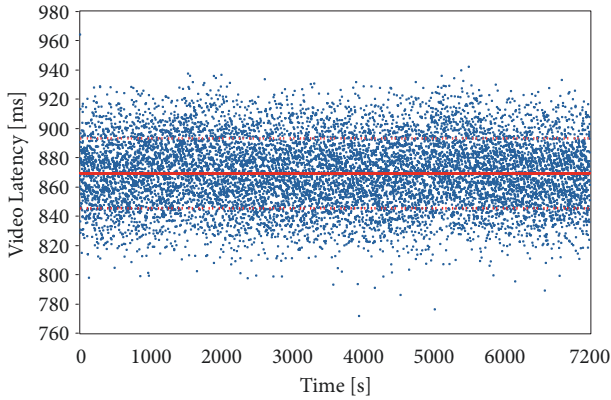


FIGURE 17: T1\_GI video latency.

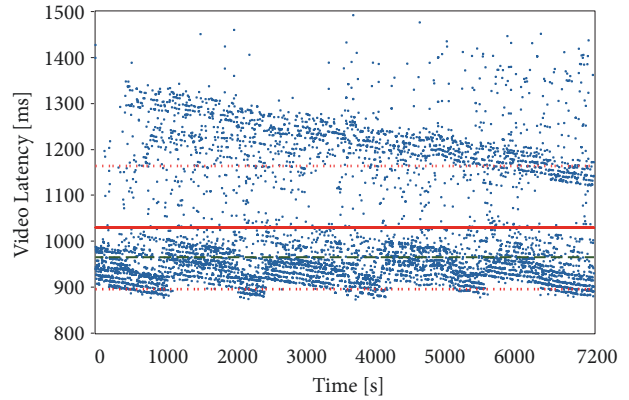


FIGURE 18: T2\_GI video latency.

deviation, and minimum and maximum latencies are listed in Table 9, while graphs are shown for the most representative tests only.

4.5. *Group I: H.264 Encoding and TCP Protocol.* Tests which pertain to Group I are listed in Table 5. The first test (T1\_GI) was conducted by using the TCP protocol with the camera encoding configured to the H.264 algorithm. Figure 17 depicts the results of T1\_GI conducted with all devices connected through a LAN.

Two other tests from the same group were conducted between two remote locations (Sarajevo, Bosnia, and Limerick, Ireland, and vice versa). When compared to the LAN tests, it is noticeable that latency values as well as jitter are much higher in the tests conducted over the Internet. Figure 18 depicts results obtained in one of these experiments where the Latency Measurement application was running on a node in Sarajevo and the camera was in Limerick. T3\_GI was conducted by streaming in another direction (the Latency Measurement application was running on a node in Limerick while the camera was located in Sarajevo). Increased latency in the tests conducted over the Internet is attributed to the distance between the locations, which introduces higher network transmission latency, and to the properties and functionality of the TCP protocol as well. An increase in latency variation is also visible in the Internet test, which is presented in Figure 19. The difference in latency measurement between T2\_GI (mean value: 1029.13ms) and T3\_GI (mean value: 1140.52ms) is caused by the difference in available upload bandwidths from Limerick and Sarajevo. The Limerick site was equipped with an Internet connection which was capable of

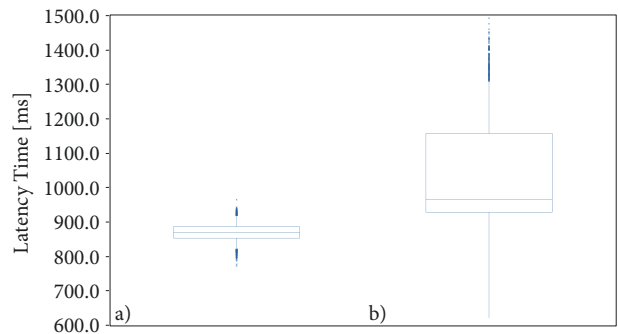


FIGURE 19: Box plots for (a) T1\_GI and (b) T2\_GI.

providing 100Mbps upload, while Sarajevo could only handle 2Mbps.

Video is a type of service which will try to take as much bandwidth as it needs to transfer the stream. Bandwidth has an indirect influence on the latency of network packets. Insufficient bandwidth can cause network packet queueing on the stream source side, which manifests as increased latency in the video. The effects of packet queueing on network QoS is an active field of research and more comprehensive insight into the effects of queueing is given in [38].

4.6. *Group II: MJPEG Encoding and TCP Protocol.* The second group of latency measurement tests deals with MJPEG stream encoding with TCP as the transport layer protocol. Tests from this group are listed in Table 6. Figure 20 depicts latency results obtained in the test where the Latency Measurement application was directly connected to the camera on a LAN. The difference between T1\_GII and T1\_GI is the

TABLE 5: Group I Test Summary.

| Test No. | Protocol | Encoding | Connection Type | Network type | Source   |
|----------|----------|----------|-----------------|--------------|----------|
| T1_GI    | TCP      | H.264    | Direct          | LAN          | N/A      |
| T2_GI    | TCP      | H.264    | Direct          | Internet     | Limerick |
| T3_GI    | TCP      | H.264    | Direct          | Internet     | Sarajevo |

TABLE 6: Group II Test Summary.

| Test No. | Protocol | Encoding | Connection Type | Network type | Source   |
|----------|----------|----------|-----------------|--------------|----------|
| T1_GII   | TCP      | MJPEG    | Direct          | LAN          | N/A      |
| T2_GII   | TCP      | MJPEG    | Direct          | Internet     | Limerick |
| T3_GII   | TCP      | MJPEG    | Direct          | Internet     | Sarajevo |

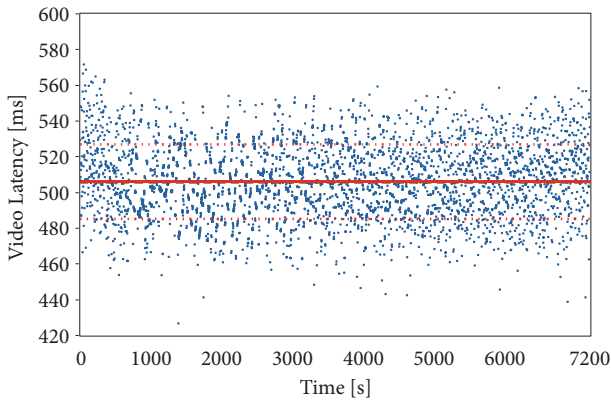


FIGURE 20: T1\_GII video latency.

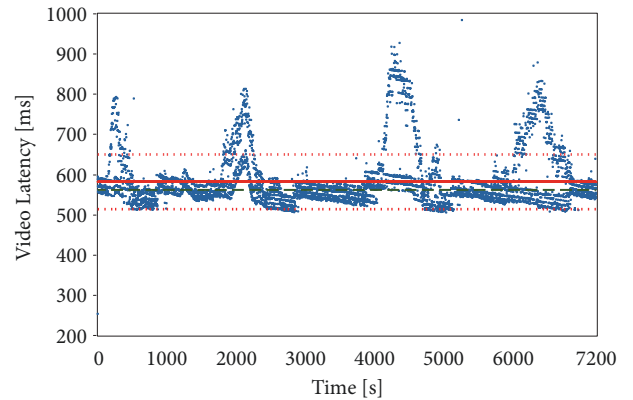


FIGURE 21: T2\_GII video latency.

encoding algorithm and it can be seen that the latency in T1\_GII is much lower when compared to the latency from T1\_GI. The H.264 encoding is complex in terms of computation and time needed for its processing which increases the overall latency of the video, while MJPEG does not require such intensive data processing [39]. MJPEG, on the other hand, requires bandwidth of several orders of magnitude higher (10-100 times higher, as will be shown in the section on bandwidth analysis) for transferring the video. The main difference between MJPEG and H.264 is in the fact that H.264 uses the concept of group of pictures (GOP) with each group divided into different frames, more specifically to intraframe (I-frame) and interframes (P and B frames). An I-frame is the leading frame in each GOP, followed by the P and B frames. The P frame depends on data from the I-frame, while the I-frame does not depend on data in preceding or following frames. The I-frame contains the core image data for a particular GOP, while P and B frames contain only information about changes in the scene relative to the I-frame. This lowers the amount of the data needed to be transferred. A lower amount of the data results in lowering the overall bandwidth requirement for transferring the H.264 encoded video [40]. In comparison to MJPEG, the H.264 provides a better compression rate as well as a smooth transition between frames. The MJPEG frames have no interframe correlation; each frame is represented with an independent JPEG image. This is the reason why MJPEG is

ideal for videos with lots of movements on the scene (e.g., videos taken in emergencies, sports, or action movies) [41].

T2\_GII and T3\_GII were conducted over the Internet. In T2\_GII, the Latency Measurement application was running on a node in Sarajevo and the camera was located in Limerick. When we compare the results shown in Figures 21 and 22 with the results obtained in T1\_GII, it is noticeable that there was an increase in video latency (by 75.58ms and 110.62ms, resp.). The difference between the Internet and LAN test results can also be observed in Figure 23 which shows that the values from the Internet tests are distributed across a wider range of values (plots b and c) when compared to the LAN test (plot a).

**4.7. Group III: H.264 Encoding and UDP Protocol.** Tests from this group are conducted using UDP as the transport layer protocol and H.264 as the encoding algorithm (see Table 7 for the list of the tests). By comparing results of tests from groups III and I, conclusions can be made about the influence of transport layer protocol on overall video latency. It is noticeable that the latency is slightly lower in tests with UDP protocol (group III) than in tests with TCP protocol (group I) i.e., 849.93ms in Test 9 vs. 869.12ms in Test 1.

Test 9 was conducted in a LAN, while T2\_GIII and T3\_GIII were conducted over the Internet. The video stream in LAN test was stable with low latency variation (jitter) which is visible in Figure 24. The first Internet test from this

TABLE 7: Group III Test Summary.

| Test No. | Protocol | Encoding | Connection Type | Network type | Source   |
|----------|----------|----------|-----------------|--------------|----------|
| T1_GIII  | UDP      | H.264    | Direct          | LAN          | N/A      |
| T2_GIII  | UDP      | H.264    | Direct          | Internet     | Limerick |
| T3_GIII  | UDP      | H.264    | Direct          | Internet     | Sarajevo |

TABLE 8: Group IV Test Summary.

| Test No. | Protocol | Encoding | Connection Type | Network type | Source   |
|----------|----------|----------|-----------------|--------------|----------|
| T1_GIV   | UDP      | MJPEG    | Direct          | LAN          | N/A      |
| T2_GIV   | UDP      | MJPEG    | Direct          | Internet     | Limerick |
| T3_GIV   | UDP      | MJPEG    | Direct          | Internet     | Sarajevo |

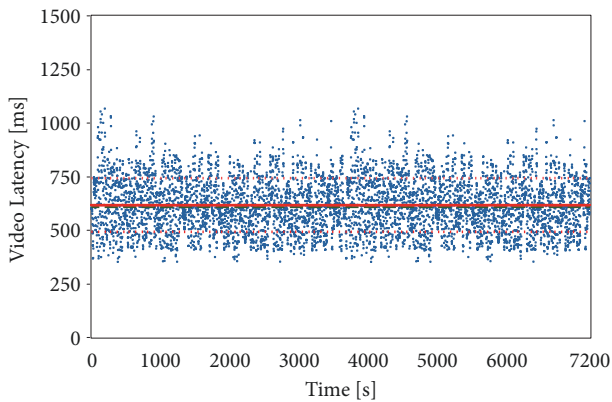


FIGURE 22: T3\_GII video latency.

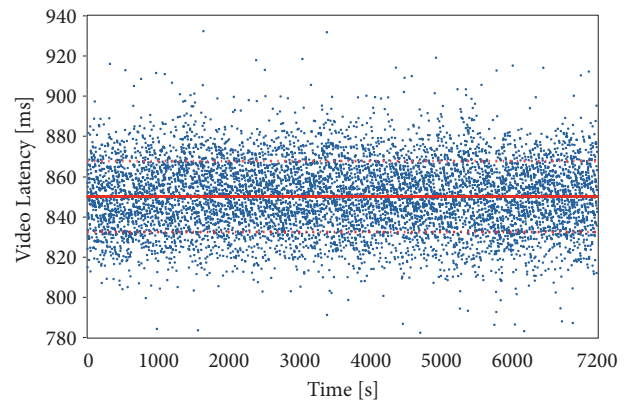


FIGURE 24: T1\_GIII video Latency.

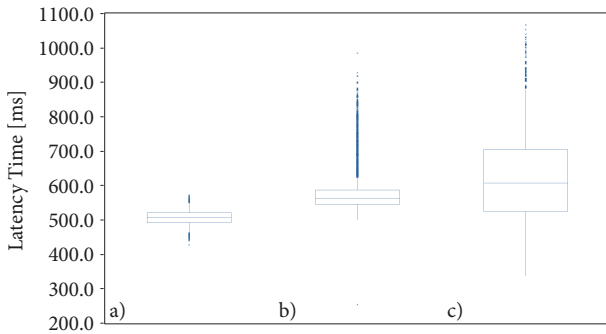


FIGURE 23: Box plots for (a) T1\_GII, (b) T2\_GII, and (c) T3\_GII.

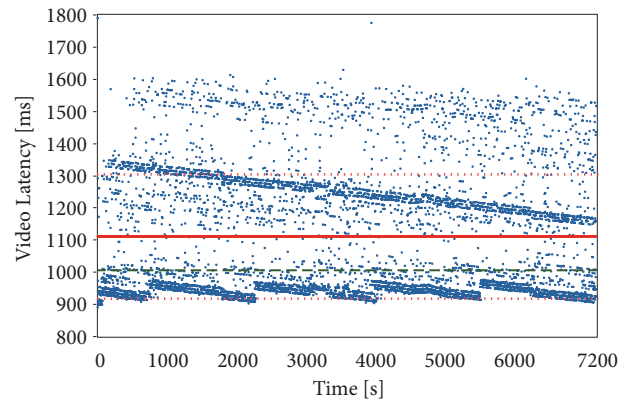


FIGURE 25: T2\_GIII video latency.

group (T2\_GIII) was conducted with an IP camera located in Limerick (video stream direction Limerick–Sarajevo), while the second test (T3\_GIII) was conducted with an IP camera located in Sarajevo. The results of these two tests are shown in Figures 25 and 26. Connection over the Internet and the connectionless UDP protocol also caused increased latency with higher jitter when compared to the LAN connection (this is best seen in Figure 27).

*4.8. Group IV: MJPEG Encoding and UDP Protocol.* The video latency tests from the last group were conducted using the MJPEG encoding algorithm and UDP for the transport layer protocol. The summary of Group IV tests is listed in

Table 8. T1\_GIV was conducted in a LAN with the results shown in Figure 28. When comparing the results obtained here with those from T1\_GII (488.48ms vs. 505.72ms in T1\_GII), it is seen that, by using the UDP protocol for stream transport, the latency can be lowered by approximately 70ms. Tests T2\_GIV and T3\_GIV are conducted between nodes connected over the Internet. T2\_GIV has a video stream direction from Limerick–Sarajevo (Figure 29) and T3\_GIV is in the direction Sarajevo–Limerick (Figure 30). While the mean latency values in the results of T2\_GIV and T3\_GIV are

TABLE 9: Proposed video latency measurement methodology results.

| Test No. | Arithmetic Mean [ms] | Median [ms] | Standard Deviation [ms] | Min Value [ms] | Max Value [ms] |
|----------|----------------------|-------------|-------------------------|----------------|----------------|
| T1_GI    | 869.12               | 869.05      | 23.90                   | 771.52         | 964.07         |
| T2_GI    | 1029.13              | 964.60      | 134.09                  | 872.59         | 1491.10        |
| T3_GI    | 1140.52              | 1053.25     | 460.96                  | 502.32         | 2499.7         |
| T3_GI    | 900.19               | 901.20      | 30.23                   | 800.15         | 1096.93        |
| T1_GII   | 505.72               | 504.97      | 20.77                   | 426.55         | 571.49         |
| T2_GII   | 581.30               | 561.26      | 68.05                   | 253.05         | 983.74         |
| T3_GII   | 616.34               | 606.54      | 125.16                  | 351.29         | 1060.00        |
| T1_GIII  | 849.93               | 850.01      | 17.66                   | 782.20         | 932.20         |
| T2_GIII  | 1109.99              | 1005.07     | 193.18                  | 889.77         | 1789.27        |
| T3_GIII  | 1211.30              | 1066.30     | 350.18                  | 802.27         | 2398.4         |
| T1_GIV   | 488.48               | 488.41      | 31.68                   | 304.74         | 585.76         |
| T2_GIV   | 558.85               | 556.18      | 29.69                   | 494.39         | 689.19         |
| T3_GIV   | 558.05               | 550.11      | 41.00                   | 465.15         | 961.81         |

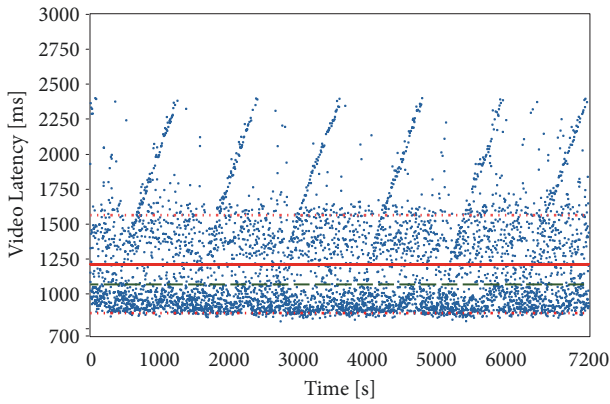


FIGURE 26: T3\_GIII video latency.

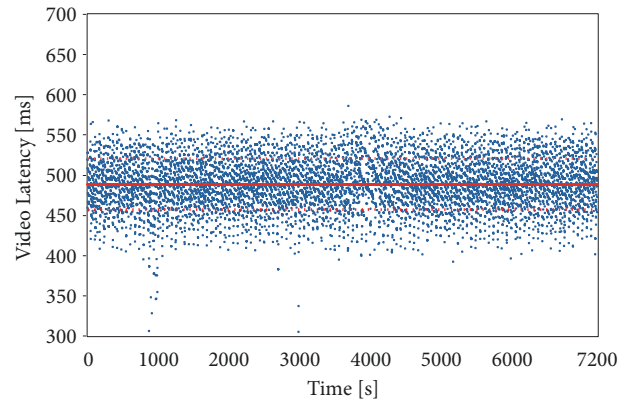


FIGURE 28: T1\_GIV video latency.

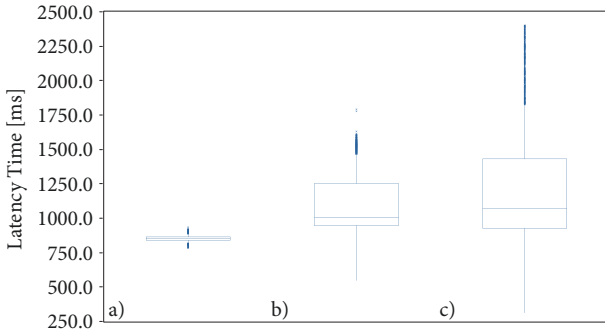


FIGURE 27: Box plots for T1\_GIII, T2\_GIII, and T3\_GIII.

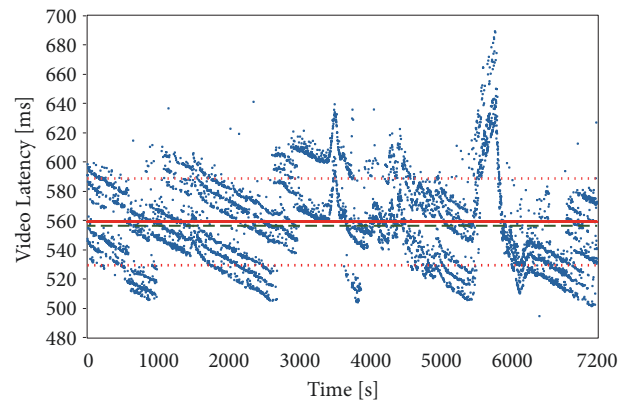


FIGURE 29: T2\_GIV video latency.

the same, both of the tests have latencies which are higher than T1\_GIV. Box plots for Group IV tests are visible in Figure 31.

*4.9. One-Way Network Transmission Latency Measurements in Video Streaming Applications.* Connection to the camera over the Video Server, which was described in previous sections, enabled network transmission delay to be measured and logged. Values presented in this section represent the

time necessary for a TCP/UDP packet to travel from the application level on one node to the application level on another node, located at the remote site. Connection between the nodes is established over the Internet, while the time on both nodes was synchronized to a local stratum-1 clock. It is worth noting that the measurements in these tests depend on neither the video encoding nor the transport layer protocol

TABLE 10: One-way network transmission latency in video streaming applications.

| Test No. | Protocol | Encoding | Connection Type | Network type | Source   |
|----------|----------|----------|-----------------|--------------|----------|
| T1_ND    | TCP      | H.264    | Video Server    | Internet     | Sarajevo |
| T2_ND    | TCP      | MJPEG    | Video Server    | LAN          | N/A      |
| T3_ND    | UDP      | MJPEG    | Video Server    | LAN          | N/A      |
| T4_ND    | TCP      | MJPEG    | Video Server    | Internet     | Limerick |
| T5_ND    | UDP      | H.264    | Video Server    | LAN          | N/A      |
| T6_ND    | UDP      | H.264    | Video Server    | Internet     | Sarajevo |

TABLE 11: One-way network transmission delays in video streaming application.

| Test No. | Arithmetic Mean [ms] | Median [ms] | Standard Deviation [ms] | Min Value [ms] | Max Value [ms] |
|----------|----------------------|-------------|-------------------------|----------------|----------------|
| T1_ND    | 43.65                | 33.95       | 24.26                   | 20.11          | 259.88         |
| T2_ND    | 7.11                 | 1.52        | 13.25                   | 0.1            | 49.7           |
| T3_ND    | 1.67                 | 1.63        | 0.56                    | 0.19           | 3.49           |
| T4_ND    | 50.94                | 44.90       | 20.79                   | 30.67          | 249.78         |
| T5_ND    | 1.52                 | 1.48        | 0.56                    | 0.19           | 3.49           |
| T6_ND    | 40.92                | 35.73       | 12.86                   | 28.30          | 176.83         |

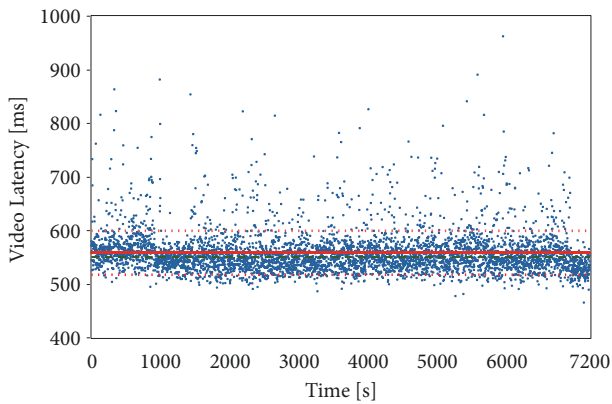


FIGURE 30: T3\_GIV video latency.

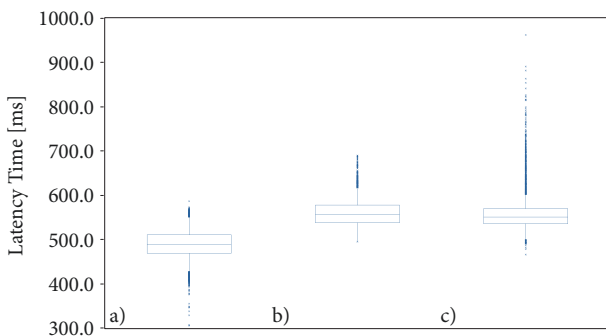


FIGURE 31: Box plots for (a) T1\_GIV, (b) T2\_GIV, and (c) T3\_GIV.

employed. Tests were conducted by using two remote nodes, one located in Limerick and the other located in Sarajevo. A list of tests in which network latency was acquired is shown in Table 10.

A summary of the results is listed in Table 11. It is worth noting that latencies of up to 260ms in overall video

latency can be caused by the network itself. To reconstruct a single frame of the video, multiple UDP/TCP packets are necessary (the exact number of network packets per video frame varies depending on the encoding algorithm used). The measurement methodology used in the experiment increases overall video latency due to the additional data processing and routing requirements. For this reason, in cases where network latency is a mandatory measurement, it is recommended to have separate applications executing in parallel which do not cause any unnecessary delays to the video stream.

As expected, the results show that latencies between devices connected on a LAN are lower than the ones between devices connected over the Internet. However, interesting comparisons can be made between the TCP and UDP measurements (e.g., between tests T2\_ND and T3\_ND). Latency measurements for these tests are shown in Figures 32 and 34, respectively. In both tests, video was encoded using MJPEG and latency was measured between the devices connected over the LAN. However, there is a difference in the acquired measurements. Tests using TCP show greater latencies with a higher jitter value and an uneven distribution of measurements, compared to the jitter and distribution of the UDP test. Box plots for T1\_ND, T2\_ND, and T3\_ND are shown in Figure 35. The difference in latency values and distribution shape can be explained based on the nature of the protocols used for transporting the stream. Distinct levels, which can be observed in most of the TCP protocol latency graphs, are caused by the packet retransmission mechanism (Figures 32 and 33). In TCP when the sender does not receive acknowledgment of reception of the packet, the packet is retransmitted. However, retransmission does not occur immediately, but after a timeout (known as Retransmission Timeout (RTO) [42]). These timeouts cause an increase in latency when using the TCP protocol compared to tests where UDP was used as the transport layer protocol. UDP is a connectionless protocol which does not employ any

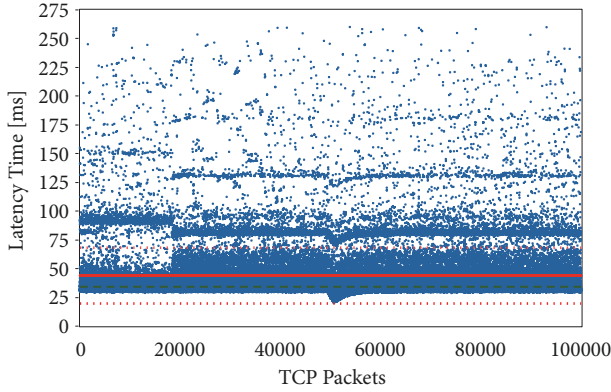


FIGURE 32: T1\_ND network latency.

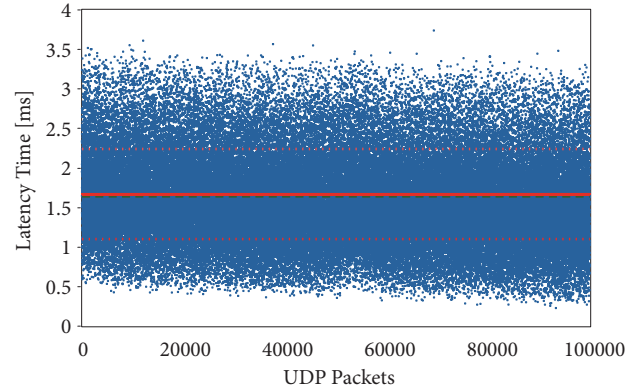


FIGURE 34: T3\_ND network latency.

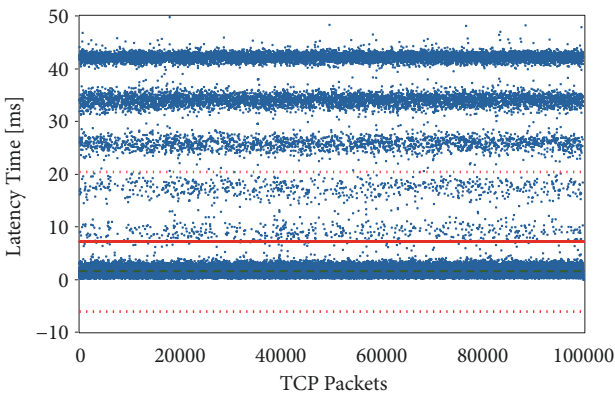


FIGURE 33: T2\_ND network latency.

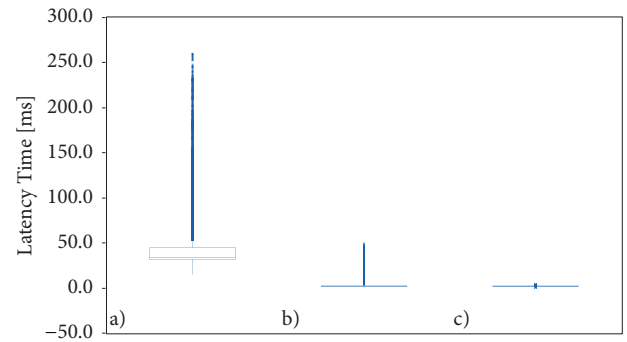


FIGURE 35: Box plots for (a) T1\_ND, (b) T2\_ND, and (c) T3\_ND.

retransmission mechanism, but there is no guarantee that all packets which are sent will be delivered to the receiving end [43].

Two more tests that can be compared are listed as T4\_ND and T6\_ND. The difference between these two tests is in the direction of the stream. While T4\_ND was conducted with the stream source in Limerick, T6\_ND was in the opposite direction; i.e., the stream source was in Sarajevo. A comparison between the two can be made using the minimum latency value. The difference between the minimum latency values was  $T_{\text{MIN}} = |T4_{\text{ND}_{\text{MIN}}} - T6_{\text{ND}_{\text{MIN}}}| = 30.67\text{ms} - 28.30\text{ms} = 2.37\text{ms}$  (see Table 10), which means that routes during the tests had similar configurations in both directions (same duration). In addition, if we consider that the PC clocks are synchronized with accuracy of 1ms, then the difference between minimum values is in the range from  $(T4_{\text{ND}_{\text{MIN}}} - 1\text{ms}) - (T6_{\text{ND}_{\text{MIN}}} + 1\text{ms}) = 0.37\text{ms}$  to  $|(T4_{\text{ND}_{\text{MIN}}} + 1\text{ms}) - (T6_{\text{ND}_{\text{MIN}}} - 1\text{ms})| = 4.37\text{ms}$ .

The presented latencies are for video which was compressed but not encrypted. Latency measurement which includes video encryption is ongoing using the methodology presented in [44, 45].

*4.10. Analysis of Required Throughput for Video Transfer.* Figures 36 and 37 show the required bandwidth for streaming with the same frame rate and resolution for both MJPEG and H.264 video. While the throughput of the MJPEG

compressed video stream is quite stable, the throughput of the H.264 stream oscillates. The reason for these oscillations lies in the fact that the scene which the camera was capturing was changing with period of  $T = 3\text{s}$ . Between two changes in the scene, the image was still and hence the required throughput dropped. However, for the MJPEG encoding whole images are being transferred all the time, regardless of changes in the scene in front of the camera. Captured throughput corresponds to the encoding mechanism functionality and behaviour. In terms of the protocol used in these test (TCP vs UDP), there was no change in the required bandwidth.

## 5. Conclusions

The methodology for latency measurement of the video stream in a marine based remote-control application with live video feedback has been discussed in this paper. When compared to the overall cost of the ROV system, this methodology represents an effective solution and provides measurement of the video latency between the control centre and the remote plant. Accurate knowledge of this latency is necessary for effective and safe control of the ROV. This methodology requires synchronization of the equipment clocks to a stratum-1 NTP server, both at the control centre and at the remote plant where the ROV is being deployed. Since the system is based on networked devices, adding two dedicated NTP servers does not require any architectural changes of the system.



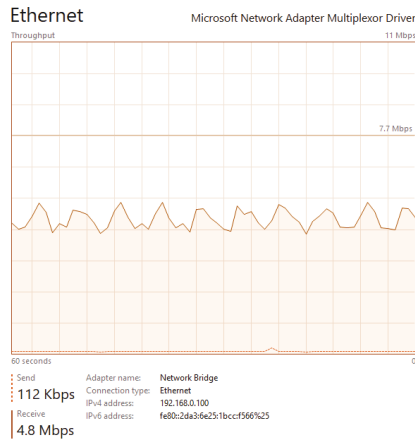


FIGURE 36: Capture of measurement of network bandwidth required for MJPEG video. Required bandwidth is within the limits of 4.6Mbps – 5.0Mbps.

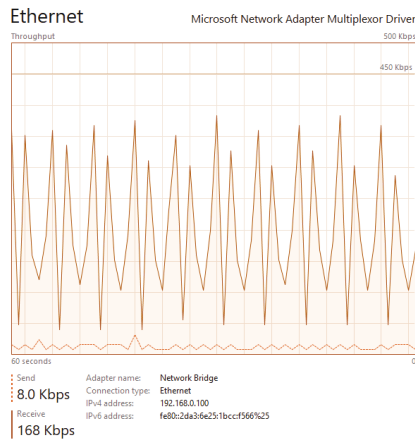


FIGURE 37: Capture of measurement of network bandwidth required for H.264 video. Required bandwidth is within the limits of 50Kbps – 380Kbps.

A real-time video stream is one of the main sources of feedback for the ROV operator; therefore, having stable and high quality video is mandatory for normal and safe system control. The results presented show that there is a trade-off between the level of latency and the bandwidth needed for the video transfer. When using the MJPEG encoding compression format, the overall latency is 300ms lower than in cases when H.264 is used for video compression. On the other hand, the throughput required for transferring MJPEG is measured in Mbps and depends on video resolution, while the throughput required for transferring of a H.264 compressed stream did not go above 500Kbps during the tests.

It is noticeable that the latencies measured here are mostly higher than the high limit values, which are usually suggested for applications running over the Internet. For example, 400ms is perceived as a high latency limit for a video conferencing application, as suggested by the ITU [46].

On the other hand, the maximum acceptable video latency for teleoperation systems depends on the systems purpose and overall dynamics. Since the remote operating for the underwater vehicle is a system with relatively slow dynamics (e.g., compared to a drone system), it can be stated that some of the measured latencies are in an acceptable range required for normal control. From [47], it is visible that average latency of up to 100ms for transferring control signals to the ROV does not impair the ROV control algorithm. Extensive analysis of impact of delay on teleurgical performance was presented in [48]. It was concluded that deterioration of performance was noticeable for latencies above 300ms and that there was increase in errors during control for latencies greater than 500ms.

The lowest latencies were measured while using MJPEG compression and hence it is recommended to use cameras without or with very low video compression. Remote control of the ROV over the Internet was successfully performed many times with existing equipment but without active video latency measurement. This work allows the remote ROV pilot to have real-time information about video delay from the remote plant and their ROV control strategy can be adapted to suit this delayed video. In addition, this research helps the remote pilot in determining whether it is possible to safely control the ROV over the provided network link, or whether a physical deployment to the site under inspection is required.

## Data Availability

The data that this work was based on was gathered by the authors and can be made available for research work by other interested parties.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Acknowledgments

The authors would like to thank [EPSPG/2011/257] the SFI Centre for Marine and Renewable Energy Ireland (MaREI), Grant references [12/RC/2302] and [14/SP/2740], for facilitating this work.

## References

- [1] Y. Chang, Q. Liu, X. Jia, and K. Zhou, "Routing and transmission scheduling for minimizing broadcast delay in multi-rate wireless mesh networks using directional antennas," *Wireless Communications and Mobile Computing*, vol. 15, pp. 87–99, 2012.
- [2] H. Li, X. Liu, W. He, W. Yang, and W. Dou, "Delay analysis in practical wireless network coding," *Wireless Communications and Mobile Computing*, vol. 14, no. 5, pp. 497–515, 2014.
- [3] C. Pereira, A. Pinto, D. Ferreira, and A. Aguiar, "Experimental Characterization of Mobile IoT Application Latency," *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 1082–1094, 2017.
- [4] J. H. Sorensen, P. Popovski, and J. Ostergaard, "Delay Minimization in Real-Time Communications with Joint Buffering and

- Coding,” *IEEE Communications Letters*, vol. 21, no. 1, pp. 52–55, 2017.
- [5] J. Saldana, J. Ruiz-Mas, and J. Almodovar, “Frame aggregation in central controlled 802.11 WLANs: The latency versus throughput tradeoff,” *IEEE Communications Letters*, vol. 21, no. 11, pp. 2500–2503, 2017.
  - [6] D. Tian and V. C. M. Leung, “Analysis of broadcasting delays in vehicular ad hoc networks,” *Wireless Communications and Mobile Computing*, vol. 11, no. 11, pp. 1433–1445, 2011.
  - [7] B. Girod, M. Kalman, Y. J. Liang, and R. Zhang, “Advances in channel-adaptive video streaming,” in *Proceedings of the International Conference on Image Processing (ICIP’02)*, IEEE, Rochester, NY, USA, September 2002.
  - [8] S. Lin, Y. Liao, Y. Tsao, and S. Chien, “Object-based on-line video summarization for internet of video things,” in *Proceedings of the 2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, IEEE, Baltimore, MD, USA, May 2017.
  - [9] R. Pereira and E. G. Pereira, “Video streaming considerations for internet of things,” in *Proceedings of the 2nd International Conference on Future Internet of Things and Cloud (FiCloud ’14)*, pp. 48–52, IEEE, Barcelona, Spain, August 2014.
  - [10] “IMAQ Color Learn,” Article ID 370281, [http://zone.ni.com/reference/en-XX/help/370281AC-01/imaqvision/imaq\\_color\\_learn/](http://zone.ni.com/reference/en-XX/help/370281AC-01/imaqvision/imaq_color_learn/).
  - [11] A. Kumcu, L. Vermeulen, S. A. Elprama et al., “Effect of video lag on laparoscopic surgery: correlation between performance and usability at low latencies,” *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 13, no. 2, 2017.
  - [12] E. Omerdic, D. Toal, G. Dooly, and A. Kaknjo, “Remote presence: Long endurance robotic systems for routine inspection of offshore subsea oil & gas installations and marine renewable energy devices,” in *Proceedings of the 2014 Oceans - St. John’s, OCEANS 2014*, pp. 1–9, St. John’s, Canada, September 2014.
  - [13] G. Almes, S. Kalidindi, M. Zekauskas, and A. Morton, “A One-Way Delay Metric for IP Performance Metrics (IPPM),” IETF, January 2016.
  - [14] A. Kryczka, A. Arefin, and K. Nahrstedt, “AvCloak: A tool for black box latency measurements in video conferencing applications,” in *Proceedings of the 15th IEEE International Symposium on Multimedia, ISM 2013*, pp. 271–278, IEEE, Anaheim, CA, USA, December 2013.
  - [15] O. Boyaci, A. Forte, S. A. Baset, and H. Schulzrinne, “vDelay: A tool to measure capture-to-display latency and frame rate,” in *Proceedings of the 11th IEEE International Symposium on Multimedia, ISM 2009*, pp. 194–200, IEEE, San Diego, CA, USA, December 2009.
  - [16] O. Boyaci, A. Forte, S. A. Baset, and H. Schulzrinne, “Demonstration of vDelay: A tool to measure capture-to-display latency and frame rate,” in *Proceedings of the 11th IEEE International Symposium on Multimedia, ISM 2009*, pp. 444–445, IEEE, San Diego, CA, USA, December 2009.
  - [17] J. Jansen, “VideoLat: An Extensible Tool for Multimedia Delay Measurements,” in *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 683–686, Orlando, FL, USA, November 2014.
  - [18] J. Jansen and D. C. Bulterman, “User-centric video delay measurements,” in *Proceedings of the 23rd ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, pp. 37–42, ACM, Oslo, Norway, February 2013.
  - [19] “IMAQ Vision Concepts Manual,” 2000, <http://www.ni.com/pdf/manuals/322916a.pdf>.
  - [20] M. C. Jacobs, M. A. Livingston, and A. State, “Managing latency in complex augmented reality systems,” in *Proceedings of the 1997 symposium on Interactive 3D graphics*, ACM, Providence, RI, USA, April 1997.
  - [21] C. Bachhuber and E. Steinbach, “Are Today’s Video Communication Solutions Ready for the Tactile Internet?” in *Proceedings of the 2017 IEEE Wireless Communications and Networking Conference Workshops, WCNCW 2017*, pp. 1–6, IEEE, San Francisco, CA, USA, March 2017.
  - [22] C. Bachhuber, E. Steinbach, M. Freundl, and M. Reisslein, “On the minimization of glass-to-glass and glass-to-algorithm delay in video communication,” *IEEE Transactions on Multimedia*, vol. 20, pp. 238–252, 2018.
  - [23] A. Kaknjo, E. Omerdic, and D. Toal, “Measurement of Network Latency in Remote Presence Applications,” *IFAC-PapersOnLine*, vol. 49, no. 23, pp. 193–198, 2016.
  - [24] T. Eylen and C. F. Bazlamacci, “One-Way Active Delay Measurement with Error Bounds,” *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 12, pp. 3476–3489, 2015.
  - [25] Y.-P. Fang, N. Pedroni, and E. Zio, “Resilience-Based Component Importance Measures for Critical Infrastructure Network Systems,” *IEEE Transactions on Reliability*, vol. 65, no. 2, pp. 502–512, 2016.
  - [26] W. Zhang and C. van Luttervelt, “Toward a resilient manufacturing system,” *CIRP Annals*, vol. 60, no. 1, pp. 469–472, 2011.
  - [27] T.-C. Wang, H.-C. Fang, and L.-G. Chen, “Low-delay and error-robust wireless video transmission for video communications,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 12, pp. 1049–1058, 2002.
  - [28] W. Molisz and J. Rak, “Quality of resilience in IP-based future internet communications,” in *Proceedings of the 2011 13th International Conference on Transparent Optical Networks, ICTON 2011*, pp. 1–4, Stockholm, Sweden, June 2011.
  - [29] S. Gjessing, “Implementation of two Resilience mechanisms using multi topology routing and stub routers,” in *Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services, AICT/ICIW’06*, pp. 29–29, IEEE, Guadeloupe, French Caribbean, February 2006.
  - [30] J. Rak and J. P. Sterbenz, “Preface: Optimization issues in resilient network design and modeling,” *Networks. An International Journal*, vol. 66, no. 4, pp. 251–252, 2015.
  - [31] C. Bachhuber and E. Steinbach, “A system for high precision glass-to-glass delay measurements in video communication,” in *Proceedings of the 23rd IEEE International Conference on Image Processing, ICIP 2016*, pp. 2132–2136, IEEE, Phoenix, AZ, USA, September 2016.
  - [32] “Libavcodec Documentation FFmpeg,” <https://ffmpeg.org/libavcodec.html>.
  - [33] “Quectel L80 Compact GPS Module Integrate with Patch Antenna,” 2018, <https://www.quectel.com/UploadFile/Product/Quectel.L80-GPS.Specification.V1.2.pdf>.
  - [34] T. Stütz and A. Uhl, “A survey of H.264 AVC/SVC encryption,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 3, pp. 325–339, 2012.
  - [35] N. Conci, M. Verza, G. Boato, and F. G. B. De Natale, “Multiple description coding using data hiding and regions of interest for broadcasting applications,” *International Journal of Digital Multimedia Broadcasting*, vol. 2009, Article ID 234360, 9 pages, 2009.

- [36] H. Schulzrinne, A. Rao, R. Lanphier, M. Westerlund, and M. Stiemerling, "Real-time streaming protocol version 2.0," IETF, December 2016.
- [37] C. Bachhuber and E. G. Steinbach, "A system for precise end-to-end delay measurements in video communication," CoRR, 2015.
- [38] T. H. Szymanski, "An Ultra-Low-Latency Guaranteed-Rate Internet for Cloud Services," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 123–136, 2016.
- [39] J. Silvestre, L. Almeida, R. Marau, and P. Pedreiras, "Dynamic QoS management for multimedia real-time transmission in industrial environments," in *Proceedings of the 12th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2007*, pp. 1473–1480, IEEE, Patras, Greece, September 2007.
- [40] L. Chen and C.-W. Lee, "The effects of mobility and redundancy on wireless video streaming over MANETs," in *Proceedings of the 22nd International Conference on Advanced Information Networking and Applications Workshops/Symposia, AINA 2008*, pp. 949–953, IEEE, Okinawa, Japan, March 2008.
- [41] L. Chen, N. Shashidhar, and Q. Liu, "Scalable secure MJPEG video streaming," in *Proceedings of the 26th IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2012*, pp. 111–115, IEEE, Fukuoka, Japan, March 2012.
- [42] V. Paxson, M. Allman, J. Chu, and M. Sargent, "Computing TCP's retransmission timer," IETF, June 2011.
- [43] L. Eggert, G. Fairhurst, and G. Shepherd, "UDP usage guidelines," IETF, March 2017.
- [44] M. Rao, T. Newe, E. Omerdic, G. Dooly, E. Lewis, and D. Toal, "An efficient implementation of FPGA based high speed IPSec (AH/ESP) core," *International Journal of Internet Protocol Technology*, vol. 11, no. 2, pp. 97–109, 2018.
- [45] M. Rao, T. Newe, E. Omerdic et al., "Bump in the wire (BITW) security solution for a marine ROV remote control application," *Journal of Information Security and Applications*, vol. 38, pp. 111–121, 2018.
- [46] "One – way transmission time," ITU-T Recommendation G.114 Series G: Transmission Systems and Media, Digital Systems and Networks, June 2003.
- [47] M. Doniec, C. Detweiler, I. Vasilescu, and D. Rus, "Using optical communication for remote underwater robot operation," in *Proceedings of the 23rd IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010*, pp. 4017–4022, Taiwan, October 2010.
- [48] M. Perez, S. Xu, S. Chauhan et al., "Impact of delay on telesurgical performance: study on the robotic simulator dV-Trainer," *International Journal for Computer Assisted Radiology and Surgery*, vol. 11, no. 4, pp. 581–587, 2016.



**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

