Measurement of Latency During Real-Time Video Transmission for Remote Supervision of Agricultural Machines

Mitchell Green

University of Manitoba

A thesis submitted to the faculty of Graduate Studies of

The University of Manitoba in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Department of Biosystems Engineering

University of Manitoba

Canada

January 7, 2021

Abstract

Autonomous vehicles will have an increasing presence in the agricultural world in the coming decades and third-party remote observation is likely to be an important element. Few previous studies have assessed video latency as it pertains to the operation of an agricultural vehicle, and understanding variations in latency is a critical factor in designing a useful system for an operator. This study aimed to investigate real-time video latency using a Raspberry Pi as a method of video capture and remote streaming, overlaying barcode timestamps into a video stream as a method of latency calculation. A system was assembled capable of transmitting real-time video using radio and cellular transmission and used to transmit video from a field vehicle to a nearby viewer to assess differences in transmission across possible platforms for field use. Statistically significant increases were observed for latency in relation to video resolution, and cellular video latency was lower when transmitting higher resolution video in areas of good network quality, but higher than radio transmission when transmitting low quality video or in areas where cellular networks were not as robust. Based on the results we conclude that Ethernet radio or a similar point-to-point transmission system is viable as an alternative to cellular transmission for edge-of-field video surveillance when using lower resolution video. We recommend assessment of local network quality prior to any transmission system selection, and the according adjustment of transmitted video resolution such that latency for the viewer is minimized while optimizing overall quality of service.

Acknowledgements

To my advisors, Dr. Danny Mann and Ekram Hossain, for their infinite patience and guidance.

Dr. Mann allowed me an incredible opportunity to be part of this experiment, and gave excellent direction while allowing me enough freedom on this project to enable me to develop new skills and talents as an engineer and as a person, providing incredible support through the unexpected challenges posed by COVID-19. I will be forever grateful for the chance I had to work with him completing this project.

Dr. Hossain provided project guidance and technical instruction which was instrumental in the completion of this thesis. His networking expertise was invaluable and I had the privilege to learn a great deal of new information from being able to work with him through this degree. Thank you for the support, Dr. Hossain.

Bell MTS provided integral support and funding over the course of this project, and without their contributions this work would not be possible. I am very grateful for their support, and for the Bell MTS Innovations in Agriculture Graduate Student Fund which made this project possible.

I am very grateful to Dr. Jitendra Paliwal and Dr. Don Petkau for being able to act as members of my thesis committee. It was a great opportunity to be able to meet, interact with, and learn from them over the course of this project.

Dr. Jason Morrison has been an amazing guide to statistical analysis and computing throughout the entirety of my work and a terrific source of knowledge and great discussions. I am grateful for the opportunities I had to speak with and learn from him.

To Uduak Edet, for being a constant friend and mentor through my work, and whose advice has never led me wrong over the course of my degree. Thanks Uduak.

I want to express my gratitude to Matt McDonald and Minami Maeda for their support with all electronic needs, and for an incredible amount of patience when it came to returning truck keys.

To Marcel Lehmann and Dale Bourns for their work on everything hardware, going above and beyond when it came to fabrication and operating the vehicle, and putting up with many unreasonable requests, thank you both.

Dedication

To my wonderful sister Aran and her family, who have supported me through all my studies and in all aspects of life.

To my father, Jim, for the years of guidance and love and for putting me on the path which I now take. Thank you for everything.

To my mother, Lori, for her unending care, love and support in everything I do. I could never have reached this point in my life without the belief you have had in me as a person and for the never-ending encouragement you provided. I love you Mom.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Overview

Equipment capable of limited self-operation on a farm has been implemented for decades using technologies such as GPS controlled autosteering, and as the technology has progressed the capabilities of the equipment have increased in sophistication and complexity, with vehicles capable of operating autonomously without direct manual operation or even without the need of a driver or cab. The financial implications of these advancements are considerable, with the autonomous farm vehicle market forecasted to be worth $1.8 billion by 2024 (Hegde, 2018). Agricultural work is often simultaneously time intensive and time sensitive, requiring significant man-hours condensed into just a few months of a season. Self-driven agricultural vehicles circumvent these shortcomings, allowing for fewer operators to conduct more work over greater spans of time, which is a significant asset to the time-sensitive nature of farm work.

Autonomous farm machines such as driverless tractors are agricultural implements which can operate independently, without the need of an operator sitting at the controls directly managing the vehicle at all times. Technology to facilitate automated guidance of a farm vehicle has existed for decades dating back to the 1950s (Yaghoubi et al., 2013) with the use of guide wires to facilitate the motion of tractors. Machine vision experiments for tractors have been conducted since the mid-1980s at the University of Michigan (Reid, Zhang, Noguchi, & Dickson, 2000), and developments in satellite guidance led to autosteer technology available to farmers utilizing GPS guided systems to control the path of the vehicle. Autonomous farm vehicles can operate entirely without a cab, based on obstacle detection mechanisms, and the ability to receive commands from the operator via touchscreen tablet or phone. While at the time of this writing, operators would not necessarily be required to be located within the proximity of the vehicle while in operation, some developers such as DOT request operators keep their machine to a viewable distance for an operator pending regulatory uncertainties (Ag Professional, 2019). Farmers can be kept apprised of current operational status of the remote machine via applications used to interface with the system, and based on feedback provided by the autonomous vehicle through a range of telemetric data can make decisions regarding the machine's operation.

Autonomous farm vehicles utilize a range of sensors to operate and relay relevant information to their operators, and cameras which are able to cover a wider range than normal for a driver

situated in a cab. Autonomous farm vehicles such as the Case IH autonomous concept tractor supplement camera systems with radar and LIDAR (Light Detection and Ranging), and are able to use the visual data both for machine vision tasks, and to convey information to the operator to use to make decisions regarding machine guidance through access to the onboard camera video (Case IH, 2017). ISO (International Organization of Standards) standard 18497 requires that, in the event that the perception functions of the machine fail, the machine either stops or yields control to the operator (Tiusanen, Malm, & Ronkainen, 2020). Previous research was conducted at the University of Manitoba surveying farmers for their response to interfaces in an autonomous vehicle to understand the qualities which would be desirable for an autonomous system. A conclusion that emerged from the study was the larger degree of comfort test subjects felt in the operation of a remote vehicle when sensor data from a remote vehicle was paired with a live video feed farmers could use to develop an understanding of how the machine was operating at that point in time (Edet, Hawley, & Mann, 2019). A follow-up study was suggested that would examine the possibility of transmitting live video for a remote observer from a vehicle. Specifically, it was desired to measure the expected latency from transmission of video for a farm vehicle as it would operate in a field environment.

Any transmission of information incurs a certain amount of latency due to various factors. These factors can include the propagation of the information across mediums including air and wires, and the creation and interpretation of the data being sent by the transmitter and receiver, as well as any intermediaries required to pass the data to the destination. In the case of autonomous vehicles, as the observer is not generally located proximal to the vehicle, some means of remote transmission is necessary to relay telemetric and observational data to the operator to enable decision making. Agricultural fields are widespread and often remote, and therefore transmission of camera or telemetric information from a vehicle to an operator has the potential to take place over large distances in the event that monitoring is not taking place somewhere adjacent to the vehicle. Transmission of information over significant distances is not instantaneous, and can be impacted by delays in propagation and computational calculation. Sending video data over an LTE connection, for example has been measured to have a two-day latency of 100 ms in previous experiments for use in remote driving (Kang, Zhao, Qi, & Banerjee, 2018). The result of this is a certain amount of experienced delay for the observer, which can have a significant impact on performance and user perception.

In order to replicate the operation of an autonomous vehicle and gauge the expected latency during transmission, a smaller system is used to obtain latency measurements in a series of different field locations. In this case, in place of a larger agricultural vehicle such as a sprayer or swather, a tractor mower is used to make passes in a plot of land similarly to a vehicle making passes through a field. While a case study based on field testing is difficult to reasonably generalize, results obtained through this method can provide an indication of attainable latencies in a field environment and provide a greater understanding of the influencing factors and challenges posed working in such an environment.

The use of cellular transmission of video is a reasonable way to first approach wireless surveillance of autonomous vehicles, by way of sheer ubiquity of the technology. Cisco predicts that by 2022, nearly 80% of the world's mobile internet traffic will be video (2019). While the capabilities of cellular video will become increasingly robust as additional towers are installed to increase coverage, and 4G and 5G become more accessible in remote or rural areas, at present cellular availability in agricultural fields is often limited and the ability to obtain even a 3G signal can be unreliable. As a result of the limited network environment, it is worthwhile to supplement evaluation of video transmission over a cellular network in a field with a direct point-to-point transmission of video, facilitated in this case through the use of Ethernet radios.

Using a Raspberry Pi as a video capture and encoding device, and overlaying barcode timestamps into video as a method of calculating latency using the video pipeline construction application GStreamer, point-to-point radio transmission and cellular transmission of video from a field vehicle to a nearby viewer were measured and evaluated across a range of varied factors, and the ability to transmit video in real-time for autonomous vehicle use across a variety of factors and conditions will be verified. In doing this, the feasibility of real-time video transmission for autonomous agricultural equipment will be examined and evaluated as a realistic possibility.

1.2 Contributions of the Work

This thesis aims to measure latency as it would apply to an autonomous farm vehicle. In this case the movements of a larger autonomous farm vehicle would be replicated through the use of a smaller lawn tractor. Through analysis of latency, the intent is to gain an understanding of

latencies as they might be expected when transmitting data locally or over a cellular connection to an observer.

While there have been several studies to characterize latency using varied methods across a range of disciplines, to the author's knowledge there has been little in the way of literature and research efforts to measure latency as it applies to an autonomous farm vehicle, particularly when operating in a naturalistic environment. Additionally, the employment of Ethernet radios as a method of video transmission, particularly for this purpose has similarly little extant literature. The use of this device as a method of video transmission in an agricultural field environment is a novel use for the device not previously found in literature.

1.3 Objectives

The objectives of the thesis were as follows:

1. To design and fabricate a system capable of transmitting video in real-time from a mobile agricultural machine to a monitoring station located adjacent to the field.
2. To measure the transmission latency associated with varying transmission distances and varying video resolutions.

2 Literature Review

I first begin the literature review by discussing the topic of remote supervision of agricultural vehicles. Subsequently, the topic of video streaming is addressed, before moving into an examination of the latency associated with video transmission. I then examine the impact this latency has on the experience for the user, before moving into an examination of methods utilized to measure latency.

2.1 Remote Supervision of Agricultural Machines

Prior research has been conducted into the relationship of farmers with autonomous agricultural implements and which components provide comfort for the operator and utility from the machine. Panfilov and Mann (2018) showed that farmers felt a higher degree of comfort in the use of an agricultural machine when telemetric sensor data was paired with a live video feed. Edet et al. (2019) found that the preferred location for farmers to observe an autonomous vehicle would be near the edge of a field, based on the results of a study conducted among farmers in Manitoba, Saskatchewan, and Alberta. To this end, it is worthwhile to analyze the encountered

video latency from a monitoring station located approximately at what might constitute an "edge of field" setting.

2.2 Video Streaming

Wireless streaming of video for latency measurement requires an examination of the state of the art for the elements required in a video stream. It is necessary to examine tools used to evaluate formats of video streaming to efficiently compress and transport video, understanding of the uses some of the applications and products being used have had in the industry and for similar work in order to be able to configure a camera streaming system.

2.2.1 Streaming Formats

Real-Time Protocol (RTP) is a network protocol used to deliver audio and video in real-time, adding sequence numbers and timestamps in an information header encapsulated with packets of data, allowing for monitoring and easier delivery of packets in the correct order. RTP makes no guarantee of timely delivery or quality of service (Zeng & Lan, 2008). A considerable downside of RTP is the IP-based nature of the protocol, which can cause security issues with certain devices, but the protocol is effective with helping synchronization of multiple streams, making it useful for services such as voice over IP transmission.

Dynamic Adaptive Streaming over HTTP (DASH, or MPEG-DASH) is a standard intended as a common platform for content streaming. Adaptive streaming works by selectively retrieving segments of video based on available data rates as requested by the client based on a supplied manifest file (Sodagar, 2011). Compared to real time protocol transmission, DASH based streaming works with system firewalls and reduces server loads by being more client based.

2.2.2 Encoding and Compression Formats

Streaming of raw recorded video is very data intensive and requires a very large throughput to maintain. For this reason, compression of video is standard in video streaming. Through a range of algorithms, video is compressed using encoder-decoder (CODEC) pairs which reduce the size of the video based on an assortment of compression schemes and later reconstruct it. While many methods of compression are lossy and will discard information through the process of encoding and decoding the incoming video feed, in a remote agricultural environment where available bandwidth is limited, compression of video becomes paramount.

Motion JPEG (MJPEG) is an encoding method in which an incoming video stream is encoded as a series of JPEG formatted images before delivery of the bitstream (Library of Congress, 2017). MJPEG trades a very fast encoding rate due to the relative simplicity of conversion compared to other methods for a large amount of required bandwidth, due to the uncompressed size of each image. Streams which can meet the required capacity can deliver video in a very timely manner, but the bandwidth demands are considerable.

H.264 is a well-established and widely supported compression format first ratified by the International Telecommunication Union (ITU) in 2003 (International Telecommunication Union, 2003). H.264 simplifies and reduces incoming video streams into blocks based on several differing algorithms using intra frames and predicted frames, removing extraneous information while still retaining a sufficient amount of data to provide a video stream of reasonable viewing quality. H.264 is still widely used today, with 92% of video developers using the technology as of 2017 (Bitmovin, 2018).

H.265 was accepted as a new standard in 2013 (International Telecommunication Union) and is considered to be the successor to H.264 compression. While H.265 executes similar block based encoding, it promises faster encoding rates than H.264 as a result of the larger block sizes, able to use 64x64 chunks as opposed to 16x16. This improvement to H.264, along with additional factors simplifies the process greatly and reduces the encoding time. While GStreamer encoders do exist for H.265 through the x265enc plugin, the relative lack of support for the considerably newer compression method, coupled with GStreamer having tools to allow for hardware acceleration in H.264 encoding offer significant advantages for using the older, more commonly adopted method.

2.2.3 Transport Layer Protocols

TCP (transmission control protocol) is a reliable data protocol which waits for an acknowledgement from the receiving device when sending packets. In the event that a packet is not confirmed to be received, the sender will retransmit the packet again and wait for an acknowledgement. While this prevents loss of data, this back-and-forth process can introduce a considerable delay if data transmission is time sensitive (Kumar & Rai, 2012). UDP consists of a continuous transmission of data without requirement of confirmation from the sender past the initial handshaking process. While this continuous transmission can result in packet loss, the

latency is significantly lower than for TCP, a property which is desirable for communications which require minimal delay and can tolerate loss such as in video streaming. Figure 1 illustrates the comparative data transaction processes post-handshake of the two communication protocols.

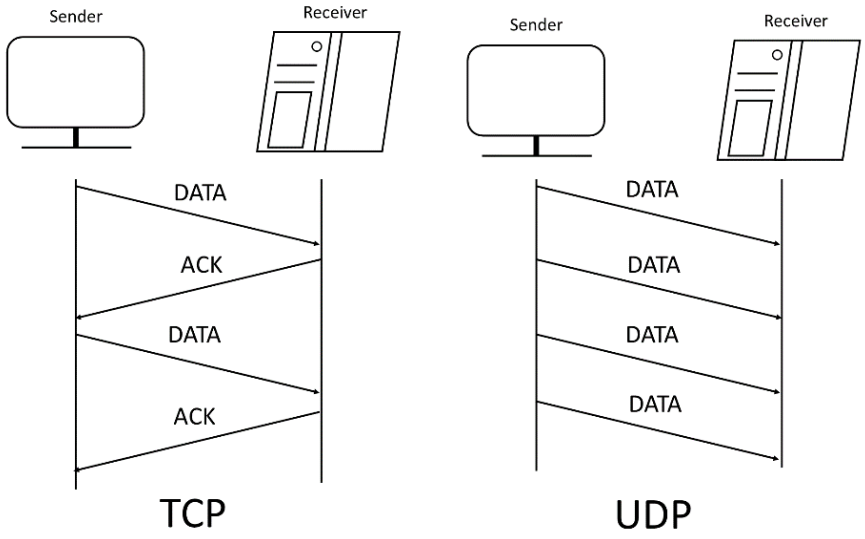

Figure 1: Comparison of TCP/UDP datagram transfers.

As the primary parameter of interest is the latency, and we wish to keep it within real-time constraints, relying on the continuous stream of UDP data will allow a minimization of the video transmission latency from the video platform to the observing user by eschewing the packet acknowledgement requirement of TCP. While TCP was initially tested in early phases of the project, it was found that it was considerably slower than UDP for delivery of video.

2.2.4 Qualitative Evaluation of Stream

Video quality was considered as a potential metric of evaluation and would be theoretically useful to verify the performance of a method of video transmission. Measurement schemes such as the MOtion-based Video Integrity Evaluation index and Multi-scale Structural Similarity Index have had a correlation with viewer perception demonstrated (Seshadrinathan, Soundararajan, Bovik, & Cormack, 2010) when livestreaming video in different formats for viewers to subjectively rate. Luma-Peak Signal to Noise Ratio is also a very commonly used metric for image quality across many compression standards (Shang et al., 2017), but it was decided not to use qualitative video evaluation metrics for the experiment, as the priority is to measure and evaluate latency rather than the perceptual quality of the stream.

2.3 Video Latency

When discussing networking, latency can mean elements such as the time taken from an event occurring to the display of the event, often referred to as the Glass-to-Glass (G2G) latency (Bachhuber, Steinbach, Freundl, & Reisslein, 2018), or the time taken for a command or bit of data to move through a network. Delays in encoding and decoding, transmission of a signal and travel time for the propagation of a signal over a network are all factors which comprise individual elements of video latency. Congested networks, poor bandwidth limits, data buffering, and data processing times all impact the time it takes to send data across a network. In the case of video data, when considering G2G latency the time taken to capture a frame of pixels with a camera and draw that same frame for the observer on their monitoring device are also factors that impact the time delay. The encoding method used can also heavily impact the latency, due to the computations required to compress the video feed. This can be seen in the work of Kaknjo et al. (2018) and their measurements of the latency of a camera system equipped to a robot used to stream video data; when directly comparing MJPEG and H.264 encoded video streams, it was found that the latency of MJPEG was nearly 300 ms less than that of H.264 compression in similar cases.

Specific constraints for real-time video latency do not appear to be well defined and requirements vary depending on how the video is used. In experiments operating a robotic surgery tool with the addition of latency, latencies of less than 200 ms were shown to be ideal, with latencies up to 300 ms also considered to be suitable for surgery with deterioration beginning past that threshold (Xu et al., 2014). Participants in a study examining the effects of latency on performance in video games found first-person-shooters and racing games have been shown to tolerate latencies only up to 100 ms, while 3rd person role-playing and sports games were able to tolerate up to 500 ms (Claypool & Claypool, 2006). Tests involving participants remotely operating a vehicle to do laps around a track showed that with a 358 ms delay, lap times for participants increased from 50-100% over completion times without the added delay (Liu, Kwak, Devarakonda, Bekris, & Iftode, 2017). Regarding latency during internet communications, the ITU recommends not exceeding 400 ms of latency (International Telecommunications Union, 2003) during video communications. Given the various limitations imposed for different requirements, it seems reasonable to aim to achieve a latency in the low hundreds of milliseconds.

2.4 Effects of Video Latency on the User Experience

Several studies have been completed that aim to tie user experience with encountered latency. A previous study into the impact of delay on a subject using humans completing simple tasks in a virtual environment with varying levels of lag (MacKenzie & Ware, 1993). MacKenzie and Ware found that with a maximum increase of 225 ms in latency compared to the null state, there was a corresponding 64% increase in required movement and a 214% increase in the rate of errors. Studies for the impact of latency in robotic laparoscopic surgery (Rodr, Kypson, Moten, Nifong, & Jr, 2006) found that even an increase as minute as 105 ms resulted in a deleterious impact in performance and overall user experience.

Davis, Smyth, & McDowell (2010) examined the effects of time lag on operation, and found that the issues encountered were more severe with variance in latency than for a fixed delay implemented during the test for participants. Similar studies (Liu et al., 2017) operating an RC car and broadcasting 240p video frames to the experimental testers over an LTE network concluded that direct operation of vehicles remotely was likely not feasible without delay compensation, with the addition of a randomized delay in video demonstrating large increases in the time taken to complete laps around a test track, along with increases in perceived mental workload and frustration for testers.

2.5 Latency Measurement

There have been a significant number of studies which have set out to characterize latency in video, utilizing a variety of methods and analyzing differing aspects. A study was conducted examining the performance of low latency MPEG-DASH formatted video (Bouzakaria, Concolato, & Le Feuvre, 2014), analyzing latency during the transmission of two standard test videos. Measurements of the network quality and latency of WebRTC video by García, Gortázar, López-Fernández, & Gallego (2016) were conducted with a custom Java application. Utilizing a transmitted video specialized for measurement applied by a web browser and which was read by a receiver browser using software written by the team, a consistent latency of approximately 145 ms was measured as long as the number of simultaneous users remained below 180. Video latency was measured by way of a flashing light source emitting light at a known time a short distance from a photo sensor using an inexpensive Arduino to calculate the difference in time (Bachhuber & Steinbach, 2016). While the Arduino was an effective method of latency

measurement, obtaining a range of latency values very close to what would be theoretically expected based on camera frame-rate and the frequency of the display used, the dependency on a low-noise environment was not compatible with the goal to examine a machine in a regular operating environment, where the presence of variable light sources would likely interfere with the measurement capabilities. A latency measurement format in a virtual environment (Friston & Steed, 2014) was completed as a way to test with finely tuned, highly adjustable parameters for the experiment, comparing two previous methods of latency analysis to a third alternative of automated frame counting designed by the authors to compare their performance.

Capturing pure G2G latency is not straightforward in a remote agricultural environment. The above methods are reliant on the refresh rate of the screen involved or require a large number of external components to be able to complete latency measurements. In this experiment aiming a camera at two screens to calculate one-way delay is very difficult due to the transmission distances and environment involved and while the addition of an external image detection mechanism is possible it will add a small time delay to register based on the processing time required to register the timestamp and the refresh rate of the screen.

Timestamps have been effectively used in several studies as a way to characterize and measure one-way delay from a source to the viewer. vDelay (Bachhuber et al., 2018) employs EAN-8 formatted barcode timestamps encoded onto video frames and measures them at the receiver to calculate delay. AvCloak (Kryczka, Arefin, & Nahrstedt, 2013) uses EAN barcoded timestamps to measure the capture-to-display latency of a transmitted video frame using Google Hangouts.

A QR-code based timestamping solution called videoLat (Jansen & Bulterman, 2013) was developed as a flexible tool for latency measurement on Apple devices, enabling a user to set up a video stream on an external device to test a variety of black-box streaming configurations. The result of this was a remote media delivery system capable of transmitting video up to 1080p video quality at 30 frames per second (fps), with latencies as low as 240 ms obtained using the method. The open-source Latency-Clock library has been previously utilized for remote latency measurement on an autonomous platform. With the aid of the video utility GStreamer to develop a utility to transmit telemetric data while reporting on the latency of transmitted video, Osman Firat (2019) measured the latency in an autonomous bus located at a university campus using a

Linux based computer and Latency-Clock plugins, using cameras installed in a self-driving bus to transmit video to a remote receiver station via WebRTC.

To gain an understanding of latency expected during transmission of video on an autonomous agricultural machine, and how it varies between these two methods of transmission we forego a pure G2G measurement, overlaying a timestamp immediately after a video frame is captured and decoding the timestamp immediately before image playout. Eq. 1 illustrates the measurement used within the experiment.

$$T_{pipeline} = T_{encoding} + T_{network} + T_{propagation} + T_{decoding} \quad (1)$$

By keeping the encoding and decoding pipelines constant for all combinations used, in addition to gaining an understanding of overall latency, an understanding can be developed of the latency added for transmission and propagation over a cellular network or over a wireless radio link.

3 Test System Assembly

The objectives of the research work were to assemble a real-time video transmission system from a mobile agricultural machine to a monitoring station located adjacent to the field in order to measure the transmission latency associated with varying transmission distances and varying video resolutions. The following chapter details the design and assembly of the system used to complete this process and evaluate real-time video streaming.

3.1 Overview

For a system to be able to communicate remotely with an observer it is necessary to select a transmission method with which to transfer data. Based on the ranges expected and available technologies, transmission over a cellular network would seem to be a reasonable approach to accomplish this task, due to the considerable range over which it is possible to communicate using video through services such as FaceTime.

While cellular networks are widely available and video communication over cellular has been common for over a decade, cellular networks at the present time encounter limitations in many areas where agricultural work is common. Ample bandwidth and stable connections are not as readily available in rural locations as they are for major cities. For video transmission, which requires a significant throughput for data, these limitations pose significant potential issues for

observation of a remote farm vehicle. More remote areas may require additional network hops for the signal to travel through, which will introduce compounding queuing delays as the number of hops increases, impacting time it takes for data to reach the user from the machine's location. For these reasons the examination of a more direct transmission method is beneficial, as many of these networking issues are circumvented if the video were to be sent directly from a vehicle to the observer.

A trade-off for this more direct transmission will be a corresponding loss of distance. Transmission from a moving vehicle will require an omnidirectional antenna to transmit the signal as it is difficult to guarantee consistent alignment from the vehicle as it moves through an agricultural field for a directional antenna, illustrated in Figure 2.

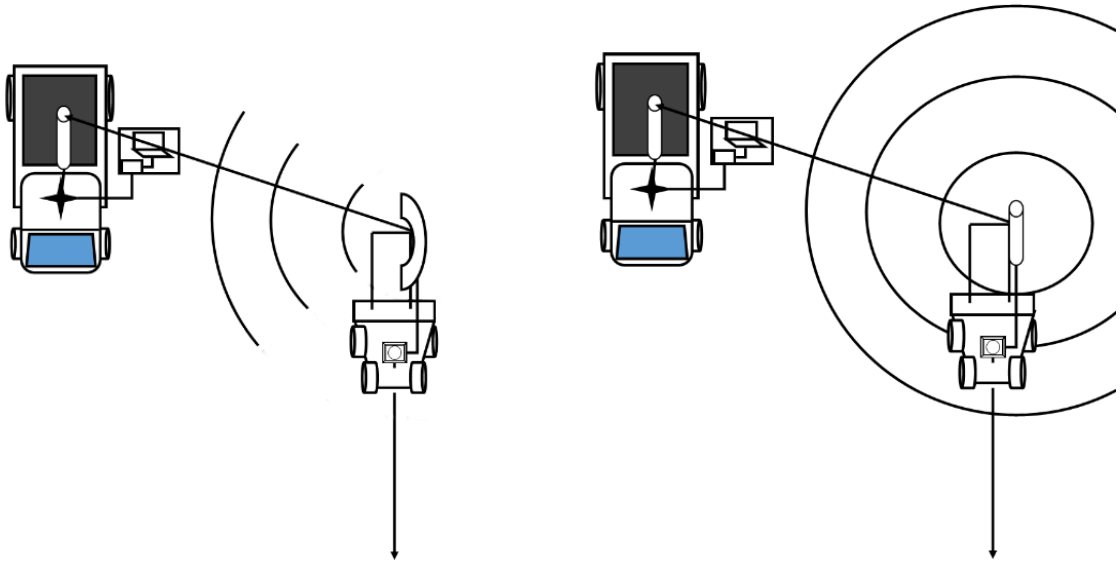

Figure 2: Image illustrating issue posed by difference between directivity of directional antenna and an omnidirectional antenna for a moving vehicle. As a directional antenna moves, alignment will eventually be lost without corrective measures.

Without being able to rely on a network to rebroadcast the signal, the range of a direct transmission will be much shorter, as video signals sent wirelessly will attenuate due to scattering and the spreading of the signal over a large distance due to the omnidirectional radiation of the data. As both of these methods have corresponding advantages and disadvantages, it becomes desirable to test both on a common platform to observe the delays

encountered with video transmission across both methods. The overall framework for the hypothetical transmission system is shown in Figure 3.

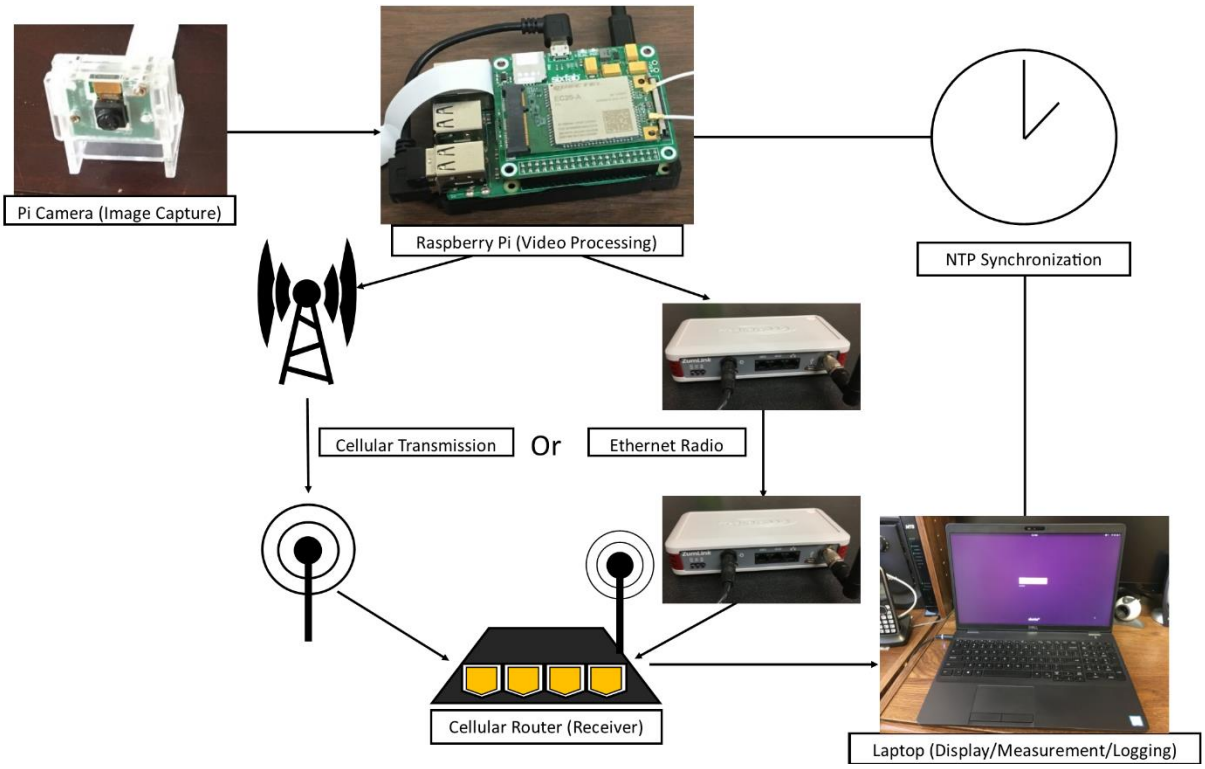

Figure 3: Diagram of the overall framework for the transmission system.

A Raspberry Pi 4 and Pi camera are mounted to a tractor mower and act as the capture device and encoder for video, transmitting a compressed live video feed from the front of the mower using a cellular or radio connection (depending on the trial being evaluated) to a cellular router attached to the other half of the radio pair. The router forwards the received video feed over Ethernet to a connected laptop, which displays the video and measures the latency by comparing the difference in local machine time to a timestamp encoded into the video frame. For this reason it is necessary to synchronize the time of both machines as closely as possible to minimize time differences.

A byproduct of the research and some discussion was the realization that an option for machine reading that would be automatic would be beneficial, as manual operations over many trials could be time consuming and require meticulous observation. Automated frame counting was considered, as was the method of aiming a camera at a connected screen such that it could record both a running timer and the displayed result from the timer. The dependence on a low light-

noise environment for automated frame counting (Friston & Steed, 2014) made it incompatible with field operation, while displaying a running timer would either require manual readings or otherwise require a specific library to be used or developed to facilitate it. As several libraries already existed to perform this kind of timestamp encoding and calculation, including barcode-based timestamp libraries like vDelay and Latency-Clock, it was decided to use an already existing and established method. An option was requested for a system that would be flexible and easily programmable; a Raspberry Pi met these standards due to the ease of configuration for the options available, including Raspivid, FFmpeg and the GStreamer library.

3.2 Raspberry Pi

The Raspberry Pi computing series are a family of single board computer developed by the Raspberry Pi Foundation in the UK. Created by the Pi Foundation as a way to promote general computer science literacy, the Raspberry Pi has become a popular tool which is used in a wide variety of computing applications as an easily accessible and portable Linux computer (Johnston & Cox, 2017). Studies in computer adjacent fields frequently employ Raspberry Pis for their flexibility and relatively low cost. While Raspberry Pis are based on the Linux Kernel and can run a variety of operating systems, for the purposes of this research the system used was the official Debian OS variant for the Raspberry Pi known as Raspbian. The relatively high adoption rate of the Raspberry Pi in comparison to other devices made it a preferred option as a processing platform for this project, as it provides a wide pool of extant implementations and solutions to various challenges to draw from.

Raspberry Pi systems have been used as a delivery mechanism for video in several previous studies. A study aiming to transmit video with under 200 ms latency using Raspberry Pi was conducted as a proof of concept for video transmission using IoT-based devices and H.264 video encoding (Jennehag, Forsstrom, & Fiordigigli, 2016). A potential application utilizing Raspberry Pi based video and comparing with an Android-based alternative for use in telecytology for medical purposes was devised by researchers where the system can deliver video of microscopic tissues to a remote researcher or observer (Bahaweres & Beni Santoso, 2016). Sub-400 ms latencies were achieved for both systems, with the Raspberry Pi ranging between 83 and 355 ms.

The Raspberry Pi 4 comes in sizes of 1, 2, and 4 GB RAM, operating off of a 3A 5V power source. It supports two micro HDMI ports, a total of 4 USB ports evenly divided between USB 2

and 3, a Gigabit Ethernet port and 40 GPIO (General Purpose In/Out) pins to attach various peripherals. As with earlier versions of the Raspberry Pi, the Pi 4 comes equipped with a Broadcom chip capable of H.264 hardware accelerated video encoding via OpenMAX. This feature is later used with the omxh264plugin in GStreamer within this project to reduce video processing load and expedite the conversion process to reduce latency introduced as a result of encoding delays.

The Raspberry Pi's low cost was a considerable asset for testing, as using it during the initial phases of the project kept options available for potential alternate options, in the event that unforeseen developments in testing would make it less than ideal. Initially it was to be used to explore working in Linux in preparation for an alternate video capture device such as an IP camera, but the considerable flexibility of the device and the ability to stream video at higher frame rates proved significant enough that it was adopted as the main video platform. A 99.6 Wh RavPower battery pack was sourced that could provide the Raspberry Pi 4 with power via a USB-C port, as the RPi4 requires a 5V 3A connection. A SixFab cellular shield was used to allow the Raspberry Pi to interface with a Quectel EC25 cellular modem. The modem was used with a SIM card to connect to a cellular network.

During initial testing with a Raspberry Pi 3B+, it was noticed the video appeared to drop frames during the recording process – while it could send video frames via the cellular network at a rate of 30 fps without the barcode timestamps, the addition of timestamps at 720p30fps would limit the average frame rate to around 15 frames, likely due to limitations in processing power for accommodating both video and the timestamping mechanism. A Raspberry Pi 4 was later tested and found to have sufficient processing power to eliminate this issue. From this came the realization that having a way to verify the framerate of the video during recording would be a considerable asset.

The Raspberry Pi was initially tested by connecting it to a Virtual Machine on a laptop running Windows 10 to test the connectivity and ability of the system to stream video data. The Raspberry Pi would transmit video over a Wi-Fi network to the laptop. Initially this was done using raspivid for testing, but later with a greater understanding of stream operation and Linux, GStreamer was used instead to allow for greater flexibility in construction of more complex pipelines and the implementation of timestamping to read latency via machine. The receiver used

was upgraded to a computer tower natively running Linux Ubuntu for test purposes, as running a virtual machine proved to be far too intensive for the laptop to decode video effectively, although it was useful as a proof of concept.

3.3 Camera Testing

Initially, a GoPro Hero5 Session available in the lab was tested as a possible camera. The camera was tested transmitting over Wi-Fi with Camera Suite and used in the lab environment to test transmission of data. As the Hero5 could only transmit video data over Wi-Fi, it was determined that it would not be suitable for testing in an environment where long range was required, although if some packet loss would be tolerated it could potentially work when paired with a cellular transmitter which could receive the wireless signal. Additionally, during this testing phase it was determined that the lack of any kind of timestamping feature would likely make it unsuitable for use during the experiments. The testing and research involved with the GoPro was useful, in that it led to the conclusion that a system would likely be required that was capable of some manner of overlaid timestamping, and more specifically able to timestamp video to at least a millisecond level of precision.

3.3.1 Pi Camera

The Raspberry Pi Camera is a 5 megapixel camera able to operate in 1080p30, 720p60, and 640x480p resolution at 60 fps (Pi Foundation, n.d.). Costing approximately $25 and having a maximum framerate of up to 90 fps, the Pi Camera captures video as a 10-bit RAW RGB image. The GPU of the Raspberry Pi can then convert the captured frames into other formats. The Pi camera attaches to the Raspberry Pi family via a CSI port, and can then be accessed through a range of applications, including raspivid and the rpicamsrc plugin library in GStreamer.

While USB webcams and cameras such as the GoPro Hero5 were examined and tested and in particular, the in-built H.264 hardware encoding found on some USB cameras was a desirable trait, the flexibility of the Pi camera for frame rate and resolution configuration, combined with the available libraries on the Raspberry Pi and the low cost made it very suitable for testing and development.

3.4 GStreamer

Once the camera system was selected it was necessary to select a method with which to control it. While initially the native raspivid functionality on Raspberry Pi was examined and tested, and

found to allow for flexible overlay of text including timestamps, it increasingly became apparent that machine readable timestamps would be desired.

GStreamer is a C-based multimedia framework used for streaming audiovisual content and has been frequently used in studies relating to video transmission for its utility and flexibility in the delivery of audio and visual content. Previous studies have made use of GStreamer as a method of implementing H.264 video encoding (Sundari, Bernatin, & Somani, 2015). GStreamer has also been used to create a real-time protocol based connection (Peltotalo, Harju, Vtminen, Bouazizi, & Curcio, 2010) acting as a bridge for additional video elements to create an alternate peer-to-peer streaming service capable of acting as a viable alternative to already extant solutions.

GStreamer multimedia pipelines are completed through the use of plug-ins which are assembled using "pads" to form an interconnected framework, moving video from the sink pad of a plugin to acting as the source of another. GStreamer's open-source development has allowed for the creation of a wide range of utilities which can stream video, live and prerecorded video and more. GStreamer allows a considerable amount of very precise control of particular aspects of incorporated plugins, and the command line functionality of the utility allowed for rapid prototyping and iteration of pipeline design for the project. While GStreamer has the ability to fork video pipelines to generate .mp4 and .mkv video files, such methods could add processing delays and on a Raspberry Pi could increase the load to the point of introducing a bottleneck in data flow, which may inhibit the data stream.

Real-time protocol-based streaming was implemented for the stream pipeline, as it was possible to access and configure security for all devices on the network to enable streaming and for video streaming from a remote farm vehicle it is most realistic to prioritize and evaluate transportation of video to a single overseer initially. WebRTC was also considered, as GStreamer does support development for the protocol and has various plugins available, however it was decided to first work with the simpler implementation of RTSP in GStreamer.

GStreamer H.264 encoding and decoding plugins were used to compress and transport the video data, making use of hardware accelerated encoding via the OpenMAX enabled omxh264 library to reduce the overall processing time compared to software encoding. The "Timestampoverlay" and "Timeoverlayparse" functions were critical for machine-enabled evaluation of latency in the

project, shown in Figure 4 in a simplified version of the overall path from capture to playout as Timestamping and Time Reading. Timestampoverlay places a 64 bit binary timestamp within the middle of the frame containing the current clock time of the processing device. Because the timestamping library requires RGB formatted video, it becomes necessary to capture video in RAW RGB format, and the pipeline was configured accordingly. Formatting differences between the GStreamer pads are negotiated using videoconvert GStreamer plugins to adjust the video to the appropriate I420 format required to enable hardware encoding.

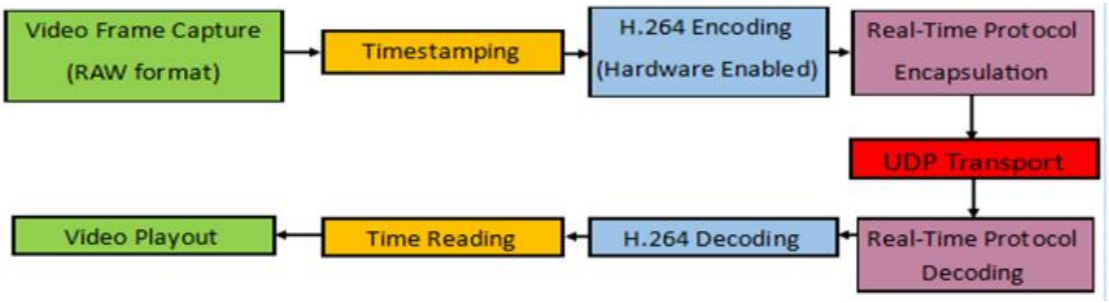

Figure 4: Diagram of key processes used to transport video and measure latency.

OpenGL is used to play the video feed for the observer via the glimagesink plugin, which was recorded using Open Broadcaster Suite to record the desktop. Latency readings were piped to a .csv file and later using a script in RStudio, along with measurements for fps and instantaneous throughput.

The GSTShark (Grüner & Soto, 2017) series of metric analyzers for GStreamer were implemented to capture the frames per second, bitrate and interpipeline latencies at each section of the GStreamer pipeline on the output side. Output for the framerate tracer corresponded to the frame count over the last second, while bitrate tracers would correspond to the instantaneous throughput, measured in bits per second. Both values were reported every second and logged along with latency data to a .csv file where they could later be extracted and analyzed. CSV files were processed using RStudio, extracting the latency and frames per second values and their associated measurement times for later data analysis. RStudio was also used to pair the auxiliary measurement data to the measured latency values, with the collected data being tied to the nearest latency value at that particular point in time.

3.4.1 Video Pipeline

For a hypothetical video delivery system to be able to meet the desired goal of being able to machine read timestamps while delivering a compressed video stream, certain between format conversions would be necessary. Figure 5 outlines the pipeline construction from the transmitter side.

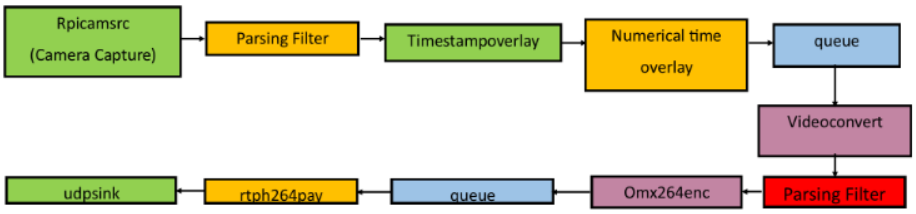

Figure 5: GStreamer transmitter pipeline.

The GStreamer pipeline receives video from the Raspberry Pi camera with the rpicamsrc plugin for GStreamer. The video is taken as RAW RGB formatted video, as that is the format for which the Timestampoverlay latency plugin is designed to work with. Once a timestamp is overlaid the video is compressed to H.264 formatted video, employing the hardware encoding capabilities of the Raspberry Pi to more efficiently process the video with the use of the omxh264 hardware acceleration plugin in GStreamer. The video is encapsulated in a real-time protocol format stream, and then packaged in a UDP datagram which is directed to the receiver. Figure 6 outlines the process for the receiving pipeline upon reception of the datagram.

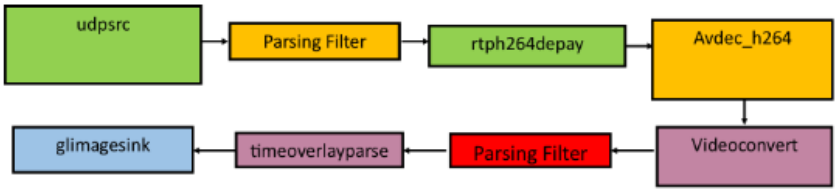

Figure 6: GStreamer receiver pipeline.

Upon receiving the datagram, either over cellular or radio depending on the instrument being used for transmission, the router is configured to forward the packet to the Ethernet connected laptop. The received RTP payload is received and decoded, with the H.264 payload further decoded leaving a RAW format stream, paralleling what was encoded on the transmitter side. This stream is then parsed for the current time from the timestamp displayed on the screen,

which is then subtracted from the current machine time. Figure 7, above, shows a constructed example command line for the receiver designed to play 480p resolution video.

```
GST_DEBUG="GST_TRACER:7" GST_TRACERS="framerate;bitrate;interlatency"
GST_DEBUG_FILE=/home/mitch/TimeTest1.csv gst-launch-1.0 -e -v --gst-debug=4  udpsrc port=4200
! 'application/x-rtp, encoding-name=H.264, width=640, height=480, payload=96'! rtph264depay !
avdec_H.264 ! videoconvert ! 'video/x-raw' ! timeoverlayparse ! glimagesink sync=false
```

Figure 7: Command line for decoding of timestamp and playing of 480p video.

GST_DEBUG controls various aspects of error logging and data collection and in the implemented pipeline logs framerate, bitrate, pipeline latency (both whole and broken down into subcomponents), and the overall measured latency output from the timeoverlayparse plugin into a .csv file. The pipeline receives video over UDP with the udpsrc plugin, unwrapping the received rtp packet and subsequently the H.264 packet containing the data frame, converting the video to a raw format, parsing the video for the timestamp, and finally outputting the video frame for OpenGL to display.

3.4.2 Latency-Clock

Latency-Clock is an open source timestamp based measurement method of latency evaluation developed by William Manley (2016), originally intended for use with the Stb-tester hardware for evaluation of Smart TVs. It consists of two libraries, Timestampoverlay and Timeoverlayparse, which are compiled as a GStreamer library with the Linux “make” command and can be included in the libraries utilized by GStreamer once compiled. Timestampoverlay inserts a 64 bit binary timestamp containing different time components measured from the GStreamer application running at the transmitter into a captured RAW video frame. Timeoverlayparse, when used in a GStreamer pipeline, decodes the timestamp and renders the output by subtracting the decoded time from the current machine time. The Latency-Clock plugin can be run on a variety of Linux platforms which are capable of running GStreamer, which makes it a flexible method of measuring latency through a machine-read approach.

The timestampoverlay library used was modified slightly to accommodate project needs. Latency values were originally rendered in UNIX time since 1970, but a modification to the receiver code used to decode timestamps, porting over a clock conversion function which was used to render a clock as the realtime value used in the transmitter code over to the receiver code rendered latency as a human readable number. While the output code for timestamp decoding initially provides

the latency values along with several other clock variables encoded within, the output code was simplified to print the system latency and the time at which the latency was read, in standard UNIX output. This time would be useful later as a reference for pairing the measured latency with factors such as the Raspberry Pi temperature and the current fps during analysis.

3.4.3 Handling of Frame Corruption

Images that have become distorted or impossible to read, such as out of order frames sent over UDP which are displayed as a grey screen in GStreamer will render values close to the maximum possible UNIX time. Assuming that any latency measurement received larger than the current stream run-time would correspond to an impossible reading, this can be used to filter video frames which arrived in a distorted state, and gauge the relative performance of different trial sets. An example of a frame that cannot be properly read is shown below, in Figure 8.

Figure 8: Example image for a dropped or corrupted frame.

3.5 Cellular Transmission

A SixFab cellular shield with a Quectel EC-25 modem was used to allow the Raspberry Pi to stream video data over cellular networks. Buffering due to poor connectivity or traffic congestion are potential issues with the cellular transmission and were observed during pre-experimental testing, and are issues that can impact the quality of service. To transmit, a pair of u.FL SMT antennas were attached to the 50Ω main and diversity antenna ports located on the modem.

A BulletPlusAC router was used to receive the video feed through the cellular connection. The router was configured to forward all data packets at port 4200 to the receiving laptop, which was connected via an Ethernet connection. GPS tracking functionalities of the router were used to

estimate receiver positioning, sent periodically over UDP to the receiver laptop along with the received signal strength for the router at the same point in time. The BulletPlusAC router was configured to be able to receive and forward packets to the connected laptop, both with the connected cellular header and the connected Ethernet radio described in the next section, to keep the packet travel path relatively consistent between devices upon reception.

3.6 Ethernet Radio – the ZumLink 900

Direct transmission between the vehicle and observer is accomplished with a device capable of radiating omnidirectionally between a transmitter and receiver. After a search through various components capable of direct transmission, the ZumLink 900 was selected as the component to test with due to a combination of range, flexibility and low cost which made it the most feasible to use. The ZumLink 900 Ethernet Radio series operates between the 900-928 MHz frequency ranges. It can use Frequency Hop Spread Spectrum technology to adjust signal transmission rapidly over differing preset frequencies, which serves to limit interference from outside signals transmitting on the same channel. While the ZumLink can function as an interconnected network comprised of multiple radios, a single pair of radios was used for testing here due to cost restrictions and for testing practicality.

While the ZumLink 900 has a lower data rate than the Quectel EC25 cellular modem (4 Mbit reported by the manufacturer, compared to the higher 25-50 Mbit of the cellular module), the radio system has the benefit of independence of any kind of cellular network quality in the delivery of data. The single required hop compared to the multi-hop traversal necessitated for cellular network transmission implies that the ZumLink 900 would theoretically have lower transmission latencies, in exchange for a shorter overall possible range of transmission as a result of the lack of devices to retransmit, and the lower height for a receiving antenna compared to already extant cellular infrastructure.

900 MHz Ethernet radio was selected as a transmission method due to the desirable properties of UHF signals relating to signal attenuation from path loss, expressed in the below formula, and the ability for the signal to penetrate through obstacles when compared with 2.4 GHz transmission.

$$\text{Path Loss } = 10\log((4\pi df)/c)^2 - Greceiver - Gtransmitter \quad (2)$$

From the above, theoretical free space loss for the higher frequency 2.4 GHz cellular communication with wavelength 0.125 m will be significantly larger than for 900 MHz with wavelength 0.33 m. While this might in theory allow for a longer available transmission distance, the longer wavelength requires additional height to fully mitigate ground and obstacle reflections of radio waves, which are difficult to provide in a flat, open environment where mobility is a significant priority for both sender and receiver. The ZumLinks were configured to transmit 30 dBm of power using the maximum available data rate of 4 Mbit during transmission, although this later was reduced to a maximum data rate of 1 Mbit due to challenges encountered when testing with the device in the field (later covered in the results section). Canadian requirements for transmitted antenna power limit the total EIRP which can be transmitted to be within 4 W, or 36 dBm. For this reason, an omnidirectional antenna with gain of 5 dBi was used, providing a maximum transmission power of 35 dBm when considering the maximum available radio power of 30 dBm. Frequency hopping was disabled per the recommendations of the manufacturer to optimize transmission range.

A large portion of the initial testing of the radio was completed in a lab setting. While cellular transmission allowed both the Raspberry Pi and the receiver system to be taken into the same vehicle for testing while technically maintaining a large transmission distance between devices, the same could not be said for radio transmission, which required 12V DC power sources at each end and a person to man each station to safeguard the equipment. Securing dedicated power supplies required additional time, and due to this and weather interference, radio transmission could not be tested outdoors at a reasonable distance until later into the project lifespan.

3.7 Receiver and Power Components

While initially a computer tower with a Linux install was used to test the functionality of the components, it became clear during the process that an alternative was required. Field testing severely limited the available power sources and required minimization of the number of outlets that might be required to power systems for testing over the course of a day. To increase portability, it was decided to use a laptop in place of a tower and monitor that would be required to work out in the field. A Dell laptop running Ubuntu version 18.04, with an Intel Core i7 processor and 8 GB RAM was selected, and used to decode and play the video. The laptop CPU

ran at 800 MHz during testing. To power the laptop, Ethernet radio and BulletPlus cellular router, an available Power-It! battery generator was borrowed from another research group.

3.8 Python Control Interface

To facilitate testing, a Python program was written to send and terminate Linux shell commands to the Raspberry Pi to run scripts which read data from peripheral devices. For cellular transmission, scripts were configured to read RSSI values using AT+CSQ commands, while for radio transmission corresponding RSSI values were read via Modbus. The core temperature of the Raspberry Pi was also polled to examine whether performance could be impacted by excessive CPU temperatures. The script establishes the appropriate GStreamer video pipeline, wrapping all of the underlying components into a simple interface using the TkInter library in Python to execute data collection scripts before initializing the stream on the Raspberry Pi. Cellular RSSI and GPS position values from the Bulletplus Cellular router were obtained through a UDP socket connection over Ethernet and polled every 5 s. The program was able to use NTP protocols for checking time, using ntpdate queries between the Pi and the laptop.

Commands for the receiver pipeline in GStreamer were initially called using the subprocess module in Python which allows for a program to relay commands directly to the shell terminal. Testing at 60 fps using the Python application showed an issue where measured latency continued to build steadily at a linear rate. This issue was not seen below 50 fps and increasing the frame rate past 60 increased the severity of the linear increase. This linear increase would persist until reaching a threshold after which a frame would appear to be dropped and the latency would dip slightly, before increasing up to the threshold. An image of the application as it looks after generation in Python is shown below, in Figure 9.

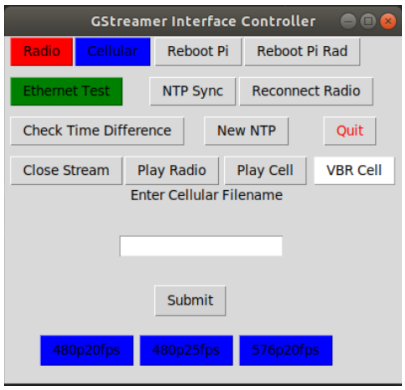

Figure 9: Image of TkInter interface.

The interface allows the user to select the method of streaming, at which point an input for a filename is called which will be used to label all generated data. After the entry of a filename the user can select the level of video quality desired. Functions for executing scripts and for controlling the remote transmission of video were controlled at the command line level using the subprocess and paramiko libraries in Python.

Options were also provided to enable reconnection to the radio using the cellular link, as transmission with the cellular network required the Ethernet interface on the Raspberry Pi to be disabled while plugged into the radio. Buttons to remotely reboot the Raspberry Pi over both the cellular and radio links were added, and once enabled a stream can be closed with the click of a generated button. No button to shut the Pi down was implemented, as a remote startup method for the Pi was not considered as part of the experiment, so the driver would need to manually reenergize the Pi in the event that power was disabled at range or the Pi became unresponsive.

3.9 NTP Synchronization

Clock synchronization for both transmitting and receiving devices was a critical component of latency measurement. The standard ntp daemon for Linux was used as a time reference at the receiver. The ntpdate -b command was used on the Raspberry Pi through a remote SSH connection to bind the clock time of the Raspberry Pi to the receiving laptop. While initially the time difference reported by an ntp query would be on the order of a fraction of a millisecond, a clock drift could be seen to occur. The clock drift was measured before and after each trial, where the Raspberry Pi was resynchronized to the receiver, in a similar fashion to latency measurements for AvCloak (Kryczka et al., 2013). While this would only reconcile the times to the level of milliseconds, clock drift during the 3 minute run-time would be minimal.

Re-synchronization was done over the ZumLink system to limit the number of required hops over a black box network and keep an approximately symmetrical network for time measurement. An ideal method of controlling the reference time on both devices in an outdoor environment would be the addition of a GPS capable of pulse per second (PPS) transmission, which has been demonstrated to allow for synchronization on the order of microseconds. The laptop used has no GPIO pins which would allow for the implementation of PPS through a serial

line, and the Raspberry Pi's TX/RX pins were necessary for the use of the cellular shield used for some transmission of video data.

3.10 System Housing

A modified lawn mower was used as the platform for the Raspberry Pi video setup. The Ethernet radio was mounted at the rear of the vehicle, and connected to the Raspberry Pi camera system via Ethernet cable, with the Pi held within a small enclosure mounted to the front of the vehicle, shown in Figure 10. The Raspberry Pi camera was placed in a box positioned at the front of the modified lawn mower, with the camera facing outward aligned with the direction of movement of the vehicle. Holes were cut in the box for the antennas, and for an Ethernet cable to connect to the Pi.

Figure 10: Front view of mower and camera box.

The ZumLink radio and antenna were affixed to an adjustable aluminum frame attached to the back of the mower as high as possible to emulate a vehicle antenna and to mitigate ground reflections as much as possible. The ZumLink radio was powered by an 110V AC inverter connected to the battery of the mower, to which the power supply that the radio came with was attached. A photo of the radio and 5dBi gain antenna used can be shown in Figure 11, with the Zumlink Radio system affixed to a project box attached to a movable metal frame from which the antenna could be adjusted. A photo of the open Pi case, showing the battery used along with the Pi, the antennas used for the cellular header and the front mounted Pi Camera can be shown attached to the mower in Figure 12.


Figure 11: View of mower from back, with radio and antenna mounted.

Figure 12: Image of the Raspberry Pi enclosure with lid removed.

On the receiver side, the Dell laptop was connected to the cellular router through an Ethernet cable. A portable power bank was used to power the router, radio, and laptop across all trials.

3.11 Ancillary Data Collection

The received signal strength (RSSI) was periodically measured on both transmitter and receiver, continuously polled from the equipment in use at the time and measured in decibel-milliwatts (dBm). The methods of RSSI reporting used by the devices varied and required additional post-processing upon collection to be converted to the appropriate dBm levels. The cellular header used on the Raspberry Pi could only report RSSI data to the nearest dBm. The ZumLink radio systems used output their values as a 16 bit signed integer, while the Bulletplus reported RSSI values in 4 bit hexadecimal format to two decimal places.

A uBlox GNSS200L GPS was connected to a USB port on the Raspberry Pi 4 to gather location data. NMEA data were sent containing the location. Python was used to interface with the uBlox over a USB connection with the serial library, with the baud rate set to 9600.

CPU Temperature of the Raspberry Pi, RSSI for transmitting and receiving devices, and the current GPS location of the vehicle and receiver station were all obtained during the experiment.

RSSI for the radios was obtained via a Modbus TCP connection requesting a periodic report, while the CPU temperature of the Raspberry Pi was obtained using the Pi's own internal temperature sensor and Raspberry Pi Libraries for Python. These parameters were each polled every 5 s. A Python script was written for the receiver and transmitter ends that would complete data collection and log data for later use in a .csv file.

Data collection scripts were synchronized to begin data collection at the same clock time in Python – arbitrarily selected to run on the 8th second of the current time (i.e., beginning at 10:10:18 or 10:10:28). To mitigate a potential timing issue where a script on the Raspberry Pi could miss the timing due to requiring a transmitted signal from the laptop at runtime which would potentially arrive with a delay, the scripts were configured to run for longer than 3 min. This provided a time buffer where, in the event that the script did miss the initial timing due to a delay in signal transmission, data collection could instead be considered to begin at the first time where all scripts were running. The transmitter script would terminate the video stream from the Raspberry Pi by inputting a "kill" command into the shell directed at the GStreamer process, which would also serve as a signal to the observer to terminate the stream on the receiver side once the receiver ran out of frames to display.

3.12 Glass-to-Glass latency measurement

Full glass-to-glass latency was measured for the system by aiming the camera at a laptop screen running at 60 Hz such that it would capture the current video as it was displayed and also a previous video frame, and manually subtracting the two timestamps. Measurements were taken every 10 s over a total of 3 min time. Table 1 documents the results of testing and the values obtained through measurement.

Table 1: Comparison of latencies measured during camera test via Ethernet using Glass-to-Glass and timestamp measurements.

Parameter	480p Trial 1	480p Trial 2	480p Trial 3	576p Trial 1	576p Trial 2	576p Trial 3
Mean G2G Latency (ms)	163.8	155.5	163.9	216.6	219.4	213.9
Mean Encode-Decode Latency (ms)	67.1	61.1	68.0	109.5	114.3	109.8
Difference (ms)	96.7	94.4	95.9	107.1	105.1	104.1

The measured differences between G2G and the encoding to decoding latency ran between 94 and 104.1 ms. At 20 fps, the total capture time for the camera would correspond to 50 ms. Similarly, a frame of video must be displayed on the monitor within 50 ms, before subsequent

frames arrive from downstream. The measured difference between the encoding and decoding of the pipeline over an Ethernet cable, passing through the router reinforces this. The difference in times measured is around 95 ms in the case of 480p at 20 fps and 105 ms for 576p at 20 fps, which is roughly consistent. At 20 fps, the margin for error between frames is significant, with each frame having an error of +/- 50 ms in the reading due to capture time, in addition to an error introduced by the screen refresh rate (+/- 16.667 ms at 60 Hz).

3.13 Road Testing

Once the system was assembled, it became necessary to verify the ability for the system to viably transmit video over long distances. While the system could safely transmit data over a cellular network within the lab environment, it was not known how it would perform in motion in areas that could have a less robust cellular environment than a university located in the middle of a city, where LTE availability was fairly reliable. Figure 13 shows an example image taken during the test drive, with the overlaid timestamp visible in the center of the video frame.

Figure 13: Screenshot of 720p30fps video test taken during test drive, near Landmark, Manitoba.

The Raspberry Pi was converted into a makeshift dashcam using a RavPower portable battery, while positioned on a temporary platform for the equipment affixed to the dashboard of the vehicle. A viewer stationed in the laboratory at the University of Manitoba with the cellular router had full control over the Pi's ability to stop and start video streaming via a secure shell connection. Coordinating via cell phone, the operator of the vehicle drove outside the city,

periodically checking in to ensure that the camera was still able to be viewed by the observer. During this time, video of the vehicle in motion was being streamed in real-time to the viewer over UDP, while latency was being measured via the timestamps encoded into the video feed. Table 2 summarizes the measured results from the initial test drive.

Table 2: Results from the testing drive from Winnipeg to Landmark.

Drive Section	Frames	Unreadable Frames	Error Rate (%)	Minimum Latency (s)	Maximum Latency (s)	Mean Latency (s)	Median Latency (s)	Standard Deviation (s)
1	24213	942	3.9	0.376	14.585	0.606	0.480	0.826
2	16166	607	3.8	0.372	17.003	0.688	0.440	1.426
3	30737	427	1.4	0.204	3.518	0.503	0.478	0.244
4	10627	185	1.7	0.363	3.769	0.517	0.511	0.109
5	15375	857	5.6	0.455	19.952	0.608	0.505	1.274
6	2565	59	2.3	0.472	0.772	0.510	0.507	0.024

In the above table, the total frames read count the total number of received frames over the duration of the test drive. If a frame could not be successfully read, identified as having an unreasonably high latency value above a particular threshold (in this instance set at 180 s), it would be added to the unreadable frames count. The error rate was defined as the unreadable frames divided by the total frames multiplied by 100, with the remaining measurements corresponding to minimum, maximum, mean and median latencies, to gauge the performance of the system.

A later road test was conducted where both sender and receiver units were mobile within the same car, transmitting video data over a cellular connection in a moving vehicle with the Pi Camera again used as a cellular streaming dash camera, with the laptop and router both powered by the portable battery generator. The operator would periodically begin a stream while the vehicle was parked in a safe location, where latency could then be recorded as the vehicle travelled down a busy highway across southern Manitoba. Periodically the driver would pull over to a safe location to end the stream and begin a new segment covering another leg of the highway as the vehicle travelled. The results were later collected and analyzed, shown below in Table 3.

Table 3: Results from the cross province drive from Winnipeg to Brandon.

Drive Section	Frames	Unreadable Frames	Error Rate (%)	Minimum Latency (s)	Maximum Latency (s)	Mean Latency (s)	Median Latency (s)	Standard Deviation (s)
1	22773	1941	8.5	0.409	81.933	3.347	0.517	9.555
2	23822	527	2.2	0.340	20.230	2.224	0.494	3.985
3	6906	87	1.3	0.397	1.019	0.438	0.437	0.020
4	25653	1336	5.2	0.375	7.490	0.438	0.433	0.064

The results from these drives showed a median latency of between 0.430 and 0.520 s with means which were highly influenced by spikes in measured system latency. Delays over the network in the worst case drove the mean latency to be as high as 3.347 s during these experiments.

From these early tests it became apparent that running 720p30fps with the timestamping mechanism active caused the frame rate of the Raspberry Pi 3B+ to be reduced to an average rate of 15 fps instead of the expected 30 fps. As a result of this, a realization was made that an automatic method of calculating framerate would be useful to gain an understanding of system performance, and the system was later upgraded to a Raspberry Pi 4, with the framerate tracers incorporated into the GStreamer pipeline using GSTShark. The adoption of the newer, more powerful Raspberry Pi 4 eliminated the framerate reduction issue posed by the earlier model.

3.14 Field Testing

Testing of the radio system in an outdoor environment proved challenging as it required a relatively large amount of distance between the antennas. Field testing of the system showed numerous issues when transmitting at the maximum bitrate of 4 Mbit (Megabit) with omnidirectional antennas. When testing with the mower configuration, it was observed that the range for quality video was extremely limited, with relatively consistent transmission of video only up to approximately 200 m, after which the video quality degraded severely and the arrival of frames became increasingly intermittent, up to a maximum of around 400 m where the video feed would cut entirely.

Consulting with the manufacturer of the radio, it was decided to drop the maximum data rate for the radios down to a lower 1 Mbit setting. To compensate for this and to reduce the likelihood of data rates exceeding the 1 Mbit capacity, the qualities and bitrates for video were adjusted accordingly, adopting a constant bitrate encoding scheme as opposed to the previously used

variable bitrate, which had been observed to exceed the 1 Mbit bandwidth limitation occasionally during streaming. To facilitate constant bitrate encoding in GStreamer, the required profile for video transmission was converted to using exclusively a baseline profile for the stream. It was observed that testing a constant bitrate required a larger amount of time to process than variable bitrate video transmission. At the previously used framerates, this had the effect of introducing a continuously mounting latency, as new frames were introduced from the camera faster than the system could process. Consequently, it became necessary to reduce the framerates and upper resolution used to allow sufficient time for the encoding process to be completed.

Conducting trials in areas of limited cell strength showed significant limitations in the ability to communicate remotely with the Raspberry Pi over SSL connections when the connection was reduced. During streams the connection might periodically become unresponsive or cease to respond at all for a continued length of time. For this reason, a command was added at the end of data collection scripts to automatically turn off the stream, at which point it would be possible to SSL into the Raspberry Pi via the Ethernet radio to remotely send commands. NTP synchronization proved to be a persistent obstacle over the course of the experiment for gathering accurate time data. Time synchronization over USB or Ethernet is reasonable in most situations, but is not ideal for the measurement of latency, where bus-level delays from transportation of signals can add milliseconds of latency which are problematic. Ideally the addition of a pulse per second source of time such as a GPS module would mitigate this, but all available options which were found either required use of the already occupied TX and RX pins for the Raspberry Pi using something like an Adafruit Ultimate GPS to synchronize time with a PPS connection, or were cost prohibitive and required access to the already occupied Ethernet port. It may have been beneficial to use a cellular router for both sending and receiving of a signal if a second router could be made available, as in addition to only requiring the use of the Ethernet port at that point, it would provide symmetry to the devices used in the experiment for both signal transmission and receiving.

Communication from a ground device to a cellular tower has the advantage of a greater overall angle limiting wave reflections from the ground or other obstructions. Depending on the position of sender and receiver at a given time, waves can constructively or destructively interfere with one another. While this is commonly mitigated through placing antennas at a height such that

they would be out of range of most interference in the Fresnel zone, this was not considered to be a viable option. The height required to be outside of most interference is larger for lower frequencies, and even at 5.9 GHz where the height required is significantly lower at approximately 9 m, the height is still wildly outside of most average farm equipment. Limitations encountered would be similar to what might be expected in vehicle to vehicle communication.

A possible explanation for this may arise from the location in which testing was completed; at the University of Manitoba to allow for a maximum distance of 600 m the testing had to take place with the mower driving parallel and proximal to a long treeline. Some manner of interference from the nearby trees may have contributed to this considerable drop in frames. The formula for receiver sensitivity is shown below, expressed as a function of temperature, bandwidth, the noise floor of the instrument, and the carrier to noise ratio.

$$Receiver\ Sensitivity = 10\log(kTB) + NF + \frac{C}{N} \quad (3)$$

The larger bandwidth may have raised the sensitivity of the radio antenna. This by itself does not necessarily explain the source of the range limitations that were encountered, as equipment such as drones are known to be able to transmit video multiple kilometers via radio connections. The height advantage given by a drone may circumvent issues posed by reflections from obstacles and from the ground that might reduce the overall transmission range of the antenna.

A test system was assembled using a Raspberry Pi that was successfully able to wirelessly transmit data over cellular and Ethernet radio links, and measure the latency incurred between encoding of the captured video frame and immediately before display of the video frame. From this it became necessary to design the testing plan for the system and how to appropriately measure latency during field operation.

4 Methodology

The following chapter details the process used to design the experiment, beginning with a brief overview before moving to a description of the testing procedure employed and the experimental factors used in the evaluation, before discussing some of the limitations experienced.

4.1 Testing Procedure

To perform testing of the equipment, a smaller vehicle would be used to make passes in a field similarly to the back and forth motions of a typical agricultural machine during spraying or harvest. The vehicle would make passes back and forth for 3.5 min while video was streamed. Only the first 3 min of video for which both data collection scripts were active would be counted, due to possible variability in sending the control signal required to start the collection on the Raspberry Pi. Ethernet radio antennas were placed on the roof of the truck used to transport equipment, suspended approximately 2.3 m (7 ft) above the ground, and the bracket holding the antenna for the mower was placed as high as could reasonably be accommodated, both to best approximate the likely positions of antennas on the equivalent vehicles in question and to mitigate as best as possible the considerable height differences between a cellular receiver tower and the available height for the radio receiver. An example image of the setup and positioning of the truck used is shown below in Figure 14.

Figure 14: Image of the truck with antenna mounted and receiver laptop with equipment.

The procedure would first begin with powering on the devices and waiting for the clock on the receiver side to settle, such that measured time differences between the two devices changed very little over a period of time when the transmitter clock was stepped to that of the receiver.

Once the clocks were within an acceptable range, the experiment would begin with stepping of the clock followed by collecting three measurements between devices corresponding to the time

difference using ntpdate –q. The experiment would be signaled to commence via cell (in the case of Elm Creek, the experiment was signaled via walkie-talkie due to periodic difficulties communicating over the cellular network). Once the vehicle was confirmed to be ready to move, the stream was initialized using the Python interface. The vehicle would make passes back and forth down the length of the available trail at the location, turning at the end of the pass and returning back to start before repeating the process again. This would last for the 210 s period programmed into the Python interface, after which the stream would terminate. The speed of the vehicle was set similarly at all trial locations, such that by the end of the run time of the stream, the vehicle would be at the starting location lining up to begin a new measurement trial. After streaming was completed, ntpdate –q would again be executed to verify by how much the clock had drifted.

It was not possible to run the same distance for the trial vehicle for all locations due to varying amounts of available space at each location. Experiments proceeded sequentially, with radio testing completed first to leave more time available for cellular transmission, for which testing showed that the process of establishing a connection could run into issues in areas with a lower signal strength, potentially requiring multiple tries to successfully begin the stream.

4.2 Experimental Factors

To successfully collect the experimental data, test plots approximately 600 m which were sufficiently wide enough to allow a tractor mower to make back and forth passes were required. It was desired to collect data at different geographical locations based on the assumption that transmission latency would likely vary due to local environmental conditions and cell signal strength. Through contacts with the Faculty of Agricultural & Food Sciences, appropriate test plots were identified at The Point on the Fort Garry Campus of the University of Manitoba (hereafter referred to as “UM campus”) and at the University of Manitoba’s Glenlea Research Station (hereafter referred to as “Glenlea”). Through personal contacts with staff in the Department of Biosystems Engineering, two additional plots were identified near the communities of Elm Creek, MB (hereafter referred to as “Elm Creek”) and Sanford, MB (hereafter referred to as “Sanford”). The GPS coordinates of each testing site are shown below, in Table 4.

Table 4: Average longitude and latitude for all locations used.

Location	Longitude	Latitude
Elm Creek	9801.71361	4936.82236
Glenlea	9708.38856	4938.21741
Sanford	9726.80371	4940.89970
UM	9707.16156	4948.80536

Testing distances were selected based on the maximum distances of the locations available for testing. In particular, while testing on the UM campus, the equipment was limited to a line of sight distance of 600 m while operating in an agricultural research area. While cellular transmission is likely able to function over significantly further distances than Ethernet radio will be able as a result of the significantly higher position and gain of receiver towers, securing space to test this was difficult. It is reasonable to state that during the test drives documented in the previous chapter, the variable distances at which the car was located along with the considerable distances the signal travelled (previously performing a traceroute with the SIM card indicated the signal passed though the Toronto area) combined were several times greater than the advertised range of the Ethernet radio of 96 km. The receiver station was placed at a stationary location, which would meet the distance requirements as best as possible while having a clear line of sight to the vehicle to reduce the impact of objects such as trees which would reduce the overall transmission range of the radio. Figure 15 shows a reduced diagram documenting the distances and the perpendicular path of movement truck and radio.

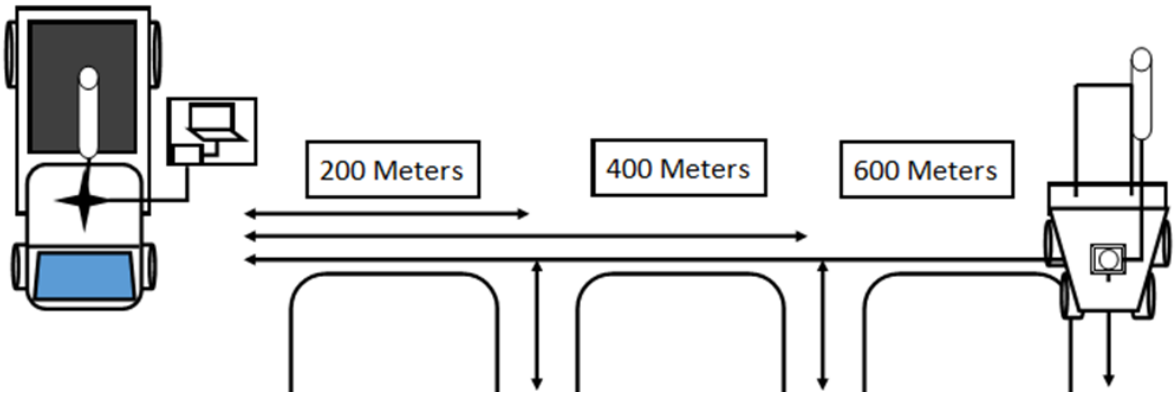

Figure 15: Simplified diagram of the test setup.

Video resolutions were selected according to typical recommended resolutions and bitrates but needed to be reduced to accommodate the lower than expected transmission bandwidth of the radio. Three different video quality standards were selected to operate within the bandwidth constraints of the device. A lower quality 480p stream running at 20 fps, a stream at the same resolution running at 25 fps, and a 576p quality video stream running at 20 fps were all adopted as measurement benchmarks for the trial after testing with the system. Trials were completed on the UM campus, Glenlea, Elm Creek, and Sanford, and were repeated three times each to match standards for similar studies. Table 5 summarizes the video test parameters used.

Table 5: Video test parameters corresponding to low, medium, and high qualities.

Quality	Video Parameters	Aspect Ratio
Low	480p20fps@400kbit	4:3
Medium	480p25fps@500kbit	4:3
High	576p20fps@600kbit	16:9

Distance away from the receiving vehicle was varied, with the mower's position varying from 200, to 400, to 600 m away. In the case of the test site at Sanford, the adjacent river restricted the third transmission distance to 540 m (rather than the 600 m tested at other geographical locations). From these factor combinations and with trial repetition a total of 214 trials were conducted, with two trials unable to be completed in Elm Creek due to cellular connection issues. Distance and resolution factors both were required to be altered due to the limitations of the Ethernet radio encountered during field testing.

Testing combinations of specific location, distance and video quality were repeated for both cellular and radio, with the mower moving along the same route to keep the scene as consistent as possible. Factors such as time of day and the current weather could impact the visuals captured by the Pi Camera, which could correspondingly impact the relative bitrate, but such factors cannot be controlled in a real-world environment and so were not considered in the experiment beyond not operating during rainy conditions outdoors.

4.3 Limitations

Cellular transmission takes place over a network that is essentially a black box – it cannot generally be modified or manipulated by the user, and while steps can be taken to minimize the

impact of additional users, cellular network testing is dependent on external factors which cannot be controlled.

The timestampoverlay function used to calculate the latency begins immediately after a video frame is already captured by the Raspberry Pi and decodes shortly before playing the frame out for observation from the user. There can be a loss of approximately 20-40 ms in the time, depending on the proportionate video quality and framerate being used in the video stream. It will therefore always underestimate a pure glass-to-glass latency measurement by at least the time taken to initially capture and display a frame of video, in exchange for higher precision, larger sampling sizes and independence from the refresh rate of a screen when compared with traditional methods of latency analysis. The lack of a checksum mechanism in the timestamp is also a possible issue with regards to corruption of a video frame.

A pair of 5 dB omnidirectional antennas were used for radio transmission to maximize coverage for the ZumLink setup without requiring specific antenna positioning that more directional antennas would require. This came at the cost of the overall range of the radio, which had a maximum power of 1 W (30 dBm) available, conforming with Canadian regulations for limitations on the amount of input and radiated power (Craig, 2014). The maximum data rate of the radio was 4 Mbit/s, which limited the overall data rates that could be explored during testing with the radio system.

To limit synchronization issues and to prioritize the measurement of visual latency, sound was not considered as a factor in the experiment. While sound is an important factor in machine operation for the detection of malfunctions and proper machine operation, coordinating sound and video can introduce synchronization issues and for the purposes of latency measurement it was preferable to prioritize video latency only. It bears mentioning that due to the relatively low data rate required for audio compared to video that the increase in required throughput would be low when compared with video data alone.

The study originally intended to use an autonomous ATV designed by the University of Manitoba agBOT team as the test platform for the camera, although circumstances did not allow for this. An RTK-GPS was to be made available through the same group which would allow for location tracking, and for similar reasons was not able to be implemented. A less accurate uBlox USB GPS was used instead as a substitute.

5 Results

Data collection took place over a one month period after testing of the radio was completed and appropriate adjustments were made to the factors used. In total, 214 latency tests were completed, with two tests in the Elm Creek area unable to be completed due to difficulties connecting over the cellular network. The remainder of the chapter covers the data which were gathered over the course of the experiment, breaking down the data by location through a mix of graphical and numerical analysis.

While the total end-to-end delay measured from the results cannot be taken as universal, due to potential variance in test equipment and conditions at different locations, the measurements obtained provide an impression of what can be possible for real-time video transmission. Collection of data at each trial location typically lasted one to two days while conducting trials. Distances were measured via GPS, and the path for movement of the mower was set to be approximately perpendicular to the distance from the measurement station.

5.1 Mean Latency Data

Incoming latency values were recorded for each incoming frame by the data acquisition system. For each experimental replicate, the mean latency was calculated as the mean of all readings recorded for that replicate. Means for each replicate (labelled as R1, R2 and R3) and the mean latency for all three replicates are provided in Tables 6 through 9.

Table 6: Mean latency values with standard deviations for UM campus.

		Latency and Standard Deviation Values (ms)											
		200 m Transmission				**400 m Transmission**				**600 m Transmission**			
Type	**Resolution**	R1	R2	R3	**Mean**	R1	R2	R3	**Mean**	R1	R2	R3	**Mean**
Cellular	480p20	122 (9)	128 (47.1)	122 (8.5)	**124 (28.2)**	131 (66.1)	128 (11.9)	129 (27.2)	**129 (42)**	131 (7.5)	141 (7.5)	129 (8.9)	**133 (6.7)**
	480p25	136 (9.9)	135 (9.9)	137 (14.2)	**136 (11.5)**	142 (15.8)	137 (9.2)	144 (33.6)	**141 (21.9)**	143 (10.6)	137 (10.8)	140 (24.5)	**140 (16.6)**
	576p20	180 (19.8)	179 (21.4)	191 (94.6)	**183 (56.8)**	193 (21)	348 (25.1)	194 (67.7)	**245 (43.7)**	180 (16.3)	180 (15)	212 (299.6)	**191 (170.1)**
Radio	480p20	97 (56.8)	107 (110.9)	104 (103.9)	**102 (94.5)**	96 (10)	93 (7.9)	93 (8.7)	**94 (8.9)**	94 (50.4)	151 (92.1)	117 (112.2)	**121 (88.2)**
	480p25	142 (117.8)	338 (85.7)	129 (54.4)	**203 (90.4)**	121 (35.9)	123 (32)	120 (37.1)	**121 (35.1)**	234 (143.3)	192 (147.5)	207 (131.6)	**211 (141.1)**
	576p20	215 (109.5)	220 (85)	235 (91.3)	**223 (95.5)**	347 (135.6)	211 (55.9)	224 (80.9)	**261 (96.7)**	354 (164.2)	349 (150.9)	312 (134.8)	**338 (150.6)**

Table 7: Mean latency values with standard deviations for Glenlea.

		Latency and Standard Deviation Values (ms)											
		200 m Transmission				**400 m Transmission**				**600 m Transmission**			
Type	**Resolution**	R1	R2	R3	**Mean**	R1	R2	R3	**Mean**	R1	R2	R3	**Mean**
Cellular	480p20	132 (8.8)	133 (9.6)	132 (37.8)	**132 (23.1)**	133 (8.4)	134 (8.3)	134 (8.4)	**134 (8.4)**	138 (14.2)	132 (8.7)	140 (115.6)	**137 (67)**
	480p25	146 (17.1)	144 (20)	141 (8.5)	**144 (16)**	151 (22.9)	146 (12.4)	153 (12.9)	**150 (16.8)**	147 (21.3)	148 (9.5)	147 (11.6)	**147 (15.1)**
	576p20	193 (73.5)	190 (18.2)	186 (27)	**190 (46.3)**	189 (15.3)	190 (15.1)	192 (16.3)	**191 (15.5)**	188 (15.1)	189 (20.5)	192 (18.8)	**190 (18.3)**
Radio	480p20	297 (256.2)	335 (258.8)	394 (288.8)	**342 (267.6)**	103 (79.4)	133 (9.6)	101 (8.1)	**112 (46.2)**	101 (9.3)	100 (8.8)	106 (114)	**102 (65.9)**
	480p25	498 (360.1)	560 (332.6)	518 (324)	**525 (338.5)**	126 (25.7)	128 (29.5)	128 (26.1)	**127 (27.1)**	134 (44.9)	130 (29.7)	128 (25.3)	**131 (34.4)**
	576p20	548 (259.7)	518 (323.3)	590 (586.4)	**552 (425.8)**	223 (64.1)	222 (68)	211 (58.7)	**219 (63.7)**	213 (65.2)	228 (64)	234 (72.5)	**225 (67.3)**

Table 8: Mean latency values with standard deviations for Elm Creek.

		Latency and Standard Deviation Values (ms)											
		200 m Transmission				**400 m Transmission**				**600 m Transmission**			
Type	**Resolution**	R1	R2	R3	**Mean**	R1	R2	R3	**Mean**	R1	R2	R3	**Mean**
Cellular	480p20	73901 (64010)	69878 (29542)	49155 (34935)	**64311 (42664)**	22625 (17484)	41195 (20828)	79808 (14453)	**47876 (20491)**	53258 (26406)	136760 (35548)	62845 (21257)	**84288 (27138)**
	480p25	122467 (31832)	N/A	N/A	**86120 (31832)**	55600 (17947)	29716 (30811)	62624 (18672)	**49314 (21922)**	58151 (21977)	46888 (27604)	40824 (11005)	**48621 (21622)**
	576p20	71203 (42029)	53899 (47229)	39365 (33999)	**54822 (41645)**	56124 (17221)	59189 (25284)	45793 (32355)	**53702 (24015)**	48615 (13415)	45180 (20260)	52409 (43442)	**48734 (27188)**
Radio	480p20	172 (196.2)	154 (152.1)	196 (243.5)	**174 (193.7)**	101 (24)	103 (15.8)	101 (8.1)	**102 (17.3)**	97 (8.2)	95 (7.8)	95 (7.7)	**96 (7.9)**
	480p25	196 (183.9)	192 (221)	205 (157.7)	**198 (187.4)**	129 (27)	131 (27)	129 (27.2)	**130 (27)**	122 (26.6)	126 (31.4)	127 (29.9)	**125 (29.4)**
	576p20	258 (207.4)	224 (115.7)	227 (153.7)	**237 (157.1)**	205 (50)	221 (73.1)	214 (57.3)	**213 (60.9)**	235 (71.2)	202 (57.6)	220 (64.2)	**219 (64.5)**

Table 9: Mean latency values with standard deviations for Elm Creek.

		Latency and Standard Deviation Values (ms)											
		200 m Transmission				**400 m Transmission**				**600 m Transmission**			
Type	**Resolution**	R1	R2	R3	**Mean**	R1	R2	R3	**Mean**	R1	R2	R3	**Mean**
Cellular	480p20	255 (137.8)	212 (124.1)	233 (238.9)	**233 (172.7)**	30590 (13456.3)	199 (69.7)	234 (272.1)	**10341 (5979.2)**	291 (280)	189 (42.5)	198 (43.4)	**226 (159.5)**
	480p25	257 (166.4)	6706 (12981.6)	40234 (16662.8)	**15732 (9128.2)**	20812 (12822.7)	221 (163.3)	396 (367.2)	**7143 (6305.6)**	213 (48.6)	214 (49.9)	211 (44.6)	**213 (47.6)**
	576p20	283 (942.9)	253 (498.7)	288 (424)	**275 (643.5)**	319 (181)	275 (53.9)	275 (53.9)	**290 (101.1)**	363 (338.9)	288 (122.1)	272 (299.3)	**308 (272.8)**
Radio	480p20	109 (54.3)	110 (46.6)	113 (49.6)	**111 (50.3)**	101 (15.1)	104 (12.3)	104 (12.6)	**103 (13.4)**	97 (8.9)	98 (10.9)	83 (8.6)	**93 (9.5)**
	480p25	193 (146.6)	184 (88.1)	198 (103.1)	**191 (114.2)**	130 (27.4)	129 (31.4)	131 (33.1)	**130 (30.7)**	133 (49.7)	128 (31.1)	126 (27.3)	**129 (37.3)**
	576p20	316 (126.2)	307 (125.8)	326 (128.7)	**316 (127)**	237 (74.7)	224 (150.7)	230 (59.6)	**230 (102.7)**	244 (80.3)	226 (63.6)	228 (63.6)	**233 (69.6)**

5.2 RSSI Receiver Data

RSSI information for the Receiver system was collected and tabulated for both the cellular and radio systems, obtained via UDP connection from the BulletPlusAC router, polled every 5 s by a Python Script for the duration of the trial. Results are as shown below in Table 10 and 11 for cellular and radio transmission respectively, which lists them along with the average value.

Table 10: Measured average cellular receiver RSSI at each location, expressed in dBm.

Location	Resolution	Cellular Receiver RSSI (dBm)											
		200 m Transmission				400 m Transmission				600 m Transmission*			
		R1	R2	R3	**Mean**	R1	R2	R3	**Mean**	R1	R2	R3	**Mean**
UM campus	480p20	-76.55	-69.43	-70.47	**-72.15**	-76.92	-77.55	-74.97	**-76.48**	-56.38	-54.60	-62.94	**-57.97**
	480p25	-73.04	-74.22	-75.49	**-74.25**	-77.13	-78.66	-73.52	**-76.44**	-62.66	-62.58	-60.66	**-61.97**
	576p20	-75.77	-74.61	-74.72	**-75.03**	-77.68	-77.10	-58.86	**-71.21**	-59.46	-52.97	-57.30	**-56.58**
Glenlea	480p20	-76.60	-76.27	-74.52	**-75.80**	-75.52	-73.99	-72.64	**-74.05**	-73.41	-74.47	-75.21	**-74.36**
	480p25	-75.83	-74.72	-71.96	**-74.17**	-74.97	-73.55	-75.80	**-74.77**	-75.60	-76.03	-76.52	**-76.05**
	576p20	-71.94	-72.13	-73.75	**-72.61**	-76.44	-77.18	-75.47	**-76.37**	-74.50	-75.46	-76.46	**-75.47**
Elm Creek	480p20	-74.54	-74.50	-97.55	**-82.20**	-74.91	-94.40	-82.00	**-83.77**	-75.83	-89.60	-92.40	**-85.94**
	480p25	-74.91	-94.19	N/A	**-84.55**	-74.77	-74.47	-74.05	**-74.62**	-94.69	-92.19	-91.47	**-93.44**
	576p20	-73.46	-74.46	-74.46	**-74.13**	-74.61	-77.56	-74.10	**-75.42**	-91.47	-91.92	-89.46	**-90.95**
Sanford	480p20	-113.29	-86.74	-86.75	**-95.59**	-86.67	-76.27	-76.46	**-79.80**	-83.85	-76.49	-83.91	**-81.42**
	480p25	-86.30	-86.29	-79.13	**-83.91**	-84.82	-76.61	-89.03	**-83.48**	-83.13	-86.19	-84.91	**-84.75**
	576p20	-76.45	-76.45	-77.41	**-76.77**	-82.36	-83.30	-83.47	**-83.04**	-84.19	-82.80	-85.57	**-84.19**

*540 m for Sanford only.

Table 11: Radio received signal strength for all trials at each location, expressed in dBm.

Location	Resolution	Radio Receiver RSSI (dBm)											
		200 m Transmission				400 m Transmission				600 m Transmission*			
		R1	R2	R3	**Mean**	R1	R2	R3	**Mean**	R1	R2	R3	**Mean**
UM campus	480p20	-76.55	-69.43	-70.47	**-72.15**	-76.92	-77.55	-74.97	**-76.48**	-56.38	-54.60	-62.94	**-57.97**
	480p25	-73.04	-74.22	-75.49	**-74.25**	-77.13	-78.66	-73.52	**-76.44**	-62.66	-62.58	-60.66	**-61.97**
	576p20	-75.77	-74.61	-74.72	**-75.03**	-77.68	-77.10	-58.86	**-71.21**	-59.46	-52.97	-57.30	**-56.58**
Glenlea	480p20	-76.60	-76.27	-74.52	**-75.80**	-75.52	-73.99	-72.64	**-74.05**	-73.41	-74.47	-75.21	**-74.36**
	480p25	-75.83	-74.72	-71.96	**-74.17**	-74.97	-73.55	-75.80	**-74.77**	-75.60	-76.03	-76.52	**-76.05**
	576p20	-71.94	-72.13	-73.75	**-72.61**	-76.44	-77.18	-75.47	**-76.37**	-74.50	-75.46	-76.46	**-75.47**
Elm Creek	480p20	-74.54	-74.50	-97.55	**-82.20**	-74.91	-94.40	-82.00	**-83.77**	-75.83	-89.60	-92.40	**-85.94**
	480p25	-74.91	-94.19	N/A	**-84.55**	-74.77	-74.47	-74.05	**-74.62**	-94.69	-92.19	-91.47	**-93.44**
	576p20	-73.46	-74.46	-74.46	**-74.13**	-74.61	-77.56	-74.10	**-75.42**	-91.47	-91.92	-89.46	**-90.95**
Sanford	480p20	-113.29	-86.74	-86.75	**-95.59**	-86.67	-76.27	-76.46	**-79.80**	-83.85	-76.49	-83.91	**-81.42**
	480p25	-86.30	-86.29	-79.13	**-83.91**	-84.82	-76.61	-89.03	**-83.48**	-83.13	-86.19	-84.91	**-84.75**
	576p20	-76.45	-76.45	-77.41	**-76.77**	-82.36	-83.30	-83.47	**-83.04**	-84.19	-82.80	-85.57	**-84.19**

*540 m for Sanford only.

5.3 Raspberry Pi Temperature Data

Operating temperature data for the Raspberry Pi were polled similarly to the data for receiver RSSI, instead polled every 5 s from the Raspberry Pi via a Python script incorporated into the transmitter protocols. Table 12 documents the average temperature measured across trials.

Table 12: Measured average temperature for the Raspberry Pi for all trials.

			Temperature (°C)											
			200 m Transmission				**400 m Transmission**				**600 m Transmission***			
Location	**Type**	**Resolution**	R1	R2	R3	**Mean**	R1	R2	R3	**Mean**	R1	R2	R3	**Mean**
UM campus	Cellular	480p20	70.24	68.54	69.68	**69.49**	66.93	66.96	67.5	**67.13**	64.82	65.2	65.89	**65.31**
		480p25	70.62	71.2	71.24	**71.02**	69.21	68.48	69.59	**69.09**	66.88	67.02	67.41	**67.1**
		576p20	72.31	72.73	72.9	**72.64**	71.24	71.52	71.98	**71.58**	67.61	67.98	68.25	**67.95**
	Radio	480p20	67.65	68.18	68.55	**68.13**	64.39	65.28	65.54	**65.07**	58.06	61.24	61.42	**60.24**
		480p25	68.92	69.24	69.55	**69.24**	66.08	66.58	66.72	**66.46**	62.62	62.86	64.07	**63.18**
		576p20	70.37	71.03	71.55	**70.99**	67.42	67.42	67.96	**67.6**	65.62	65.54	66.01	**65.73**
Glenlea	Cellular	480p20	73.97	74.69	73.98	**74.21**	69.65	69.52	69.97	**69.71**	66.88	67.01	67.58	**67.16**
		480p25	75.11	75.85	75.99	**75.65**	70.64	71.74	72.63	**71.67**	67.66	67.5	68.21	**67.79**
		576p20	77.19	77.95	78.41	**77.85**	73.59	74.06	74.43	**74.03**	69.41	69.22	69.22	**69.28**
	Radio	480p20	72.48	73.09	73	**72.86**	65.19	65.96	66.53	**65.89**	60.81	62.73	63.32	**62.29**
		480p25	73.27	73.47	73.1	**73.28**	67.81	68.55	69.11	**68.49**	63.54	63.76	63.97	**63.76**
		576p20	74.73	75.39	75.69	**75.27**	70.75	71.24	71.24	**71.08**	65.7	66.34	67.16	**66.4**
Elm Creek	Cellular	480p20	61.55	61.93	62.2	**61.89**	59.52	59.99	61.23	**60.25**	69.87	71.6	71.79	**71.09**
		480p25	61.47	49.01	N/A	**55.24**	61.38	63.27	63.32	**62.66**	72.54	72.35	73	**72.63**
		576p20	60.78	61.61	61.93	**61.44**	63.68	64.42	64.28	**64.13**	72.94	73.75	62.84	**69.84**
	Radio	480p20	66.99	69.14	48.39	**61.51**	60.02	60.39	60.48	**60.3**	69.17	69.65	70.34	**69.72**
		480p25	70.91	59.67	61	**63.86**	61.44	61.72	60.47	**61.21**	70.78	70.71	70.93	**70.8**
		576p20	62.88	63.51	64.16	**63.52**	61.73	62.45	62.84	**62.34**	72.67	72.7	73	**72.79**
Sanford	Cellular	480p20	67.68	67.95	71.86	**69.16**	72.63	74.08	77.07	**74.59**	68.27	52.22	71.29	**63.93**
		480p25	74.52	75.1	73.51	**74.38**	78.65	79.2	78.98	**78.94**	73.01	73.75	75.19	**73.98**
		576p20	51.69	53.34	79.61	**61.55**	73.46	75.45	N/A	**74.45**	75.58	76.28	77.13	**76.33**
	Radio	480p20	68.99	67	66.15	**67.38**	71.69	71.37	70.61	**71.22**	71.01	68.84	66.46	**68.77**
		480p25	67.72	68.52	68.73	**68.32**	70.83	72.24	73.06	**72.04**	66	66.56	66.96	**66.51**
		576p20	69.56	67.52	68.02	**68.36**	74.01	74.06	74.21	**74.1**	68.17	68.59	68.95	**68.57**

*540 m for Sanford only.

6 Discussion

6.1 Effect of Video Resolution on Latency

Trial results across locations were averaged, and arranged graphically by video quality used and the distances used, across all locations and transmission types. The graphs for these are shown

below in Figures 16-19. A total of 4 values were identified as outliers and omitted from the Sanford results and following analysis, as their magnitude was vastly larger than the remaining values obtained in the area and their inclusion would obscure differences at that location.

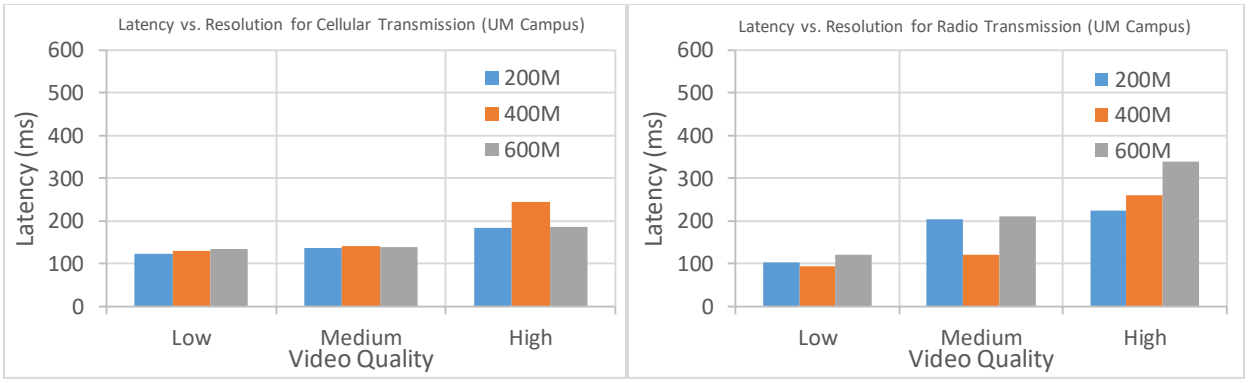

Figure 16: a) Results for mean latency separated by resolution for cellular transmission, with lines differentiated by distance at the University of Manitoba campus. b) Equivalent results for radio transmission.

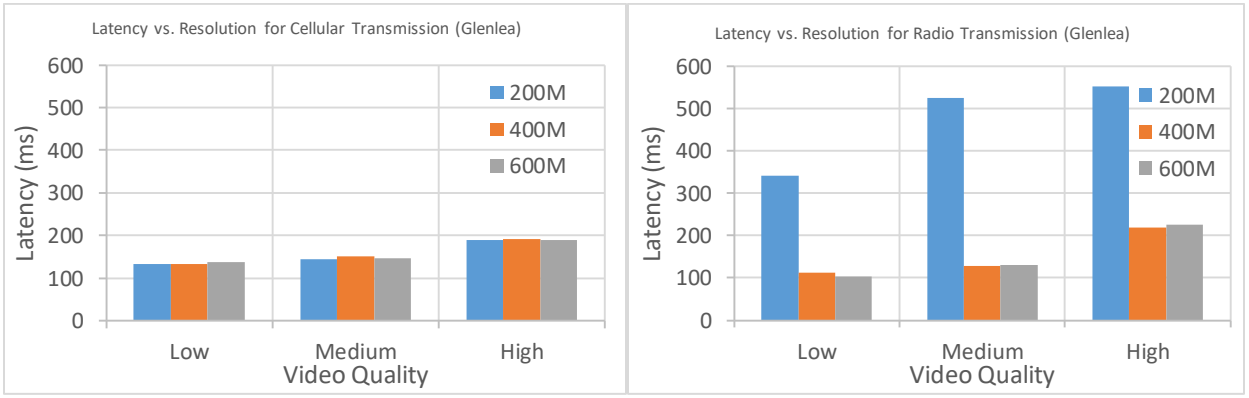

Figure 17: a) Results for mean latency separated by resolution for cellular transmission, with lines differentiated by distance in Glenlea. b) Equivalent results for radio transmission.

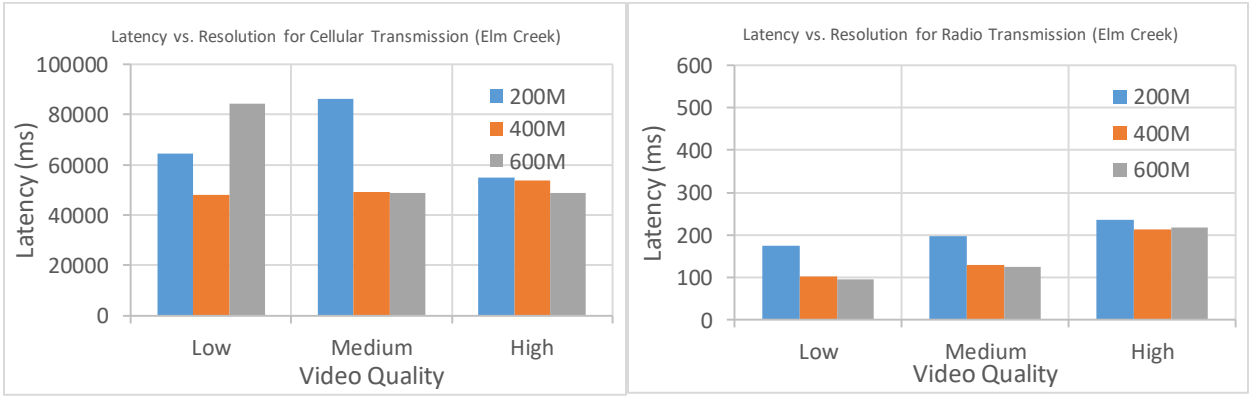

Figure 18: a) Results for mean latency separated by resolution for cellular transmission, with lines differentiated by distance in Elm Creek. b) Equivalent results for radio transmission.

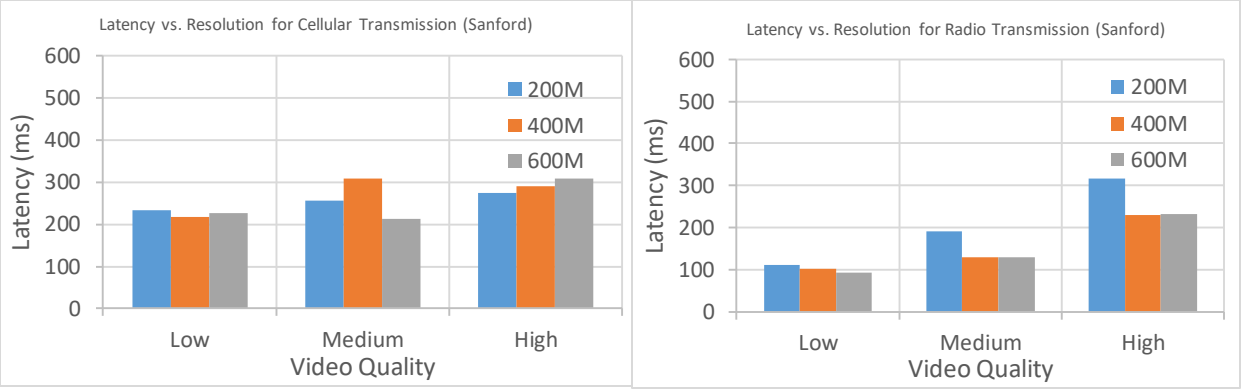

Figure 19: a) Results for mean latency separated by resolution for cellular transmission, with lines differentiated by distance in Sanford. b) Equivalent results for radio transmission.

From examination of the graphs, it becomes apparent that the mean latency for most locations tended to fall under 300 ms, with prominent exceptions for high video quality with radio transmission in Sanford at 200 m, medium and high video quality for radio transmission in Glenlea at 200 m, medium quality video at 400 m in Sanford and cellular transmission of all video qualities for all transmission distances at Elm Creek. While it should be mentioned that the camera and display will each add their own delay to the total latency when looking glass-to-glass latency, the comparative measurements done outside of the experiment demonstrated the additional latency added to be around 100 ms. It can be reasonably stated that the camera system was able to transmit video in real-time in most locations with the exception of Elm Creek.

One-way ANOVA was conducted at each location, examining the significance of resolution as it applied to the mean latency when broken down into the respective transmission types used. At the UM campus, statistically significant effects were found for both cellular ($F_{2, 24} > 16.03$, $p < 3.80e\text{-}5$) and for radio ($F_{2, 24} > 19.49$, $p < 9.37e\text{-}06$) transmission. Tukey's honestly significant difference (HSD) test was then used to characterize the differences between means and their significance. For cellular transmission, a difference of 77.5 ms was found between low and high quality video ($p < 0.0775$), and a statistically significant difference of 67.3 ms between medium and high ($p < 3.96e\text{-}4$). For radio transmission, statistically significant differences were found between means for all resolutions at the UM campus, with a difference of means of 72.8 ms between low and medium ($p < 0.0328$), a difference of 168 ms between low and high ($p < 5.72e\text{-}6$) and a difference of 95.6 ms between medium and high resolution ($p < 0.00467$).

In Glenlea, statistically significant differences between means were observed only for variations in video resolution during cellular transmission ($F_{2, 24} > 889.5$, $p < 2e\text{-}16$). Specifically, a significant difference of 12.9 ms was observed between low and medium resolution ($p < 5.63e\text{-}09$), 56 ms between low and high resolution ($p < 2.09e\text{-}14$), and 43 ms between medium and high resolution ($p < 2.09e\text{-}14$). In Sanford, video resolution was found via ANOVA to have a significant effect for both cellular ($F_{2, 20} > 4.16$, $p < 0.0309$) and radio ($F_{2, 24} > 60.41$, $p < 4.29e\text{-}10$) transmission. For cellular transmission, significant differences were observed only between low and high video resolution ($p < 3.67e\text{-}10$), where a difference of 157 ms was found through Tukey's HSD. Radio transmission comparisons lead to the observation of significant differences for all resolutions, with a difference of 47.9 ms reported between low and medium resolution ($p < 0.00896$), 109.6 ms between medium and high resolution ($p < 3.67e\text{-}10$), and 157.5 ms between low and high resolution ($p < 3.12e\text{-}7$).

In Elm Creek, cellular transmission was not found to have a significant differences between resolutions for the mean latency. Radio transmission ($F_{2, 24} > 23.1$, $p < 2.50e\text{-}06$) analysis with Tukey's HSD indicated a significant difference between means for low and high video resolution of 99 ms ($p < 2.43e\text{-}06$), while the statistically significant difference between medium and high video resolution means was reported to be 72 ms ($p < 0.000206$). Across all graphs, it can be seen that an increase in the video quality corresponds with an increase in latency. At certain distances and locations this trend was not observed, in particular for cellular transmission at Sanford where values do not seem to have an appreciable trend between them. Radio transmission tended to have an average latency under 300 ms, with the exception of tests performed at 200 m distances or under the higher quality video setting, for which average latencies were significantly higher in some cases.

One-way ANOVA examining variations in mean latency as it related to distance indicated statistically significant variation only for radio transmission in the locations of Glenlea ($F_{2, 24} > 55.478$, $p < 1.00e\text{-}9$) and Elm Creek ($F_{2, 24} > 4.13681$, $p < 0.286$). In these locations, there was a noticeably higher latency at 200 m compared to 400 and 600 m, for which Tukey HSD analysis at Glenlea indicated the difference from 200 m to be 320 ms, for both testing at 400 m ($p < 0.00754$) and 600 m ($p < 0.00753$). In Elm Creek, the distance from 400 m to 200 m yielded a

between means increase of 54 ms ($p < 0.0550$), while the distance between 600 m to 200 m yielded an increase of 56 ms ($p < 0.471$).

Overall, distance was not observed to have a strong effect on the latency, as might be expected. A possible explanation might come from the somewhat shorter distances used, when incredibly high propagation speeds of radio and cellular waves ($3.00*10^8$ m/s). Even a jump from 1 to 10 km will result in a theoretical increase in propagation time of just 30 µs, and the method used to synchronize the clocks likely was not adequate to capture a difference on that scale.

Temperature measurements for the Raspberry Pi were taken during each trial, and graphed with the corresponding mean latency in a scatterplot graph, shown below in Figure 20.

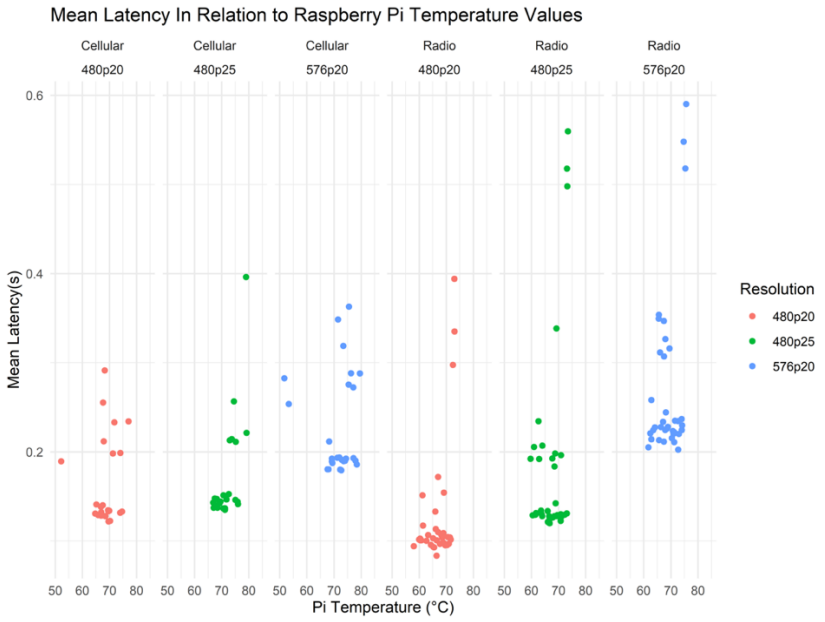

Figure 20: Scatterplot graph of latency with temperature data for Glenlea and the University of Manitoba.

While each measurement varies between 60-75℃ approximately, and increases were observed with higher resolutions, an increase in latency was not generally observed with increases in temperature. In general, significant effects were observed on temperature from both resolution ($p < 0.00702$) and transmission type ($p < 0.00168$) when examined with a two-way ANOVA evaluating both factors. In particular, Tukey's HSD analysis for resolution showed a significant 2.14℃ increase from low to medium resolution ($p < 0.0312$) and an increase of 2.45℃ from low to high resolution. Differences between medium and high resolution were not statistically significant.

Additional Tukey's HSD tests conducted with the samples showed a significant 2.18°C increase between radio and cellular transmission ($p < 0.00168$). This is likely because the burden associated with data transmission was on the Pi itself through the attached cellular header, rather than an external radio pack connected over an Ethernet cable. The Raspberry Pi is known to throttle performance when temperature becomes sufficiently high to prevent damage due to excessive temperatures, with throttling beginning at a critical value of 80°C (Longbottom, 2019). The results observed show some increased latency with mean temperatures nearing the throttling temperature for medium and high quality video, although this was not present for all trials near the temperature threshold. The effect of throttling, if any, appears to have been relatively small or non-existent during testing.

While examination of the overall magnitude of latency is important, it becomes worthwhile to evaluate the variance of the data to gauge how much a potential user could be impacted by the performance. Xu et al. (2014) found that in remote telesurgery, the negative effects posed by latency on study participants were mild in the 200 to 300 ms range of delay, increasing steadily from 300 to 700 ms. Experiments with time lag on individuals participating on a driving simulator by Davis et al. (2010) indicated that variations in latency had a more severe effect than the magnitude of the mean latency after experiments involving the introduction of a variable latency for subjects which was expressed as the sum of multiple sine waves, and that time lags including and exceeding 700 ms could severely impair the ability to operate a vehicle while driving. In the work of Perez et al. (2016) the completion time of required tasks using a robotic simulator for telesurgical implementations began to increase above 300 ms of latency, with the rate of errors increasing after 500 ms. As a method of assessing the variance of the data, the coefficient of variation was measured for each trial set by dividing the standard deviation for latency of each collected value by the mean latency, compiled in Table 13.

Table 13: Average coefficient of variation for each location and transmission type.

Type	UM	Glenlea	Elm Creek	Sanford
Cellular	19.8%	13.7%	52.1%	80.2%
Radio	48.3%	43.5%	44.4%	32.3%

From comparison of the mean values, it can be seen that the coefficient of variation for cellular was lesser than that of radio transmission at the UM and Glenlea, but larger in Elm Creek and Sanford, where the cellular connection was found to be overall worse. Average radio variance tended to remain relatively consistent across locations, with the exception of Sanford where it was noticeably reduced in comparison. Figures 21 and 22 show the respective coefficients of variation, split by transmission type and location for each location.

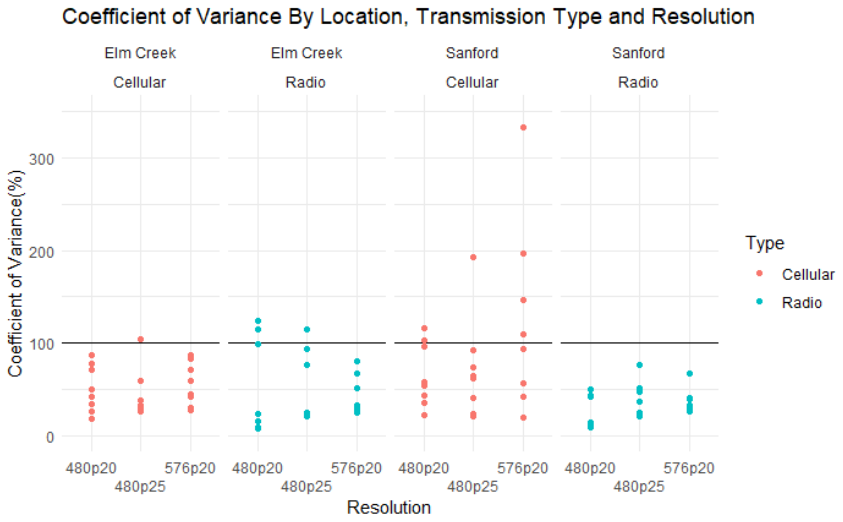

Figure 21: Graphical arrangements of the coefficient of variation for Sanford and Elm Creek, with each dot representing one measurement trial.

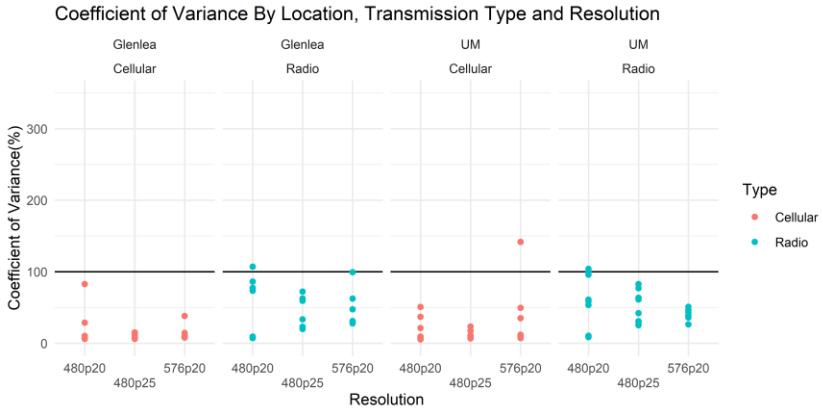

Figure 22: Graphical arrangements of the coefficient of variation for Glenlea and UM campus.

From the above graph it can be observed that most values for the coefficient of variation fall within less than 100%, indicating that the standard deviations for most samples were less than the mean. Exceptions were observed prominently for cellular transmission in the Sanford area, where some mean latency values were measured several hundred times higher than the average

of most other measured values. This increase was not observed within the Elm Creek area, where mean latency values for cellular were universally measured as very large. This suggests that cellular transmission in Sanford was prone to significant spikes in variation outside of the typical range that might be expected. Overall, variance in latency was observed to be unexpectedly large for some samples, particularly when distance of the radio put it close to the receiver or near a fixture such as a wall near the 600 m line at the UM campus. Examination of the impact of such variance on the user experience likely warrants further study.

Better resolutions offer advantages for viewers including clarity of vision and a more distinct picture of what is happening in the field at the cost of a potential increase in latency. To reduce the impact from this and from potential stuttering, one may wish to prioritize the sending of a lower quality, lower framerate video in a bandwidth restricted environment. Adaptive bitrate encoding is also an option to consider that can allow for flexible delivery of video to provide the best quality possible for the user at varying points in time. The inclusion of an adaptive bitrate modification of the streaming pipeline may also have been able to reduce the latency spikes which were observed. Adaptive bitrate technology has been successfully incorporated into GStreamer pipelines for use on drones to minimize latency, modifying the bitrate based on the difference between received and transmitted packets of video (Domi, 2019).

6.2 Effect of Transmission Method on Transmission Latency

Cellular transmission appears overall more consistent and with a lower latency at the UM campus and Glenlea locations, although the data collected at Elm Creek shows there are locations where the technology may encounter difficulty without some remediation. Welch's T-tests were conducted for the gathered values at each location, breaking down the data into the three utilized resolutions to compare whether differences in means existed between video transmission methods. Cohen's d analysis was completed after to analyze the direction and effect size of the difference between mean populations across each location and resolution. On the UM campus, cellular transmission was found to have a statistically significant lesser latency than that of radio transmission for high video quality ($t = -2.4142$, $df = 15.567$, p-value < 0.0142), with a large negative effect size ($d = -1.14$). For low resolution, the reverse was found ($t = 3.5373$, $df = 9.342$, $p < 0.00299$), with a large effect size, $d = 1.67$. A difference between means appears to be

indicated at medium resolution (d = 0.76), but it was not found to be significant (t = -1.6089, df = 8.0324, p < 0.146).

In Glenlea, cellular video was found to universally have a lesser overall mean than for radio transmission at low (t = -1.2803, df = 8.0098, p < 0.118), medium (t = -1.7202, df = 8.0046, p < 0.0618) and high (t = -2.5602, df = 8.0032, p < 0.0168) resolution, with moderate (d = -0.60 for low resolution) to large (d = -0.81 for medium resolution) (d = -1.21 for high resolution) effect sizes indicated through the analysis. In Sanford, the reverse was true, where cellular video universally had higher mean latencies when compared to radio transmission, for low (t = 9.9036, df = 7.8325, p < 5.30e-6), medium (t = 3.2414, df = 6.2475, p < 0.00834), and high (t = 1.7286, df = 14.784, p-value < 0.0524) transmission, with the respective Cohen's d effect of each compared sample set having large effects (d = 4.93, d = 1.82, d = 0.81, for small, medium and large resolution, respectively). Elm Creek's substantially higher latency values were reflected in the analysis, with low resolution (t = 6.12, df = 8, p < 0.00014) having a large effect size (d = 2.88), with medium resolution being similarly significant (t = 5.24, df = 6, p < 0.00097) (d = 3.02). High resolution again proved to be statistically significant, (t = 16.8, df = 8, p-value < 8e-08), with the largest overall measured effect size for the 3 locations (d = 7.92). Table 14 summarizes the measured values.

Table 14: Summary of comparison between transmission types at varying resolutions.

Location	Summary of Latency Comparisons by Type For Video Quality and Location		
	Video Quality		
	Low (480p20fps@400kbit)	**Medium (480p25fps@500kbit)**	**High (576p20fps@600kbit)**
UM	Cellular > Radio (p < 0.00299) 1.67	No difference (p < 0.146) -0.76	Cellular < Radio (p < 0.0142) -1.14
Glenlea	Cellular < Radio (p < 0.118) -0.60	Cellular < Radio (p < 0.0618) -0.81	Cellular < Radio (p < 0.0168) -1.21
Sanford	Cellular > Radio (p < 5.295e-6) 4.93	Cellular > Radio (p < 0.00834) 1.82	Cellular > Radio (p < 0.0523) 0.81
Elm Creek	Cellular > Radio (p < 0.00014) 2.88	Cellular > Radio (p < 0.00097) 3.02	Cellular > Radio (p < 8e-08) 7.92

Significantly lower latencies were observed for cellular compared to radio for high video quality at both Glenlea and the UM campus, with low quality radio transmitted video having a lower latency at the UM. In Sanford, the means for cellular video were found to be universally higher than for those of Ethernet radio transmission. Tukey HSD analysis at the University of Manitoba showed a statistically significant increase of 23 ms (p < 0.00274) from radio to cellular transmission for low quality, while the mean for cellular transmission was measured to be 68 ms less (p < 0.0281) than for that of radio transmission when transmitting high quality video.

In Glenlea, statistically significant differences between means were observed only for high transmission, where the difference between means was measured to be 142 ms ($p < 0.0210$). In Sanford, the mean for radio transmission at low resolution was measured to be 124 ms lower than for cellular transmission ($p < 2.74e-8$) and 102 ms at medium resolution ($p < 0.00234$). Significant differences were not observed for high resolution.

Empirical cumulative distribution functions (CDFs) were generated at each location for the measured mean latency across all trials, showing the total proportion of measurements that fell within a particular range of values. 95% confidence envelopes for each location were generated based on the Dvoretsky-Kiefer-Wolfowitz inequality (1956). Figure 23 shows the constructed CDF with confidence envelopes. It can be seen that radio transmission is initially lower latency, with a significant jump in latency occurring just before 50% of the variation is accounted for. Cellular latency completely falls within 200 ms, but the variation of radio latency is not entirely accounted for until well nearly 600 ms.

Figure 23: CDF by equipment type at Glenlea.

Figure 24 shows the CDF for the UM campus. Examination of the CDF by equipment type for the UM campus also has radio latency lesser than that of cellular until reaching close to 50% of the variation, after which cellular latency tends to be lesser. The effect is not as pronounced as in Glenlea, with maximum radio latency being just over 350 ms and cellular latency reaching a similar point.

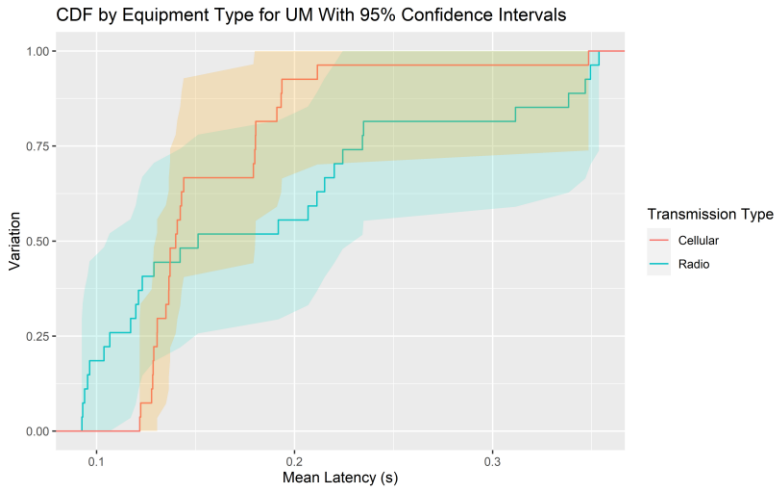

Figure 24: CDF by distance for the UM campus, with 95 percentile markers.

In both the above graphs radio transmission initially gives a lesser latency than cellular transmission, with distinct points of intersection at which cellular transmission begins to outpace radio transmission. The results in Table 14 suggest that at the UM campus, resolution seemed strongly associated with which type had lower latency. Elm Creek and Sanford were both areas for which cellular quality was expected to be mixed, and visual analysis of the CDF for both locations seems to confirm this, with Figure 25 illustrating the difference in Elm Creek.

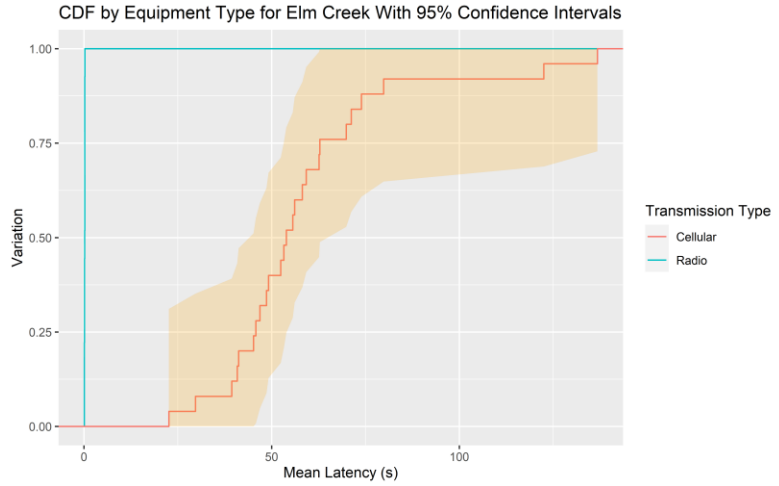

Figure 25: CDF by equipment type for Elm Creek.

The difference between cellular and radio transmission in Elm Creek is so considerable that it makes it very difficult to evaluate certain characteristics, with the cellular latency values measured being so staggeringly high that the CDF for radio and the accompanying 95% confidence envelope are rendered as a straight vertical line. Average latency values for cellular

transmission begin at close to 25 s and run well into the 150 s range. It is clear that cellular transmission in this location ran into severe connection difficulties. In Sanford, it can be seen that the empirical CDF consistently shows a lower latency for radio transmission than for that of cellular transmission (Figure 26).

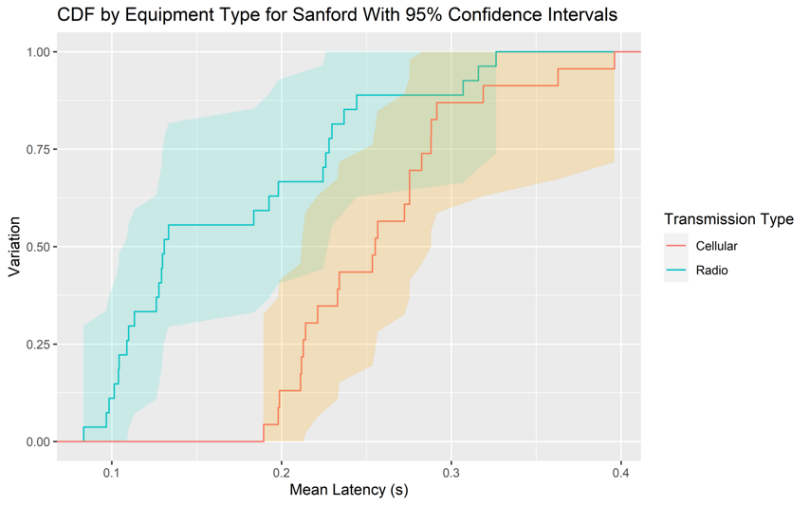

Figure 26: Sanford CDF by transmission equipment type.

Comparison with the Ethernet radio exposed shortcomings with cellular transmission and significant issues that will need to be addressed to allow for observation in areas with poor network quality. Ethernet radio with omnidirectional transmission at even moderate ranges required a considerable drop in transmission bandwidth to accommodate, which demanded concessions in the quality of video to be able to transmit without exceeding bandwidth limitations. For transmission over the shorter field-adjacent ranges used in the experiment and for lower resolutions of video, Ethernet radio has been shown to offer reduced latencies while functioning in areas where cellular transmission encounters severe difficulties. The data at Elm Creek coupled with some of the very large measured values in Sanford suggest that there are areas where the transmission capabilities for cellular data are limited, where it may be worth examining alternate transmission methods. In particular, satellite internet such as SpaceX's StarLink represents a potential alternate method of transmission to circumvent issues with ground-based obstacles which could pose problems during transmission. In consideration of the obtained results, it seems reasonable to suggest that a prospective owner consider their surveillance needs, and the cellular network availability in areas of operation prior to device selection.

6.3 Relation Between Signal Strength and Transmission Latency

A potential relationship of interest that was desirable to examine was whether any relationship could be found between the average signal strength for a location and the mean latency observed. In addition to the mean latency values broken down by location, other parameters were measured including RSSI for both radio and cellular, utilizing on-board device mechanisms to gather the output of each as a .csv document. While radio RSSI values would remain relatively constant from location to location due to the control over their relative locations, cellular towers are in a fixed position over which we have no control and our distance – and relative signal strength – from them is somewhat fixed by the testing location. Table 15 shows the mean RSSI at each location.

Table 15: Mean cellular RSSI values for the UM campus, Glenlea, Sanford and Elm Creek.

Location	Mean RSSI (dBm)
Elm Creek	-82.18
Glenlea	-74.77
Sanford	-83.78
UM	-68.84

Latency was graphed against RSSI for all locations and is as shown below in Figure 27. It should be mentioned that the substantial size of the Elm Creek latency measurements eliminates much of the granularity found in the other locations and it is necessary to examine them further.

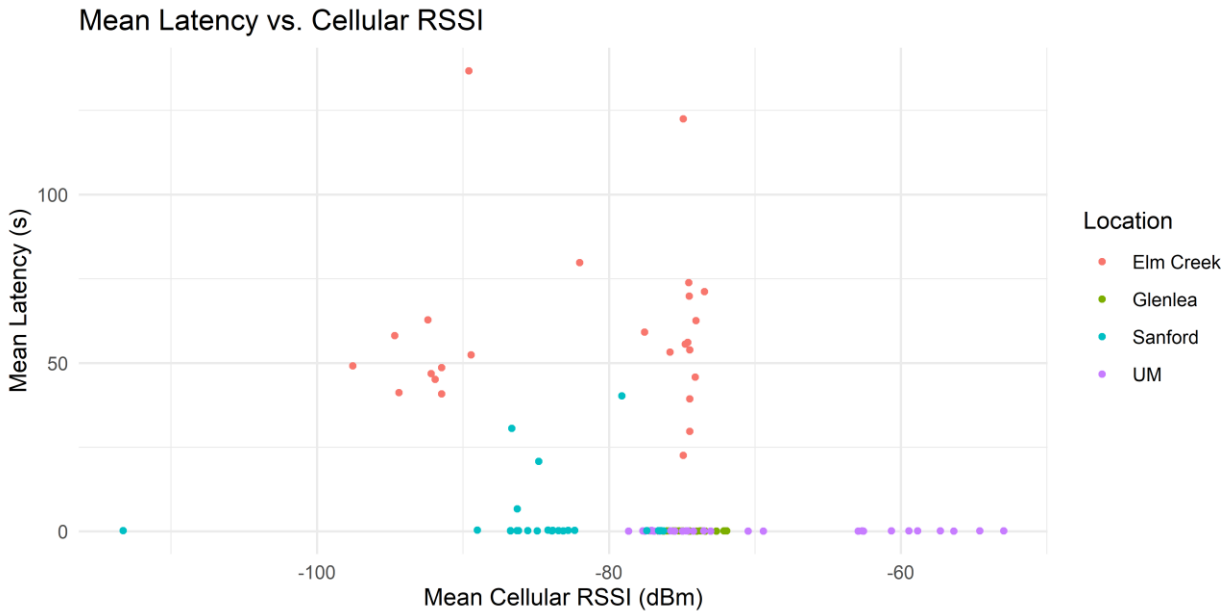

Figure 27: Scatterplot of mean latency values for trials vs. cellular RSSI, for all locations.

While the observed RSSI at that location certainly appeared to be higher in Elm Creek, there are a considerable number of values for which RSSI was lesser where the dramatic difference in latency was still observed. A second plot of cellular latency vs. RSSI, shown below in Figure 28, excludes the values from Elm Creek to highlight the differences between the remaining values.

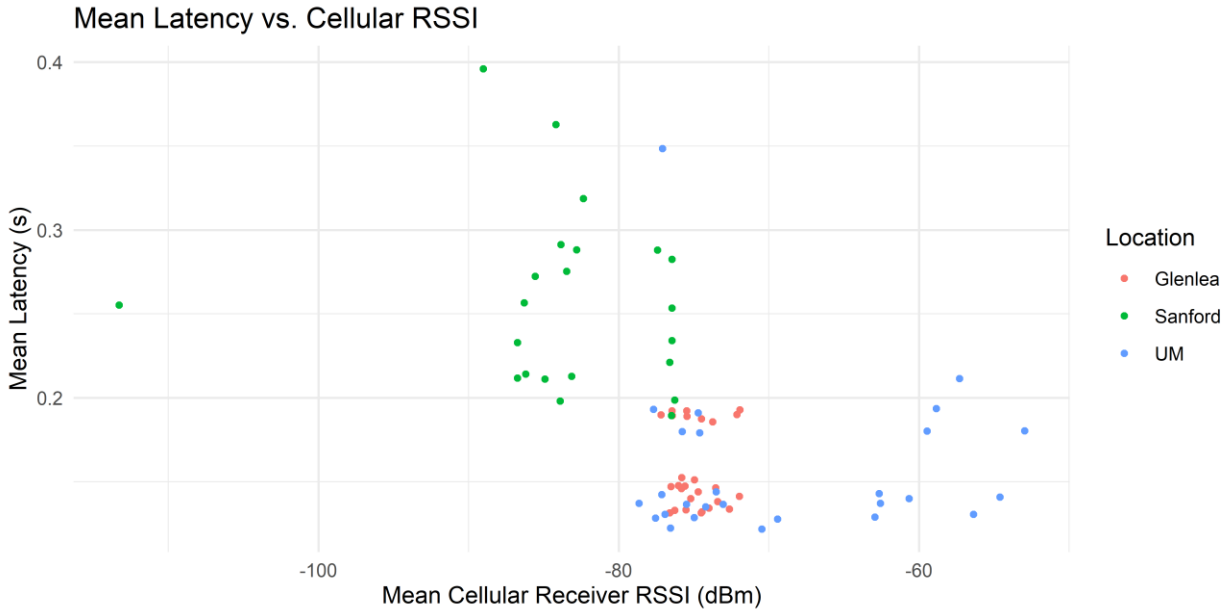

Figure 28: Scatterplot showing latency values, with Elm Creek data excluded.

The arranged data initially would suggest a correlation between cellular receiver RSSI and the observed mean latency. It should be mentioned that the data seems more clearly stratified by location, rather than the magnitude of the RSSI values, as measurements from UM campus do not appear to exhibit much of an increase with RSSI, and in Glenlea similar latency values were obtained to those of the UM campus despite larger exhibited receiver RSSI values. A different factor tied to the location of test sites may have been a greater source of influence on these values than the cellular RSSI.

Figure 29 contains the radio RSSI values for all locations. Visually, there does not appear to be relationship between latency and the RSSI at the receiver for the radio. The measured RSSI values do appear to be stratified into distinct groups, likely corresponding to the different distances at which measurements were taken, with the larger RSSI measurements which would correspond with 200 m distance having some noticeably higher values in Glenlea. It is interesting to note that despite being recorded at the same distance, RSSI for the UM campus at 200 m appear noticeably lower than for those of the other locations.

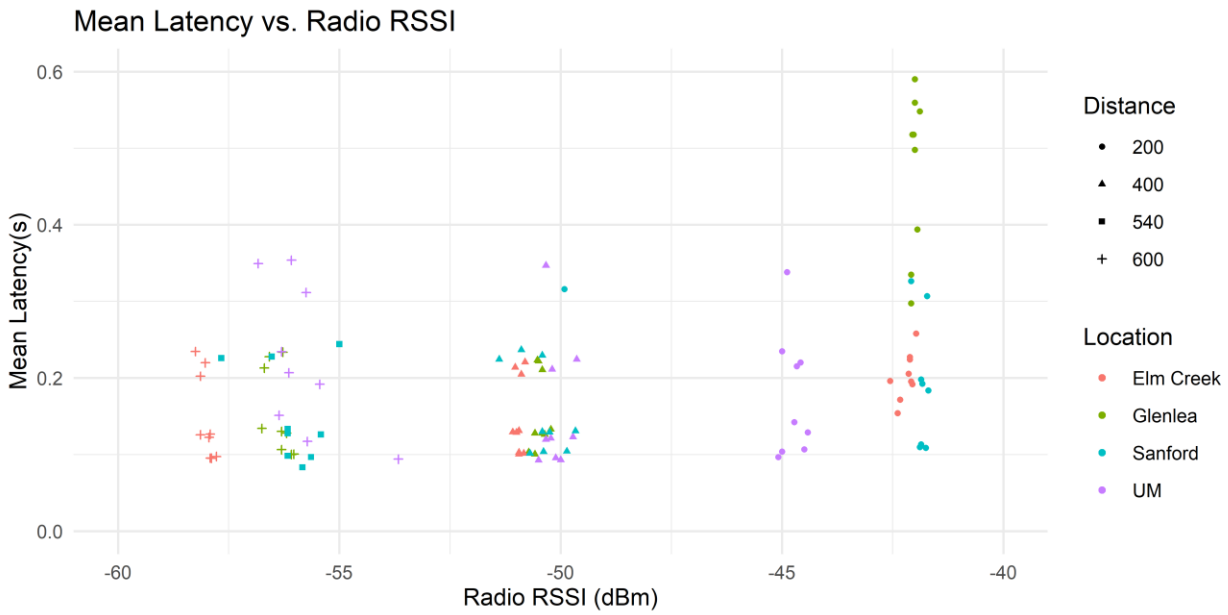

Figure 29: Scatterplot showing latency values in relation to radio RSSI.

Spearman Rank Correlation tests were used to evaluate the relation between mean latency with both cellular and radio transmission. For cellular transmission a moderate negative trend emerged between RSSI and latency which was statistically significant (r_s = -0.463, p < 7.96e-07), while radio transmission showed a smaller positive trend which was found to be significant. (r_s = 0.2633, p < 0.00589). The results seem to line up visually with the graphs of RSSI with mean latency. Cellular transmission in areas where RSSI was lower did appear noticeably higher in latency, while radio transmission does have higher latency values as the strength of the signal increases, which indicates that testing at 200 meters may have been too close to the receiver system.

The cause of the severe latency spikes observed in both Sanford and Elm Creek for cellular transmission is currently unknown. The relative lack of proximal cell towers in those areas may have contributed, based on an examination of the locations. The Elm Creek testing location, in particular, was placed at an almost equidistant point between two towers on the same network used to stream the video connection. Network congestion over the relatively limited connection may have been a potential factor. System testing tended to encompass an entire day so it would be probable that a significant amount of variation in the amount of network traffic occurring in the local area over that span of time. Testing at consistent times may have helped mitigate any

variation in conditions, but would have been very difficult to schedule due to the considerable travel distances required for certain locations.

6.4 Frame Corruption and Error Rate

Analysis of the received video frames, specifically how many are received and how many are considered discarded due to corruption, will provide insight into potential issues that were encountered. Frames might be discarded due to arrival out of order, or distortions caused during transmission, or simple timeouts. Figure 30 measures the number of read video frames by location and transmission type at each location for 25 fps video.

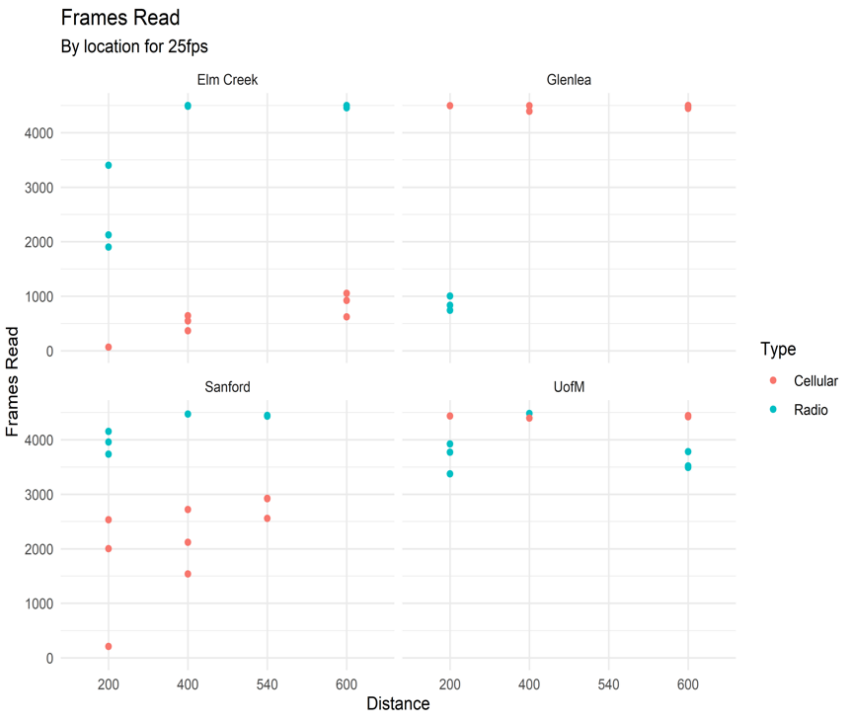

Figure 30: Scatterplot graph of number of successfully read video frames at 25 fps.

For 25 fps, significant frame drops were observed at both Elm Creek and Sanford universally for cellular, with Elm Creek in particular frequently falling below 1000 read frames. Radio transmission at 200 m can be seen to have a considerable drop in frames, with the effect particularly pronounced in Glenlea. At 600 m, a significant drop in frames was observed for the UM campus relative to all other locations. Figures 31 and 32 show the number of read video frames for both low and high resolution at each location.

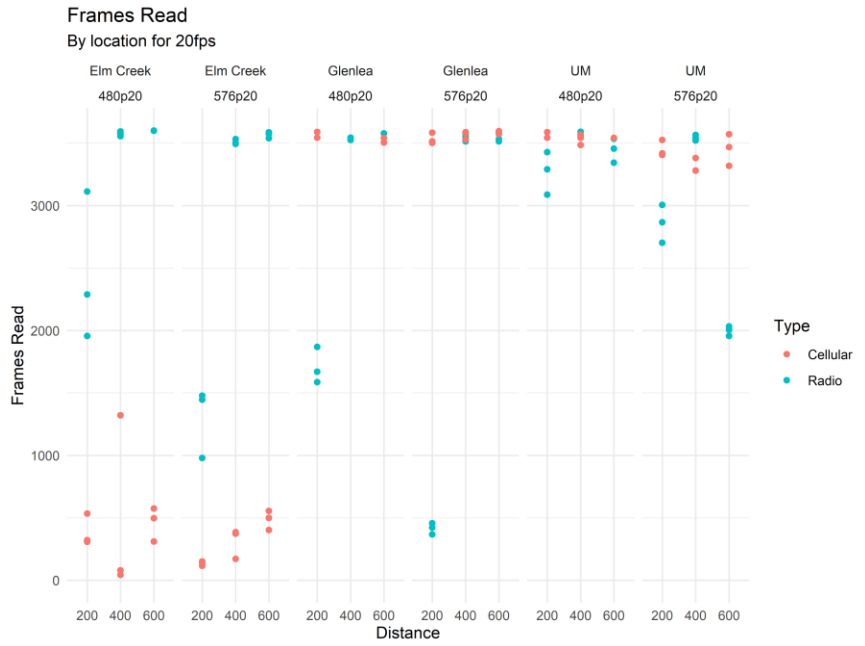

Figure 31: Scatterplot graphs of received video frames for 480p and 20 fps for Elm Creek, Glenlea and the UM campus.

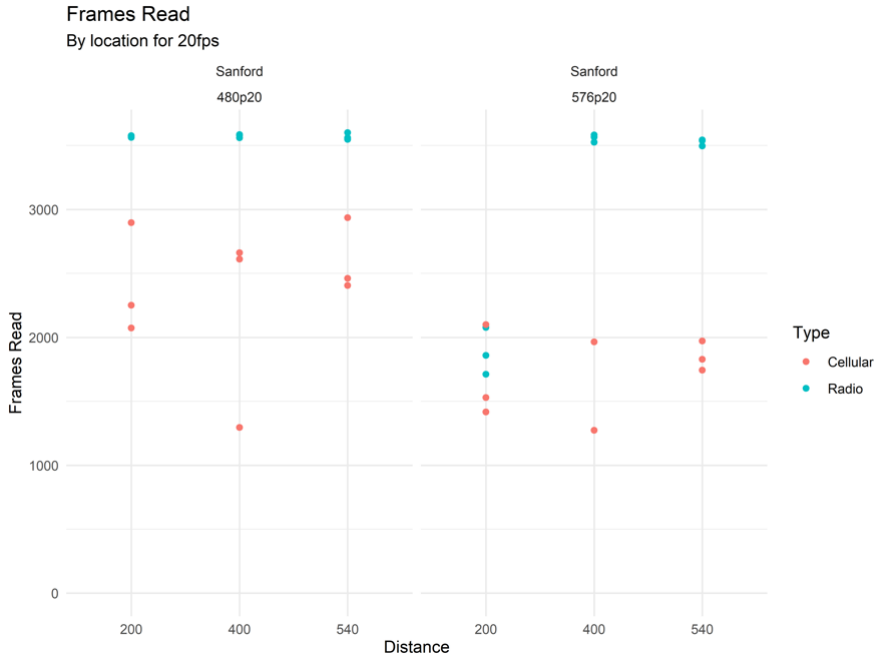

Figure 32: Scatterplot graphs of received video frames for 480p20fps and 576p20fps in Sanford.

For both 480p and 576p at 20 fps, the differences between distances are again observed similarly to 480p at 25 fps, with a noticeable decrease in frames at 200 m for radio transmission and severe drops in received frames at both Elm Creek and Sanford for cellular transmission. For Sanford's cellular connection and at 200 m radio for Elm Creek, Glenlea and the UM campus, a

pronounced decrease in received frames can be observed when moving to a higher resolution video. While radio transmission had more unreadable frames of video at the UM campus and Glenlea, for Elm Creek and Sanford cellular video transmission led to a considerably high number of unreadable video frames.

A mixed model was initially constructed using the factors varied in the experimental process, with type and resolution set as fixed effects and location and distance set as random effects. The initial model produced a conditional coefficient of determination which did not appear to match well with the data ($R^2 = 0.375$) with only resolution being statistically significant within the model ($p < 2e\text{-}16$). A subsequent version incorporated the percentage of unreadable video frames as a fixed effect, at which point the conditional coefficient of determination jumped considerably ($R^2 = 0.845$), with resolution again being significant ($p < 2e\text{-}16$) and the unreadable frames showing a similarly high level of significance ($p < 2e\text{-}16$). A similar model incorporating the total number of received frames with the original factors again showed high significance for resolution ($p < 2e\text{-}16$) and the incorporated variable ($p < 2e\text{-}16$), with transmission type also showing significance ($p < 0.004$), but a decrease in the conditional coefficient of determination when compared to the unreadable video frame percentage ($R^2 = 0.638$). A final model incorporating both unreadable video frame percentage and the total received video frames again showed high significance for both resolution ($p < 2e\text{-}16$) and frame readability ($p < 2e\text{-}16$), along with the received video frames ($p < 0.025$), with only a very slight increase ($R^2 = 0.847$) over just the unreadable video frame percentage. These results carry a strong implication that the percentage of unreadable video frames was a strong predictor for the overall latency.

Table 16: Summary table for mixed model marginal and conditional fit values.

Mixed Model	**Model R^2 (Marginal)**	**Model R^2 (Conditional)**
Initial Model	0.245	0.375
% Readable Frames	0.817	0.845
Total Video Frames	0.522	0.638
% Readable Frames, Total Video Frames	0.824	0.847

6.5 Summary

Statistically significant differences were observed in relation to resolution for the mean latency across all analyzed locations, with effect sizes ranging from moderate to large. Cellular latencies were generally seen to be less than radio latencies for high resolution video and larger for low resolution video for both the UM campus and Glenlea, while in Sanford, radio latencies were

observed to be significantly smaller than cellular latencies, and radio latencies were vastly under those of cellular transmission in Elm Creek. From these results it would appear that for the purposes of edge of field surveillance of a vehicle, Ethernet radio or a similar system capable of direct video transmission is worth consideration, particularly in areas where cellular transmission may be unavailable.

The techniques and equipment used for gathering video data were relatively unoptimized. With the implementation of elements such as dedicated specific hardware, adaptive bitrate encoding and H.265 compression it would likely be possible to reduce latency further by reducing the time required for encoding and decoding. Similar studies have been able to obtain latencies under 200 ms with Raspberry Pis (Jennehag et al., 2016) in different environments.

7 Conclusion

A testing apparatus was successfully created capable of measuring latency while remotely streaming real-time video data, assembling two different remote transmission systems, one capable of transmitting using cellular video and another transmitting video data over an Ethernet radio connection. Using common industry standards of H.264 video compression and the RTP network protocol to enable video streaming, we were able to configure a test pipeline to evaluate latencies in a practical field environment for both cellular transmission and Ethernet radio as a method of video delivery. By measuring latency in a field environment, we were better able to develop an understanding of latencies one could expect for streaming video over a cellular network or over direct radio transmission.

Statistically significant differences were observed between video resolutions at all locations for transmission latency. Increasing video quality tended to result in an increase in mean latency. Based on the resolution measurements obtained and the analysis of the differences between them, lower latencies can be obtained by reducing the corresponding video quality during transmission. It would be beneficial to be able to set the resolution such that a reasonable amount of delay can be achieved while minimizing compromises in the relative video quality for the remote observer.

Cellular transmission latencies were overall found to be lower than those of radio transmission in areas which had better overall cellular network quality such as Glenlea and the UM campus,

when transmitting high resolution video, while radio transmission had a reduced latency when transmitting lower quality. In areas where signal strength was lower and connection quality was worse, the mean latency of radio transmission was observed to be significantly lower than that of cellular transmission across all resolutions.

We have been able to identify that in areas of poor cellular quality the transmission of video with reasonable latency can be difficult and inconsistent. In order to provide services relating to observation of remote vehicles it is recommended that the quality of service in those areas be strengthened, or in areas where this is not possible, for alternative methods of transmission such as Ethernet radio to be considered and tested. While Ethernet radio may not necessarily be the optimal delivery method of choice given the concessions in bandwidth which were required to supplement the range, it seems clear that it would be viable for use in edge-of-field observation, particularly in areas where cellular network quality is poor and lower throughput is acceptable for video. It would be beneficial to assess the quality of networks in the area before selection of any system to ensure that it can operate with a reasonable latency under the local conditions.

Statistically significant different observations between distances were not generally observed as might be expected, only apparently for radio transmission in two locations, where latency had been observed to increase as the distance from the transmitter was reduced to 200 m. The distance used may have been too small to observe meaningful differences in link speed; distance tests conducted using a 802.11g protocol link (Chiew et al., 2005) showed a decrease in link speed and an increase in the rate of packet errors as a mobile client moved further from the server, with a corresponding RSSI decrease from -50 dBm to -90 dBm.

Some possible outlier values were identified in the data, first by searching results for mean latencies less than 500 ms but with maximum values larger than 50 s, and by further searching for values where the standard deviation was 25% higher than the mean to filter values that were unreasonably large relative to the rest of the values in the sample. Graphical examination of the plots showed that, in some cases there were singular observed points with latencies within the range of 40-160 s. These points were likely the result of a reading error with the timestamper, as the values found were generally hundreds to thousands of times larger than the next highest values, and removal of these single or small number of points would drop the measured maximum values to more reasonably expected levels (within 1 or 2 s). While these outliers were

few in number, it does suggest filtering out values with a latency greater than 3 min was not sufficient by itself, particularly when most of the suspected outliers were observed when the frequency of unreadable video frames was higher.

An increase in ground reflections from the antenna at 200 m may have impacted the results for radio transmission and yielded the observed increases in latency at a close distance. These differences were not consistently observed, but appeared most prominently in the Glenlea location, which was a relatively typical plot of flat agricultural land. Testing at a further away distance of 800 m may have been able to eliminate or reduce these reflections, but we were unable to consistently secure access to locations at that distance with a clear line of sight. A very limited test at 800 m in Elm Creek demonstrated transmission was possible from the full length of a half-mile field. While this could not be done due to access and time constraints, the next logical test would be to see if video transmission was possible from approximately 1100-1200 m, as that would encompass the full corner-to-corner distance of a standard half-mile plot of land.

7.1 Future Work

Satellite technology has continuously improved and allows for transmission of video in environments where clear line of sight is not attainable, in exchange for higher overall latency during transmission. In the future it would be worth repeating similar experiments via the use of satellite technology to determine if the latency incurred would be acceptable for use as a method of transmission for video data. The relatively constant distance between ground and satellite could keep latency somewhat consistent over trials.

While the original project plan involved testing over distances up to 100 km, this was not practical for radio transmission, given the probable requirement for either retransmission or directional antennas to overcome signal attenuation limitations. Adding an additional, directionally transmitting radio pair to the configuration which would retransmit the signal from the omnidirectional connection could greatly boost the overall distance, on the condition that the receiver be located somewhere at a fixed point. The use of an intermediary bay station using some manner of directional antenna may allow for much larger transmission distances than what was tested.

It would be interesting to test the design with multiple cameras. The assembly with the cellular header, timestampoverlay and Raspberry Pi is theoretically scalable for a low cost, and would be

able to work for the connection of multiple devices. Using real-time control protocol would allow for additional cameras to be used while synchronizing video across multiple streams. The low cost of the Raspberry Pi and cellular HAT would allow for easy scaling of the system over a cellular network and streams can be synchronized using RTCP. While variable bitrate encoding was tested, constant bitrate encoding was selected and implemented in the final experimental design to account for potential bandwidth constraints in each location. Testing with an adaptive bitrate encoding scheme would be worthwhile, as a more intelligent encoding mechanism adjusting to the bandwidth and other streaming conditions may allow for a more optimized stream. Implementation of methods of error concealment may also improve the quality and latency of delivered video.

Network conditions in particular locations may vary considerably throughout the course of a day. While the time of day for testing was recorded during each trial, it was not utilized to draw any observations from how the mean latency or any other factors varied with time. It would be worthwhile to examine changes in conditions in a single location as network usage fluctuated, and to be able to observe whether these changes in conditions had any noticeable effects on the latencies which would be measured as a result.

While the timestamping method used was likely adequate for the nature of the project, in environments where the quality of received video frames was reduced (either due to a reduction in bandwidth, dropped frames or interference caused by ground reflection) the latency observed may have been higher than the true value, particularly when comparing the median values to the received mean values. The incorporation of a checksum mechanism or similar method to ensure frames arrived without corruption would have been beneficial. This could likely be accomplished with a minor modification to the timestamping mechanism to repeat the received time on two separate lines of the timestamp, and comparing the two received values to see if they are equal, discarding the frame in the event that the two values don’t match.

References

Bachhuber, C., & Steinbach, E. (2016). A system for high precision glass-to-glass delay measurements in video communication. *Proceedings - International Conference on Image Processing, ICIP*, *2016-Augus*, 2132–2136. https://doi.org/10.1109/ICIP.2016.7532735

Bachhuber, C., Steinbach, E., Freundl, M., & Reisslein, M. (2018). On the minimization of glass-to-glass and glass-to-algorithm delay in video communication. *IEEE Transactions on Multimedia*, *20*(1), 238–252. https://doi.org/10.1109/TMM.2017.2726189

Bahaweres, R. B., & Beni Santoso, N. (2016). A preliminary study: Implementation and QoS analysis of dynamic telecytology system on broadband and 3G/4G network in Indonesia. *Proceedings of 2016 4th International Conference on Cyber and IT Service Management, CITSM 2016*, 1–6. https://doi.org/10.1109/CITSM.2016.7577469

Bitmovin. (2018). *Welcome to the 2018 Video Developer Report!*

Bouzakaria, N., Concolato, C., & Le Feuvre, J. (2014). Overhead and performance of low latency live streaming using MPEG-DASH. *IISA 2014 - 5th International Conference on Information, Intelligence, Systems and Applications*, 92–97. https://doi.org/10.1109/IISA.2014.6878732

Case IH. (2017). Autonomous tractor technology shows way forward for farming: enhancing efficiency and working conditions in agriculture. Retrieved November 27, 2020, from https://www.caseih.com/emea/en-za/News/Pages/2017-02-26-Autonomous-tractor-technology-shows-way-forward-for-farming-enhancing-efficiency-and-working-conditions-in-agric.aspx

Chiew, T., Ferre, P., Agrafiotis, D., Molina, A., Nix, A., & Bull., D. (2005). Cross-Layer WLAN Measurement and Link Analysis for Low Latency Error Resilient Wireless Video Transmission. *Analysis*, 177–178.

Cisco,(2019). Cisco visual networking index (VNI) global mobile data traffic forecast update, 2017-2022 white paper. *Ca, Usa*, 3–5. Retrieved from http://www.gsma.com/spectrum/wp-content/uploads/2013/03/Cisco_VNI-global-mobile-data-traffic-forecast-update.pdf

Claypool, M., & Claypool, K. (2006). Latency and player actions in online games. *Communications of the ACM*, *49*(11), 40–45. https://doi.org/10.1145/1167838.1167860

Craig, A. (2014). Regulatory status for using RFID in the EPC Gen 2 band (860 to 960 MHz) of the

UHF spectrum. *Gs1*, *928*(May), 1–22. Retrieved from http://www.gs1.org/docs/epcglobal/UHF_Regulations.pdf

Davis, J., Smyth, C., & McDowell, K. (2010). The effects of time lag on driving performance and a possible mitigation. *IEEE Transactions on Robotics*, *26*(3), 590–593. https://doi.org/10.1109/TRO.2010.2046695

Domi, A. (2019). *L OW L ATENCY A DAPTIVE VIDEO ENCODING*. (May).

Driverless Goes Big in Farming 2019 | Ag Professional. (2019). Retrieved October 10, 2020, from https://www.agprofessional.com/article/driverless-goes-big-farming-2019

Dvoretzky, A., Kiefer, J., & Wolfowitz, J. (1956). Asymptotic Minimax Character of the Sample Distribution Function and of the Classical Multinomial Estimator. *The Annals of Mathematical Statistics*, *27*(3), 642–669. https://doi.org/10.1214/aoms/1177728174

Edet, U., Hawley, E., & Mann, D. D. (2019). Remote supervision of autonomous agricultural sprayers: The farmer's perspective. *Canadian Biosystems Engineering / Le Genie Des Biosystems Au Canada*, *60*(1), 2.19-2.31. https://doi.org/10.7451/CBE.2018.60.2.19

Firat, O. (2019). *School of Information Technologies TELEMETRY ON ROBOT OPERATING SYSTEM BASED SELF-DRIVING Telemeetria operatsioonisüsteemil ROS*. TALLINN UNIVERSITY OF TECHNOLOGY.

Friston, S., & Steed, A. (2014). Measuring latency in virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, *20*(4), 616–625. https://doi.org/10.1109/TVCG.2014.30

García, B., Gortázar, F., López-Fernández, L., & Gallego, M. (2016). Analysis of video quality and end-to-end latency in WebRTC. *2016 IEEE Globecom Workshops, GC Wkshps 2016 - Proceedings*, 2–7. https://doi.org/10.1109/GLOCOMW.2016.7848838

Grüner, M., & Soto, D. (2017). GstShark profiling : a real-life example. *GstShark Profiling : A Real-Life Example*, 68. Prague.

Hegde, A. (Global I. (2018). Autonomous Farm Equipment Market worth $180bn by 2024. Retrieved October 7, 2020, from https://www.globenewswire.com/news-release/2018/02/19/1361230/0/en/Autonomous-Farm-Equipment-Market-worth-180bn-by-2024.html

International Telecommunication Union. (2003). *H.264 (05/2003)*.

International Telecommunications Union. (2003). *International telephone connections and circuits – General Recommendations on the transmission quality for an entire international telephone connection*. https://doi.org/10.1201/b16540-14

International Telecommunications Union. (2013). *ITU-T Recommendations : ITU-T H.265 (04/2013)*.

Jansen, J., & Bulterman, D. C. A. (2013). User-centric video delay measurements. *Proceedings of the International Workshop on Network and Operating System Support for Digital Audio and Video*, 37–42. https://doi.org/10.1145/2460782.2460789

Jennehag, U., Forsstrom, S., & Fiordigigli, F. V. (2016). Low delay video streaming on the internet of things using raspberry Pi. *Electronics (Switzerland)*, *5*(3). https://doi.org/10.3390/electronics5030060

Johnston, S. J., & Cox, S. J. (2017). The raspberry Pi: A technology disrupter, and the enabler of dreams. *Electronics (Switzerland)*, *6*(3). https://doi.org/10.3390/electronics6030051

Kaknjo, A., Rao, M., Omerdic, E., Robinson, L., Toal, D., & Newe, T. (2018). Real-Time Video Latency Measurement between a Robot and Its Remote Control Station: Causes and Mitigation. *Wireless Communications and Mobile Computing*, *2018*. https://doi.org/10.1155/2018/8638019

Kang, L., Zhao, W., Qi, B., & Banerjee, S. (2018). Augmenting self-driving with remote control: Challenges and directions. *HotMobile 2018 - Proceedings of the 19th International Workshop on Mobile Computing Systems and Applications*, *2018-Febru*, 19–24. https://doi.org/10.1145/3177102.3177104

Kryczka, A., Arefin, A., & Nahrstedt, K. (2013). AvCloak: A tool for black box latency measurements in video conferencing applications. *Proceedings - 2013 IEEE International Symposium on Multimedia, ISM 2013*, 271–278. https://doi.org/10.1109/ISM.2013.52

Kumar, S., & Rai, S. (2012). Survey on Transport Layer Protocols: TCP & UDP. *International Journal of Computer Applications*, *46*(7), 975–8887.

Library of Congress. (2017). MJPEG (Motion JPEG) Video Codec. Retrieved October 28, 2020, from https://www.loc.gov/preservation/digital/formats/fdd/fdd000063.shtml

Liu, R., Kwak, D., Devarakonda, S., Bekris, K., & Iftode, L. (2017). Investigating remote driving

over the LTE network. *AutomotiveUI 2017 - 9th International ACM Conference on Automotive User Interfaces and Interactive Vehicular Applications, Proceedings*, 264–269. https://doi.org/10.1145/3122986.3123008

Longbottom, R. (2019). *Raspberry Pi 4 CPU MHz Throttling Performance Effects Raspberry Pi 4 CPU MHz Throttling Performance Effects*. (December). https://doi.org/10.13140/RG.2.2.25612.67205

MacKenzie, I. S., & Ware, C. (1993). Lag as a determinant of human performance in interactive systems. *Conference on Human Factors in Computing Systems - Proceedings*, 488–493. https://doi.org/10.1145/169059.169431

Manley, W. (2016). *Measuring Video Capture Latency*. Berlin.

Panfilov, I., & Mann, D. D. (2018). The importance of real-Time visual information for the remote supervision of an autonomous agricultural machine. *Canadian Biosystems Engineering / Le Genie Des Biosystems Au Canada*, *60*, 211–218. https://doi.org/10.7451/CBE.2018.60.2.11

Peltotalo, J., Harju, J., Vtminen, L., Bouazizi, I., & Curcio, I. D. D. (2010). RTSP-based mobile peer-to-peer streaming system. *International Journal of Digital Multimedia Broadcasting*, *2010*. https://doi.org/10.1155/2010/470813

Perez, M., Xu, S., Chauhan, S., Tanaka, A., Simpson, K., Abdul-Muhsin, H., & Smith, R. (2016). Impact of delay on telesurgical performance: study on the robotic simulator dV-Trainer. *International Journal of Computer Assisted Radiology and Surgery*, *11*(4), 581–587. https://doi.org/10.1007/s11548-015-1306-y

Pi Foundation. (n.d.). Camera Module - Raspberry Pi Documentation. Retrieved November 23, 2020, from https://www.raspberrypi.org/documentation/hardware/camera/

Reid, J. F., Zhang, Q., Noguchi, N., & Dickson, M. (2000). Agricultural automatic guidance research in North America. *Computers and Electronics in Agriculture*, *25*(1–2), 155–167. https://doi.org/10.1016/S0168-1699(99)00061-7

Rodr, E., Kypson, A. P., Moten, S. C., Nifong, L. W., & Jr, W. R. C. (2006). Robotic mitral surgery at East Carolina University : *International Journal*, (April), 211–215. https://doi.org/10.1002/rcs

Seshadrinathan, K., Soundararajan, R., Bovik, A. C., & Cormack, L. K. (2010). A subjective study to evaluate video quality assessment algorithms. *Human Vision and Electronic Imaging XV, 7527*, 75270H. https://doi.org/10.1117/12.845382

Shang, X., Wang, G., Zhao, H., Liang, J., Wu, C., & Lin, C. (2017). A new combined PSNR for objective video quality assessment. *Proceedings - IEEE International Conference on Multimedia and Expo*, (April), 811–816. https://doi.org/10.1109/ICME.2017.8019494

Sodagar, I. (2011). The MPEG-dash standard for multimedia streaming over the internet. *IEEE Multimedia*, *18*(4), 62–67. https://doi.org/10.1109/MMUL.2011.71

Sundari, G., Bernatin, T., & Somani, P. (2015). H. 264 encoder using Gstreamer. *IEEE International Conference on Circuit, Power and Computing Technologies, ICCPCT 2015*, 1–4. https://doi.org/10.1109/ICCPCT.2015.7159511

Tiusanen, R., Malm, T., & Ronkainen, A. (2020). An overview of current safety requirements for autonomous machines-review of standards. *Open Engineering*, *10*(1), 665–673. https://doi.org/10.1515/eng-2020-0074

Xu, S., Perez, M., Yang, K., Perrenot, C., Felblinger, J., & Hubert, J. (2014). Determination of the latency effects on surgical performance and the acceptable latency levels in telesurgery using the dV-Trainer® simulator. *Surgical Endoscopy*, *28*(9), 2569–2576. https://doi.org/10.1007/s00464-014-3504-z

Yaghoubi, S., Akbarzadeh, N. A., Bazargani, S. S., Bazargani, S. S., Bamizan, M., & Asl, M. I. (2013). Autonomous robots for agricultural tasks and farm assignment and future trends in agro robots. *International Journal of Mechanical and Mechatronics Engineering*, *13*(3), 1–6.

Zeng, W., & Lan, J. (2008). Real Time Multimedia. *Encyclopedia of Multimedia*, 757–762. https://doi.org/10.1007/978-0-387-78414-4_193