Development of a Robotic Wireless Network for Underground Mine Rescue

A thesis
submitted in fulfilment
of the requirements for the Degree
of
Master of Engineering
in Electronic and Computer Systems Engineering
at
Victoria University of Wellington

By
Jeffrey Wu

Victoria
UNIVERSITY OF WELLINGTON

*Te Whare Wānanga
o te Ūpoko o te Ika a Māui*

www.vuw.ac.nz

Victoria University of Wellington

2018

Abstract

HADES is a mine scout robot designed to be deployed by first responders to assess mine conditions post-disaster. HADES requires the capability of wireless communication between HADES and the operators on the surface station. Post-disaster mine conditions pose significant challenges to communication systems. This thesis discusses a wireless solution using 802.11b ad-hoc radios. The system developed in this thesis, named HERMES, comprises of the node system RF electronics, internal firmware, and ROS interface to the endpoints. A node design is developed in conjunction with a 2.4 GHz radio module capable of transmitting basic video data. The wireless HERMES nodes are stacked in the HADES robot and deployed as the robot traverses the mine. These wireless nodes are operable for at least 8 hours and have a range of 80 m. The wireless network formed by HERMES allows both video data and sensor data to return to a base station outside the mine. A bespoke decimation in time compression video compression strategy is implemented which provides a basic monochrome video stream with 320 x 240 resolution. This enables video to be streamed through the HERMES network with an overall through-put of 160 kbps on the application level. This basic compression is investigated and evaluated and a video stream with a peak refresh-rate of 40 frames per second with an 800 millisecond response delay is achieved.

Acknowledgements

This thesis would not be possible without the efforts and guidance of several key individuals whom I wish to acknowledge here.

The technical expertise of the VUW technicians, Tim Exley, Jason Edwards, and Brandon Pothoven appeared bottomless at times. Their efforts have been invaluable and have allowed this project to be what it is. Professor Dale Carnegie's advice and guidance has been highly inspirational over the course of this project. His attention to detail is legendary.

Special mentions to Nick. He is a constant source of technical advice.

I thank all of you for your support.

Table of Contents

Abstract.....	i
Acknowledgements.....	iii
Table of Contents.....	v
List of Tables	xi
List of Figures	xiii
Digital Appendix.....	xvii
List of Abbreviations	xix
Chapter 1 Introduction	1
1.1 Motivation.....	1
1.2 HADES	1
1.3 Principal Objectives	2
1.4 Area of Operations	3
1.5 Thesis Structure	4
Chapter 2 Literature Review	7
2.1 Previous Work	7
2.1.1 Prior work on communications by Molyneaux: A dual tether-node solution	7
2.1.2 Tethered solutions	8
2.1.3 Wireless nodes by Heesterman	10
2.2 Existing Wireless Communication Protocols	10
2.2.1 802.11b/g/n	10
2.2.2 802.15.4.....	11
2.2.3 2.4 GHz RF characteristics	12
2.2.4 700-900 MHz sub-GHz devices.....	12
2.2.5 Commercial wireless mine communication devices	13
2.2.6 Modern wireless summary	14
2.3 Wireless Node Power Source.....	14
2.4 Support Technologies	15
2.4.1 Antenna hardware	15
2.4.2 Video transmission characteristics.....	16
2.4.3 Video compression.....	16
2.5 Results.....	17
2.5.1 HERMES node RF characteristics	17
2.5.2 HERMES node shape.....	19
2.5.3 HERMES teleoperation characteristics.....	20

2.5.3 Summary of existing wireless mine devices	20
2.6 HERMES Specifications.....	21
2.6.1 HERMES RF properties.....	21
2.6.2 HERMES Data-link properties	21
2.6.3 HERMES Network topology	22
2.6.4 HERMES Data transfer.....	22
2.6.5 HERMES Power source, peripheral support and error management.....	22
2.6.6 HERMES Components cost.....	23
2.6.7 HERMES Size and weight restrictions	23
2.6.8 HERMES Desirable characteristics	23
2.7 Summary	24
Chapter 3 Initial Design, Prototyping and Experimentation.....	27
3.1 Initial Component Exploration.....	27
3.2 ERIC9 by LPRS	29
3.2.1 900 MHz link budget analysis	30
3.2.2 Evaluation testing and results	32
3.3 WGM110E by Silicon Labs.....	32
3.3.1 WGM110 dual solution.....	33
3.3.2 Summary	35
3.4 WizFi220 by WizNet (w/ ATmega2560 MCU interface)	35
3.4.1 2.4 GHz link budget analysis	36
3.5 Comparative Evaluation.....	38
3.6 WizFi220 Prototype	38
3.6.1 Experimentation method	41
3.7 Improvements to Node Design	41
3.8 Prototype Results Evaluation.....	42
3.8.1 Initial conclusions	44
3.9 WizFi220 MCU Requirements	44
3.10 Summary	45
Chapter 4 HERMES Electronics Design	47
4.1 Overview.....	47
4.2 Top PCB design	48
4.2.1 WizFi220 RF module.....	49
4.2.2 Power LEDs	50
4.3 Bottom PCB Design.....	51
4.3.1 Microcontroller unit: Teensy 3.2 and MK20DX256.....	52

4.3.2 Power regulator	55
4.3.3 Node diagnostic peripherals.....	56
4.3.4 Deployment detection accelerometer.....	57
4.3.5 System Power draw.....	58
4.4 Lithium-polymer Battery system	59
4.4.1 Solution one: 11.1V 3 cells.....	59
4.4.2 Solution two: 8.4V 2 cells.....	59
4.4.3 Battery solution summary	60
4.5 Electronics enclosure and board placement.....	60
4.5.3 PCB mounting.....	60
4.5.2 Enclosure.....	61
4.5.3 Antenna	62
4.6 Summary	63
Chapter 5 HERMES Embedded Software and Network Design	65
5.1 Overview.....	65
5.2 HERMES node embedded software	65
5.2.1 Deployment and setup process.....	65
5.2.2 Network design	68
5.2.3 Network setup	72
5.2.4 Data transfer protocol.....	74
5.2.5 Main process	75
5.2.6 Peripheral handling and monitoring.....	80
5.2.7 Endpoint design	82
5.3 ROS interface software	84
5.3.1 Base Station node structure.....	85
5.3.2 HADES node structure.....	85
5.3.4 Control input	87
5.3.5 Data packet protocol	89
5.4 Summary	90
Chapter 6 HERMES System Video Processing	93
6.1 Video acquisition hardware	93
6.1.1 Naïve video transmission	93
6.3 Video transfer strategy	94
6.3.1 Still Images for Observation	95
6.3.2 Moving Video for Driving	96
6.3.3 Compression Analysis.....	96

6.4 Single-Threshold 1-bit Run length encoding	97
6.5 Decimation in time compression.....	98
6.6 Summary	100
Chapter 7 HERMES Node evaluation	103
7.1 Evaluation environment	103
7.1.1 Testing environment 1: Table tests	103
7.1.2 Testing environment 2: Cotton Corridors	103
7.1.3 Testing environment 3: Wright's Hill Fortress	104
7.2 HERMES performance metrics and software testing environment	105
7.2.1 Inter-arrival delay.....	106
7.2.2 Raw average data rate	106
7.2.3 End-to-end delay	106
7.2.4 Error rate	107
7.3 HERMES node static characteristics	107
7.3.1 WizFi220 delay	107
7.3.2 MCU processing delay	108
7.3.3 Combined node delay.....	108
7.3.4 Power supply monitor	109
7.3.6 Temperature monitor.....	110
7.4 Single-hop RF performance	111
7.4.1 Experimental method	111
7.4.2 Range-Data Rate Result	111
7.4.3 Extended Range Result	112
7.4.4 RSSI in tunnels	113
7.4.5 Non-line of sight test: 90 degrees.....	114
7.4.6 Non-line of sight test: 30 degrees.....	116
7.5 Application performance	117
7.5.1 Decimation in time software performance	117
7.5.2 Decimation in time video quality performance.....	119
7.6 Evaluation of HERMES performance.....	122
7.6.1 RF range	122
7.6.2 RSSI	123
7.6.3 Application level data-rate	124
7.6.5 HERMES video performance	124
7.7 Summary	126
Chapter 8 Conclusions and Summary of HERMES performance	129

8.1 Overview	129
8.1.2 Hardware performance	129
8.1.3 Network and software performance	131
8.1.3 Bill of materials	131
8.2 Future Work	132
8.2.1 User Operability	132
8.2.2 Software Improvements	133
8.2.3 Hardware Improvements	134
8.2.4 Video system Improvements	134
8.3 Final Summary	135
Chapter 9 References	137
Chapter 10 Appendix	145
A.1 1-bit RLE Compression	145
A.1.1 Video Processing Algorithms	145
A.1.2 1-bit RLE system overview	147
A.1.3 Encoding process	148
A.1.4 Decoding process	152
A.1.5 Video Output	155
A.1.6 1-bit RLE Video compression static results	155
A.1.7 Adjustable threshold – 1-bit RLE	156
A.2 DIT compression scheme	157
A.2.1 Encoding Process	157
A.2.2 Decoding Process	158
A.2.3 Decimation in time compression performance	160
A.2.4 DIT vs. 1-bit RLE performance comparison	162

List of Tables

Table 2-1: Wireless mine devices	13
Table 2-2: Wireless technology summary	14
Table 2-3: Battery technology summary.....	15
Table 3-1: Evaluation of wireless devices	28
Table 4-1: Teensy 3.2 current draw	54
Table 4-2: HERMES estimated power draw.....	58
Table 5-1: IP address assignment scheme.....	70
Table 5-2: 3-byte header format.....	75
Table 5-3: Error monitoring thresholds	80
Table 5-4: Controller bindings	88
Table 5-5: Teleoperation control packet format	88
Table 5-6: Data packet message.....	89
Table 5-7: Sensor packet message	90
Table 5-8: Control packet message	90
Table 7-1: DIT encoding delay performance	117
Table 7-2: DIT decoding delay performance	118
Table 7-3: Control packet delay characteristics	118
Table 8-1: Bill of materials – HERMES node	132

List of Figures

Figure 1-1: HADES	2
Figure 2-1: The HADES node deployment system.....	8
Figure 2-2: HADES tether detachment system.....	9
Figure 2-3: Wireless node designs	10
Figure 3-1: The ERIC9 Development board.....	30
Figure 3-2: The WGM110 with the SiLabs Development Board	33
Figure 3-3: Dual Node WGM110 network arrangement	34
Figure 3-4: Dual node channel assignment.....	34
Figure 3-5: WizFi220 and individual module.....	35
Figure 3-6: WizFi220 Testing platform	36
Figure 3-7: Level shifter schematic	36
Figure 3-8: WizFi220 prototype board with node enclosure	39
Figure 3-9: Prototype PCB design	40
Figure 3-10: Prototype node enclosures. Left: The first node enclosure. Right: A second node enclosure prototype.....	41
Figure 3-11: Third iteration of the HERMES node enclosure	42
Figure 3-12: ATmega2560 prototype data-rate performance	43
Figure 4-1: HERMES electronics design.....	47
Figure 4-2: HERMES hardware block diagram.....	48
Figure 4-3: Altium mockup of HERMES top PCB	49
Figure 4-4: Top PCB Block diagram	49
Figure 4-5: HERMES constructed top PCB board	50
Figure 4-6: WizFi220 Schematic	50
Figure 4-7: HADES Bottom PCB	51
Figure 4-8: Full HERMES bottom board construction	52
Figure 4-9: Block diagram of the bottom HERMES PCB board.....	52
Figure 4-10: HERMES main MCU unit	53
Figure 4-11: Teensy 3.2 UART output at 921.6 kbaud.....	54
Figure 4-12: HERMES power block schematic.....	55
Figure 4-13: Efficiency of the NCP6306 regulator.....	56
Figure 4-14: Accelerometer Schematic.....	57
Figure 4-15: HERMES battery solution.....	59
Figure 4-16: HERMES battery height comparison.....	60
Figure 4-17: PCB electronics fit in enclosure	61
Figure 4-18: HERMES node with electronics size comparison.....	62

Figure 4-19: HERMES node antenna placemnt.....	62
Figure 4-20: HERMES antenna	63
Figure 5-1: Accelerometer processing code in waitForDeployment	67
Figure 5-2: Breakpoint statement for deployment detection.....	68
Figure 5-3: Deployment angle vs. Accelerometer output	68
Figure 5-4: Linux HADESNET setup.....	69
Figure 5-5: IP address setup code	70
Figure 5-6: UDP port assignments.....	72
Figure 5-7: WizFi220 boot sequence	72
Figure 5-8: HADESNET network search routine	73
Figure 5-9: HADESNET connection routine	73
Figure 5-10: Example data packet	74
Figure 5-11: IP address scheme block diagram	76
Figure 5-12: HERMES MCU main process block diagram.....	76
Figure 5-13: Main process implemented as cpp code	77
Figure 5-14: Error checking and gating statements	77
Figure 5-15: CID routing function	78
Figure 5-16: 3-byte header processing.....	79
Figure 5-17: Packet data processing and re-routing.....	80
Figure 5-18: WizFi220 transmission of node monitoring packet	81
Figure 5-19: Decleration of ROS libraries	82
Figure 5-20: ROS global variables.....	83
Figure 5-21: ROS node advertisements on endpoint hardware	83
Figure 5-22: Transmitting a packet from the RF radio to the endpoints.....	83
Figure 5-23: ROS callback for RF data transmit	84
Figure 5-24: HERMES base station node network.....	85
Figure 5-25: HERMES launch file.....	85
Figure 5-26: HADES robot node network	85
Figure 5-27: Internal data bridge callbacks.....	86
Figure 5-28: HADES ROS node launch file	87
Figure 5-29: XBOX360 controller.....	87
Figure 6-1: RealSense SR300	93
Figure 6-2: Basic Encoding	97
Figure 6-3: Run length encoding	97
Figure 6-4: 1-bit RLE encoding	97
Figure 6-5: DIT compression block diagram.....	99
Figure 6-6: DIT Frame decimation scheme	99

Figure 7-1: HERMES table test	103
Figure 7-2: Cotton corridors	104
Figure 7-3: Area of Wright's Hill tunnel testing area.	104
Figure 7-4: Wright's Hill tunnel interior.....	105
Figure 7-5: HERMES output debug terminal	105
Figure 7-6: MCU delay vs. packet size.....	108
Figure 7-7: Battery monitor characteristic	110
Figure 7-8: Temperature monitor characteristic	111
Figure 7-9: Range data-rate result.....	112
Figure 7-10: Extended data-rate result.....	113
Figure 7-11: RSSI result for single-hop range	113
Figure 7-12: Non-LOS 90 degree cornering test	114
Figure 7-13: Data-rate result, non-LOS test.....	115
Figure 7-14: RSSI result, non-LOS test.....	115
Figure 7-15: Angled tunnel experiment	116
Figure 7-16: Data-rate result, 30-degree bend	116
Figure 7-17: RSSI range result, 30 degree bend	117
Figure 7-18: DIT with differing thresholds.....	119
Figure 7-19: Mine interior using DIT	120
Figure 7-20: Human identification using DIT compression	121
Figure 7-21: Archway tunnel	121
Figure 7-22: Obscured visual field.....	122
Figure 7-23: Rockfall shilouette	122
Figure 8-1: HERMES node.....	130

Digital Appendix

The appendix of this thesis is split into two sections. The first section is attached to the end of this thesis and comprises appendix sections A.1 and A.2. The remainder of the appendix is contained digitally and the contents are listed in this section.

- **A.3 HERMES Compression Performance Samples.** This section comprises four videos and four compressed video clips.
- **A.4.1 HADES_BASE_STATION_ENDPOINT:** Base station endpoint node code
- **A.4.2 HADES_COM_ENDPOINT:** HADES endpoint node code
- **A.4.3 HERMES_NODE_REV3:** HERMES node code
- **A.5.1 BASE_OP:** Base station ROS process
- **A.5.2 COMOP:** HADES ROS process
- **A.5.3 hades_base_stn_proc:** This code is the Decimation in Time decoder process
- **A.5.4 hades_base_stn_proc_1BIT:** This code is the 1-Bit RLE compression decoder.
- **A.5.5 hades_video_proc:** This code is the Decimation in Time code. However, this version has been modified to allow the results in section 7.5.2.2 to be acquired.
- **A.5.6 hades_video_proc_1BIT:** This code is the 1-Bit RLE compression encoder.
- **A.5.7 hades_video_proc_WORKING:** This code is the Decimation in Time encoder process.
- **A.6 BOM:** Bill of materials.
- **A.7 HERMES setup routine.**

List of Abbreviations

This thesis references the names HADES and HERMES. These are names referring to only the robot and the associated wireless node as part of this project. Despite being abbreviations, they are used as names for the purposes of this thesis. HADES stands for “Hazard Analysis Disaster Environment Scout” and HERMES stands for “HADES Electronic Radio Module Ejection System”.

- RF: Radio Frequency
- ROS: Robot Operating System
- RTT: Round Trip Time
- RSSI: Received Signal Strength Indicator
- MCU: MicroController Unit
- PCB: Printed Circuit Board
- SPI: Serial Peripheral Interface
- UART: Universal Asynchronous Receiver / Transmitter
- ISM: Industrial, Scientific, Medical
- UDP: User Datagram Protocol
- MAC: Medium Access Control
- OSI: Open Systems Interconnection
- OFDM: Orthogonal Division Frequency Multiplexing
- LED: Light-Emitting Diode
- FPS: Frames Per Second
- NZD: New Zealand Dollars
- RLE: Run Length Encoding
- DIT: Decimation In Time

Chapter 1 Introduction

1.1 Motivation

The unknown and rapidly changing conditions encountered post-disaster in a mine environment pose huge risks to human rescue teams. HADES is a mine scout robot that can be used to safely and quickly determine the condition of mines post-disaster. HADES requires a communications system capable of streaming video data for tele-operation. Video transfer requires specific timing constraints to be met in order for the video feed to be useful to an operator. Furthermore, video transfer also requires significantly higher data-rates than simple text transfer. This raises the data-rate specification of the communication system.

Other challenges are identified in the deployment of a wireless network system inside a mine post-disaster. It cannot be assumed that existing monitoring devices inside a mine can aid in communications or provide any other kind of support. Under worst case conditions, all mine electronics systems and normal-operation communication lines can potentially be cut. The communication system hardware must therefore be brought into the mine via the HADES scout robot itself and deployed from the robot. The communication system is also constrained by weight and by size. As these are devices sealed from the outside world, battery life is limited, and therefore the transmit power and range is also limited. The real-time deployment of communications devices in a disaster environment means that the communication nodes must be tolerant to changing external conditions or suboptimal positioning. Post-disaster hazards such as rock-falls, fire, and water are all hazards that can damage or destroy electronic equipment such as wireless nodes. The range and unknown conditions can cause interference with RF equipment, and this can significantly impact on the effective data-rate of the network.

This thesis specifies, designs and tests a wireless node system is designed to operate as the HADES wireless communication system. This wireless communications system is required to transmit driving video alongside other data intended for the end user to allow the HADES robot to perform its role as a mine scout robot.

1.2 HADES

HADES is shown in figure 1-1 [1]. HADES offers rescue teams a fast and accurate means of assessing the mine conditions post-disaster. This presents mine operators with a bespoke cost-effective device that is suited to unique conditions encountered in a mine disaster situation. HADES features no external actuators but implements a wide array of sensors and cameras that allows it to assess mine conditions. Included on the HADES system are gas level sensors, temperature and

humidity sensors. A camera feed is also provided which is vital in allowing an operator to drive HADES into the mine. The camera also provides a visual indicator of mine conditions.

Figure 1-1: HADES

HADES has a top speed of 5 km/hr. HADES' physical features include its small body size with large whogs. Figure 1-1 shows the whog (wheel-leg) design that allows HADES to climb obstacles. The spokes and gripped feet allows the whogs to grip uneven surfaces and climb difficult terrain. A small body minimizes HADES' form and this allows it to enter areas that can be blocked off to larger robots. These features allow it to traverse terrain that is likely found in mines post-disaster. The HADES robot is both waterproof, invertible and pressurized. The HADES system is sealed and hence this eliminates the risk that HADES becomes an ignition hazard whilst in the mine. HADES essentially forms an intrinsically safe moving sensor array capable of scouting a disaster location. Its features allow it to quickly identify mine conditions to aid rescue efforts.

1.3 Principal Objectives

A critical feature of HADES that is required is the communication link between HADES and the outside world. This thesis focusses on the network design and the development of a wireless network for the HADES robot. The developed solution is termed HERMES, a wireless node system which HADES uses to communicate to the surface station operator. The problem explored in this thesis examines how a wireless network could be built between HADES and the outside world using small deployable nodes. The network system deployed faces several challenges summarised below.

- **A wireless node system requires an internal battery supply.** This battery supply limits the power available to each node.
- **The nodes can be deployed into unknown ground and tunnel conditions.** A mine may have uneven ground or obstacles within the tunnel interior. Cave-ins and debris can block line of sight needed for effective radio connections.
- **As data-rates of a network can be impacted by bad tunnel conditions, it is possible that the overall data-rate of the network can be reduced.** It is desirable to use compression to reduce the amount of data transmitted over the network to maximize the video stream's loss tolerance.
- **The size and shape of the nodes needs to be such that a cost-effective number can be carried by HADES and be deployed.** This limits the size and weight of the nodes so that the range of HADES is not impeded by the need to return for more nodes.
- **The placement of the nodes inside the mine is unlikely to be deterministic.** Due to HADES having no actuators that can affect node positioning after initial deployment, the position of the node being deployed is essentially final. A method of autonomously detecting successful deployments is desired.
- **A video-stream is to be supported by the wireless network, and this must be synchronous with control signals.** This allows for effective teleoperation of the HADES robot. Periodic non-time-sensitive sensor measurements should also be supported.

1.4 Area of Operations

The Pike River Mine disaster in 2010 demonstrated the lack of specialized scouting vehicles that delayed rescue efforts and proved the non-viability of non-bespoke robots deployed for mine rescue applications. The conditions faced in Pike River were such that multiple bomb disposal robots and other tele-operated robots were unable to gain access into the interior of the mine [1]. The Pike River mine is used as an example of mine topology for the purposes of establishing a likely operating use-case in this section.

The Pike River mine features 2.3 km of entryway tunnels alongside more than 4 km of mine interior. The main entry-way is used as the access point for the deployment of HADES, and therefore the 2.3 km of entry tunnels must first be traversed to reach the mine interior.

The HADES robot has a stated 4 hour operating time [1]. Therefore, the wireless node must be operable for at least 4 hours. Since HADES cannot cover the entire mine over this 4 hour timespan, it is reasonable to expect that HADES would need the nodes to continue to be operable for multiple re-entry attempts. The nodes would be expected to operate for 8 hours or better to facilitate two re-entry attempts. It is estimated that HADES could cover 3.3 km over 4 hours. This amounts to 1 km of tunnel interior and 2.3 km of entrance tunnels. This round-trip range of 3.3 km (penetration range

1.65 km) offered by HADES can be achieved using a combination of node count and node range. Fewer nodes can be used if each node has longer range, or more nodes can be used if each node has shorter range. Heesterman's work estimates that each tunnel within the mine has an average distance of 34 metres and a standard deviation of 20 metres [2]. Most tunnels described in his work fall under 80 metres. To conserve nodes, it is advantageous to place nodes only where required. Optimally, this implies that the range of each node should be around 80 metres to account for cornering. It is desirable for these nodes to have a slightly longer range to provide good video transmission at 80 metres, and to also provide a safety buffer to the range. A target node range of 100 metres is desired. Assuming 20 nodes are carried per run, a minimum range of 1.6 km is attained (20 x 80 m) which approximately matches the estimated travel range.

1.5 Thesis Structure

This chapter defined the problem faced by a mine scout robot, HADES. In this chapter, aspects of HADES' performance, networking requirements, and general approach was discussed. Chapter 2 discusses the related work and existing mine communications schemes that this project draws inspiration from. Chapter 2 also states the specifications of the wireless node solution named HERMES. Chapter 3 examines the process of finding an RF solution and the evolution of the HERMES design. Chapter 4 discusses the physical PCB electronics suite in the HERMES system, including the selection of components and the battery power source. Chapter 5 discusses the software operating system on HERMES. This section also examines the endpoint processors and the ROS interfaces that allows the data to be passed between HADES and the user. Data transfer strategies and control mechanisms are also discussed in this section. Chapter 6 discusses a bespoke compression strategy employed to allow basic video to be streamed across this network. Chapter 7 evaluates the performance of the HERMES nodes. Chapter 8 summarizes the findings of this work. This chapter also discusses avenues of future work that would improve on the HERMES wireless node system and concludes with an overview of HERMES' capabilities.

Chapter 2 Literature Review

Existing communication technologies that can be adapted for use in the HADES communication system are described in this section. Section 2.1 explores previously developed communications systems for HADES and emphasizes the need for a wireless solution. Prior work on the problem is detailed in this section.

Section 2.2 explores commonly deployed wireless networking technologies that could potentially be adapted for use in a mine environment. A focus of this section is on technologies that are already proven to carry video data, a capability vital to this communication system. These devices and the underlying technology discussed in this chapter are already widely implemented and well-understood. Other advantages of the technologies described in this chapter are their commercial and industry support. As wireless nodes need to operate standalone, a power source technology for the HERMES node system is investigated in section 2.3. Section 2.4 discusses the vital support systems required by this communication system. This section explores topics of RF antenna design and video quality metrics for teleoperation. Section 2.5 summarizes the results of this literature review. Section 2.6 outlines the specifications of a wireless node system named HERMES with reference to the problem paradigm described in chapter 1.4 and the results presented in this chapter.

2.1 Previous Work

This section discusses two prior solutions that attempt to design a communications link between HADES and the outside world. Molyneaux proposes a dual solution featuring a tether and a wireless node design [1]. An attempt to build a purely wireless network with pill-shaped nodes is conducted by Heesterman using Zigbee modules detailed in section 2.1.3 [2].

2.1.1 Prior work on communications by Molyneaux: A dual tether-node solution

Previous work on the HADES communication system by Molyneaux examined a solution featuring a wireless network sustained via wireless-relay nodes that are deployed from the HADES chassis with a tethered support [1]. This allows HADES to build its own wireless network as it traverses the mine. A wireless signal from the HADES robot is repeated across the nodes and reaches the outside of the mine. This is intended to allow a real-time camera feed to be available to the operator. Figure 2-1 shows this proposed design carrying up to 5-7 nodes per deployment tube. Each tube is external to the sealed environment of the HADES robot, with a motorized screw-like node deployment system to drop each node in position.

Figure 2-1: The HADES node deployment system

This proposed solution featured a dual wireless node and tether design. Most mine entry tunnels are straight and long before the extraction areas are reached. This common mine topology raises the possibility of being able to conserve nodes during initial deployment during the transit into the main sections of the mine. This proposed idea features a tether for the initial transit, detaching the tether after a certain distance. This allows HADES to transition to a node system appropriately where a tether becomes less suitable in the labyrinth-like and obstacle-strewn tunnels of the interior of a mine. This system saves nodes during sections of a mine where a tether would have an advantage, leaving the limited supply of nodes to be deployed in the mine interior.

The initial design of the wireless repeater nodes featured a tear-drop shaped enclosure that provided a mono-stable resting point. This design measured 4.5 cm x 7 cm. This node property was achieved using non-uniform mass distribution as shown in figure 2-3 (left), allowing the node to deploy with an upright antenna. However, further development on this solution was limited. It was theorized that this solution would use an existing wireless standard to achieve wireless communication.

2.1.2 Tethered solutions

Figure 2-2 shows a tether mount designed for the HADES robot, using a sliding detachable mechanism [1]. This provides a mount for the dual node-tether solution proposed in section 2.1.1. The tether transmission medium provides up to 1 Gbps using fibre optic shielded with a Kevlar jacket. This provides ample bandwidth for transmission of data. Unlike copper cored cable, fibre optic cannot provide power to the robot. This design trades this additional functionality for weight and durability improvements. The detachment system jettisons the tether once it is no longer suitable and the HADES robot switches to a wireless network system. The detachable section of the tether would then also provide the first wireless node.

Figure 2-2: HADES tether detachment system

Fully tethered solutions are unsuitable for mine communications between HADES and the outside world. Historically, tethered solutions pose a high risk of immediate mission failure once the tether is severed or otherwise immobilizes the attached scout robot. A tether adds significant weight and limits the agility of the attached robot. Furthermore, a tether severely limits the robot's deployment range, and therefore also the search range as a result. A tether has mass and this mass is dragged along the ground and increases as the distance travelled increases. Therefore, a tether cannot be drawn by HADES indefinitely. While a tether can supply a high data-rate compared to wireless technologies, the physical limits of a tether are too high for such a design to be both reliable and practical, especially in the type of terrain present in mines post-disaster. Since a tether can be easily caught, a human operator must attempt to manoeuvre HADES to ensure that obstacles are cleared. This requires maintaining a spatial awareness that may not be possible in a mine environment post-disaster. In a variable and volatile mine environment, it is almost impossible to guarantee physical tether safety.

A vital flaw in the dual node-tether solution exists in the need for HADES to exit the mine. Under the assumption that the tether range is significantly longer than the wireless node range, taking the tether into the mine through the access tunnel leaves a dead zone where no wireless coverage exists in the entry section. If the tether range is not significantly longer than the wireless node range, the tether is pointless. Because the tether cannot be picked up by HADES after it is detached, (as HADES has no external actuators) HADES cannot traverse the section of the mine past the tether unless it also deploys nodes on the exit path. Overall, the dual tether-node solution does not save nodes if HADES needs to exit the mine, and hence the tether for this solution is redundant. For HADES to exit the mine, wireless nodes need to be deployed from the entrance of the mine for full wireless coverage. Therefore, a fully wireless communication solution is desired.

2.1.3 Wireless nodes by Heesterman

An alternate investigation by Heesterman into this problem resulted in the design of compact pill-shaped nodes with a flat antenna shown in figure 2-3 right [2]. Two nubs are added to the side of the node to prevent the pill from rolling when deployed on angled surfaces. The minimal size of the node designed by Heesterman was not only made possible by the small whip antenna, but also the small size of the AAA power source and associated electronics. This combination of battery source and a low-power transmitter allowed each node to be reduced in size to a length of 75 mm and a diameter of 40 mm. By abandoning the need for an upright antenna, the node is reduced in shape and the volume is further minimized compared to Molyneaux's design. The deployment and storage mechanism of the nodes designed by Heesterman was not examined.

Figure 2-3: Wireless node designs

Figure 2-3 shows the originally proposed teardrop node by Molyneaux [1]. The size of this node is 45 mm wide and 70 mm high. The centre image shows a dome shaped-node which was eventually dropped in favour of the pill shaped node shown on the right.

This pill-shaped node design uses a wireless system based on the IEEE 802.15.4 standard implemented with Zigbee. It was shown that a throughput of 1.2 kbps was attainable with a 1 second latency. This was predicted to provide one frame every two seconds with a frame resolution of 400 x 300 pixels. Several improvements to this solution are desired. A higher framerate is required for HADES to be controllable by a human in real-time. It was shown that this 2.4 GHz system was able to attain a range of 60 m.

2.2 Existing Wireless Communication Protocols

2.2.1 802.11b/g/n

Specifications for wireless networks and commonly employed commercial devices are developed, standardized, and made marketable by IEEE. In particular, the wireless standards of 802.11b/g/n constitutes a major part of wireless technologies that are currently implemented.

The basic 802.11b standard specifies a transmission scheme of up to 11 Mbps with a theoretically specified maximum range of 100 metres, depending on conditions [3]. The 802.11b radio protocol specifies the physical layer mechanisms for encoding and decoding of RF signals and standardizes the channel assignment and bandwidth allocation of the radio transmission inside the 2.4 GHz band Industrial, Scientific, and Medical (ISM) band. The exact channel assignments vary from country to country, but generally encompasses the frequency range 2412 MHz to 2462 MHz. The 802.11b standard separates these frequencies into 11 overlapping 22 MHz channels.

The 2.4 GHz ISM band is not exclusively reserved for 802.11b. Newer standards such as 802.11g/n feature more sophisticated encoding methods to allow for higher data throughput up to 600 Mbps for 802.11n or 72 Mbps for 802.11g. To facilitate wireless connectivity between various devices, most 802.11b/g/n radios are back-compatible. The widespread nature of the 802.11 technologies among wireless devices also provides a common physical layer for internet devices. 802.11 devices are commonly implemented alongside higher level internet protocol standards such as 802.3 Ethernet. This allows the 802.11 technologies to be easily implemented into existing internet protocol frameworks. Other radio protocols also use this 2.4 GHz ISM band, including those that are not well suited for use as the wireless technology for this application. One such protocol is Bluetooth which instead uses frequency-hopping spread spectrum over the entire ISM band [4]. The transmission power of between 1 and 100 mW allows Bluetooth to be implemented in very small packages and provides a maximum data-rate of up to 1 Mbps. The maximum range is generally low at 10 m.

The performance of 802.11b at open-air ranges is explored by Anastasi et al. [5]. Their work explores the performance of the User Datagram Protocol (UDP) under 802.11b ad-hoc conditions. They show that the effective data-rate of 802.11b is limited depending on the range of the device. The 802.11b performance at 1 Mbps is shown to be effective up to 100 metres. In contrast, 11 Mbps transmission was observed to be limited to a range of around 25 metres before heavy packet loss was observed. Due to the additional overhead introduced by the higher layers of the Open Systems Interconnection (OSI) internet protocol, actual utilization was shown to be roughly 44%. On the 1 Mbps coding rate of 802.11b, the maximum RF data rate under UDP is expected to be 440 kbps.

2.2.2 802.15.4

802.15.4 is another physical layer protocol which manages the transmission of data over low-rate wireless personal area networks [2][6]. Unlike the 802.11b/g/n standards which focus on internet traffic with data-rates up to 11 Mbps, 802.15.4 provides a means for low power transmission, primarily for battery powered devices. This standard specifies RF data rates of up to 250 kbps, an order of magnitude lower than 802.11 compliant devices. In practice, data-rates of 125 kbps are attained due to overhead. As such, devices operating on this protocol focusses on lower data rates. The 802.15.4 protocol operates on three channels, 800 MHz, 900 MHz, and 2.4 GHz. The 802.15.4

protocol is intended to be used for wireless sensor networks. The Zigbee protocol is developed by the ZigBee Alliance provides high level networking functions for data routing [7]. The Zigbee protocol is designed for versatility, non-user intervention, and reliability.

2.2.3 2.4 GHz RF characteristics

Heesterman's solution features the use of a network based on the 802.15.4 standard, operating on the 2.4 GHz band. Despite the poor wireless data-rate, the range attained of around 60 m provides opportunities for improvement in performance by exploring better options on the 2.4 GHz ISM band.

Modern wireless technologies that provide reliable video streaming are already widely adopted. Video streaming relies on bandwidth and low-latency, two major concerns that are addressed with the commonly employed 802.11 family of wireless protocols. The use of the 802.15.4 protocol in mines allows simple monitoring data to be transmitted in mine conditions as shown in the works of Bandyopadhyay et al., and Patrie and Minaje [8][9]. The commonly used Zigbee protocol (in conjunction with 802.15.4) provides a maximum of 250 kbps which is a very low data rate compared to the 802.11b/g/n family of protocols. Despite the advantage of 802.11 wireless over 802.15.4, Heesterman [2] notes that commercial hardware required to implement 802.11 radios is significantly larger and requires more physical space. Modern routers and high-power routing devices can provide the data-rate required for video streaming, but size and cost remain an issue with commercially complete solutions. A small 802.11b device is desired for low size implementation if possible.

The practical effectiveness of the 2.4 GHz band in mine tunnels is tested by Patri and Mimaje [9]. This work evaluates the effectiveness of 2.4 GHz signals in a mine environment. Their results establish how a 2.4 GHz signal strength changes when propagating through mine tunnels. Their work leads to the conclusion that long tunnels allow the signal to travel further than in an area with more obstacles. It was again shown that the default power Zigbee transmission had an effective range of around 85 m before signal loss. The work of Zhou et al. and others infers the average attenuation loss of a 2.4 GHz signal to be around 20 dBm per 100 m in tunnels [9][2][10], although it is difficult to evaluate the exact received signal strength at any particular point in a tunnel.

2.2.4 700-900 MHz sub-GHz devices

The newer IEEE 802.11ah standard describes a 900 MHz radio system that maintains an RF data rate in the range of megabits per second. Common proprietary 900 MHz transmission devices only attain up to 500 kbps RF data-rate [11][12][13]. Performance of this standard in practice was examined by Sun et al [14]. This work explores the performance of 802.11ah in comparison to the other 802.11 family of standards. In this paper, 802.11ah is demonstrated to have longer range but slightly worse RF-data rate compared to 802.11b in urban environments. However, the data rate remains higher than the data rates provided by the 802.15.4 family of protocols, allowing for up to 1 Mbps depending on

external conditions. This reveals the viability of 900 MHz systems when trading off range against data rate. This data-rate at range is achieved through more sophisticated Orthogonal Division Frequency Multiplexing (ODFM) channel encoding and a larger bandwidth. Zhou et al.'s [10] work explores the effectiveness of 900 MHz and this work shows that the attenuation loss of 900 MHz is inferred to be at roughly 14 dBm per 100 m.

2.2.5 Commercial wireless mine communication devices

Table 2-1 shows that currently available mine devices are only constructed for the bespoke purpose of transmission and routing of monitoring data. This use case is reflected in the design and deployment of such devices. Most devices in table 2-1 demonstrate data-rates of 100 kbps or lower, allowing for text or sensor data transfer. These devices are not intended to be emergency devices, and hence the size of these devices allows for large power supplies and allows them to be low-maintenance general-purpose monitoring and data repeater devices.

Table 2-1: Wireless mine devices

Device	Manufacturer	Operating frequency	Use case	Target Range with data-rate	Power supply	Function
PT 7000 [15]	Orbcom	Cellular band with optional satellite	Equipment and asset management	Not Stated	9 V - 23 V DC Jack with backup battery system.	Vehicle monitoring in all mining applications
MDT_RTU/ SAA_RTU/ [16]	Mine Design Technologies	900-928 MHz Tx +14 dBm transmit power	Data acquisition device – sensor interface	Underground: Up to 150 m Opencast: Up to 600 m Data Transfer rate: 2.6 kbps	Battery capacity not stated: But minimum 1-year operation, typical 3 years.	Both underground mines and opencast.
Crescendo Router [17]	RFInnovations	148 to 174 MHz and 380 to 520 MHz Tx +37 dBm transmit power	Data Router	Up to 19.2 kbps.	12 V power supply	Both underground mines and opencast.
NS40 Wireless Network Switch [18]	MST – Impact series	2.4 GHz ISM with Tx +24 dBm	Data Router	802.11b/g/n 1 - 54 Mbps	8 - 15.1 V battery supply	Both underground and surface
Impact Mobile Mesh Network Node [19]	MST – Impact series	2.4 GHz ISM with Tx +16 dBm	Mesh node router	802.11b/g/n with 2.4 GHz mesh routing	Lithium 96 Wh	Both underground and surface

The availability of external power provides significant advantages to mine monitoring devices, as the device can be powered indefinitely. Both the Crescendo router and the PT7000 operate on a 12 V external power supply. This offers the PT7000 the capability to connect with satellite and cell towers. Furthermore, the Crescendo router is afforded 37 dBm transmit power with dual-band capability. Both

devices feature low rate transmitters but significant operating range. Devices such as the NS40 and the MST Impact Mobile Mesh Network Node feature significantly high transmit powers but are limited by requiring a power supply and a weight of 3.45 kg respectively. As such, it is shown that a high power router with 25 dBm and high transmit rates requires a large 96 Wh battery. While the transmit power and data-rate matches expected router-level speeds, the weight and size of such a solution remains a problem, with the Mesh Network Node having dimensions of 223 x 310 x 92 mm. It is clear that devices of this size are unsuitable for deployment of HADES nodes. The large device size is mainly due to the 96 Wh battery system which grossly exceeds the operating time requirements stated in section 1.4.

2.2.6 Modern wireless summary

Table 2-2: Wireless technology summary

Wireless Protocol	RF channel	RF data-rate	Commonly implemented Effective Range	Common Implementations
802.11b/g/n	2.4 GHz ISM	1-72 Mbps	Up to 100 m	WiFi cards, adaptors and hardware
802.11ah	900 MHz	0.65 to 8 Mbps	Up to 1000 m	Not commonly used. (released late-2017)
Bluetooth	2.4 GHz ISM	BT3.0,4.0: 25 Mbps BT5.0: 50 Mbps	10 m (BT3.0) 60 m (BT4.0) 240 m (BT5.0)	Cell phones – short range devices
802.15.4	800 MHz, 900 MHz, 2.4 GHz ISM	250 kbps	Up to 100 m	Low-rate/Low Power wireless sensors
LTE/4G	700 MHz to 2.6 GHz	1.4 to 40 Mbps	Variable depending on transmit power.	Cell phone technology, large area networks
Various proprietary 900MHz devices [11][12][13]	900 MHz	Up to 500 kbps	Up to 1 km	Wireless sensors/ low-rate networking

2.3 Wireless Node Power Source

A wireless node is isolated from the outside world and requires an internal power supply. Several commonly available battery technologies are explored to investigate various battery systems for implementation as a power source. The results are shown in table 2-3. Important characteristics of the node electronics are the power density per unit size and power density per unit weight. The wireless node design constricts the size of the node, and hence the size of the battery. Also of concern is the weight restriction. Since HADES is designed to be agile and lightweight, the node weight should be minimized so that the number of nodes that can be carried is maximized.

Table 2-3: Battery technology summary

Battery Technology	Operating voltage (per cell)	Power capacity per weight (Wh/kg)	Power capacity per volume (Wh/L)	Common applications	Common form factors
Alkaline	1.5 V	85-190 [20]	250-430[20]	Household appliances – low current draw devices	AA, AAA non-rechargeable batteries
Nickel-Metal-Hydride	1.2 V	60-120[21]	401[22]	Rechargeable devices	Cell form, AAA, AA
Lithium-polymer	3.7 V	100-200	170-460	Hobby electronics	Hobby power packs, stick/block form.

Table 2-3 shows that the lithium battery provides the best combination of power per volume and power per weight, with the highest power per density of each of the technologies. An advantage of lithium polymer cells is that they can be constructed to fit any form, as the cells can be made to any shape. Many commercially available batteries such as AA and AAA are cylindrical, which limits their packing ability. A wide swing in energy capacity is observed with standard alkaline batteries. Alkaline AA and AAA batteries have a capacity that is inversely proportional to their current draw. These batteries hold more charge when slowly discharged as opposed to when they are quickly discharged [23]. Therefore, their current capacity can only be accurately expressed once the target current draw is known. However, most alkaline batteries are unable to deliver more than 1 A instantaneous current [23]. Current draw at this level significantly limits battery life. Therefore, the upper capacity limit is observed in alkaline batteries when the current draw is roughly around 25 mA [23]. External environmental conditions also have a marked effect on alkaline batteries' performance [23]. In contrast, lithium technologies can source significantly higher currents, with capacities independent of their current draw over their lifetime. A predictable and constant current sourcing capability is an important advantage. Battery life is also affected by factors such as temperature and humidity. High heat and irregular atmospheres are commonly encountered in mines. This makes lithium technologies more desirable as the current capacity is not as heavily affected by outside conditions as other battery chemistries [24]. Differing manufacturers and processes are responsible for the wide power-capacity range of lithium polymer batteries. The battery capacity, size and weight is hence highly dependent on the battery source.

2.4 Support Technologies

2.4.1 Antenna hardware

The use of a wireless device raises the need for an associated antenna system. Industry guides [25] show the importance of antenna deployment and selection as antenna performance significantly impacts RF performance. RF range is a significant factor in the performance of the HERMES node. Due to the deployed nature of the antenna, the location of the antenna cannot be made optimal as nodes can only be dropped and not precisely placed. The precision of the deployment cannot be

guaranteed, and therefore directional antennas (such as dish or Yagi antennas) cannot be relied on to be accurately positioned to maximize transmission gain. The work of Da Vita [26] describes the antenna types that are available and concludes that the cheapest high gain antenna comes in the form of a dipole antenna. Other types such as helical or spiral antennas are shown to be highly dependent on positioning or form factor. The positioning of the node guarantees that the antenna will be placed near the ground. This poses several problems as line of sight is important for 2.4 GHz transmission over distance. Furthermore, space is a concern for antennas, as antenna gain is generally achieved with larger antenna units. Due to the need for the wireless nodes to be able to be carried by HADES, a method of storing the units with an antenna is desired and hence an antenna cannot be arbitrarily large.

2.4.2 Video transmission characteristics

As video is a stream of images presented to a user, a video stream is reliant on certain conditions in order to be effective. One important metric is the frame-rate of the video. The work of Chen et al. states that a lower limit of video frame-rate for teleoperation of robots is roughly 10 Hz [27]. It was shown that frame-rates below this point resulted in significantly degraded performance of tele-operated actuation. Conversely, higher-frame rates of 10 Hz or better were observed to have little additional effect on navigation. It was shown that in this case, improved performance was achieved with higher video quality as opposed to an increase in frame-rate. This shows that a minimum frame-rate is a vital condition for robot control, but diminishing returns are observed with significant frame-rate increases over 10 Hz. This is further supported when situational awareness, and identification was examined; the frame-rate did not affect object and field of view identification. The work of Laine also corroborates this 10 Hz threshold limit for visual motion [28].

Chen et al.'s examination of teleoperation also discusses a second vital metric for robotic control, end to end delay. Delay is defined as the time between an action and a response to that action being visible. It is shown that delay has a large effect on the control actions of the human operator. While a delay of 10 - 20 ms was noticeable by the operator, real-time operation deteriorated at delays approaching 1 second. The work of Apostoloupoulos et al. estimates that interactive delay should be limited to 150 ms [29]. When the delay exceeded this delay length, the operators employed a stop and wait strategy. The operator would wait for a command to propagate through to the robotic device and this operator delay ensures that the state of the robot is consistent with the state that the operators intended it to be.

2.4.3 Video compression

Video compression involves taking video data and representing the video stream in a manner that eliminates redundancy and reduces the overall size of the frames being transmitted. By doing this, video can be transmitted over networks in an efficient manner. Video compression will be required by

this system. The long-range of the network will incur some data-rate loss due to signal degradation. This will cause the effective data-rate to be reduced further. Modern compression methods use a multitude of methods to reduce the data-rate of a raw video stream. Lossless compression involves re-representing data in a shorter form, whilst lossy compression involves the reduction of data-rate with an acceptable loss in image quality. Lossy compression generally achieves greater compression ratios. This is especially important for video, as some image quality loss is often acceptable.

The HVEC video coding scheme achieves a data-rate savings by only transmitting full frames sparingly [30]. Prediction frames are used in order to instead transmit the difference of a frame from the full frame. By doing this, small changes in a full frame does not necessitate the transmission of a full frame with slight differences, but a reduced size frame containing only the change that has occurred. A data-rate savings is attained by sending frames that reference off of previous frames and next frames and this avoids having to transmit the raw data.

An analysis of modern compression algorithms shows that exact performance of the best modern compression schemes such as H.265/HVEC, VP9 or even AVC are unknown at very low data-rates. Exact compression ratios are difficult to estimate. For example, HVEC provides a minimum compression ratio of 2 using main profile level 1. This attains a maximum data-rate of 128 kbps [31]. Assume that the low-quality video needs to fit constraints shown below.

- The frame-rate of at least 10 Hz needs to be maintained with low latency (as per 2.4.2).
- A minimum resolution of 320 x 240 needs to be maintained as the video needs to have sufficient quality to identify obstacles.

The aforementioned compression systems are typically implemented in cases where the data-rate allowable by the network is significantly higher. As shown by [30], medium level compression attains around 0.7 to 1 Mbps for HVEC, the newest compression scheme. There are only a few unverified examples of these compression algorithms being used at 120 kbps or lower. YouTube considers the minimum recommended bitrate for 360p (the lowest quality) to be 1 Mbps [32]. Modifications to the aforementioned commonly used compression algorithms or new algorithms to suit low-data-rate are discussed by Aghagolzadeh et al. [33] using a modified H.264/AVC and Talluri et al. [34] using an object based encoding system. These algorithms are highly specialized and although low bit rates up to 10 kbps can be achieved, it is unknown whether the methods described in these works can be scaled to fit the conditions required by the HERMES system.

2.5 Results

2.5.1 HERMES node RF characteristics

A review of existing mine communication schemes and wireless technologies reveals several avenues of investigation that may be suitable for the HERMES node system. The most pressing problem is the

requirement for the network to return driving-quality video to the tele-operator at home base. This capability is vital to HADES' function as a mine scout robot. The commercial existence of mine monitoring devices proves the viability of wireless devices' capability of returning numerical data, such as text or controller commands. To achieve the performance required to stream video, certain performance requirements must be met:

1. The communication system needs to have sufficient bandwidth for a video stream to be recovered.
2. The video must be simultaneously streamed alongside the other data streams, of which the most important are the controller commands. Teleoperation requires both controller commands and video to be transmitted synchronously as specified in section 1.3. This must be achieved with low latency.
3. The communication system can operate in substandard conditions, including but not limited to, near-line of sight, non-line of sight, and suboptimal deployment. Other external conditions such as temperature, humidity, and the drop shock of deployment must be considered in the design.

The above factors place performance requirements on the network. Factor 1 is a feature of the system bandwidth, and hence a function of the RF frequency. This factor limits the carrier frequency of the HADES network to 900 MHz or above to achieve this requirement using off-the shelf, commercially available technologies. Literature describing 2.4 GHz technologies both currently existing in mines and in use in other sectors demonstrates that there is an opportunity in 802.11b/g/n radios if such devices can be made in a small package that is suitable for HADES. If so, factor 2 is a requirement readily achieved by such devices. Application-level throughput in an 802.11b ad-hoc environment discussed in section 2.2.1 shows that an overall maximum transport throughput of around 44% can be achieved with UDP. Given an RF data-rate of 1 Mbps at the lowest coding rate using 2.4 GHz 802.11b, it is expected that the practical maximum data-rate utilizing UDP at the application level is around 440 kbps. A reduced data-rate is specified to account for potential data-rate loss in mine conditions at range. It is likely that an effective data-rate of around 250 kbps would be viable. A bespoke compression algorithm will be employed to handle transmission of video data over the HERMES network as currently available compression algorithms are not suitable as shown in section 2.4.3.

The result in section 2.2.3 shows that 802.15.4 is usable in mines at ranges up to 60 m. Despite this range being proven for 802.15.4, range characteristics for 802.11b/g/n inside mines is unknown. Since both technologies use the same 2.4 GHz ISM band, (and therefore have the same signal propagation properties) there may be an opportunity to extend the range of existing 802.11b radios to provide the requisite network coverage. Table 2-2 shows that most 2.4 GHz technologies can technically be

operable at close to 100 m but applications in these ranges are not common. While existing works prove the viability of transfer of simple text or numerical data, the video streaming capability of the technologies on the 2.4 GHz band in a mine environment has not yet been established. It is likely that longer range on 2.4 GHz can potentially be attained using more powerful radios. It is difficult to estimate the data-rate attained by these devices in a tunnel environment as both range and difficult terrain can affect the data-rate.

Section 2.2.4 discusses the advantage of 900 MHz technologies. A range advantage in an enclosed space is attained by common 900 MHz implementations in low-rate networks but cannot attain high-rate data transfer compared to 802.11b/g/n 2.4 GHz. Factor 2 is affected by the performance of the routing algorithms and software processing capabilities of the network elements, and is heavily affected by the network protocols that are used to route the data. Commonly implemented wireless networks (mostly using 802.11/g/n) already routinely implement protocols that can sustain low delay video streams. This is generally done using the established Open Service Interface 7-layer stack that comprises most internet infrastructure. Common 900 MHz transceivers do not implement such routing features, which may make real-time video routing difficult. The tunnel-like structure of a mine imposes significant range penalties to high-frequency transmitters. 802.11ah proves the performance of medium-rate (roughly 1 Mbps) networks at lower operating frequencies (sub-2.4 GHz) at range. The use of a lower operating frequency offers significant range benefits with good data rates after high rate encoding. The 802.11ah standard presents a viable 900 MHz transmission scheme providing data rates of around 1 Mbps. The performance of this technology appears to be uniquely suited for HERMES nodes. Despite 802.11ah having favourable characteristics, this technology is unsuitable at this time as 802.11ah is not currently in wide scale deployment. Non-custom components for general purpose use implementing this technology is rare. This technology is unsuitable for the HERMES node currently, but may become viable in the future.

2.5.2 HERMES node shape

The shape of the wireless node is an important aspect of the node design as the node shape limits the size of the internal space. Three aspects of the node design are affected by the internal space available:

- **The size of the node electronics.** The node electronics size is affected as well as the overall shape of the design. The internal electronics must be designed around the node enclosure.
- **The size of the physical networking hardware.** The antenna size is affected by the size of the node. The antenna provides the range of the device and hence must be accommodated.
- **The internal battery capacity.** The internal battery capacity is directly related to the volume of node interior space that is available. The internal battery capacity then determines the range-operating time product.

An exploration of antenna technologies reveals the need for an antenna to be deployed on the HERMES node. Due to the need for line of sight for 900 MHz+ technologies, antennas provide two significant benefits. An antenna provides innate equivalent gain, which can improve transmission strength and adds range to a link budget. The second benefit is that the antenna raises the transmitting element above the ground. This has the benefit of providing additional ground clearance to maintain line-of sight between adjacent nodes in the network. However, an antenna has significant downsides as it adds additional size to the device. Directional antennas are shown to be unsuitable for wireless nodes. Due to the adverse terrain present in mines, the exact positioning of nodes such that a directional antenna is focussed at 80 m or better is impossible. This limits antenna design to dipole or omni-directional antennas as these antenna designs do not rely on exact positioning to function. Other ideas such as using a cave wall to reflect a signal to achieve non-line of sight communications also relies too heavily on external conditions and precise positioning to be viable.

2.5.3 HERMES teleoperation characteristics

The video stream provided by the HERMES network must maintain a viable frame-rate for video transfer. The minimum frame-rate for teleoperation is shown to be roughly 10 Hz. If possible, a higher frame-rate may be desired to account for losses of frames in-transit. The work examined in section 2.4.2 shows that the delay of the video stream is also important in determining how controllable a tele-operated robot is to a human operator. It may be the case that a large delay can be tolerated with a stop and wait-like strategy. A delay length of 1 second was observed in section 2.4.2 to be the absolute limit of delay. It is desired minimize round-trip delay to 1 second at most, with a focus on improving this where possible.

2.5.3 Summary of existing wireless mine devices

The node size directly impacts the operating time of a sealed wireless unit. Unlike wired networking devices, sealed wireless units operate from battery power, and this limits the power available to a node. The node operating time is limited by the battery power provided to the node, and hence the larger the size of the node, the greater the operating time. Existing mine wireless devices presented in section 2.2.5 are not limited in size as these devices are deployed by hand and do not need to be carried on a robot. Therefore, the size and weight of existing mine wireless devices are not constrained. Other advantages that mine monitoring devices have are the fact that external power supplies are available. This affords existing devices an almost limitless transmit power coupled with high gain antennas. The hand-installation of these wireless monitoring devices in mines provides a significant advantage as devices can be placed in optimal positions for best RF propagation.

Currently existing devices differ from the function of a sealed RF wireless node for HADES teleoperation. As the mine environment is isolated post-disaster, a deployed wireless node cannot rely

on existing infrastructure for wireless repeating or power. This means that the limited battery power must be considered in the design or selection of an RF transmitter.

The MST Impact series of mining communications devices implements a scalable 2.4 GHz 802.11 mesh system to allow for high-data-rate communication. However, while these devices are developed for mining communications, the use case of these devices are vastly different to the use case of HERMES nodes. As an all-purpose network system, the MST-Impact series devices can rely on positioning and external power to build such a mesh network. A major challenge for the HADES communication nodes involves the reduction in size of these devices for HADES to carry enough to minimize node replenishment runs, while still allowing sufficient battery power for sustained operation.

2.6 HERMES Specifications

As informed by the results of this chapter, the following sections specify the capabilities and characteristics of the designed HERMES wireless node solution.

2.6.1 HERMES RF properties

The HERMES node will employ a wireless network on an RF frequency between 2.4 GHz and 900 MHz. A focus is made on devices operating on 2.4 GHz 802.11b/g/n with high transmit power and 900 MHz devices with high data-rate. Devices using these RF frequencies are currently already used in commercial wireless mine equipment and are already commonly deployed in other sectors. The widespread nature of the technology increases the likelihood of a suitable off-the-shelf component being found as the core RF module. The target distance for full node operation is 80 m within line of sight inside the mine tunnels, extending to 100 m if possible. Current literature shows that 900 MHz devices can easily exceed this range, with 2.4 GHz being proven to be effective from 60 to 85 m in practice. The range target of 80 m is hence shown to be viable by most 2.4 GHz and 900 MHz technologies.

The HERMES node will deploy with an external dipole antenna. A dipole antenna is suitable as the node deployment process is variable and hence direction cannot be guaranteed that would enable precise positioning of directional antennas as discussed in section 2.4.1. The antenna provides a radiating element that rises above ground level, which improves line-of sight for RF propagation.

2.6.2 HERMES Data-link properties

The HERMES node's primary function is to allow data to transfer between HADES and the outside world. Currently employed devices for mine sensing and networking provide an application-level data rate of up to 100 kbps. The target application-level data-rate is 250 kbps with the aim of increasing this if possible. The examination of 802.11b in section 2.2.1 shows that up to 440 kbps on the

application level is viable. This target constitutes around 50% of the maximum data-rate accounting for potential data-rate losses due to signal degradation in a mine environment.

1. The round-trip-time latency from HADES to the outside world over 40 hops shall not exceed 1 second to allow for near-real time control of the HADES robot.
2. The frame-rate of the video feed on the user-side shall be at least 10 Hz, with an aim to increase this if possible to compensate for data loss in transmission.
3. The resolution of the resultant video stream shall be specified to 320 x 240 pixels, with a standard 4:3 aspect ratio to provide basic video.

2.6.3 HERMES Network topology

HERMES shall implement a network system allowing nodes to join the system on an ad-hoc basis. The basic paradigm network connection shall allow nodes to join the network without prior coordination. Each node must permit the network connection to be offered to new nodes that are deployed. The network connection must be maintained even if one node were to fail.

2.6.4 HERMES Data transfer

The HERMES data transfer system shall support the transfer of video data with support for synchronous controller messages to allow HADES to be tele-operated in near real time. The HERMES system shall also support the transmission of sensor data as sensing of the mine environment is a critical function of the HERMES system as described in section 1.3. As sensor data is not time-sensitive like video, the sensor data does not need to meet any time constraints.

2.6.5 HERMES Power source, peripheral support and error management

The HERMES node shall operate on a single battery charge for at least 8 hours. The power source should have support for protection of internal systems in the event of failure. As shown in section 2.3, the battery technology with the highest power density is lithium technologies. This technology is shown to have superior power density and current sourcing capability. Existing mine systems such as the Impact WRM wireless repeater node source their power from lithium batteries, as does the HADES robot itself [1].

Due to the critical function of the wireless node, the node shall have monitoring to ensure that the operator is aware of a failing or potentially failing node to minimize the probability of node failure causing immediate mission failure. Similar features in existing mine devices include battery voltage monitoring and temperature monitoring.

The relevant protection, peripheral support, and error management protocols are specified to be a combination of the following features listed below.

1. **Over-current/surge protection:** Over-current is defined by the discharge of the battery system past its instantaneous current draw capacity. This protection prevents the battery from becoming a hazard whilst in the mine.
2. **Under-voltage protection:** The under-voltage protection is defined by the empty cell discharge voltage. This protection defends against current draw when the battery charge specification is met. The under-voltage protection allows the node to detect conditions where it is approaching the end of its battery charge.
3. **Temperature detection:** The circuitry on-board the node shall detect temperature to determine the safety of node operation. This temperature detection circuitry prevents thermal runaway that can lead to battery failure (and consequentially node failure).
4. The support system shall have the capability to provide a warning to the operator to indicate anomalous events.

2.6.6 HERMES Components cost

The HERMES node shall be low cost as node recovery may be infeasible. The HERMES node electronics shall cost no more than \$5000 for a set of 20 nodes. The HERMES node is therefore specified to \$250 NZD per node.

2.6.7 HERMES Size and weight restrictions

HADES is a scout robot with an operating weight of 40 kg. This limits the node weight as HADES must carry 20 nodes per run. Assuming that 8 kg is afforded to the node system and the deployment mechanism, then the node system should comprise no more than 5 kg of this. Over 20 nodes, each node therefore cannot weigh more than 250 g. The HADES chassis size is limited as shown in section 1.2, and must be redesigned to fit the deployment mechanism. Previous HADES node designs have featured a node shape of no larger than 70 mm in the largest dimension. While it is difficult to pre-estimate the shape of the node, the primary node case should not exceed 15 cm in any dimension as this is the size of the HERMES chassis size. The antenna size is exempt from this requirement as the antenna must satisfy RF performance conditions as stated in section 2.6.1. It is desirable to minimize the size of the battery and internal electronics.

2.6.8 HERMES Desirable characteristics

Several other characteristics are desirable for the implemented HERMES node. These capabilities are detailed below.

- LED indicators on the top PCB board are desired to provide a visual indicator of the node state. The LED indicators are externally visible, providing illumination to the deployed position of the node. Minor auxiliary functions are provided by these LED indicators, such as providing path lighting for rescuers or survivors.

- The HERMES wireless node requires a means of detecting its own state as it forms a part of the HADES system. This involves detecting when the HERMES node is deployed and automatically applying functions such as routing, indication and node initialization whilst it is in the mine. As HADES cannot actuate the node after it is deployed, this level of automation is desired.

2.7 Summary

This chapter explored the background theory related to the problem of mine communications. Topics of existing solutions and modern wireless technologies were discussed, including an exploration of existing mine communications devices. Prior work by Monlyneaux and Heesterman was discussed in detail. A wireless system communications framework and associated protocols was examined in conjunction with supporting technologies such as video compression and antenna hardware. Finally, this section laid out specifications for the proposed HERMES wireless node. The results of this literature review revealed several key technologies that are potentially suitable for implementation as the RF core of the HERMES wireless node in the 900 MHz to 2.4 GHz band. The examination of common wireless standards and radio systems raised the possibility of using an off-the-shelf wireless module to achieve radio communications. Chapter 3 discusses the search for such an off-the-shelf wireless module to function as the RF radio for the HERMES node.

Chapter 3 Initial Design, Prototyping and Experimentation

This chapter describes the testing and evaluation of various RF modules that suit both the RF and application-layer requirements as described in section 2.6.1 and 2.6.2. Section 2.6.1 described the requirement of an RF radio operating between 2.4 GHz and 900 MHz. Off-the-shelf components are selected for devices in this RF operating range and are narrowed down to three modules as detailed in sections 3.2 through 3.4. Two 2.4 GHz solutions and one 900 MHz solution are examined in detail. A comparative evaluation of these devices is outlined in section 3.5. Basic testing on a viable candidate RF module is described in sections 3.4 through 3.9. The selection of the final WizFi220 RF device is described in section 3.5 with an initial node design. A prototype printed circuit board is designed in section 3.6 for initial evaluation. The work of Thompson [35] and Jones [36] improves the node design described in section 3.7 and a new enclosure is designed to improve the node shape characteristics. A list of changes to the design of the HERMES node is described in sections 3.8 and 3.9. A major area of change exists in the MCU unit implemented alongside the WizFi220 and a new MCU unit is specified in section 3.9. This experimentation informs the design of the final HERMES electronics discussed in Chapter 4.

3.1 Initial Component Exploration

Several off-the-shelf RF modules are examined for use as the core radio device in HERMES. The outcomes of the search are detailed in table 3-1. The commonality of the technologies in the 2.4 GHz and 900 MHz spectrum opens the possibility that an existing wireless module already implements (or partially implements) features desired in section 2.6.

Table 3-1: Evaluation of wireless devices

Device	Manufacturer	Operating RF frequency	Maximum RF operating data-rate	Advertised Range / Transmit Power	Power draw at max Tx	Device Size	Price (NZD)
ERIC9	LRPS	915 MHz	500 kbps	7 dBm	33 mA	15 x 20 mm	\$22.14
ZETA915	RF Solutions	915 MHz	500 kbps	2 km / 13 dBm	18 mA	16 x 16.5 mm	\$18.70
Xbee SX	Digi	915 MHz	250 kbps	100 km / 30 dBm	900 mA	33.8 x 22.1 mm	\$156.42
WizFi630A	WizNet	2.4 GHz	72 Mbps	X	X	43 x 33 mm	\$43.59
WizFi220	WizNet	2.4 GHz	11 Mbps	600 m / 17 dBm	250 mA	32 x 23.3 mm	\$58.84
WF121	Silicon Labs	2.4 GHz	54 Mbps	X / 16 dBm	400 mA	26.2 x 15.4 mm	\$45.36
WGM110	Silicon Labs	2.4 GHz	72 Mbps	450 m / 16 dBm	261 mA	21 x 14.1 mm	\$21.57
WLNN-ER-DP551	B+B SmartWorx	2.4 GHz	72 Mbps	X / 17 dBm	370 mA	40.6 x 29.6 mm	\$140.62
RN171	Microchip Technology	2.4 GHz	54 Mbps	X / 12 dBm	190 mA	26.6 x 17.78 mm	\$49.77

Table 3-1 shows the results of an initial component search. This table focusses on devices that are readily available and can be obtained through reputable suppliers. The devices shown in table 3-1 describe a series of 900 MHz and 2.4 GHz radio modules. Devices operating between these two frequencies are explored but not included. An example of a device that meets this criterion is an LTE module implemented by Digi, operating in the cellular bands of 1.8 GHz [37]. Due to the need for regulatory approval from licence holders (and payment of associated fees) to operate devices in this band, development on these devices is difficult and was hence abandoned.

The examination of 900 MHz RF devices shows a wide variety of roles that such radios can fulfil. Devices such as the XBee SX is shown to have excellent range performance and very high transmit power. However, it is observed that such systems are designed for outdoor open-air communication, and the device is suited for this purpose. The high-power of this device results in an equally undesirable high transmit current draw. Despite the long range, it is shown that this device does not achieve high throughput, being capped at 250 kbps. The high power draw and comparatively high cost of this device means that this particular device is unsuitable. The remaining 900 MHz devices shown in table 3-1 show a significantly reduced cost. Associated with this is a smaller device package and an improved RF operating data-rate. The reduced power draw correlates to a reduced transmit current. Both devices focussed on low cost, low power data transmission. The ZETA915 device was initially selected as the superior device. However, the ERIC9 is selected instead due to the presence of an on-board MCU, eliminating the need for other components. Because of this added feature, the ERIC9 can then perform sensing functions as specified in 2.6.5 whilst minimizing external component count. The ERIC9 package is also smaller than the ZETA915 package, thus reducing the size of the node electronics.

Multiple modules implement a 2.4 GHz RF radio operating an 802.11b/g/n protocol as shown in table 3-1. Most modules have a cost of roughly \$50 NZD except for one module costing three times as much, the DP551. The WizFi220 is chosen as it features the highest transmit power with the lowest power draw for a device offering 15 dBm+ transmit power. Despite being the second most expensive device in its class, the WizFi220 advertises an open-air range of 600 m. The second unit chosen is the WGM110 as this device is numerically superior in cost and size, with a transmit range of 450 m as advertised by the manufacturer. This device also offers the smallest package size, as well as an integrated MCU unit that allows the WGM110 to be operated standalone with a custom script language. These two devices are selected due to their differing maximum specifications. The WGM110 features an RF rate as specified by 802.11g of a maximum of 72 Mbps. The WizFi220 only features a data-rate of 11 Mbps. Since the maximum data-rate is an indicator of performance under optimal conditions, it is unclear whether these devices can attain the application data-rate specification at the target range in mine conditions, particularly with ground proximity and possible loss of line of sight. Further investigation is required and these devices are chosen for experimentation. The WGM110 is chosen for its optimal data-rate, and the WizFi220 is chosen for its advertised long range.

3.2 ERIC9 by LPRS

The LPRS ERIC9 radio system is a low power, low cost radio module. The main module is pictured in figure 3-1. This figure depicts the ERIC9 evaluation platform that was used to test this device. The ERIC9 operates on the unlicensed 915 MHz to 928 MHz band, with an antenna mount for a U.FL connector for external antenna connections. To save chip space, minimizing the size requirement of the electronics as stated in section 2.6.7, the ERIC9 implements an on-board 4 MHz MCU. This eliminates the need for other control circuitry. The ERIC9 therefore also features standard microcontroller features such as timers, on-board ADC channels and interrupt functionality. The ADC channels allow the external environment to be sensed using analogue circuitry, a requirement specified in section 2.6.5. The ERIC9 device draws a current of 33 mA transmitting at maximum power. With an operating voltage of 2.4 to 6 V, the power drawn over 8 hours of operating time is 871 mWh. This operating current and voltage is easily attained with most batteries discussed in section 2.3. The ERIC9 features an internal regulator that supplies 3.3 V which further reduces the need for external circuitry and allows a variable voltage input.

Figure 3-1: The ERIC9 Development board

3.2.1 900 MHz link budget analysis

Friis' equation is used to estimate the signal loss due to distance in an open-environment. This is used as a tentative approximation of the range that can be viably attained by an RF device. The results of the literature review in chapter 2 shows that tunnel attenuation varies depending on range, type of tunnel and operating frequency, but can be estimated as a linear function of distance. Other factors such as the Fresnel zone obstruction is also applied in this range estimate. This link budget analysis determines the received signal strength given the transmitter and receiver characteristics and the antenna gain with reference to the operating distance and bandwidth. The link margin is evaluated with respect to the minimum signal strength. This analysis assumes no obstacles impeding the line of sight of the devices and no particular tunnel shape is assumed.

The maximum transmit power of the ERIC9 is 7 dBm [13] with a receive sensitivity of -81 dBm at 500 kbps. This gives the ERIC9 an 88 dBm link margin. The effective received signal power of the ERIC9 using the Friis' equation [39] in free-space is evaluated in equation 3.1. This equation evaluates the loss of signal strength due to the distance between a transmitter/receiver pair. For this analysis, standard 900 MHz 2 dBi antennas are used for the transmitter and receiver gain.

$$P_{rx} (dB) = P_{tx} + G_{tx} + G_{rx} + 20 \log_{10} \left(\frac{\lambda}{4\pi D_r} \right)$$

$$P_{tx}: \text{Transmitter power (Amplifier output power)} = 7 \text{ dBm}$$

$$G_{tx}: \text{Transmitter gain} = 2 \text{ dBi}$$

$$P_{rx}: \text{Recieved power at RF decoder}$$

$$G_{rx}: \text{Reciever gain} = 2 \text{ dBi}$$

$$\lambda: \text{Wavelength of } 0.9 \text{ GHz} = 333 \text{ mm} = 0.3 \text{ m}$$

$$D_r: \text{Distance between reciever and transmitter} = 100 \text{ m}$$

$$P_{rx} (dB) = -60.53 \text{ dBm} \quad (3.1)$$

When applying the range penalty of 900 MHz in tunnels, this results in a total received power of -75 dBm as shown in equation 3.2 which falls above the receive sensitivity of -81 dBm. Section 2.2.4 states the tunnel loss of the direct-path 900 MHz signal to be -14 dBm at 100 m.

$$P_{rx} (dB) = -60.53 - 14 = -74.53 \text{ dBm} \geq -81 \text{ dBm} \quad (3.2)$$

A node deployed onto the ground of the mine can experience interference from the ground itself. Since the ground forms an obstacle, some signal can be lost due to ground proximity. The Fresnel zone forms a conical area between two antennas that must be kept clear in order to form line of sight communication. Partial obstruction of field of view can still cause signal loss. The Fresnel zone is evaluated for two 900 MHz radios at 100 m. Common 2 dBi antennas for 900 MHz systems measures 11 cm long. The centre-point of the antenna is therefore 5.5 cm off the ground, and an estimation of loss due to obstruction with the ground is made with the Fresnel zone calculation as shown in equation 3.3 [39]. The distance between antennas is given by $d = 100 \text{ m}$ and $f_c = 900 \text{ MHz}$, the Fresnel radius at 50 m (equidistant from both nodes) results in a zone that is 2.88 m wide at its largest. The ground clearly intrudes with the Fresnel zone and losses in the signal strength are expected. The work of [39] shows that losses due to ground interference is approximately -6 dBm.

$$\text{Radius (m)} = 17.31 \sqrt{\frac{100 \text{ m}}{4 \times 0.9 \text{ GHz}}} = 2.88 \text{ m} \quad (3.3)$$

$$P_{rx} (dB) = -60.53 - 14 - 6 = -80.53 \text{ dBm} \geq -81 \text{ dBm} \quad (3.4)$$

The received power will therefore be reduced to around -80 dBm as shown in equation 3.4. This shows that the received signal strength of -80.53 dBm is only marginally stronger than the receiver

sensitivity of -81 dBm. It is likely that 500 kbps at this range will be unachievable with this device. A range experiment is conducted in section 3.2.2 that examines the receiver strength at range.

3.2.2 Evaluation testing and results

The ERIC9 module is tested by placing two modules at either ends of a hallway (see testing environment section 7.1.2) with both modules placed on the ground within line-of sight. The ERIC9 is observed to maintain a signal up to 150 m. This experiment was conducted in the hallway with light foot-traffic. This resulted in occasional line of sight obstruction. While the ERIC9 was able to receive a signal up to 150 m, the ERIC9 was unable to maintain a data-rate of 500 kbps as specified in 2.6.1. Several software issues caused the ERIC9 to crash with instability issues on the MCU when data is returned. The interface speed of 115.2 kbaud significantly bottlenecks the data speed on the endpoints, rendering the 500 kbps RF speed unachievable. The de-facto effective data-rate is therefore 115.2 kbaud. Networking features such as routing and node addressing functions are not implemented in the ERIC9. Instead, these functions are intended to be handled inside the MCU software. This is because the ERIC9 is intended to be a simple radio communicator. The extension of the ERIC9 software to implement these features is required. It is concluded that the low application level data-rate of this device renders the ERIC9 unsuitable for the HERMES node as it cannot satisfy the data-rate specification in section 2.6.1.

3.3 WGM110E by Silicon Labs

A 2.4 GHz module is evaluated to determine if improvements can be attained over the ERIC9 module that satisfies the data-rate specification in section 2.6.1. The WGM110 2.4 GHz 802.11b/g/n module by Silicon Labs is evaluated and shown in figure 3-2. This features one of two 2.4 GHz modules as discussed previously in section 3.1 with a maximum data-rate of 72 Mbps under optimal conditions. This module advertises a transmission range of up to 450 metres under ideal conditions [41]. The WGM110 features an internal 48 MHz ARM Cortex M3 MCU operating on a custom script language named BGScript. Like the ERIC9, 10-bit ADCs are available on the WGM110 to satisfy external measurement specifications detailed in section 2.6.5.

Figure 3-2: The WGM110 with the SiLabs Development Board

A vital flaw is found during the experimentation of the WGM110 device. Due to its design as an endpoint device with some capability as a wireless access point, the WGM was unable to perform distributed networking functions required by nodes for the HADES communication nodes as specified in section 2.6.5. A vital networking feature of the HERMES node is to allow nodes to join the network without prior coordination. The WGM110 has a client mode that allows connection to access points, and an access point mode, which allows nodes to join its network. However, the client mode does not allow other nodes to join. The WGM110 is a device designed explicitly for infrastructure mode networks. To circumvent this, a dual solution is envisaged as shown in section 3.3.1.

3.3.1 WGM110 dual solution

An alternate solution using dual-WGM110 devices is conceived. Heesterman's [2] work theorizes the use of dual radios on different channels to achieve higher throughput, preventing interference between channels and improving overall throughput. Each node features two WGM110 devices with one device acting as an access point and the second device acting as a client. This chain allows each node to connect to each other node though the arrangement shown in figure 3-3. This system sets up a new network for each WGM110 module and each node connects to the network preceding it. This forms a data-chain that allows data to pass between the devices. Each WGM110 device communicates to the other using high-speed UART. However, this solution requires double the area and power consumption with two antennas in close proximity and this contributes to RF swamping on the 2.4 GHz band.

Figure 3-3: Dual Node WGM110 network arrangement

The channels handled by each node are allocated on a dynamic basis to minimize RF interference. The channel allocation method aims to prevent adjacent nodes from operating on the same or overlapping channels. This must be avoided as this can cause interference. As each node bridges two networks, the 2.4 GHz spectrum is divided up to allow separation between channels of adjacent nodes. There are eleven 22 MHz channels in the 2.4 GHz ISM band defined by the 802.11b protocol. Each node builds its own network five channels away from the last node to prevent interference. The base station operates on channel 1, and hence the next node will operate on channel 6. When the last six channels are encountered, the channel wraps around to the first channels. Figure 3-4 shows that as the eleven 2.4 GHz channels do not evenly divide into five, this process will ensure that all channels are used with each node having at least a five channel spacing between adjacent nodes.

$$1 \rightarrow 6 \rightarrow 11 \rightarrow 2 \rightarrow 7 \rightarrow 3 \rightarrow 8 \rightarrow 4 \rightarrow 9 \rightarrow 5 \rightarrow 10 \rightarrow 1$$

Figure 3-4: Dual node channel assignment

The mechanism described in figure 3-4 for deployment of a dual-module system utilizes a modular channel assignment that prevents channel contention and maximizes throughput by transmitting data on non-overlapping channels.

Several problems are observed with this dual-module solution. The WGM110 is observed to draw 250 mA per module, and therefore this solution draws 500 mA at worst case. This device power draw is deemed to be unacceptable. Furthermore, the requirement for two modules also requires two antennas. Since the antenna version is used, this requires the placement and stacking of two 2.4 GHz antennas. This vastly increases cost and complexity of the designed node system. A significant lead-time on the WGM110E (with an external antenna mount) results in uncertainty with regards to component acquisition.

3.3.2 Summary

The WGM110 features both a chip antenna variant and an external antenna variant. The WGM110 model desired is the external antenna variant. Due to the excessive lead time on acquisition of this model, the desired variant would be too difficult to attain. A proposed dual solution requires both double the power requirement and double the component cost. A revised WGM110 solution is deemed too expensive and complex to be considered viable for a low-cost node. The WGM110 is hence not fit for purpose and abandoned.

3.4 WizFi220 by WizNet (w/ ATmega2560 MCU interface)

The WizNet WizFi220 is a 2.4 GHz module that features standalone network connectivity [41]. This solution is chosen due to its advertised long range and high output power with comparatively lower current draw of all devices as discussed prior in section 3.1. The WizFi220 features an ad-hoc mode. This feature is necessary in allowing the WizFi220 to allow units to freely form and connect to a network. This ad-hoc feature is discussed further in section 5.2.2. The application note states that the best range of the WizFi220 device was attained using a 2.5 dBi dipole antenna, reaching a distance of 620 m with the ping utility [42].

Figure 3-5: WizFi220 and individual module

Figure 3-5 shows the WizFi220 module as implemented on an evaluation board and the module itself. This evaluation board features the standard interfaces to the WizFi220. To interface to this evaluation board for testing, the WizFi220 is connected to an external MCU unit as shown in figure 3-6. An ATmega2560 is used for initial testing of the WizFi220 unit. The ATmega2560 is a 5 V, 8-bit 16 MHz microcontroller with 100 I/O pins. This design is chosen due to its multiple serial UART I/O pins which allows the ATmega2560 to easily interface to the WizFi220.

Figure 3-6: WizFi220 Testing platform

Figure 3-6 shows the evaluation board with the Arduino Mega MCU development device on a constructed testing platform. The WizFi220 operates on a 3.3 V power supply whilst the Arduino Mega operates on a 5 V power supply. This necessitates the use of level shifters when connecting the two devices together as this mismatch in high-level voltage can cause damage to the WizFi220. Connections are made for the UART and SPI lines for experimentation of the different interfaces to maximize data transfer speed. The level shifters are built into Veroboard as shown on the two boards in figure 3-6 with the circuit shown in figure 3-7. Power is provided to these boards through two separate USB connections on the Arduino Mega and the WizFi220 Evaluation board. This MCU provides debug lines with a serial connection to the PC to allow a user to manually send commands to the WizFi220.

Figure 3-7: Level shifter schematic

3.4.1 2.4 GHz link budget analysis

The WizFi220 has an average transmit power maximum of +17 dBm with a receive sensitivity of -94 dBm at 1 MBps, the lowest RF data-rate setting for 802.11b. The Friis' equation estimate as shown in equation 3.5 shows that at 100 m, the received power between nodes should be

roughly -59 dBm. A 2 dBi antenna is used on this testing platform and this adds to both the transmitter and receiver gain.

$$P_{rx} (dB) = P_{tx} + G_{tx} + G_{rx} + 20 \log_{10} \left(\frac{\lambda}{4\pi D_r} \right)$$

$$P_{tx}: \text{Transmitter power (Amplifier output power)} = 17 \text{ dBm}$$

$$G_{tx}: \text{Transmitter gain} = 2 \text{ dBi}$$

$$P_{rx}: \text{Recieved power at RF decoder}$$

$$G_{rx}: \text{Reciever gain} = 2 \text{ dBi}$$

$$\lambda: \text{Wavelength of 2.4 GHz} = 122.45 \text{ mm}$$

$$D_r: \text{Distance between reciever and transmitter} = 100 \text{ m}$$

$$P_{rx} (dB) = -59.53 \text{ dBm} \quad (3.5)$$

The equation for the Fresnel zone is shown in equation 3.6. The distance between antennas is given by $d = 100 \text{ m}$ and $fc = 2.4 \text{ GHz}$, the Fresnel radius at 50 m (equidistant from both nodes) results in a zone that is 1.8 m wide. As before, this clearly interferes with the ground and thus causes additional loss of signal, and a -6 dBm loss is added to the margin. The received power drops to -65.53 dBm.

$$\text{Radius (m)} = 17.31 \sqrt{\frac{100 \text{ m}}{4 \times 2.4 \text{ GHz}}} = 1.8 \text{ m} \quad (3.6)$$

A -65.53 dBm receiver sensitivity results in a link margin of -25 dBm before loss of receptivity. Over 100 m, the signal fading due to the tunnel must be limited to -0.25 dBm per metre of tunnel. Testing is still required to determine exact performance, but literature estimates place signal loss at around -20 dBm at this distance, leaving a link margin of roughly 8.5 dBm as shown in equation 3.7.

$$Rx_{rec} = -65.83 - 20 \text{ dBm} = -85.53 \text{ dBm} > -94 \text{ dBm}$$

$$\text{Link Margin} = -94 \text{ dBm} + 85.53 \text{ dBm} = -8.47 \text{ dBm} \quad (3.7)$$

The link margin at 100 m is 8.5 dBm above minimum signal strength with a received signal strength of -85.53 dBm. The receiver sensitivity of the WizFi220 is -94 dBm. A link margin of 8.5 dBm above the receiver sensitivity shows that the transmission of 802.11b at 1 Mbps is theoretically viable at this range.

3.5 Comparative Evaluation

The ERIC9 device is evaluated and deemed unsuitable due to the bottlenecking of the target data-rate at the UART interface. The ERIC9 is shown to have a transmission range of up to 150 m but the target application level data-rate was unattainable. The MCU unit operating at 4 MHz is also a significant impeding factor to the data-rate performance of the ERIC9. Finally, routing functions are required to allow data to be intelligently passed between nodes. However, the ERIC9 module did not implement the mechanisms that would allow this to be possible. It is expected that such functions would be implemented inside the MCU. This demonstrates the weakness of using proprietary radio modules; The ERIC9 does not support networked solutions past providing the physical radio functions. Addressing, header and data delimiting are all handled by the MCU unit, and custom software is required.

The WGM110 unit implements an 802.11b backbone with a radio advertising a range of 450 m. This solution improves on the 900 MHz option as it provides a high-rate data-input interface (up to 6 Mbps) with the implementation of a standard OSI internet protocol stack to provide routing and dataflow control. However, routing functions were not available as the WGM110 is designed to be an endpoint unit or wireless access point. A dual-module solution is examined but is deemed too expensive and uses too much power. Difficulties in sourcing the correct variant of the device resulted in a long lead time. The WGM110 unsuitable for implementation for the HERMES node.

Several RF modules are evaluated with the WizFi220 device selected for final implementation. The manufacturer states that the range of the WizFi220 unit is up to 620 metres. An ad-hoc mode is supported to allow nodes to self-configure the network and this allows the WizFi220 to perform ad-hoc advertisements which allows new nodes to join when needed, a key requirement for the HERMES node. The WizFi220 has a maximum interface rate of 921.6 kbaud which exceeds the data rate specification as stated in section 2.6.1. An ATmega2560 device is initially selected to interface with the WizFi220 to provide initialization commands and provide network management functions. An initial node design is constructed with the WizFi220 with this MCU as the core electronics. This prototype is discussed in section 3.6.

3.6 WizFi220 Prototype

A prototype is constructed for the WizFi220 system. This board implements the features that are present in the evaluation board setup as shown in figure 3-8. The PCB board measures 9 x 7.5 cm. The wireless node measures roughly 11 x 15 mm.

Figure 3-8: WizFi220 prototype board with node enclosure

A USB-micro B connector is implemented on the bottom edge of the chip to allow the ATmega2560 interface to the PC through an FTDI chip. This allows the MCU to interface to the PC as a serial device. The attached FTDI chip is capable of 1 Mbaud communication with the MCU. This USB connector is connected to serial 0 of the ATmega2560, which is the position of the USB serial on the Arduino Mega MCU board. This ensures that the conditions for the Arduino libraries to function are met.

The internal construction of the node enclosure provides the battery space underneath the PCB board, and hence no components were placed on the bottom layer of the PCB. The battery connectors connect to the top of the board through gaps between the PCB and the node enclosure wall. The antenna is designed to be connected in one of four orientations on the bottom edge of the node enclosure as marked in figure 3-8. The battery space is provided underneath the electronics with the PCB riding on tapped steps.

Figure 3-9: Prototype PCB design

Figure 3-9 shows the PCB design for the WizFi220 prototype. The PCB prototype features twin buck regulators as shown on the bottom-left hand edge of the board. This provides the 3.3 V and 5 V power rails for the on-board components. A ground-plane separates the power regulators from the RF module implemented on the top-left hand corner of the board, minimizing signal and ground noise between the regulators and the RF unit.

The prototype board implements several features that are designed for prototype testing. Figure 3-9 shows the headers that separate circuit board elements. This allows individual elements to be tested before allowing them to interact with other board components. Faults are isolated to particular components with no risk of causing faults to other components. Notable board elements that are isolated in this way are the power rails, which are tested independently. The regulators are isolated from the main 3.3 V and 5 V rail through the P5 and P6 headers. The MCU features a 6 pin SPI header (P15 figure 3-9) that allows the MCU to be programmed. Similarly, the UART and SPI connections are separated via U3, a level-shifting chip. This level-shifting chip performs the same functions as the level shifter circuit discussed in section 3.4. However, this solution eliminates the need for discrete components. Headers P4 and P3 separate the Level shifting chip from either the RF and MCU unit, and this allows the signals from the WizFi220 and MCU unit to be debugged. Power and ground points are provided through the ground headers on P8, and the power headers on P9 and P10, allow the board to power external circuitry for testing.

3.6.1 Experimentation method

An experiment is conducted to test the performance of the ATmega2560 processing speed when functioning as a packet processing device. This experiment aims to evaluate the prototype system's range and data-rate performance in a tunnel-like environment. The viability of data transfer under 2.4 GHz is also explored in this experiment. Two prototype nodes are placed a variable distance apart from 20 m to 120 m at 20 m intervals along a large open corridor (see section 7.1.2). 500 packets are processed by the receiver node. The transmit rate is limited to 64 kbps (discussed in section 3.8) and hence this transmit rate is set to be the data-rate ceiling.

3.7 Improvements to Node Design

Improvements to the node design are conducted by Thompson and Jones. The initial prototype node design presented in section 3.6 shows a node design that is 15 cm tall and 11 cm wide. Despite the size of the device conforming to specifications as stated in 2.2.2, a node of this size still presents problems to the function of the HADES robot. The node case size of 15 cm is roughly the thickness of the HADES chassis and hence improvements are desired. A consequence of the size of the wireless node is the stacking volume and weight of nodes, as well as the size of the deployment device. A large node size also requires a larger deployment mechanism size to store the nodes. The positioning of the deployment mechanism on the sides of the chassis presents issues to the width of the HADES robot. The increased size of the deployment mechanism significantly increases the risk of bottoming out. This is because the space on the side of the chassis no longer provides clearance for obstacles. This impedes the agility of the HADES robot.

Figure 3-10: Prototype node enclosures. Left: The first node enclosure. Right: A second node enclosure prototype.

A second prototype shown in the right on figure 3-10 is designed that features slightly smaller dimensions. This is facilitated by thinner side walls and a smaller bottom edge. The PCB is reduced in size to match the internals of this new design. This is shown by the orange acrylic insert on figure 3-10 right. This node measures 11 x 14 cm with an identical width which is an improvement on the volume and mass of the node whilst still maintaining the battery power and functionality.

A further improvement is made with the final node design that allows the width of the node to be reduced to 7 cm wide, a significant improvement on the previous prototypes. This further reduction in node size is demonstrated in figure 3-11. By segregating the batteries and the node electronics, the need for cables to come up from beneath the PCB is eliminated. The node design features a PCB compartment and a node battery compartment. The compactness is achieved by allowing the antennas to be threaded through the PCB area. This shortens the distance between the antenna and the RF input on the WizFi220 chip. The electronics of this node design are designed to be placed on the top and bottom of the PCB compartment, allowing a stacked PCB design linked by headers.

Figure 3-11: Third iteration of the HERMES node enclosure

The prototype implementation of the WizFi220 node system as discussed in section 3.6 consumes a full 9 x 7.5 cm of area. A smaller surface area is desired to allow the node to reach the smaller dimensions of a new node enclosure designed by Thompson and Jones shown in figure 3-11. This node design significantly reduces the size of the node with a requisite size reduction of the internal compartment. This reduction in size also translates to a reduction in weight, and a new node electronics design is desired.

Several improvements as discussed in section 3.8 can be used to achieve a smaller PCB design. A significantly smaller design can be achieved using a microcontroller operating on 3.3 V. Elimination of the logic level shifter circuitry and the redundant regulator reduces the size of the design significantly. In addition to being too slow and operating on a non-3.3 V voltage rail, the ATmega2560 is also large as it features 100 I/O pins. Since the I/O utilization is low, it is desired to change the MCU to a smaller unit overall. Other improvements include the search for smaller power inductors and capacitors.

3.8 Prototype Results Evaluation

The prototype PCB designed in section 3.6 reveals several key flaws that renders the prototype electronics not suitable for final HERMES implementation. Major changes in the node shape require a

complete redesign of the PCB. Several components chosen for prototyping proved to be unfit for purposes for final implementation. Figure 3-8 shows especially large inductors and capacitors. The capacitors are roughly 1 cm tall and exceeds the height dimension of the interior. Smaller power components are desired in general. The use of both a 5 V and a 3.3 V rail for powering the on-board components is both redundant and contributes to a significant amount of wasted PCB space due to the necessary inclusion of logic level shifters. It is desirable to eliminate the logic level shifters and the 5 V regulator by changing the MCU to a unit that operates at 3.3 V to further reduce the area consumed by the node electronics.

Figure 3-12 shows that the prototype system attains a peak data-rate performance of around 67.7 kbit/s up to 100 m. The ATmega2560 unit demonstrates a processing time of around 50 ms per 400 bytes. The delay variance is observed to increase as the node separation distance increases. This result shows that a stable data-rate is more difficult to maintain at longer ranges. The standard deviation of the delay is observed to be roughly 2 ms per hop. This processing time extends to a best case of 2 seconds over 40 hops excluding endpoint processing time. This delay vastly exceeds the specified 1 second round trip time (RTT) as described in section 2.6.2. This delay is due to the processing lag occurring aboard the ATmega2560 unit. The WizFi220 unit is shown to be a suitable RF unit with good performance up to around 100 m and hence the range specification in 2.6.1 is met. The WizFi220 unit is capable of a serial speed of 921.6 kbps link speed. However, this UART speed is unattainable with the current microprocessor unit.

Figure 3-12: ATmega2560 prototype data-rate performance

The low clock speed of the implemented MCU limits the attainable UART speeds. It is desired to operate the UART at the maximum speed of 921.6 kbaud. Due to the limits of the AVR 16 MHz clock divider, the closest UART clock rate attained by the MCU to match this is 1 MHz. The UART

protocol can resolve a clock rate mismatch up to 3.75% [43]. However, the UART error at this mismatch is 8%, which vastly exceeds the clock rate tolerance. This effectively limits the serial link speed to 115.2 kbaud. An MCU that can match a 921.6 kbaud is desired.

3.8.1 Initial conclusions

The WizFi220 performance in the hallway test as shown in figure 3-12 shows good receptivity up to 100 m at full data-rate. The data rate transfer is limited by the slow 16 MHz 8-bit MCU. The maximum data-rate is observed to be 67.7 kbps due to UART limits on the host MCU system. This slow MCU system also results in a delay of 50 ms per hop. Figure 3-12 shows the data-rate performance of the two-node system as a function of separation distance. This figure shows that the variance of the data-rate increases as the separation distance increases. Note that the error bars at 120 m are not marked, as the data-rate was variable between 0 kbps and 50 kbps. A stable data-rate is hence observed to be lost past 100 m.

3.9 WizFi220 MCU Requirements

The WizFi220 requires a microcontroller to operate as the RF module on the HERMES node. Section 3.8 shows that the WizFi220 module has a transmit rate of up to 100 m. The MCU unit was initially chosen to provide basic setup and monitoring functions to the WizFi220. However, it is shown that the MCU needs to also provide data streaming and decision-making actions. The MCU is both too slow and requires a significant amount of support circuitry to effectively provide the intelligence functions to the HERMES node. The testing shown in section 3.8 shows that the MCU forms part of the processing chain and the limited clock speed slows down the processing of packets. The results showed that the ATmega2560 MCU processes a packet with an average time of 50 ms and a standard deviation of 2 milliseconds. As discussed in section 3.8, this delay culminates to an unacceptable delay length after 40 hops. A faster processor is desired to minimize delay and improve on other factors.

The WizFi220 is shown to have the requisite range and routing functions suitable for implementation in the HERMES node as shown in section 3.7 with a 921.6 kbaud maximum transmission speed on the serial input line and a 1 Mbps minimum RF rate. The range attained is shown in figure 3-12 and satisfies the range criteria discussed in section 2.6.1. A new MCU unit is desired that satisfies the following requirements:

1. The operating I/O voltage shall be 3.3 V. This eliminates the need for logic level shifters and an additional 5 V regulator. An MCU operating on 3.3 V then allows the entire board to be powered on a single power supply. This saves PCB space which is vital in miniaturizing the design for the new node enclosure discussed in section 3.7.

2. The operating MCU needs to be able to maintain a serial link of 921.6 kbaud, the maximum baud rate that is attainable by the WizFi220. This will allow the WizFi220 to transmit with a data-rate that satisfies the specification in section 2.6.2.
3. The MCU unit is currently large and features 100 I/O pins. An MCU with a smaller form factor and fewer pins is desired as most pins on the ATmega2560 are redundant. This also saves space on the new smaller PCB.

Requirements 2 and 3 are achieved with an MCU that has a higher clock speed that allows data to be processed at a faster rate. A new MCU is chosen and implemented in the design of the hardware as discussed in the next chapter.

3.10 Summary

This section discussed the selection of a module to function as the core radio for the HERMES wireless node. Initial experimentation showed that the WizFi220 was capable of a range of 80 to 100 m. A prototype was designed using an ATmega2560 microcontroller unit. Several issues were found with the MCU unit and a faster MCU unit is desired. A node design evolution was discussed that resulted in a reduction in form factor of the physical node enclosure. Despite the shortcomings of the designed prototype, the WizFi220 was selected as it is shown to have both long range and an ad-hoc mode. However, changes in physical node design and MCU results in the need for a revised wireless node electronics design. Chapter 4 discusses this new HERMES electronics design. This design features a new MCU unit and PCB design to fit inside the new node enclosure. Deployment control and sensing circuitry is also added to this hardware design for implementation functionality. Other minor improvements motivated by the results in this chapter are also implemented.

Chapter 4 HERMES Electronics Design

4.1 Overview

This chapter discusses the hardware of the HERMES node, including the PCB board, circuit design, and microcontroller selection. The HERMES node hardware features a stacked PCB design. The top PCB board is discussed in section 4.2. The bottom PCB board is discussed in section 4.3. Section 4.4 discusses two battery power solutions to meet the system power draw characteristics examined in section 4.3.5. Section 4.5 examines the node enclosure designed by Jones and Thompson in detail. The HERMES PCB design, battery placement and enclosure design is shown in figure 4-1. The HERMES hardware performance is summarized in section 4.6.

Figure 4-1: HERMES electronics design

The prototype pictured in figure 4-1 shows an enclosure designed and printed using 3D printer ABS plastic. The final manufactured enclosure will be milled from solid nylon plastic. The lid is manufactured from laser-cut acrylic and bolted onto the enclosure with set screws. This figure shows an approximate fit of the electronics components, the battery system, and the antenna placement.

The HERMES electronics design implements improvements identified from the prototype design as discussed in section 3.9. A change in the external physical node enclosure discussed in section 3.7 necessitates a change in the node electronics and design of the PCB. A smaller node footprint with double-stacked PCBs is designed to fulfil the smaller space requirement of the new enclosure. This smaller HERMES node features the reduction of the node size width, but maintains the height and depth of the node. The node is segregated into separate compartments, with the batteries stored separately from the node electronics shown in figure 4-1.

HERMES node Overview

Figure 4-2: HERMES hardware block diagram

Figure 4-2 shows a block diagram of the functionality present in the HERMES electronics design. The WizFi220 unit is implemented on the top board as discussed in section 4.2.1. Section 3.9 discusses the shortfalls of a previous prototype board. Several limitations are identified such as the inclusion of a redundant 5 V regulator and the poor performance of the initially selected ATmega2560 microcontroller. A new MCU is selected and improvements are discussed in section 4.3.1. Minor changes in the circuit include the selection of smaller power components for power supply filtering and this is discussed with the power regulator in section 4.3.2. An accelerometer is chosen to detect the successful deployment of the module and this is discussed in section 4.3.4. Additional improvements over the prototype examined in section 3.6 include the addition of peripheral node diagnostic components. These components allow the node to fulfil monitoring functions specified in section 2.6.5. Battery power and temperature detection peripherals are discussed in section 4.3.3.

4.2 Top PCB design

The top PCB features the WizFi220 for the primary radio functionality and provides the external indicator LED position. The top area is the only position where the LEDs are externally visible as the lid is transparent. The power LEDs are placed in the centre of the top PCB, away from the top edge and avoiding the high speed signal pins on the edge of the WizFi220 on the underside of the board to minimize signal noise. Both LED gate transistors are connected to the Teensy digital output pins on the bottom board. The LED transistors and land patterns are visible in figure 4-3. This figure also shows the positioning of the WizFi220 module on the underside of the board.

Figure 4-3: Altium mockup of HERMES top PCB

An 8 x 2 header (P1 figure 4-3) is used to connect all top board signals to the bottom PCB board, consisting of both the UART (4-pin with CTS/RTS) and SPI (5-pin) interfaces, as well as the feedback from the WizFi220 through dedicated interrupt outputs. The header configuration is shown in block diagram figure 4-4, depicting the top to bottom connections.

Figure 4-4: Top PCB Block diagram

4.2.1 WizFi220 RF module

The WizFi220 unit is a 49-pin module with surface mount edge connectors measuring 32 x 23.5 mm. The underside of the top board is dedicated to this RF module. The inverted positioning of the WizFi220 module allows the U.FL connector from the antenna mount to connect from underneath the top board, with the RF signal cable traversing between the two boards. Figure 4-5 shows how the entire bottom board is reserved for the WizFi220 unit allowing a non-obstructed connection of the single U.FL connector to the external antenna.

- The LED status provides a visual indicator when the network connection is gained. This indicator can help the operator determine if a node has failed on deployment. This light indicator can be easily detected by the side-mounted HADES cameras.

One orange LED and one green LED are provided to differentiate between the two indicators. The schematic for these LEDs is shown in figure 4-4. The LED is intended to be a configurable LED indicator, and therefore can be driven with a varying PWM cycle to adjust the brightness of the LED. The power LED recommends a DC forward current of between 10 mA and 70 mA. A 27 Ω resistor is placed in series with the LED to provide the PWM with a large range of brightness settings and limits the output current to 50 mA. The maximum power output by the LED is given by equation 4.1.

$$V_f = 2.0 V \rightarrow 3.3 V = 2.2 A + IR$$

$$I = \frac{1.3 V}{27 \Omega} = 48 mA$$
(4.1)

In practice, to limit the current draw of the LEDs, the LED will be flashed at 1 Hz with 50% duty cycle to maintain high brightness when active. This limits current draw to 25 mA.

4.3 Bottom PCB Design

The bottom PCB features the intelligence and power management components, as well as deployment indicators and additional monitoring features specified in section 2.6.5. The MK20DX256 chip is chosen on-board the Teensy 3.2 as the central MCU as shown in figure 4-8. The Teensy 3.2 is implemented on a separate PCB module and this module is joined to the bottom PCB board with header pins. These header pins are cut down to allow the board to ride as close to the bottom board as possible, minimizing the vertical space consumed.

Figure 4-7: HADES Bottom PCB

Figure 4-8: Full HERMES bottom board construction

The bottom PCB provides connections for the power rails, ground and 3.3 V supply though to the top PCB. Provided are the interfaces for UART and SPI, alongside system change interrupts and LED indicator pins. These signals are routed to the top board through the P2 header visible on figure 4-9 and 4-8. The regulator is placed on the front tab closest to the battery system with the power components placed together to prevent the coupling of the power supply ripple to the high-speed digital signals.

Figure 4-9: Block diagram of the bottom HERMES PCB board

4.3.1 Microcontroller unit: Teensey 3.2 and MK20DX256

The MK20DX256 MCU system implemented on the HERMES resides on the Teensey 3.2 board. A 14 pin 2.54 mm pitch header package is designed in Altium and used for the footprint to connect the Teensey to the board as shown on figure 4-7. Only the main 14 pins on the long side of the board are used as the other pins on the edge and inside of the board are surplus to circuit requirements and are left unconnected. The MCU is placed in the centre of the board to expose all component pins to the

peripherals. Standard through-hole connectors are used to connect the Teensy 3.2 to the main board as opposed to surface mount versions. Due to the vibration-inducing motion of HADES and the impact of deployment, large masses hung via surface mount headers have the danger of disconnecting due to a shock load. The use of a through-hole header allows greater structural strength to minimize the likelihood of pin disconnection during operation. The digital pin assignment is shown in figure 4-10.

The MCU has input power pins with an on-board linear LDO regulator provided by the Teensy 3.2 evaluation board. This regulator is connected to the USB voltage bus by default. The Teensy 3.2 offers an exposed track which is cut on the underside of the board. This bypasses the on-board regulator and allows the on-board MCU to be powered directly from the 3.3 V power supply. Like the WizFi220, this MCU operates on a 3.3 V power supply and hence this removes the need for a second power regulator and all components in this design operate from the 3.3 V power system.

Figure 4-10: HERMES main MCU unit

Section 3.9 discusses the need for an MCU unit that can attain a nominal 921.6 kbaud. This MCU is shown to attain a baud rate that matches with this at roughly 915.75 kbaud. This baud rate is observed to successfully interface the MCU with the WizFi220. The baud rate mismatch is 0.6% and this is an acceptable mismatch for successful UART transmission [43]. An oscilloscope is used to check the UART baud rate of 915.75 kbaud and the result is shown in figure 4-11.

Figure 4-11: Teensy 3.2 UART output at 921.6 kbaud

The power consumption of the MCU varies depending on the application, clock-rate, and number of I/O pins and peripherals used. A maximum power draw estimate from the datasheet of this device is given in table 4-1. These figures show the maximum current ratings that the MCU operates at under normal operation. The system runs on a 3.3 V rail with 15 digital I/O pins used. Two dedicated transmission blocks are used for the UART and SPI split among 9 pins. 2 ADC pins are polled at 5 s increments and the power draw is considered negligible compared to constant the I/O drive current of the digital pins. The peripheral clocks are enabled due to the use of the UART and SPI modules.

Table 4-1: Teensy 3.2 current draw

MCU operation	Current draw of operation
Code executing from flash @125 deg C, Vdd = 3 V, ALL peripheral clocks enabled	Up to 39 mA
Output High current (all pins)	100 mA
Output Low current (all pins)	100 mA

The MK20DX256 chip features an LQFP package of 64 pins. With 15 pins, the expected drive current for the MCU is shown to be approximately 25 mA [44]. This current draw in addition with the maximum draw of the processor under worst case conditions is 39 mA and hence the total operating current of the MCU is shown in equation 4.2.

$$\frac{15}{64} \times 100 \text{ mA} \cong 25 \text{ mA draw on I/O}$$

$$25 \text{ mA} + 39 \text{ mA} = 64 \text{ mA MCU power draw at 72 MHz} \quad (4.2)$$

As the MCU is overclocked at 120 MHz, this increases the current draw of the MCU unit but not the digital I/O pins. Since the over-clocking of the chip raises the operating frequency to 160% of the spec, it is expected to raise the MCU current worst case by a factor of 1.6. The revised power draw is shown in equation 4.3.

$$25 \text{ mA} + (39 \text{ mA} \times 1.6) = 88 \text{ mA} \rightarrow \text{MCU power draw at 120 MHz} \quad (4.3)$$

Figure 4-13: Efficiency of the NCP6306 regulator.

The battery connector is chosen to be a JST-PH connector. The JST-PH connector offers several major advantages, such as the sturdy construction and small form and its latching features. The vibrations from HADES' motion and the shock impact during deployment can cause connectors to disconnect. Since a power connection is vital to the function of the node, the latching property of the connector prevents this failure condition from manifesting.

4.3.3 Node diagnostic peripherals

The node PCB contains two diagnostic circuits as shown in figure 4-10. Resistive temperature and power sensing circuitry is included on the PCB for basic monitoring purposes as specified in section 2.6.5. The MCU unit has ten, 10-bit analogue to digital inputs attached to the board, referenced to the 3.3 V supply voltage. The resistive temperature and power sensing circuitry included on the bottom PCB features four resistors in two separate resistive chains. These resistive sensing devices are in the 100 k Ω + range to limit power draw to micro-ampere levels to limit the power consumption. The footprint is also minimized, as 0603 resistors are used for the resistive chain.

The resistive chain measures 430 k Ω total from 3.3 V to ground for the battery monitoring circuit. The resistive chain measures at 168 k Ω at 25 degrees Celsius for the temperature sensing circuit. Under these conditions, the current draw of the two diagnostic peripherals can be evaluated in equation 4.4.

$$\text{Temperature Circuit Current Draw: } I_t = \frac{3.3 \text{ V}}{168 \text{ k}\Omega} = 19.6 \mu\text{A at 25 degrees Celcius}$$

$$\text{Battery Circuit Current Draw at maximum charge: } I_b = \frac{12.4 \text{ V}}{430 \text{ k}\Omega} = 73 \mu\text{A at full charge}$$

$$\text{Total Peripheral Current draw} = I_t + I_b = 92.6 \mu\text{A} \quad (4.4)$$

Combined, the average current draw by the peripheral monitoring components does not exceed 1 mA, and hence the power draw of these peripherals are negligible compared to the main processor and RF module.

The presence of these diagnostic circuits allows fault conditions to be detected, such as overheating, low battery, or excessive power use. These peripherals can detect fault conditions before the fault causes the node to be inoperable. Forward warning allows the node to achieve the target level of fault tolerance stated in section 2.6.5.

A small low-power LED (LED1 in the schematic on figure 4-7) is provided to the operator as an indicator of node power. This LED draws negligible current and is connected directly to the 3.3 V line. This LED is powered on as long as the board is powered. The power draw is minimized to less than 2 mA, as shown in equation 4.5.

$$\text{Voltage drop over LED: } 3.3\text{ V} = V_f + IR \rightarrow V_f = 2.2\text{ V}$$

$$I \times 560\ \Omega = 1.1\text{ V} \rightarrow I = 1.9\text{ mA} \quad (4.5)$$

4.3.4 Deployment detection accelerometer

The HERMES node is incapable of communication unless the radio is active. The deployment of the node serves as a trigger for the node's activation of the radio module to initiate communication. As will be described in section 5.2.1, the radio functions are left off in transit in order to reduce interference on the 2.4 GHz band. Power is conserved when the radio functions are switched off when not required. The accelerometer schematic is shown in figure 4-14.

Figure 4-14: Accelerometer Schematic

An ST Microelectronics LIS3DSH accelerometer is implemented to detect the node's orientation relative to space. When the node is deployed, the accelerometer detects the direction of Earth's

acceleration to determine its deployment orientation. This device operates on the same 3.3 V power supply as the remainder of the node system. Due to the one-use nature of this node component, it is desired for this system to draw minimal current. The LI3DSH draws 225 μA at active mode, and draws 2 μA at stand-by [46]. The accelerometer interfaces to the MCU with the SPI protocol. The 4-pin SPI system consists of four lines connected as shown in figure 4-14. This accelerometer has a dynamically selectable full scale output of up to $\pm 16\text{ g}$, which exceeds the target acceleration detection of 1 g due to earth's gravity. The accelerometer withstands up to 10000 g with high shock survivability to prevent failures due to the vibration of the HADES chassis in transit.

4.3.5 System Power draw

The target operating lifetime of the HERMES node is 8 hours as specified in section 2.6.5. Each system on the HERMES node operates on a 3.3 V supply as supplied by the regulator. The table of current draw is given in table 4-2. The node's operation can be given in two states: The node is in transit aboard HERMES when it is idle, and the node is active when it has been deployed.

Table 4-2: HERMES estimated power draw

Component	Current Draw (mA) node idle	Current Draw (mA) node active
WizFi220 RF module	0.035	250
MK20DX256VLH7	100	100
Peripheral Sensors + on LED	< 2.5	< 2.5
Accelerometer LIS3DSH	0.225	0.002
Power Indicator LED	0	25
TOTAL	103	378

When the HERMES node is idle, the system draws at most 103 mA from the 3.3 V line. When the node is transmitting, the system draws 378 mA at worst case. Note that the MCU unit is estimated to always draw 100 mA as it must monitor the accelerometer to determine deployment conditions and hence it cannot be placed into sleep mode. Over 8 hours of active operation, the total current draw capacity is required to be 3.1 Ah. On a 3.3 V system, the requisite power capacity of the power source is calculated to be 9.97 Wh as shown in equation 4.6.

$$3.1\text{ Ah} \times 3.3\text{ V} = 9.97\text{ Wh} \quad (4.6)$$

The regulator discussed in 4.3.2 has an efficiency of roughly 75% at 12 V. An additional 33% is added to the estimate to account for regulator power loss. The target capacity of the battery system is then raised to 13.3 Wh as shown in equation 4.7.

$$9.97 \text{ Wh} \times \frac{4}{3} = 13.3 \text{ Wh} \quad (4.7)$$

The estimated power draw of 13.3 Wh at worst case allows the node system to remain operational for the target 8 hours. This power draw is supplied by the power system detailed in section 4.4.

4.4 Lithium-polymer Battery system

The main power supply of the nodes requires at least 8 hours of operation as per the specifications described in section 2.6.5. Multiple technologies are examined in section 2.3, with lithium polymer being the best technology in terms of cost, charge per volume, charge per weight and availability given a specific form factor. Two different battery solutions are selected for the node. Figure 4-15 shows the two node power packs side by side. Two similar solutions are discussed to take advantage of similar power capacities and nearly identical form factor.

Figure 4-15: HERMES battery solution

4.4.1 Solution one: 11.1V 3 cells

Solution one features three 1200 mAh 3.3 V cells placed in series. The nominal voltage of the pack at full charge is 12.6 V, falling to 9 V when fully depleted. The cells measure 6 x 34 x 56 mm and therefore are roughly 18 mm thick when stacked top on top as shown in figure 4-16, right. The total power capacity averages 4.44 Wh per battery cell. In total, this translates to 13.32 Wh power capacity per node. The height is given to be 18 mm, but the cavity for the batteries equates to 22 mm to account for battery pack swelling that occurs during normal lithium-polymer operation.

4.4.2 Solution two: 8.4V 2 cells

Solution two features two 2000 mAh 3.3 V cells placed in series. The nominal voltage of the cells at full charge is 8.4 V, falling to 6 V when fully depleted. The cells measure 10 x 34 x 56 mm, and is therefore nearly the same shape as solution one when considering the width and length of the overall stacked package. The height of the new node solution is given to be 20 mm, which provides 2 mm of height clearance. The cells placed in series provide a full 14.88 Wh power capacity per node.

4.4.3 Battery solution summary

The two solutions provide the minimum target of 13.3 Wh as discussed in section 4.3.5. Both solutions feature inbuilt over-current, under-voltage, and short protection. This function is shown by the small, thin circuit boards by the output tabs of each cell in figure 4-15. These inbuilt boards limit the current draw of all the cells to 1 A per cell at 12 V potential. This translates to a 3 A draw with solution one and a 2 A draw with solution two. The current maximum that can be drawn is at least 2 A, and both solution one and solution two can deliver at least double the maximum power under normal operation. This satisfies section 2.6.5 for over-current protection.

Figure 4-16: HERMES battery height comparison

Both battery solutions provide better than the 13.3 Wh of power required to power the node for a full 8 hours as specified. Solution one is preferred as this solution provides a power rail voltage of around 12.6 V as opposed to 8.4 V. This matches the expected optimal efficiency of the regulator as discussed in section 4.3.1.

4.5 Electronics enclosure and board placement

4.5.3 PCB mounting

The node enclosure offers two positions for the external dipole antenna (discussed in section 4.5.3). The antenna is connected to a waterproof U.FL to SMA connector with the U.FL output mounting to the top board. The top to bottom board connection is achieved with a matched pair of header pins and header sockets. As the bottom board is the source board for the power supply, the bottom board contains the sockets. The header pin arrangement is shown in figure 4-9 and matches with the header pin arrangement on figure 4-4. Header pins of through hole-type are used due to the structural strength advantage over surface mount headers. The boards are held away from one another via inserts and M3 screw holes as shown in figure 4-17. The outside end of the board is bolted into the node enclosure with M3 standoffs. The size stacking advantage of the PCB components allows the node electronics to be packed within the 5 x 4 cm node compartment as shown in figure 4-17.

Figure 4-17: PCB electronics fit in enclosure

4.5.2 Enclosure

The HERMES wireless node enclosure features a compact plastic enclosure with dimensions 7 x 2.5 x 13.5 cm. The entire HERMES node fully loaded weights 225 grams. The node enclosure allows the tight packing of both the PCB system and the battery system. These compartments are segregated by a central pair of screw mounts for the HERMES node electronics. The antenna mount is provided on the electronics side as the U.FL connector is confined to this area.

Deployment of this node design is explored by the work of Thompson [35] and Jones [36]. The node is held inside the deployment mechanism by screw-rods that stacks the nodes. The holding torque of this system provides two unique advantages. The first advantage is that due to the worm principle, the node cannot move inside the deployment tube unless the rods themselves are turned. The second advantage of this screw rod system is the increased deployment torque. As the node is pushed out of the deployment system, it is possible that the deployment system is blocked by debris or is otherwise impeded by hostile mine conditions. The high torque of this system maximizes the push strength of the nodes. This factor maximizes the deployment reliability in sub-optimal mine conditions. The brass push-spiral mounts are shown in figure 4-18.

Figure 4-18: HERMES node with electronics size comparison

The HERMES node enclosure provides two positions for the antenna system as shown in figure 4-19. The antenna positioning clears the top of the bottom PCB on both orientations. Combined with 180 degree rotational symmetry, the total number of possible node orientations when viewed from the deployment tube is four. This provides a four-fold symmetry in the node positioning inside the deployment tube.

Figure 4-19: HERMES node antenna placement

4.5.3 Antenna

A 2.1 dBi dipole antenna is selected as the antenna unit. The WizFi220 is a radio module with only one transmit/receive path and hence only one antenna is required as discussed in section 4.2.1. Due to the non-suitability of directional antennas and the limited space available for an antenna, a generic dipole antenna is chosen. This antenna is 9 cm long and extrudes away from the top of the node by

around 3 cm. The antenna is pictured in figure 4-20. The red RF cable is coaxial. This wire is looped underneath the WizFi220 unit to prevent interference from the battery section. The compartmentalized node design allows the high-speed RF and the power components to be separated.

Figure 4-20: HERMES antenna

4.6 Summary

This chapter described the hardware design for the HERMES node. Topics discussed in this chapter included the implementation of a dual PCB design with a new MCU unit suitable for implementation with the WizFi220. Challenges of a limited size compartment were overcome with a stacked PCB solution. Node peripherals such as temperature detecting elements and deployment were implemented to provide the node with functions vital to end-case operation. This chapter also discussed a battery solution that was capable of providing the requisite run-time. Chapter 5 discusses the software of the HERMES node, including the operating system on-board the MCU unit as well as endpoint support software. The HERMES node interfaces to endpoint processors and this function is explored. Chapter 5 also discusses ROS code that implements a data bridge between the hardware and the software.

Chapter 5 HERMES Embedded Software and Network Design

5.1 Overview

This section describes the design of the HERMES node operating system and the associated interface ROS software on the endpoint processors. The HERMES operating system implements routing functions and interfaces to the node sensing peripherals. These HERMES operating system functions are described in Section 5.2. The network traffic that the HERMES MCU processes conforms to the WizFi220 packet transfer protocol and a custom software interface is developed to handle data transactions between the two modules. A custom application level header and software interface is designed to minimize packet processing on the MCU. The endpoint hardware differs from the main node hardware, and several changes to the HERMES node software to suit the endpoint devices are discussed in section 5.2.7. This examines the addition of the ROS framework to the embedded software and the implementation of the interface bridge between the hardware and the endpoints. Section 5.3.4 discusses control signals from the XBOX360 controller which is used as the control device. The data message types and the ROS implementation of these messages are discussed in section 5.3.5. Section 5.4 summarizes the work in this chapter.

5.2 HERMES node embedded software

The HERMES node features the MK20DX256 chip as its primary microprocessor unit. The main function of this MCU is to interface with the WizFi220 on the application level. Major tasks performed by the MCU are boot management and data forwarding. In addition to these major functions, an important additional function is provided by the Teensy 3.2 to allow it to communicate to the two endpoints, the HADES robot and the base station. This is provided for by the associated USB connection on the Teensy 3.2 board. The MCU implemented on the deployable nodes also performs peripheral functions such as reading the temperature, battery voltage, and detecting node orientation as specified by section 2.6.5.

5.2.1 Deployment and setup process

The HERMES node features a boot setup procedure that initializes the features of the node and this is performed by the MCU. The code implementation of the initialization process is shown in appendix A.4.3. Several interfaces are initialized when a stable 3.3 V voltage is gained:

- The UART interface to the WizFi220 is set to 115.2 kbaud to initialize the serial link to the WizFi220 to prepare the RF setup process (see section 5.2.3).

- The SPI interface is initialized to a 2 MHz clock to setup the communication with the accelerometer unit for deployment detection.
- One of the two power LEDs (see section 4.2.2) is set high to indicate that the initial setup has been complete. This provides the operator with an indicator that the node has entered the setup process.

After the initial setup process, the HERMES node waits for its deployment. The deployment trigger is provided via a blocking function querying the on-board accelerometer. The WizFi220 is held in a reset state before deployment. By holding the WizFi220 in reset mode before the node is deployed, several advantages are attained:

- The WizFi220 is set to idle and draws less than 35 μA [41] in this state. This minimizes the power draw from the batteries during transit. Each node is only connected to the network once the node is fully deployed.
- The WizFi220 must be held off whilst in transit due to the proximity of every node during deployment and transit. By disabling the RF functions of each node in transit, interference on the 2.4 GHz channel is avoided. This is desirable as each node is set to the maximum transmit power when deployed in the mine.

Because the wireless system is non-functional while the node is in transit, the RF module cannot be used to query or change the state of the node. The node must therefore independently evaluate its deployment status to activate the RF functions of the node when appropriate. A successful node deployment is detected by the hardware accelerometer, which is polled at a 10 Hz rate and filtered with a moving average filter of length 10. Two conditions are checked to determine whether a node has deployed:

1. The first condition is the orientation of the node. The hardware accelerometer can detect acceleration due to the earth's gravity. In transit, the nodes are stacked at a 90-degree angle. When deployed, they become upright. The axis of gravity changes for the accelerometer, which allows the node to determine its own orientation relative to the earth.
2. The second condition is the variance in the reading of the node. When the HERMES node is inside the deployment tubes of HADES, HADES' motion causes the node to vibrate due to the non-linear whleg motion. When the variance in the HERMES acceleration reading is high, this indicates that HADES is in motion. When the variance is low, this indicates that HERMES is still, either due to a deployment in steady state, or HADES is still.

A combination of the two conditions can be used to infer the state of the node. When the orientation of the node is such that the node is upright as per condition one, the node has been deployed to the ground. When the variance of the node orientation reading is low as per condition two, this indicates

that the node has reached a stable position. The node is both upright and not moving. This provides the trigger to advance the internal state machine of the node to allow RF functions to boot. A custom function *waitForDeployment* is used after the setup of the Teensy interfaces in order to poll the accelerometer for the orientation information. The main function body is shown in figure 5-1.

```
while (true){

    z_accel= 0; //reset the acceleration on this axis.

    //get data z
    low_byte = SPITransferData(Z_LOW,0x00);
    high_byte = SPITransferData(Z_HIGH,0x00);

    //add it to the array
    z_buff[v_pos] = (high_byte << 8) + low_byte;

    //if the readings a are within standard deviation of the array reading.
    if ((z_buff[v_pos] > z_accel - ACCEL_DEV) && (z_buff[v_pos] < z_accel + ACCEL_DEV)) stable = 1;
    else stable = 0; //we are no longer stable.

    ++v_pos; //increment the array position
    if (v_pos == 10) v_pos = 0; //reset v position

    for (int k = 0; k < 10; ++k){ //add the elements of the acceleration
        z_accel += z_buff[k];
    }

    //divide through to get acceleration
    z_accel /= 10;

    //if the accelerometer is at a valid angle, we increment the number of valid reading points.
    if (z_accel > ACCEL_FLAT && z_accel < ACCEL_TILT_MAX) ++valid;
    else valid = 0;
}
```

Figure 5-1: Accelerometer processing code in *waitForDeployment*

The deployment system reads the data from the accelerometer using the *SPITransferData* function. The *SPITransferData* function accesses two 8-bit variables corresponding to the high and low byte of the z-axis acceleration reading, and is addressed by the register addresses *Z_LOW* and *Z_HIGH*. This is added to an array of length 10 to smooth out the individual readings. The two checks for stability and valid angle are also shown in figure 5-1. The variable *stable* checks each register to ensure that each reading is of an acceptable deviation away from the mean value through a boundary statement that sets the variable. If the value is high, then the *stable* flag is de-set, indicating a high variance. This implies that the node is subject to high accelerations as is expected when HADES is in motion. The valid angle check is the last statement of figure 5-1. This section of code maintains a counter variable named *valid*, and this ensures that the average is within acceptable accelerations over a period of time. This *valid* counter is compared to the variable named *ACCEL_STABLE_THRESH* in figure 5-2 which checks if the angle has been held for long enough.


```
//break if we are both valid and stable at the same time
if (valid == ACCEL_STABLE_THRESH && stable == 1) break;

delay(ACCEL_DELAY); //do readings every 200ms
```

Figure 5-2: Breakpoint statement for deployment detection

Figure 5-2 shows the logic statement that allows the code to continue to the WizFi220 setup functions. The break statement breaks out of the blocking perpetual loop that prevents the code from continuing until after the device has detected a successful deployment. An accelerometer delay is implemented to only poll every 200 ms. This minimizes power draw during the polling process and ensures that the stability of the node is maintained for a significant length of time before accepting a successful deployment.

Due to the uneven ground of the mine it is desired to allow the node to tolerate some deviation from upright in case the node deploys on ground that is not exactly flat but still attains a stable position with low variance. A plot of integer data to deployment angle on the z-axis is shown in figure 5-3. Figure 5-3 shows that the standard deviation of the node angle is observed to be 50 (integer points) from resting. The standard deviation requirement is hence set to be 100 to determine whether a node is in rest state and this is implemented as the threshold for condition 2. This is the value of *ACCEL_DEV* in figure 5-1.

Figure 5-3: Deployment angle vs. Accelerometer output

While parts of the mine can attain slopes of up to 30 degrees, it is not actually desirable to place nodes in such sloped areas. Figure 5-3 shows that the *ACCEL_TILT_MAX* variable is set to around 1300 integer points to set the threshold to 20 degrees. When the correct stable orientation is obtained, the network setup procedure is then performed as shown in 5.2.3.

5.2.2 Network design

The WizFi220 wireless RF module utilizes the 802.11b wireless protocol operating on the 2.4 GHz ISM channels. The 802.11b protocol comes with specifications for an ad-hoc network. This ad-hoc specification allows each node to advertise its connection and allows any nodes that receive the signal

to join the network. Each new node that gains access to the network also gains the ability to advertise the network and allows others to join. This property allows the deployed nodes to join the network and expand as HADES penetrates further into the mine.

Physical and MAC layer functions

The static association of the HADES network is provided for by a base station Linux PC. The base station Linux system provides the ad-hoc SSID “HADESNET” operating on channel 1 of the 2.4 GHz ISM band. This association is then provided to the HADES endpoint and the base station wireless endpoint. This process is generally performed before the deployment of HADES into the mine. Both HADES and the base station endpoint module gain the ability to pass the network parameters to future nodes. The Linux steps for setting up this network is provided in appendix A.7. The ad-hoc network setup dialog is shown in figure 5-4.

Figure 5-4: Linux HADESNET setup

IP addressing

The WizFi220 operates on a static network addressing system so DHCP is disabled on the network. The ad-hoc paradigm means that there is no coordinator of the network; each node maintains the network connection independently of others. Due to the flat network hierarchy required for the ad-hoc paradigm, there is no DHCP server to query for IP addresses. IP addresses are used to identify individual nodes, and therefore IP coordination is required. To resolve IP addressing, each entity is assigned an IP address prior to deployment. This IP address is burned into the flash memory, and is configured prior to loading. The IP address is provided to the WizFi220 on boot. Since all addresses are static and loaded uniquely at node setup, the need for IP address querying and assignment is avoided. This also prevents IP address conflicts.

The IP address setting function is shown in figure 5-5. This function evaluates the IP address of the node using a *node_Number* variable, and this number is placed in the IP set command string to be parsed to the WizFi220 as an AT command using the *send_command* function.

```
//Set self IP address.
void SetHostIPAddr() {

    //first calculate our IP address.
    char node_number_ASCII[4] = "000";
    char gateway_ASCII[4] = "000";

    Int2AsciiExt(node_Number, node_number_ASCII); //to current node
    Int2AsciiExt(node_Number - 1, gateway_ASCII); //to last node

    //set our current node
    COMMAND_SET_SELF_IP[16] = node_number_ASCII[2];
    COMMAND_SET_SELF_IP[17] = node_number_ASCII[1];
    COMMAND_SET_SELF_IP[18] = node_number_ASCII[0];

    //set the next node
    COMMAND_SET_SELF_IP[42] = gateway_ASCII[2];
    COMMAND_SET_SELF_IP[43] = gateway_ASCII[1];
    COMMAND_SET_SELF_IP[44] = gateway_ASCII[0];

    //AT command: IP address set.
    send_command(COMMAND_SET_SELF_IP);

    delay(inter_command_delay); // delay 300ms
}
```

Figure 5-5: IP address setup code

The IP address range is 10.42.0.0/24 providing 256 unique IP host addresses. The maximum and minimum address of host 255 and host 0 are not used due to their unique use as network and broadcast identifiers in the IPv4 protocol, leaving 254 unique IP addresses. Table 5-1 shows the IP definitions for each entity on HADESNET. As shown in table 5-1, the largest legal address host 254 is reserved for HADES, and host 1 is reserved for the base station. The first node deployed has host address 2 and so on for consecutive nodes. This provides 252 legal node addresses. When nodes are deployed, each node in the chain has an ascending IP address to HADES, and a descending IP address to the base station.

Table 5-1: IP address assignment scheme

Address	Definition	Address use
10.42.0.255	Broadcast Address	Reserved – not used
10.42.0.0	Network Address	
10.42.0.254	Highest legal host address	HADES robot
10.42.0.1	Lowest legal host address	HADES base station
10.42.0.2 through 10.42.0.253		HERMES nodes

As the nodes are placed in IP address order, each node has a specific position in the deployment stack. The last nodes are placed into the HADES deployment tubes first, with the lowest IP address nodes placed into the stack last. This ensures that the nodes are deployed into the mine in ascending IP address order. This node order facilitates the main process as described in section 5.2.5.

UDP transport layer

The physical and MAC layers of the 802.11b protocol handles the RF and channel control functions of HADESNET. IPv4 static addressing is used as the IP layer. This uniquely identifies each node in the network for data-routing purposes. To facilitate data transfer between nodes, UDP is used as the transport layer protocol to perform data streaming from the RF functions to the application MCU.

The WizFi220 provides both TCP and UDP socket support with up to 16 unique transports being supported at once [41]. The WizFi220 provides these 16 sockets in the form of uniquely addressable CID (Connection **I**Dentifier) numbers. Each CID identifies a socket as TCP or UDP. Each node connects to its neighbours using a system of UDP connections.

A UDP connection is a transport protocol that provides no data transmission control and no flow control. UDP provides data checking through a checksum field. Due to the lack of transmission control, lost packets are not retransmitted. The lack of advanced features negates the need for buffering of packets for retransmission, and hence the performance of UDP in wireless networks in terms of raw data-rate is improved. The use of UDP also minimizes packet transmission delay as fewer processing tasks are performed by the RF module before the data is presented to the MCU. Video data relies more on aggregate data-rate as opposed to accurate reproduction of frames at the end device. Video data is loss tolerant, which matches the transport services UDP provides.

The WizFi220 provides two types of UDP sockets, server sockets and client sockets. Server sockets are many to one, but the clients are only one to one. Many clients can connect to each server, but each client can only connect to one server. Each node is initialized with a single UDP server on port 5000 as the listening server to receive packets, and two UDP client ports to serve as transmitter ports. One client port is initialized to a lower IP address and hence provides the downstream link to the base station. One client port is initialized to the higher IP address, and provides the upstream link to HADES. Each node hence has two transport connections; one to the previous node, and one to the consecutive node. This provides full duplex data transfer between HADES and the Base station. Figure 5-6 demonstrates how each node assigns its UDP sockets.

Figure 5-6: UDP port assignments

Figure 5-6 shows that transmitting on CID 2 will always cause a packet to be transmitted up the line to HADES, and packets transmitted on CID 1 will always result in a packet being transmitted to the base station. This system allows efficient transfer of data as no decision making is required. Since each node operates a UDP server on port 5000, each node does not need to query the other nodes to determine the destination port. Each node also knows the IP address of its neighbours, as nodes are deployed in sequential order. Each node always knows the parameters it needs to forward data, and the need for nodes to query for port numbers or IP addresses is avoided. This strategy allows the maximum data-rate to be attained for transmission of useful data, and avoids the need for coordination between nodes that would consume bandwidth.

5.2.3 Network setup

The MCU sends the WizFi220 a series of commands to setup the network once the WizFi220 has left the reset state and is accepting commands. This process happens once the deployment process detailed in 5.2.1 has been complete. The WizFi220 is only put into the active state once the node has been fully deployed. The WizFi220 boot sequence is shown in figure 5-7.

```

send_command(COMMAND_AT);

send_command(COMMAND_DISCO);
send_command(COMMAND_AD_HOC_ENABLE);
send_command(COMMAND_PASSWORD);
send_command(COMMAND_BULK_MODE);
send_command(COMMAND_EXT_POWERAMP_ON);

```

Figure 5-7: WizFi220 boot sequence

The network setup routine shown in figure 5-7 first sets up the parameters of the WizFi220 operation. The first command disconnects the WizFi220 and this resets the WizFi220's network association. Ad-hoc mode is selected as specified in section 5.2.2. The transfer protocol is enabled using bulk mode. Finally, the external power amplifier is enabled for maximum transmit power.

When the basic WizFi220 static variables are set, the network configuration is performed as shown in figure 5-8. This involves finding the HADESNET network, connecting to it, and then configuring the UDP sockets and raising the baud rate of the communication link to 921600 baud. The full code for achieving this is shown in appendix A.4.3.

```

HADESNET_Scan_Connect(); //connect to HADESNET
SetHostIPAddr(); //Set self IP address 10.42.0.xxx

//setup UDP sockets
NodeServerSetup(); //Setup our UDP server: GIVES CID 0
NodeDownstreamSetup(); //Setup our link to home base station: GIVES CID 1
NodeUpstreamSetup(); //Setup our link to HADES: GIVES CID 2

delay(1000);

UART_Baud_UP(); //raise baud rate to max

```

Figure 5-8: HADESNET network search routine

In order for the HERMES node to connect to the network, the WizFi220 unit must first search for the network association. The name of the connection is HADESNET, and this is polled at a 2 Hz rate. This cannot be a static function to connect to HADESNET as the radio association may not be immediately visible to a HERMES node that has been recently deployed. The OK_Check function searches the buffers for the string “OK” which denotes a successful command. When this is received, the while true loop breaks. At this stage, the HERMES node has gained the connection to the rest of the network, and hence the WizFi220 can continue with the network setup routines.

```

void HADESNET_Scan_Connect(){

    Serial.println("Connecting to HADESNET...");

    //loop until we return (after getting the OK.)
    while (1){

        clear_Ser1_Buffers(); //first clear the buffers.
        //put the request in.
        send_command(COMMAND_CONNECT_HADESNET); //AT: CONNECT TO HADESNET

        if (OK_Check()){ //If we get an active OK
            if(testing) Serial.println("HADESNET connection obtained.");
            break; //If we get a satisfactory response, break.
        }

        if(testing) Serial.println("No HADESNET connection...Continue Scanning");
    }

    //clear all buffers, we are connected
    clear_Ser1_Buffers();

    delay(500);
}

```

Figure 5-9: HADESNET connection routine

The HADESNET scan connection function shown in figure 5-9 polls the 2.4 GHz channels for the HADESNET connection. This function hangs the MCU until the HADESNET network is found. Once it is found, the IP address is set from memory. The default 115.2 kbaud is used to initiate the setup process as no data transfer is involved. After this process, the MCU and WizFi220 are in the correct state to accept data transfer and the program advances the internal state of the HERMES node to the main process.

5.2.4 Data transfer protocol

When data is received over the radio, the node processes the packet through the IP stack and presents the data to the MCU over the provided serial link in the WizFi220 data format. This data output serial protocol is described in the WizFi220 manual as *bulk mode* [41]. An implementation for data processing is shown in section 5.2.5.

An example of the incoming data format is shown in figure 5-10. The data format provides the host MCU with the source of the data as well as the data itself. As shown in figure 5-10, the <esc> character (char 27) is used as a delimiter indicating a start of packet. As ASCII does not support a visual <esc> character, the square character is used to denote this. The character ‘u’ denotes that the transport protocol of the data is over UDP. As all packets on the HADESNET network operate on UDP, this byte is meaningless to the MCU system. This is followed by the IP address of the transmitter node. A tab character denotes the separation of the header to the data portion of the serial stream, which is represented by the simple space after the last ‘1’ of the incoming data packet. In this case, there are 10 bytes in the data consisting of the 10 byte string “1c12345678”. The data stream is presented to the user and is delimited by the <esc> character followed by the ‘E’ character. These two characters form the escape sequence indicating end of data.

Figure 5-10: Example data packet

Custom 3 byte header

A custom 3 byte header is added to the start of the packet in the application level. This reduces the information payload of each packet by 3 bytes. This custom header contains the length of the packet in multiples of 10 bytes, the type of the packet, and leaves an additional byte for reserved purposes. For example, this field may provide a packet indicator flag in the future. The packet type definition is important to each endpoint since packets are processed differently depending on the packet type – video packets are processed differently to control or sensor packets. Table 5-2 summarizes the format of the custom 3 byte header interface. In table 5-2, the first two bytes conform to this header. As this packet is a dummy packet example, the reserved field is set to ‘1’. The first byte shows ‘1’, which indicates the packet length of (1 x 10) 10 bytes. The ‘c’ character denotes the type of the packet. In this case, this ‘c’ character indicates a generic control packet.

Table 5-2: 3-byte header format

Byte Offset	Definition	Value	Meaning
0	Data packet length (including header)	1-99	The data size is value x 10 in units of bytes.
1	Control function	See 5.3.5	The type of packet contents.
2	Reserved	0	Not Currently used

Since the data length is the length of the packet in multiples of 10 bytes, the minimum data packet size is locked to 10 bytes. Furthermore, only packets in multiples of 10 bytes are allowed. This simplification allows the length field to be expressed as a single byte, minimizing the packet space that is consumed by this custom header. Processing this header only requires a single integer multiplication by 10. The maximum data size is capped to 990 bytes due to buffer limits observed on the WizFi220.

Several advantages are gained by adding this custom 3 byte header to the data contents. When a packet is received by the MCU from the WizFi220, the packet header provided by the base WizFi220 does not provide the length of the data. The length field is instead provided in the 3 byte header. This data transfer protocol allows the packet length to be determined before the data transfer instead of waiting for the explicit <esc> + E delimiter sequence. This has major advantages as this removes the need to check for the escape character on the MCU. This simplifies the receiver state machine by negating the need to add additional states to check for the delimiter escape sequence. Furthermore, the need for comparison to the delimiter is removed, and hence all bytes are passed to the output buffers without a need to perform costly *if* statement checks on the data delimiter. Finally, the length field avoids the need for differentiation of an escape sequence in data. The MCU simply reads a valid number of bytes, treating all bytes as data. No additional packet encoding is required to separate the <esc> + E sequence in data from the end of packet delimiter. Byte stuffing techniques are not needed to demarcate packet boundaries.

5.2.5 Main process

The MCU processes the incoming data on the serial line and retransmits the data by changing the target IP address. By manually changing the packet recipient, data can be passed from node to node. The application level routing observes the IP address of received packets and forwards packets on an IP address according to the logic block diagram shown in figure 5-11. This figure shows that a packet received from a higher IP address is forwarded to a lower IP address, and vice versa. Packets are forwarded from one node to the other in one direction until the packet reaches an endpoint device, either HADES or the base station.

Figure 5-11: IP address scheme block diagram

The MCU routing process relies on the IP addressing scheme described in section 5.2.2. This scheme assigns a specific IP address to each node in descending order from HADES. This arrangement is shown in figure 5-11. Routing is simplified with this method as all packets originating from HADES will eventually reach the base station, and all base station packets will eventually be routed to HADES.

The main process of the routing function is shown in figure 5-12. This block diagram shows the main process of processing an incoming packet. The MCU process acts as a repeater that simply repeats the packet to a different destination. The only processing task that the main process performs involves checking the packet destination and forwarding the packet appropriately. This process is agnostic to packet contents. Each packet of whatever type is treated the same and this is reflected in a simple single input buffer architecture used in the packet reception logic.

Figure 5-12: HERMES MCU main process block diagram

The main loop is implemented with a *while true* statement as shown in figure 5-13. The *serial1* buffers receive incoming data from the WizFi220 and the process continues until there is no data remaining in the serial buffers.

```
while (1){

    //we recieve some data over the WizFi serial, so we get it into the Serial buffers.
    if (Serial1.available()){
        while (Serial1.available()){

            //get the next packet.
            inByte = Serial1.read();

            //CHECK FLAGS and packet
            if ((set_flags == 0x00) && (inByte == delimiter)){

                analogWrite(LED_CONNECTED,0); //set Orange LED OFF

                set_flags = 0x01; //set the flag

                while (Serial1.available() < 5); // do nothing until we reach 5 bytes
            }
        }
    }
}
```

Figure 5-13: Main process implemented as cpp code

The section of code in figure 5-13 waits until serial data arrives from the WizFi220. As the WizFi220 can also output error messages and status prompts, the state machine only waits for the start of the data delimiter. This only checks for packet data and all other data will be discarded. This is used in the rare case of a buffer overflow when a packet delimiter is truncated. This causes the loss of a packet and the state machine simply discards the contents. A *set_flags* variable in figure 5-14 is used as an 8-bit flag buffer that allows the state machine to advance to the next sections of the state machine when the correct delimiter is triggered.

```
//Check delimiter
else if (set_flags == 0x01 && inByte == delimiter){
    set_flags = 0x07; //set second flag.
    #ifdef DIGEST
        Serial.write(inByte);
    #endif
}
else if (set_flags == 0x01 && inByte != delimiter){
    set_flags = 0x00; //reset state machine.
    #ifdef DIGEST
        Serial.write(inByte);
        Serial.println("DELIMIT BAD");
    #endif
}
```

Figure 5-14: Error checking and gating statements

When the packet does not conform to the packet format, the packet is discarded using the gating logic in figure 5-14. When a bad delimiter is detected, the state machine will reset the set flags and this will return the state machine function to the main loop. This also protects against buffer overruns in the event that the UART serial buffers experience unexpected burst transmissions, preventing bad packets from propagating through the network.

In order to determine the address of the node to route to, the IP address is decoded and stored in an integer *IP_HOST* as shown in figure 5-15. This IP address is compared to the IP address of the current node. The last comparison in figure 5-15 checks the target CID and determines the direction of packet routing.

```
//1 digit
//..0.1
//2 digit
//..0.11
//3 digit
//..0.111

IP_HOST = 0; //rezero HOST.

//do 1st digit
IP_HOST += (inByte - '0'); //ones position

//do 2nd digit
inByte = Serial1.read(); //try for tens.

if (inByte >= '0' && inByte <= '9'){ // if tens

    IP_HOST *= 10; //new tens
    IP_HOST += (inByte - '0'); //new ones
}

//do 3rd digit
inByte = Serial1.read(); //try for hundreds.

if (inByte >= '0' && inByte <= '9'){ // if valid number.

    IP_HOST *= 10; //new hundreds and ones
    IP_HOST += (inByte - '0'); //ones
}

//decide target CID.
if (IP_HOST < node_Number) target_CID = 2; //send north to HADES
else target_CID = 1; //send south to base station
```

Figure 5-15: CID routing function

The logic in figure 5-15 uses the UDP CID arrangement discussed in section 5.2.2 and calculates the recipient of the packet based on the IP host setup. A process of decoding the IP address involves determining whether the incoming characters are numbers after the third period denoting IPv4 addresses. Due to the difference between 10.42.0.1 and 10.42.0.101, the packet host number can differ by up to 2 characters in length. To remedy this, each character is chosen in order with the process stopping once a non-character is detected. Using feedback to determine the next delimiter boundary and calculating the receiver IP address is the main function of the process in figure 5-15.

A tab character delimits the end of the IP address against the start of the user data. This process sets the set flags to 0x1F indicating start of data. The process of data parsing from the buffers is shown in figure 5-16.

```

else if (set_flags == 0x1F){ //Start getting the data. We have reached the end of the header

    while (inByte == C_R || inByte == N_L) inByte = Serial1.read();

    //The first byte is the length byte divided down by 10, so we can grab it and put out the length.
    pack_len = inByte * 10; //multiply by 10.

    #ifdef DIGEST
    //print the packet length.
    Serial.println();

    Serial.print(pack_len,DEC);
    Serial.println();
    #endif

    if (pack_len > 990){
        set_flags = 0;
        #ifdef TESTING
        Serial.print("PACK_LEN_ERR");
        #endif
        break;
    }

    #ifdef DIGEST
    Serial.println("====DATA START====");
    #endif

    //Send the data to array form
    Int2AsciiExt(pack_len,length_buffer);

```

Figure 5-16: 3-byte header processing

Figure 5-16 shows how the packet data is handled after receiving the start of data delimiter. Section 5.2.4 describes the use of a custom header designed to optimize the data reading process. The processing of this custom header is demonstrated in the code in figure 5-16. By allowing a packet size to be transmitted in the actual data, the packet size is easily decoded. This value is loaded into the state machine to determine end of data boundaries. Error checks are conducted in this section to ensure that data does not exceed the 990 byte packet size. This function sets the *pack_len* variable which is used to count the number of bytes that are received by the WizFi220. This *pack_len* variable is used to compare to a running count as the data is acquired by the data parsing process.

Error checking is performed on the lower levels of data reception on-board the WizFi220. The WizFi220 receives a packet on the RF interface and CRC checks are present on the physical layer and on the MAC layer. UDP implementations provide a header + data checksum as the last error check before the data is presented to the MCU unit. If the data is corrupt, then the packet will not be presented to the MCU.

```

while (true){ //new while

    if (bytes_encountered == pack_len){ //if we reach the end of the packet.
        set_flags = 0x00; //clear the flag

        //put the last packet into the stack
        TxBuff[TxPos++] = inByte;
        //Serial1.write(inByte);

        //Send the completed packet to Wizfi.
        for (loop_v = 0; loop_v < TxPos; loop_v++){

            #ifdef DIGEST
                //Mirror to the serial.
                Serial.write(TxBuff[loop_v]);
            #endif

            //multihop problem here
            Serial1.write(TxBuff[loop_v]); //send the packet out to serial.
        }

        #ifdef DIGEST
            //Mirror to the serial.
            Serial.write(N_L);
        #endif
    }
}

```

Figure 5-17: Packet data processing and re-routing

The MCU reads in *pack_len* bytes and stores the data in a packet variable *TxBuff*. This copies the data over to a new data buffer with a new destination parameters. Figure 5-17 shows that when the number of bytes that are read into the MCU has reached the packet size, the entire *TxBuff* with the new packet is sent to the WizFi220. A newline character is used to indicate to the WizFi220 that the packet is complete. This process clears the flag and allows the state machine to be reset to prepare for the next packet from the WizFi220.

5.2.6 Peripheral handling and monitoring

The HERMES node contains several monitoring functions that periodically checks the state of the node. Specification 2.6.5 specifies that the node has basic monitoring functions to monitor temperature and battery voltage. Additional data is transmitted to the user, including network transfer rate and error rate and this provides a full state of the node environment and network characteristics. The external peripheral data and the internal statistical data is amalgamated into a single control packet. Table 5-3 shows the sensor data types and data field definitions.

Table 5-3: Error monitoring thresholds

Sensor type	Error threshold level	Indicator function
Battery Voltage Monitor	3 V per cell	Low battery
Temperature Monitor	70 degrees Celsius	High temperature
Data-rate Monitor	0 packets	No network connectivity
Error-rate Monitor	>10% error rate	Loss of signal

In normal operation, the node polls each peripheral and determines if the monitor has exceeded a safety limit specified in table 5-3. This occurs every 10 seconds. The raw values are presented to the user through a control packet (see section 5.3.5) that contains the node address and the node peripheral values. When the safety limit is exceeded, the node reports at a rate of 1 Hz. This indicates a serious failure, and the rapid transmission ensures that a node warning signal reaches the operator instantaneously.

```
//Send update to home base station
void SendUpdate()

    //Write the sequence to the Wizfi
    Serial1.write(esc);
    Serial1.write('Z');
    Serial1.write('1'); //send home
    //Write Packet length
    Serial1.write('0');
    Serial1.write('0');
    Serial1.write('1');
    Serial1.write('0');
    //write data. 10 bytes
    Serial1.write(1); //LENGTH 1x10 bytes
    Serial1.write(PKT_UPDATE); //send the packet index.
    Serial1.write(NODE_NUMBER); //source
    Serial1.write((char)temperature); //TODO:CHECK HOW TO CONVERT
    Serial1.write((char)batteryVoltage);
    Serial1.write(errorTraffic); //signal loss
    Serial1.write(lowTraffic); //low traffic: Link loss
    Serial1.write('0'); //reserved bytes
    Serial1.write('0');
    Serial1.write('0');

    //finish with newline
    Serial1.write(N_L);
}
```

Figure 5-18: WizFi220 transmission of node monitoring packet

Figure 5-18 shows the commands that the MCU transmits to the WizFi220 in order to make a data transmission. The MCU starts the data transmission command by using the escape sequence of the esc character followed by the ‘Z’ character as specified in the programmer’s guide [41]. The data is routed to the base station, so a CID of 1 is used, indicating transmission to the base station. The next four bytes indicate the packet length in ASCII, and this is the string 0010, indicating that the data transmission will consume 10 bytes. The next 10 bytes indicates the data that is being transmitted.

The data begins with the custom 3 byte header indicating the packet length, the control packet type, and the source of the node as specified in section 5.2.4. The following four fields are used to transmit char type data to the base station, allowing for 255 different values for the temperature and battery voltage. The traffic signals uses either char 1 or char 0 to indicate the state of the communication link. The remaining 3 bytes are left reserved and are not used, although future monitoring capabilities can

be added in these fields. The transmission command is concluded with a newline character to indicate that the data transmission has finished.

5.2.7 Endpoint design

The HADES and Base station endpoints differ from the normal nodes that are deployed into the mine. The endpoint nodes are tethered to the relevant endpoint, and hence the design of the endpoint software is simplified. On both HADES and the base station, node peripheral features are not implemented. The HADES robot features internal sensors of its own, and the base station is situated outside the mine. The peripheral circuits detecting temperature, battery voltage, and accelerations (discussed in 4.3.3 and 4.3.4) are redundant on the base station and HADES RF devices.

Due to the nature of the endpoint devices, certain simplifications can be made to the main packet handling of the WizFi220 interface. When data arrives at an endpoint, no further forwarding is involved. As such, routing functions (described in 5.2.5) are not required. Data received from the endpoint host processor is sent directly to radio. Data received from the RF interface is sent directly to the host processor.

The endpoint RF modules serve as a direct tethered extension to a host system, and hence a separate protocol is developed to interface the endpoint hardware with the host processor to forward the received RF data. The host processor operates on the Robot Operating System (ROS). ROS provides a scalable and portable software operating system framework that allows efficient integration and management of robotic devices. ROS offers a Rosserial node that implements efficient serial communication from the endpoints to the MCU.

The base station and HADES robot endpoint nodes must contain the relevant libraries to interface to the ROS system on board their respective endpoints. The base station endpoint library additions are shown in figure 5-19. Identical libraries are implemented for the endpoint nodes on the HADES system.

```
//Include statements for ROS.
#include <ros.h>
#include <std_msgs/String.h>
#include <hades_base_stn/data_packet.h>
#include <hades_base_stn/control_packet.h>
#include <hades_base_stn/sensor_packet.h>
#include <string.h>
#include <stdlib.h>
```

Figure 5-19: Declaration of ROS libraries

Figure 5-20 shows the ROS initialization step. Declarations of the variables that hold the values of the incoming and outgoing packets are performed including the setup of the publishers and subscribers. As ROS has a dedicated message system, specialized variables in memory are used to represent the individual packet types as will be defined in section 5.3.5. These variables are shown in figure 5-20.

```

//build the ROS node
ros::NodeHandle nh; //create handler of type ros::NodeHandle

//packets for publishers
hades_base_stn::data_packet incomingVideoData;
hades_base_stn::sensor_packet incomingSensorData;
hades_base_stn::control_packet incomingControlPacket;

ros::Publisher fromHADES("upstream_video",&incomingVideoData);
ros::Publisher fromHADESSensor("upstream_sensor",&incomingSensorData);
ros::Publisher fromHADESControl("upstream_control",&incomingControlPacket);

```

Figure 5-20: ROS global variables

The subscribers and publishers on the endpoint nodes map to the subscribers and publishers on the endpoint devices. Figure 5-20 shows the individual topics that the base station node publishes to. These publishers and subscribers must be initialized by advertising them to the endpoint devices, and this is performed in figure 5-21.

```

//=====IMPLEMENTATION=====
#ifdef IMPL
    nh.initNode();

    //Subscribe and advertise to the ROS connections
    nh.subscribe(fromBase); //the controller commands from the base station.
    nh.advertise(fromHADES); //the video stream to be sent to base station
    nh.advertise(fromHADESSensor); //sensor message from HADES
    nh.advertise(fromHADESControl); //control message from HADES
#endif
//=====END IMPLEMENTATION=====

```

Figure 5-21: ROS node advertisements on endpoint hardware

The main process of the endpoint nodes are modified to bridge the data incoming on the RF interface to the endpoint hardware ROS node system. When data is received on the RF interface, this data is mapped to the memory addresses on the dedicated video packet as shown in figure 5-22. This function then sends the data over USB to the endpoints.

```

if (pack_type == 'X' || pack_type == 'V' || pack_type == 'K'){
    //video packet
    //set video size
    incomingVideoData.pack_size = 50;

    //plug the data in.
    //Send the completed packet to HADES.
    for (loop_v = 3; loop_v < rxPos; ++loop_v){
        incomingVideoData.data[loop_v] = rxBuff[loop_v];
    }

    //send the data to the base station endpoint.
#ifdef IMPL
    fromHADES.publish(&incomingVideoData);
#endif

    rxPos = 0; //reset the rx position
}

```

Figure 5-22: Transmitting a packet from the RF radio to the endpoints

Figure 5-22 shows the code that is on the base station that checks the packet type and then processes the packet. The packet type uses the identifiers in the second byte of the 3 byte header described in section 5.2.2. A loop is used to copy the *rxBuff* data to the ROS buffer. The data is then published using the ROS publish function as standard in the ROS framework.

When a data packet arrives on the USB interface to be sent out to the radio, a callback function is used to send the data over the serial to be transmitted by the WizFi220. This is an interrupt based function that transmits the packet immediately after reception. This minimizes the delay of a packet having to wait in the MCU buffers. Figure 5-23 shows the callback that is used to transmit data to the WizFi220. Like the peripheral data transmission process, this callback also invokes the WizFi220 data transfer protocol.

```
void toBase_callback(const hades_com_sys::data_packet& data_pack){

    if (setup_rdy == 1){

        analogWrite(LED_POWER,0);

        transmit_length = data_pack.pack_size * 10; //get the packet
        transmit_length += 4; //4 bytes added for the length field.

        //We send the data down to the WizFi220.
        Serial1.write(esc);
        Serial1.write('2');
        Serial1.write('1');

        //loop over all of the packets and transmit the data.
        for (int i = 0; i < transmit_length; ++i){
            Serial1.write(data_pack.data[i]);
        }

        //send the newline character
        Serial1.write(N_L);

        //delay a bit to prevent overload on WizFi.
        delay(12);
    }
}
```

Figure 5-23: ROS callback for RF data transmit

Figure 5-23 shows the callback on-board the HADES robot that transmits a packet according to the WizFi220 protocol. The escape sequence is used to indicate data transfer. A loop over the entire data packet is used to pass the data into the UART transmission buffers. A newline character is transmitted at the end in order to allow the WizFi220 to transmit the data. A delay is added to the callback in order to prevent the WizFi220 from transmitting too much data back-to-back, as this can cause the WizFi220 to disconnect from the HADESNET network. The minimum delay before the WizFi220 is liable to disconnect is found to be 12 milliseconds.

5.3 ROS interface software

A critical function of the endpoint nodes is to present received data to the endpoint device, either HADES or the HADES base station. ROS Node networks are designed for both the base station and

the HADES robot. These node networks implement functions that allows the lower level WizFi220 hardware interface to the host processors on the endpoints.

5.3.1 Base Station node structure

Figure 5-24 shows the ROS node structure of the base station ROS network. This figure demonstrates the main features that allow the base station to send and receive data from the serial connection to the WizFi220 MCU. The serial connection is represented by serial_node0.

Figure 5-24: HERMES base station node network

The BASE_COM_OP_NODE shown in figure 5-24 denotes the main communication node responsible for managing the communication between the serial node and the rest of the ROS system. The *compressed_vid* topic is the compressed video stream and is discussed in chapter 6. The downstream topic contains the robot remote control instructions provided by the JOY_NODE. The *upstream_sensor* topic processes the incoming sensor data and the host node sends this sensor data to the user. The JOY_NODE implements an XBOX controller to control HADES, and this is discussed in section 5.3.4. Figure 5-25 shows the launch file, which launches the nodes in the configuration as shown in figure 5-24. This file boots the Rosserial node to 1 Mbaud and serves as the data-link between the hardware and the endpoint processor.

```

<launch>
  <node pkg="joy" type="joy_node" name="joy_node"/>
  <node pkg="hades_base_stn" type="BASE_COM_OP_NODE" name="BASE_COM_OP_NODE" output="screen"/>
  <node pkg="roscpp" type="serial_node.py" args="_port:=/dev/ttyACM0 _baud:=1000000" name="serial_node0"
respawn="true" output="screen"/>
  <!--Launch the video processing node-->
  <node pkg="hades_video_proc_base" type="HADES_VID_PROC_NODE_BASE" name="HADES_VID_PROC_NODE_BASE" output="screen"/>
  <node pkg="image_view" type="image_view" name="image_view" args="_image:=/HADES_video_stream" output="screen"
respawn="true">
    <remap from="image" to="HADES_video_stream"/>
  </node>
</launch>

```

Figure 5-25: HERMES launch file

5.3.2 HADES node structure

Figure 5-26 shows the node structure of the HADES ROS network. The COM_OP_NODE provides coordinating and management functions that encapsulates the data incoming from the video system and sends this data to the serial node.

Figure 5-26: HADES robot node network

The serial connection is represented by SERIAL_NODE0, connecting to the low-level HERMES hardware. The COM_OP_NODE denotes the main communication node responsible for managing the communication between the serial node and the rest of the ROS system. The COM_OP_NODE only processes headers and delimits the datastream into packets to send to the RF hardware. The realsense/realsense_stream acquires the video feeds and sends them to the processing block. Video processing functions and acquisition is conducted using the HADES_VID_PROC_NODE and this is discussed in chapter 6.

Figure 5-27 shows the way that the `hades_com_sys` callbacks function. The `hades_com_sys` features implementations for processing incoming data and provides call-back for publishing data to other nodes. The callback function for processing incoming sensor data and other processing is left blank to allow for future implementation.

```
//Callback for sensor data
void COM_OPERATOR::incomingSensorData(const hades_com_sys::sensor_packet input_data){
    //TODO: FOR LATER IMPLEMENTATION: FILL OUT int_sensor_pack and ext_sensor_pack.
    //NOT IMPLEMENTED: get sensor data and fill out the sensor data.
}

//data is coming out from HADES to base station
void COM_OPERATOR::incomingCallBack(const hades_com_sys::control_packet input_data){
    //mirror this packet to the video processing and the control system. Publish to upstream control.
    //Any node that wishes to see the ocmmand controls will need to subscribe to upstream_control.
    command_stream_pub.publish(input_data);

    //ROS_INFO("END");
}

//callback when image comes in.
void COM_OPERATOR::videoCallback(const hades_com_sys::data_packet data_pack){
    data_stream_pub.publish(data_pack);

    //ROS_INFO("START");
}
```

Figure 5-27: Internal data bridge callbacks

Figure 5-28 shows the launch file that boots each node in the ROS `hades` node structure. This figure shows the `hades_com_sys` node with the `Rosserial` node being booted. A second node, the `hades_video_proc` node is booted to process the incoming video. This launch file also defines video statistics such as frame-rate, size of the video frame, and provides the name of the camera.

```

<launch>
  <node pkg="hades_com_sys" type="COM_OP_NODE" name="COM_OP_NODE" output="screen"/>
  <node pkg="roserial_python" type="serial_node.py" args="_port:=/dev/ttyACM0 _baud:=1000000" name="serial_node0"
  respawn="true" output="screen"/>
  <include file="$(find video_stream_opencv)/launch/camera.launch" >
    <arg name="camera_name" value="realsense" />
    <arg name="video_stream_provider" value="0" />
    <arg name="fps" value="12" />
    <arg name="frame_id" value="realsense" />
    <arg name="camera_info_url" value="" />
    <arg name="flip_horizontal" value="false" />
    <arg name="flip_vertical" value="false" />
    <arg name="width" value = "320"/>
    <arg name="height" value = "240" />
    <arg name="visualize" value="true" />
  </include>
  <!--Launch the video processing node-->
  <node pkg="hades_video_proc" type="HADES_VID_PROC_NODE" name="HADES_VID_PROC_NODE" output="screen"/>
</launch>

```

Figure 5-28: HADES ROS node launch file

5.3.4 Control input

A vital function of the constructed HERMES network is to allow HADES to be tele-operated by the base station operator. HADES is controlled via a user-friendly XBOX 360 gamepad as shown in figure 5-29. This device was selected by Molyneaux as an operator-friendly method of teleoperation. This device is easily supported as a generic joystick device and is ubiquitous, non-expensive, and easily interfaces with the Linux operating system through bespoke libraries. ROS also provides easy interfacing with this device through the joystick message. The control scheme is shown in table 5-4, with multiple buttons mapped to the packet contents as shown in table 5-4.

Figure 5-29: XBOX360 controller

The packet generated by the control input is gated at a rate of 10 Hz, meaning each control has a 100 ms period. The packet generated by the control input has the form of the control packet (see section 5.3.5) of length 10 bytes. This minimal length packet minimizes the data that is transmitted down the channel, maximizing bandwidth for other vital functions such as video.

Table 5-4: Controller bindings

Button	Type	Function
Cross-pad up-down	Digital	Drive/Reverse
Cross-pad left-Right	Digital	Pivot left/pivot Right
LB	Digital	Reduce speed
RB	Digital	Increase speed
A	Digital	Brake
X	Digital	Toggle Camera
LT	Analog	Smooth Tilt left
RT	Analog	Smooth Tilt Right
Right Analog Stick	Analog – up-down direction	Camera threshold (see 5.4)
Back	Digital	Stop transmitting control signals – disable remote control
Start	Digital	Start transmitting control signals – enable remote control

The control input data protocol consists of several commands delimited into 8-byte lengths. The control data packet format generated by the XBOX controller is shown in table 5-5. This table shows the packet contents as well as reserved fields that can be implemented in the future. The XBOX controller also contains unused buttons that can be used in future implementations of the control scheme. The packet shown in table 5-5 utilizes the custom 3 byte header as discussed in section 5.2.4. The remainder of the packet contains basic drive commands for the HADES robot.

Table 5-5: Teleoperation control packet format

Byte	Byte Data	Default Setting	Meaning
0	1		Char 1 x 10 = 10 bytes – Length header of message
1	'C'		Character "C", meaning control packet
2	0	0	Transmission control type (0 for do not retransmit)
3	0-200	100	Drive Direction
4	0-200	100	Tilt
5	0-200	100	Turn Strength
6	1-4	1	Camera Feed Source
7	0	0	Reserved
8	0	0	Reserved
9	0-255	128	Camera Threshold (see section 5.4)

5.3.5 Data packet protocol

A dedicated protocol for the handling of data transfers within the ROS interface node system is implemented using the ROS message system. ROS uses the message system to pass information from node to node. The message system is facilitated by subscribers and publishers, where nodes advertise and subscribe to desired data messages. Three data message types are defined for the transmission of data between the ROS nodes and the W220 hardware that fulfils the specifications stated in section 2.6.4

Data_packet message

The data_packet message represents the largest amount of data that can be transmitted over the HERMES network. The data_packet message is used for transmission of video data from HADES to the base station. The packet size of 500 bytes is chosen due to the requirement for the data length to be a multiple of 10 bytes as discussed in section 5.2.4. Since the data delay is proportional to its data length as shown in section 7.4.2, a data packet size of 500 bytes is chosen. The fields in the data_packet message are shown in table 5-6.

Table 5-6: Data packet message

Field	Data Type	Meaning
Length	Int32	Length of data in 32 bit form.
Data	uint8_t[500]	500 bytes of 8-bit data

For the purposes of reception by the WizFi220 module and processing by the endpoint MCU, the data_packet intended for video has three packet identifiers when the 3 byte header is included as discussed in section 5.2.4. These are discussed in section 5.4.4.

The data_packet message is used exclusively for the transmission of video data. However, the data_packet message does not contain any data-structures or is designed in any way that prevents its use for transmitting other forms of bulk data. The data_packet message is left generic in order to allow it to perform functionality that may be implemented for HADES in the future, such as transmitting sound files for voice communication or point-cloud 3D data.

Sensor_packet message

The sensor_packet message represents the sensor data that HADES transmits back to the base station. The sensor_packet message is a 60-byte message with 10 sensor-value mapping pairs. The sensor packet is therefore extendable and can describe all sensor values and sensor types. The sensor packet provides 32-bit integers for fine grained resolution of the values and provides 8-bit values for the sensor types, allowing for 254 different sensor reading types. In addition to this, a 9-byte reserved data field allows transmission of other HADES robot parameters. This data field can be used to

indicate urgent status if thresholds are too high or to provide other indicators. The fields in the sensor message are shown in table 5-7.

Table 5-7: Sensor packet message

Field	Data Type	Meaning
Length	int8	Length of packet: always 6 to indicate a 60 byte packet.
Reserved	Int8[9]	Reserved bytes for miscellaneous use.
sensor_type	Int8[10]	10 values of sensor type. Sensor_type[0] has reading in data[0]
data	Int32[10]	10 values of 32 bit integer data.

The sensor packet is intended for use by both the internal environment sensors as well as the external environment sensors. By providing fields for both data and sensor type, the sensor packet allows a greater range of sensing capabilities than with a packet with delimited hard-coded fields. The use of packet sensor identifiers and name-value pairs allows the sensor packet to be extendable and accommodate all sensor readings with just one generalized message.

Control_packet message

The control_packet message provides a generic method for the base station and HADES to communicate with one another. The control_packet message is a 10-byte message with 7 bytes of free message space after applying the 3 byte header. The minimal length of the control_packet allows the endpoint entities to communicate without significantly impeding the performance of the primary video feed. The 10-byte message minimizes the delay incurred at each hop. Section 7.6.4 shows that a control packet message is processed in the MCU at a speed of 0.8 ms per node. This results in a total control message delay of roughly 170 ms over 40 hops. The packet content format is shown in table 5-8.

Table 5-8: Control packet message

Field	Data Type	Meaning
data	Int8[10]	Control packet contents, 10 8 bit fields.

The control packet is designed to perform any miscellaneous communication that can be achieved using 7 bytes. This packet can be used by both HADES and the base station to transmit simple text data.

5.4 Summary

This chapter detailed the software of the HERMES wireless node and associated software on-board the endpoint devices. Two major topics were discussed in this section. The first was the software

operating system on-board the HERMES wireless node. This explored topics of data routing, network management, and peripheral management. The second was the ROS endpoints and the ROS node network that provided data-bridging functions between the low-level hardware and the high-level user application. Important topics such as data format and user control were discussed. Extensible interfaces and control schemes were provided for future development and discusses the video acquisition and processing system. Chapter 6 explores a basic compression algorithm used to evaluate low data-rate video performance on the HERMES network.

Chapter 6 HERMES System Video Processing

This section describes the video acquisition, processing, and compression required to perform video transfer over the HERMES network. Section 6.1 describes the RealSense hardware and configuration. Section 6.2 describes the software pre-processing and algorithms employed in the video compression techniques. Section 6.3 describes the video system and the strategy for reducing the video to a lower quality to allow an acceptable frame-rate suitable for driving. A bespoke decimation in time compression scheme is detailed in section 6.5 to attain a minimum quality video stream. Section 6.6 summarizes this chapter.

6.1 Video acquisition hardware

The primary camera used in the forward-facing sensor suite on HADES is the RealSense SR300 camera as pictured in figure 6-1. The colour camera features a maximum 1920 x 1080 active pixels with a 16:9 aspect ratio. This camera is capable full colour with a frame-rate of up to 60 FPS [47]. This device is interfaced to the NUC endpoint using a USB 3.0 cable.

Figure 6-1: RealSense SR300

The video acquisition technique discussed in this section for the front-facing camera is also appropriate for any of the side facing cameras. This section primarily discusses the camera feature and not the depth cloud or IR camera feature as the video feature is the primary function required for teleoperation. The RealSense camera interfaces with the ROS operating environment through the `/realsense/realsense_camera` node. This node is shown in section 5.3.2. The camera stream outputs raw image messages of type `/msg/image_raw`. Each message represents a single image frame.

6.1.1 Naïve video transmission

For the implementation of the video system on HADES, a standard 4:3 aspect ratio with resolution 320 x 240 pixels is selected with the lowest possible video frame-rate of 10 frames per second. The raw data output by the RealSense is in bgr8 format. This format describes each pixel as a set of three colour hues, red, green and blue. Each hue is described by an 8 bit character, so each pixel is represented by 3 bytes. A full colour stream is acquired from the Realsense camera as the full quality

image can then be stored onto the hard drive disk. The raw image frame costs just under 2 Mb per frame as shown in equation 6.1.

$$frame\ size = 320 \times 240 \times 8 \frac{bits}{byte} \times 3\ bytes = 1.84\ Mbits \quad (6.1)$$

At the desired data rate of 10 frames per second, the associated data rate of just one frame is well over the rate that is achievable by the HERMES system as shown in equation 6.2.

$$1.84\ Mbps \times 10\ fps = 18.4\ Mbps \quad (6.2)$$

Chapter 7 shows that the maximum effective data-rate of the network is a maximum of 320 kbps. In order to ensure that a constant data-rate is maintained on the network, the maximum video rate that can be carried is set to 160 kbps, exactly half of the maximum bandwidth. This allows a 50% margin on the data transfer in case of degraded signal performance. The video rate shown in equation 6.2 vastly exceeds the 160 kbps link.

To decrease the amount of data being transmitted, it may be acceptable to transmit the messages in just monochrome. Under these conditions, if the monochrome is 8 bits then the savings attained is linear by a factor of three as the three colour channels are amalgamated into one byte. The cost of each frame falls threefold as shown in equation 6.3.

$$frame_{mono} = 320 \times 240 \times 8 = 614.4\ kb\ per\ frame \quad (6.3)$$

The decrease shown in equation 6.3 still does not allow the video data to be transmitted at a high enough rate. Even reducing the information to 1 bit per pixel does not provide sufficient decreases in packet size to provide a data-rate of 10 Hz. With unacceptable 1 bit (black or white) per pixel, each frame will cost 76.8 kb at 10 frames per second, as shown in equation 6.4.

$$10\ fps \times 76.8\ kb = 768\ kbps = 0.77\ Mbps\ at\ 10\ frames\ per\ second \quad (6.4)$$

The attempt to reduce the resolution by simply reducing the amount of information per pixel as shown in equation 6.4 does not reach the desired 160 kbps data-rate that is achieved by the HERMES network. Transmitting each frame in this way would attain a framerate of under 3 frames per second with 1 bit per pixel. This frame-rate is insufficient to allow near-real-time driving video. It is also unclear whether 1 bit per pixel provides an image quality that is useful for a base station operator. Video compression is therefore necessary in order to further reduce the data-rate.

6.3 Video transfer strategy

The HADES camera feed provides two main functions to HADES' operation:

1. The HADES camera feed can provide video to allow the base station to control HADES in near-real time.
2. The HADES camera feed can assess the situation within the mine and can identify physical conditions inside the mine.

These two functions necessitate the inclusion of the camera feed into HADES. While both functions require the video feed, there exist significant differences between these two principal functions. The driving video feed (1) provides the user with driving-quality video. The literature in section 2.4.2 shows that the minimum frame-rate for controllability is 10 Hz. Low image quality is acceptable as long as HADES can be driven accurately with a fast response time. As long as the robot can be navigated around obstacles, exact features in the mine are relatively unimportant. In this case, spatial resolution can be traded for temporal resolution. The opposite is true with the assessment video feed (2). In order to pick out features within the mine and its conditions, it is likely that the operator will likely first stop HADES. A moving object is more difficult to observe, so it is natural to halt HADES to observe a static landscape. If HADES is stopped, then the frame-rate becomes less important as the image will not be changing. It is less important for the same static image to be returned at a high frame rate. In this case, it is acceptable to reduce frame-rate in exchange for higher image quality.

6.3.1 Still Images for Observation

It is clear from equation 6.1 that quality colour video data is still too difficult to attain. Transmission of videos of higher resolution than 320 x 240 pixels would still result in a significantly long waiting time. It is assumed that 5 or 6 seconds is the maximum waiting time that would be tolerated by an operator. The still observation frames will feature an identical resolution of 320 x 240 using 8-bit monochrome. The raw image data is reduced down to monochrome 8-bits per pixel. As shown in equation 6.5, the number of bits per frame will be roughly 614 kbits.

$$320 \times 240 \times 8 \frac{\text{bits}}{\text{pixel}} = 614400 \text{ bits} \quad (6.5)$$

$$\frac{614,400 \text{ bits}}{160,000 \text{ bits/second}} = 3.84 \text{ seconds} \quad (6.6)$$

The data transfer rate is 614 kbps and hence the entire image can be transmitted in roughly 3.84 seconds as shown in equation 6.6. Still image transmission is achieved by splitting the raw video data into packets 500 bytes long. No transmission post-processing or pre-processing is required, and no receiver processing is required.

6.3.2 Moving Video for Driving

The greatest challenge presented by the available application level data-rate is the division of the available data rate to provide video. The target driving data-rate is given to be at least 10 FPS, as specified in section 2.6.2. At this frame-rate, each frame has a delay of up to 100 ms. With a given data-rate of 160 kbps, and transmitting at a constant bitrate, the data limit per packet can be calculated as shown in equation 6.7.

$$packetSize_{max} = \frac{160 \text{ kbit}}{10 \text{ frames}} = 16,000 \text{ bits per frame} = 2000 \text{ bytes per frame} \quad (6.7)$$

6.3.3 Compression Analysis

A poll of existing compression algorithms conducted in section 2.4.3 showed that modern compression algorithms commonly employed are not designed to function at low-data rates. The 160 kbps limit shown in 6.3.2 shows that the required compressed stream rate needed 2000 bytes per frame on average for a frame-rate of 10 frames per second. A bespoke compression for the HERMES system that matches the requirements stated above is desired. In particular, the HERMES system operates in a closed underground mine environment with low-lighting conditions and is limited to the aforementioned 10 frames per second with a frame size of 320 x 240 pixels. It is advantageous to build a compression system specifically to deal with these conditions as opposed to using a generalized video compression scheme. As discussed earlier in section 6.3, a video transfer strategy can be applied to increase quality when HADES is stationary and make frame-rate unimportant. This compression will only be required for driving video where frame-rate with quality is required. A basic compression scheme is desired as it will be useful for testing, investigating and implementing a bespoke video compression system. This system also allows preliminary testing of the video transmission hardware in conjunction with the ROS software.

1 bit video is investigated as this is the minimum quality that can possibly be attained by the video system. This encoding represents each pixel as either black or white, with no intermediary shades. This 1 bit video system uses the silhouette effect to allow a user to infer spatial information after the video is displayed. By using a silhouette of the internal conditions, an operator can use the limited information provided by the video feed to reproduce the spatial information that is missing from the 1 bit conversion. This effect leverages the user intelligence and intuition by allowing the human operator to piece together the specifics of the image using cues from visual field.

As this process relies on the relative brightness and darkness of the internal tunnel conditions of the mine, an adjustable threshold is implemented to aid the user with the image reconstruction. A moving threshold allows a user to change the value of the threshold that determines which pixels are light and dark. The conditions in the mine and the optical brightness of the interior can be changing. It is useful

for the user to be able adjust the threshold in response to the changing mine conditions to improve the resultant image. This can be used to allow a user to emphasize light or darker objects as required.

Run length encoding is used as a method of encoding 1-bit frames. As pixels are either value 0 (for black) or 255 (for white) there is significant redundancy post threshold encoding. For example, expressing 10 pixels of white and 5 pixels of black can be done naively in an array of values as shown in figure 6-2.

1,1,1,1,1,1,1,1,1,0,0,0,0,0 ...

Figure 6-2: Basic Encoding

This encoding costs 15 bytes in order to express 15 pixels. This encodes each pixel as a byte. As 1-bit encoding requires two states, a better approach would be to encode the data one bit at a time, allowing each byte to represent 8 bits. However, this does not fulfil the requirement that each byte needs to represent 40 bytes. Run length encoding is used to encode the length of a string as opposed to representing each byte in memory. This run-length encoding scheme is shown in figure 6-3.

10,5

Figure 6-3: Run length encoding

The run length encoding reduces each data byte to a particular count of consecutive pixels. In this way the 15 pixel count is represented by 2 bytes.

6.4 Single-Threshold 1-bit Run length encoding

A 1-bit RLE video compression process is implemented and the result is shown in figure 6-2. Appendix A.1 presents the full implementation details and analysis of this system's performance.

Figure 6-4: 1-bit RLE encoding

Figure 6-4 shows that while the general field of view can be resolved, individual features are too obscured. The image quality is too low for effective resolution of most objects. The 1-bit RLE causes

a loss of spatial awareness as shadowing, gradients, and distance information is lost in the 1-bit thresholding process. The poor image quality demonstrated in this initial video compression motivates the need for a compression scheme with a higher spatial resolution. Similarly coloured objects are difficult to distinguish from their neighbours as shown in figure 6-4 on the pile of boxes. The objects that are identifiable are generally because of large luminescence changes or because dark objects are set against bright objects. The overall shape of the table and drawers are shown and this is clearly resolved. The general shape of the field of view is revealed with reasonable quality, but some features are unable to be revealed.

The 1-bit RLE system is not very loss-tolerant. The sequential encoding scheme has a weak point in the starting reference packet. The frame is lost if this initial packet is lost and hence this is a critical point of failure in a wireless environment that is potentially lossy. A significant amount of delay was observed when the packets were received by the decoder process. This was due to the need for all packets to be received for a frame to be shown. The consecutive frame delay severely increased the delay of the system. All of the frames needed to be received by the receiver. The receiver process was waiting for 4 to 6 packet times before a new frame could be displayed to the user.

6.5 Decimation in time compression

Section 6.4 discussed a basic video compression strategy with serious drawbacks that limits the effectiveness of the resultant video stream. Serious flaws of the 1-bit RLE system include the poor visual quality, the waiting delay for all packets to be received by the decoder process, and a poor loss tolerance. A decimation in time (DIT) scheme is designed and implemented as a compression algorithm that extends the 1-bit RLE algorithm to improve on the aforementioned factors. The DIT method block diagram is shown in figure 6-5. This system uses the run-length encoding with different thresholds to decimate the frame and encode four sub-frames with a different threshold to attain more colour information.

Figure 6-5: DIT compression block diagram

The video acquisition method and monochrome conversion is shown in figure 6-5. The monochrome filter step also applies a low-pass filter to the image that allows the image to be decimated without aliasing (shown in appendix A.1.1). The DIT lossless frame decimation process reduces each frame to four smaller frames called sub-frames. The process for the frame decimation involves finding every second pixel in every second row and moving the data to four separate arrays containing the frame information. The process to achieve this is demonstrated in figure 6-6. Each of the 76800 pixel positions are then moved to four arrays of size 19200. These four arrays of data (the sub-frames) are then encoded and transmitted separately to be reconstructed by the decoder.

Figure 6-6: DIT Frame decimation scheme

The sub-framing and time decimation procedure provides three major advantages over the standard sequential 1-bit RLE encoding scheme. By separating each frame into four sub-frames, the video quality can be improved. Each sub-frame can be encoded using run length encoding with differing

thresholds, allowing greater image resolution. As each sub-frame contains information about the entire image, a loss of a single sub-frame does not constitute the loss of the entire frame. Because each sub-frame now holds information about the entire image, a lost frame spreads the entire lost-contents across the entire frame. Finally, a sub-frame can be decoded immediately into the receiver image as the decoding of a sub-frame does not rely on the reception of other sub-frames. As the 1-bit RLE is a sequential process, a single frame has a dependence on the starting packet and subsequent previous packets for a full frame to be decoded on the receiver. Delay characteristics are improved over the 1-bit RLE system as the system does not need to wait for all sub-frames of a given frame to be received before refreshing the image. A further advantage of this is that as the sub-frames contain a full down-sampled image, the interleaved images can be refreshed four times as quickly, raising the refresh-rate of the compression scheme to 40 Hz.

Exact implementation details of the DIT scheme are provided in appendix A.2.

6.6 Summary

This chapter detailed the video acquisition software and hardware for HADES teleoperation. This chapter motivated the need for compression and the means to achieve this was discussed. A bespoke DIT compression scheme allows the frame to be sent in sub-frames with multiple thresholds. This allows for an interleaved video stream of 10 frames per second at 160 kbps with a refresh rate of 40 Hz. Chapter 7 discusses the full system testing and evaluation of the designed HERMES wireless node. Chapter 7 also evaluates resultant video stream for teleoperation and improvements are suggested over this basic video compression method. Improvements for future work are suggested with respect to the results in this chapter.

Chapter 7 HERMES Node evaluation

This section details the processes and evaluation techniques used to determine the performance of the HERMES nodes. The physical environment for HERMES node testing is described in section 7.1. This details the experimental setup, methods and locations used for experimentation. Section 7.2 describes the performance metrics and the software environment used for logging of data.

The results of the experimentation is discussed in sections 7.3 through 7.5, with static characteristics such as power supply and packet delay examined in this section. Section 7.6 discusses the performance of the HERMES wireless nodes. Section 7.7 summarises the results presented in this chapter, including performance evaluations of the node and the limitations of the designed HERMES system.

7.1 Evaluation environment

7.1.1 Testing environment 1: Table tests

Preliminary testing is conducted by placing nodes within close proximity to one another, shown in figure 7-1. The antennas are connected through U.FL to SMA connectors attached to inverted dipole antennas on the bottom of the photo. This environment provides a convenient location to perform signal and basic functionality checks. For the purposes of testing, the nodes are powered by an adjustable constant voltage power supply. The power supply mimics the battery power and hence the constant voltage is set to 12 V to power the nodes through the internal regulators.

Figure 7-1: HERMES table test

7.1.2 Testing environment 2: Cotton Corridors

The Victoria University Cotton Corridors provide a hallway environment that is long and wide. The Cotton corridors have a size of 12 m wide by 10 m high in cross-section, with various obstacles occasionally impeding onto the hallway floor. These corridors have vinyl floors with reinforced

concrete support beams on the ceiling with overhung light mounts. Glass panes line some areas of the length of the hallway as shown in figure 7-2. The length of this full corridor is roughly 125 m from the end of the Laby building to the end of the Cotton building. This testing environment has WiFi devices operating on the 2.4 GHz band, and therefore some RF interference is observed during testing in this environment. This environment provides an indicative performance of the HERMES nodes in a tunnel environment. Ground level line of sight is guaranteed in empty hallway conditions. Figure 7-2 shows the environment from the halfway point.

Figure 7-2: Cotton corridors

7.1.3 Testing environment 3: Wright's Hill Fortress

The Wright's Hill fortress tunnels is used as a testing environment for replication of a tunnel environment. The Wright's hill fortress tunnels feature an elliptical cross section size of 2 m wide and 3 m tall. The tunnel walls are constructed of brick and concrete composite. The floor is concrete with steel plate inserts covering the drainage tunnels. The length of the main tunnel is around 120 m with the angled tunnel measuring 50 m. A 30 degree and 90 degree turn is provided by these tunnels. This is as shown in figure 7-3 beside marker B (for 90 degrees) and the angle past marker B (for 30 degrees). Due to being underground, 2.4 GHz interference is not present. Range experiments were conducted in the straight areas between A and B and this distance was 150 m.

Figure 7-3: Area of Wright's Hill tunnel testing area.

The Wright's hill fortress tunnels are straight, but some irregularities persist due to the uneven ground. The concrete pour is uneven with some sections of ground raised and angled around 5 cm in some places. The HERMES nodes are powered from battery power as designed. An image (taken by Thompson [35]) of the Wright's Hill tunnel interior is shown in figure 7-4.

Figure 7-4: Wright's Hill tunnel interior

7.2 HERMES performance metrics and software testing environment

Figure 7-5 shows a debug terminal with vital statistics of the packet being received using debug code. C compiler directives are used to demarcate core node functionality from debug functionality.

```

TeensyMonitor: COM19 Online
Send
DELAY: 10 DEAD time: 7 ERROR: 0 DRP/DUP: G Packets. ID: 1 PACK_SIZE: 450 FWD_TO: 1
DELAY: 9 DEAD time: 7 ERROR: 0 DRP/DUP: H Packets. ID: 1 PACK_SIZE: 450 FWD_TO: 1
DELAY: 10 DEAD time: 7 ERROR: 0 DRP/DUP: I Packets. ID: 1 PACK_SIZE: 450 FWD_TO: 1
DELAY: 9 DEAD time: 6 ERROR: 0 DRP/DUP: J Packets. ID: 1 PACK_SIZE: 450 FWD_TO: 1
DELAY: 9 DEAD time: 8 ERROR: 0 DRP/DUP: K Packets. ID: 1 PACK_SIZE: 450 FWD_TO: 1
DELAY: 10 DEAD time: 6 ERROR: 0 DRP/DUP: L Packets. ID: 1 PACK_SIZE: 450 FWD_TO: 1
DELAY: 9 DEAD time: 7 ERROR: 0 DRP/DUP: M Packets. ID: 1 PACK_SIZE: 450 FWD_TO: 1
DELAY: 9 DEAD time: 7 ERROR: 0 DRP/DUP: N Packets. ID: 1 PACK_SIZE: 450 FWD_TO: 1
DELAY: 9 DEAD time: 7 ERROR: 0 DRP/DUP: O Packets. ID: 1 PACK_SIZE: 450 FWD_TO: 1
DELAY: 9 DEAD time: 8 ERROR: 0 DRP/DUP: P Packets. ID: 1 PACK_SIZE: 450 FWD_TO: 1
DELAY: 9 DEAD time: 7 ERROR: 0 DRP/DUP: Q Packets. ID: 1 PACK_SIZE: 450 FWD_TO: 1
DELAY: 9 DEAD time: 7 ERROR: 0 DRP/DUP: R Packets. ID: 1 PACK_SIZE: 450 FWD_TO: 1
DELAY: 9 DEAD time: 7 ERROR: 0 DRP/DUP: S Packets. ID: 1 PACK_SIZE: 450 FWD_TO: 1
Autoscroll Both NL & CR 1000000 baud Clear output

```

Figure 7-5: HERMES output debug terminal

Figure 7-5 shows the output of the debugging process. Fields relevant for traffic analysis are discussed below.

- **Delay:** The delay field shows the amount of time (in milliseconds) the MCU has spent processing the packet from beginning of serial reception to end of serial transmission.
- **Dead Time:** The dead time shows the amount of time the MCU spends in idle mode. During this time, the WizFi220 serial buffer is idle.
- **Error:** If an error in the packet is detected in the packet contents, this field will be 1.

- **DRP/DUP:** The drop/duplicate field detects occurrences of dropped or duplicated packets. When a packet is dropped, a skipped character in this field is observed. For example, the packet stream; A, B...D indicates that the C packet was dropped.
- **ID:** The ID is simply the 3rd byte and this can be changed by the transmitter. Each test location can be enumerated with a specific ID.
- **Pack_Size:** This is the packet size in bytes.

7.2.1 Inter-arrival delay

The inter-arrival time can be computed using the data returned by the delay characteristics as described in the debug digest. The inter-arrival time is given by equation 7.1.

$$Interarrival = DELAY + DEAD TIME \quad (7.1)$$

The method of calculating instantaneous data rate is the packet size divided by the inter-arrival time, equation 7.2.

$$R = \frac{Packet\ Size \times 8}{Interarrival} \dots \left(in \frac{Bits}{second} \right) \quad (7.2)$$

7.2.2 Raw average data rate

The average data rate of a network link is given by equation 7.3.

$$R = \frac{D}{t} \quad (7.3)$$

R is the aggregate data rate. D is the amount of data transferred over the network over the specified time, t (seconds). If D has units of binary bits, then R has units of bits per second or bps. The data rate of the link determines the raw amount of data that can be transferred over the network in a given time. The raw average data-rate can be evaluated by counting the bytes that are received to find D , the overall data transmitted. The time, t can then be evaluated by summing all of the inter-arrival times in order to find the time elapsed. The data rate is then the amount of data received in total divided by the full time elapsed.

7.2.3 End-to-end delay

The end to end delay is given by the delay of the entire system, the time spent for data to leave ROS on HADES to the time it is received by ROS on the base station. The end to end delay is given by equation 7.4.

$$Ave_delay_{ete} = PROC_{HADES} + PROC_{Base} + N_{nodes} \times PROC_{Node} \quad (7.4)$$

$$PROC_{Node} = PROC_{WIFI} + PROC_{MCU}$$

$$RTT = 2 \times Ave_delay_{etc} \quad (7.5)$$

The average end to end delay is the sum of the processing time incurred by HADES (preparing the packet) and the base station (decoding the packet) and the sum of all the nodes' processing time, where N_nodes is the current number of nodes deployed. The processing time of each node is effectively the sum of the processing time of the WizFi220 and the MCU. The round trip time (RTT) is given by twice the end to end delay as shown in equation 7.5.

It is clear from the above equation 7.4 that as HADES ventures deeper into the mine, the number of nodes deployed will increase, and each new node will incur an additional delay penalty. As HADES progresses through the mine, HADES' delay increases.

7.2.4 Error rate

When the transmitter transmits packets at a specified rate, packets may be lost due to network conditions. Errors can cause a packet to be corrupted mid-flight, causing a failed CRC or checksum on the radio hardware. When a packet is corrupted, the WizFi220 will not report the data present in the lost packet. The receiver MCU will not detect this packet, resulting in a gap in traffic. This results in the receiver observing a reduced data rate. Error rate is thus calculated as a ratio of the received data-rate to the transmitted data-rate as shown in equation 7.6.

$$Error\ Rate(\%) = \left(1 - \frac{R_{Recieved}}{R_{Transmitted}}\right) \times 100 \quad (7.6)$$

$$Conditions: R_{Recieved} \leq R_{Transmitted}$$

7.3 HERMES node static characteristics

Discussed in this section is the performance of the HERMES node in a two node configuration. This section evaluates the performance of the nodes' range, data-rate and delay performance. This section also discusses the battery power draw characteristics as well as the performance of the peripheral functions as discussed in sections 4.3.3.

7.3.1 WizFi220 delay

The WizFi220 internal delay time can be estimated using the ping utility. A packet travelling under the ping utility will utilize the layer 4 TCP protocol on IPv4 when transmitted over the ad-hoc network. This packet travels through the OSI stack to TCP using the ICMP protocol. Since UDP packets also reach layer 4, the time the packet spends in the WizFi220 unit can be estimated. 1000 ping packets are transmitted from a transmitter/receiver setup on the table tests described in section 7.1.1, and the RTT time is noted and averaged. The result shows that the internal processing time on

the WizFi220 averages 3.44 ms as shown in equation 7.7, with a sample standard deviation of 0.83 ms over this sample.

$$Proc_{WizFi} = 3.44 \pm 0.83 \text{ ms} \quad (7.7)$$

7.3.2 MCU processing delay

An experiment is conducted to determine how the MCU delay varies with data packet length. Packets of varying lengths are transmitted from a transmit/receive pair at -16 dBm with antennas placed 10 cm apart. 1500 packets captured for a varying packet length of 200 to 600 bytes, each with identical payload of repeating 0-9 characters. Nine tests are conducted in byte increments of 50 bytes to determine the processing performance of the Cortex M4 MCU. The minimum timing resolution on the MCU is 1 ms.

Figure 7-6: MCU delay vs. packet size

Figure 7-6 shows an increasing processing load as the packet size increased. The average processing time at the largest size of 600 bytes was 12 ms. Longer packets translate to larger processing times, and therefore it is advantageous to find an optimum trade-off between packet size and processing delay. For further tests, a packet size of 500 bytes is chosen, with an average 10.45 ms processing time.

7.3.3 Combined node delay

The delay each hop incurs is the sum of the MCU and WizFi220 delay as shown in equation 7.4. These delays were found in section 7.4.1 and 7.4.2. The total average per-hop delay of the HERMES node given a 500 byte packet size is shown in equation 7.8.

$$\mu Proc_{Node} = 10.45 \text{ ms} + 3.44 \text{ ms} = 13.89 \text{ ms} \quad (7.8)$$

As the WizFi delay and the MCU delay are independent from one another, (the WizFi220 delay does not impact on the MCU delay and vice versa) the overall processing variances are summed to determine overall node processing time variance as shown in equation 7.9.

$$\sigma_{Proc_{Node}} = 1 + 0.83 = 1.83 \text{ ms} \quad (7.9)$$

Overall, the processing time incurred by the node is 13.89 ms with a standard deviation of 1.83 ms. The total average node delay is shown in equation 7.10 to be just under 555 ms.

$$40 \times 13.89 \text{ ms} = 555.60 \text{ ms} \quad (7.10)$$

7.3.4 Power supply monitor

A static table test is established in a two node configuration with one transmitter node and one receiver node. Both units are placed on battery power. The voltage readings of the battery in this section are given by the battery voltage on the transmitter. As transmission costs more current than reception, this experiment examines the case where a node is in the transmit state 100% of the time. This scenario represents the absolute worst case current draw over 8 hours.

The transmitter is tested with both the 3 cell solution and the 2 cell solution. The transmitter transmits at a maximum 300 kbps rate with 400 byte packets continuously over 10 hours. The battery voltage is observed over time using the on-board ADC on the battery pack with 2000 samples. The battery voltage is extrapolated from the ADC readings by applying a mapping as shown in equation 7.11.

$$V_{bat} = \frac{ADC_{out} \times 3.3 \text{ V}}{1024} \times \left(\frac{1.15}{0.223} \right) + 0.404 \text{ V} \quad (7.11)$$

Equation 7.11 evaluates the battery voltage as a function of the ADC reading. A 10 bit ADC has 1024 different voltage levels which is then transformed to a voltage. A linear scaling factor is used to undo the voltage divider to obtain a voltage between 0 V and 3.3 V. A small 0.4 V voltage bias is added to compensate for slight resistance deviations in the resistor chain. The results of this experiment is shown in figure 7-7.

Figure 7-7: Battery monitor characteristic.

Figure 7-7 shows the node monitor readings with respect to the true voltage values as shown on the trend lines. The top set of data points indicates the result for battery solution 1, a 3-cell solution with a maximum voltage of 12.4 V, and the bottom set of points indicates the result for battery solution 2, a 2 cell solution with a maximum voltage of 8.4 V. The error bars show 5% deviation from the average reading value.

The node is observed to be continuously operable for at least 10 hours. The node voltage monitor shows that the accuracy of the reading is within 5% of the true value. The reading value indicates that the battery voltage for both tests remained above the 9 V and 6 V safety thresholds respectively, and therefore the battery safety limit was not reached in either test, indicating that the battery power is sufficient to maintain node operation for at least 10 hours.

7.3.6 Temperature monitor

The temperature sensor aboard the HERMES nodes provides a voltage divider that is read by the MCU ADC. The input voltage vs temperature characteristic is shown in figure 7-8. The thermistor characteristic is approximated linearly around the 30 degree point. A decrease in voltage translates to an increase in temperature. This allows the internal PCB temperature to be monitored. An auxiliary function that is made possible by the temperature monitor is the ability to detect anomalous temperatures around a particular node, allowing the HERMES node to provide indicative temperature readings of the surrounding environment.

Figure 7-8: Temperature monitor characteristic

7.4 Single-hop RF performance

7.4.1 Experimental method

Two experiments are conducted to determine the range and data-rate performance of the HERMES wireless nodes in a mine environment. A pilot test is conducted using the HERMES electronics at 80 – 95 m in the VUW Cotton corridors at a nominal 320 kbps. This test is used to determine the maximum data-rate and range attained by the HERMES node. Preliminary testing of the WizFi220 system in section 3.8 showed that the prototype node was able to maintain a data-link up to 80 m. This experiment aims to verify that the range of 80 – 95 m is still viable.

The second experiment conducted in the Wright's Hill tunnels examines the node's performance in an underground tunnel environment up to 130 metres. This experiment aims to verify the result of the experiment conducted in section 3.8 in a closed tunnel environment. In all experiments, the nodes are laid flat on the floor within line of sight. 500 packets are captured at each distance by the software logger in section 7.2.

7.4.2 Range-Data Rate Result

The result for the range testing is shown on figure 7-9. This figure shows the results of three tests. The first set of data-points demonstrates the performance of the pilot trials at 320 kbps. This is denoted by the green data-set. The second set of data points demonstrates the performance of the HERMES node in the second experiment. This is represented by the blue data-set. The final sets of data-points are reproduced from figure 3-12 indicating the performance of the ATmega2560 prototype system. This is represented by the red data-set.

Figure 7-9: Range data-rate result

Figure 7-9 shows that the final HERMES system attains a reliable data rate of around 240 kbps which is slightly under the nominal 250 kbps output rate. The effective throughput of one hop at this range is shown to be around 96%. The use of the high-speed UART allows a significant improvement over the prototype system which was able to attain 64 kbps. The range that the WizFi220 is able to attain is observed to be around 100 metres in the case of the open tunnel in the VUW Cotton Corridors. A slight range penalty found in the Wright's hill tunnels as the range was limited to 95 metres. The connection was lost at 100 metres.

7.4.3 Extended Range Result

Due to the limit of 100 metres observed in the Wright's Hill tunnels, an additional experiment is conducted to explore whether the range of the HERMES nodes is primarily constrained by range or by the proximity of the node to the ground. This additional test is conducted under identical conditions of the second experiment as described in section 7.4.2. In this experiment, the transmitter node is raised above the ground by 1 metre and the data-rate is measured. The experiment starts at 100 metres where the aforementioned results predict no data transmission. The separation distance is increased at 100 metres by 10 metres until the connection is lost again. The result of this experiment is shown in figure 7-10.

Figure 7-10: Extended data-rate result

Figure 7-10 shows the result of the extended range experiment. The connection is maintained from 100 metres to 130 metres at an approximate data-rate of around 220 kbps. A slight drop in data-rate is observed owing to the increased noise and lower received signal strength. The maximum observed data-rate variation is around 100 kbps. The instantaneous data-rate varied and the absolute minimum data-rate at any time was hence around 120 kbps.

7.4.4 RSSI in tunnels

An RSSI test is conducted in the Wright's hill tunnels using the same experimental conditions as described in section 7.5.2. The results are shown in figure 7-11. This experiment starts at 50 metres and extends to 130 metres as was observed to be the limit in 7.4.3.

Figure 7-11: RSSI result for single-hop range

The RSSI result in figure 7-11 shows that the average loss of signal is approximately 2 dBm per 10 m. This seems to agree with the literature examination 2.4 GHz signal loss of roughly 20 dBm over

100 metres. The RSSI indicator predicts an RSSI of -89 dBm at 130 metres which is within the stated -94 dBm minimum signal reception provided by the WizFi220.

7.4.5 Non-line of sight test: 90 degrees

A mine environment post disaster can cause debris to be introduced into the mine tunnels from events such as cave-ins. This affects the line-of sight required for wireless transmission. The changing and unpredictable environment inside a mine motivates the need to investigate the performance of the HERMES node under these sub-optimal conditions.

This experiment is conducted inside the Wright's Hill Tunnels. This particular test examines the consequences of the placement of nodes outside of line of sight on a 90 degree corner. This single-hop experiment is conducted with two nodes being placed in a corridor intersection. The experimental setup is shown in figure 7-12. The transmitter node is placed 10 metres from the intersection and transmission is set to 250 kbps. The receiver node is placed ninety degrees away from the transmitter node. This node is placed 1 m from the crossroad. The distance between the receiver node and the crossroads is incremented by 1 m per test until the signal is lost. 1000 packets are recorded by the receiver node. This experiment is then repeated with RSSI readings at 2 m intervals until the signal is lost.

Figure 7-12: Non-LOS 90 degree cornering test

The result of the data-rate experiment is shown in figure 7-13. This figure shows the data-rate with respect to the receiver node distance. As the node is moved away from the crossroads, figure 7-13 shows that the data-rate is maintained until around 5 metres away from the intersection. When the node is moved further away, the data-rate drops to zero and the signal is lost. A variance in data-rate

is observed of roughly 50 kbps. The expected minimum instantaneous data-rate is therefore around 200 kbps.

Figure 7-13: Data-rate result, non-LOS test

The RSSI of the readings shown in figure 7-14 reveals that the loss of line of sight, even at 1 metre, constitutes a drop of around -66 dBm, which is a significant signal loss. In contrast, when the node is in full line of sight at 10 m as shown by the 0 m data point, the signal strength is -16 dBm. It is apparent from figure 7-14 that the greatest factor of signal loss is due to the loss of line of sight altogether. The loss of signal due to distance is observed to be a minor factor compared to the loss of signal due to line-of sight loss. The RSSI readings around 5 m do not show a significant drop in RSSI around the point of data-rate loss observed in figure 7-14 indicating that the RSSI does not reliably reveal any discontinuities that would cause connection loss.

Figure 7-14: RSSI result, non-LOS test

7.4.6 Non-line of sight test: 30 degrees

This experiment aims to determine the performance of the HERMES node when a tunnel is angled with no line of sight. Most mine tunnels are straight but there may be cases where a tunnel has a bend that can cause partial line of sight loss. A transmitter is placed 10 m around the corner outside of line of sight on marker B on figure 7-3 inside the Wright's Hill Tunnels. The receiver node is placed at 10 m way from the corner of the bend outside of line of sight and 1000 packets at a data rate of 250 kbps are captured. The receiver node is moved in intervals of 10 metres away from the bend until loss of signal is observed. RSSI readings are performed up to a point of signal loss. The 30 degree angled tunnel experiment is shown in figure 7-15.

Figure 7-15: Angled tunnel experiment

The data-rate result of this experiment is shown in figure 7-16. This figure shows that a signal is maintained by the receiver/transmitter pair up to 30 metres.

Figure 7-16: Data-rate result, 30-degree bend

Figure 7-17 shows the RSSI readings of the node as the receiver distance is increased. As was observed in figure 7-14, the loss of line of sight is a significant contributing factor to signal loss. The RSSI readings shows a significant variation as the node distance is increased.

Figure 7-17: RSSI range result, 30 degree bend

7.5 Application performance

7.5.1 Decimation in time software performance

This section discusses the software delay performance of the DIT compression system examined in section 6.5. 500 frames are generated and table 7-1 shows the breakdown of the encoding delay.

Table 7-1: DIT encoding delay performance

	Delay contribution	Average Delay length (ms)	Average Delay Variance (ms)
A	Monochrome Processing	3.37	0.88
B	Decimation Pre-filter	2.16	1.70
C	Frame Decimation	0.63	0.44
D	Frame Encoding	0.54	0.35
E	Video packet parsing to transmitter	0.15	0.12
F	Video packet parsing to WizFi220	10.00	10.00
	Total	17.85	13.49

The packet delay incurred on the receiver endpoint is observed. 500 packets are captured on the receiver and the delay for each decoding process is shown in table 7-2.

Table 7-2: DIT decoding delay performance

	Delay contribution	Average Delay length (ms)	Average Delay Variance (ms)
A	WizFi220 to NUC	10.00	10.00
B	Packet Decoding	17.24	20.63
C	Spatial filter	7.10	4.86
D	Temporal moving average filter	5.14	2.76
	Total	38.48	38.25

The total value of the delay as shown in table 7-1 and 7-2 is substituted into the value of the processing delay as shown in equation 7.12.

$$Proc_{Endpoints} = Proc_{Rx} + Proc_{Tx} = 55.33 \text{ ms} \quad (7.12)$$

$$Proc_{tx} = 17.85 \text{ ms}$$

$$Proc_{Rx} = 38.48 \text{ ms}$$

Equation 7.13 shows that the DIT system attains an end to end delay of 611 ms.

$$Ave_delay_{DIT} = 55.33 \text{ ms} + 555.60 \text{ ms} = 611 \text{ ms} \quad (7.13)$$

The video data is transmitted synchronously with the control signals as specified in section 2.6.4. The end to end delay characteristic for the control signals is evaluated by logging 1000 10-byte control packets. The result of this experiment shows that the control packet consumes 0.8 ms on the main MCU process. When combined with the WizFi220 delay as shown in section 7.3.1, the total average delay per hop for the control packet is 4.24 ms (0.8 + 3.44 ms) per hop. This delay improvement shows that the smaller control packet actually attains less delay due to the shortened packet length. The total packet delay incurred by a control packet is shown in table 7-3. This table evaluates the delay of the control packet, accounting for the delay of each control packet over 40 hops. 10 ms is added to each endpoint to account for hardware-software interface delay as is expected when data leaves or enters any of the endpoints, and this delay is observed in both table 7-1 and table 7-2. The total delay of the control packet is hence shown to be 189.6 ms from the base station to HADES.

Table 7-3: Control packet delay characteristics

Delay Source	Value (ms)
Base station transmit	10
40 hops	4.24 x 40 = 169.6
HADES receive	10
Total	189.6

Table 7-3 shows that the time required for a control packet to be received by HADES is roughly 190 ms. The total RTT for a control packet to react in response to an on-screen event is shown in equation 7.15. This shows the time for a sub-frame to be received by the user added to the time for a control packet to reach HADES, assuming that the human reaction time is instantaneous.

$$RTT = 189.6 \text{ ms} + 611 \text{ ms} = 800.6 \text{ ms} \quad (7.15)$$

Equation 7.15 shows that the total RTT for user response is around 800.6 ms, which falls under the 1 second requirement for operator response delay as specified in section 2.6.2. This result shows that the RTT response time length for user reaction is made possible by the short control signals that results in less per-hop delay than a video packet due to its length.

7.5.2 Decimation in time video quality performance

7.5.2.1 DIT Adjustable Threshold

The adjustable thresholding method discussed in section 6.3.3 examines how a user may be able to adjust the DIT thresholds in order to emphasize lighter or darker features by setting the threshold for white and black dynamically. This effect is shown in figure 7-18.

Figure 7-18: DIT with differing thresholds

Figure 7-18 shows a combination of four thresholds of 60, 90, 120 and 150 on the top left-hand corner. Each successive frame increases all of the thresholds by 30. This figure shows that an increase in the quality of the image has allowed more of the background objects to be resolved. While not all objects were able to be fully identifiable with just one setting, the adjustable threshold allowed more objects to be identified when the threshold was changed. In particular, the box on top of the left-most shelf is resolved through consecutive darkening of the thresholds. This object becomes apparent when the threshold is moved. The multiple boxes in front of the shelf over the left shoulder of the figure become apparent when the threshold is reduced.

7.5.2.2 Dynamic DIT video performance

The DIT compression scheme is evaluated in a mine environment. The video algorithm is applied to a series of YouTube spelunking clips (SAMPLE_X A.3) by a user named *Exploring Abandoned Mines and Unusual Places* [50][51]. These videos primarily feature underground mine environments with a limited light source. These images are run through the compression (COMPRESSED_SAMPLE_X A.3). Four clips are examined with a DIT algorithm operating with thresholds of 60, 90, 120, and 150. An adjustable threshold is applied to these frames to allow a human operator to determine the exact colouring to adjust the image colouring in real-time. These compressed video clips are recorded and slowed down by a factor of 3 as the human camera operator moves quickly which can make detail and features difficult to observe.

Sample A shows a simple traversal of an underground mine tunnel. The field of view of this simple tunnel is shown in figure 7-19 taken at 0:08. The human operator is using a single torch and this mine sample shows the view as the camera moves deeper into the mine. The compressed version of this video shows that the mine interior is well resolved. The adjustable threshold method is used to accurately and quickly re-identify the end of the tunnel whenever the camera-angle moves. Using the DIT method, the gradient of the walls and the general outline of the tunnels are revealed. Objects that were easily identified were the chains hanging from the ceiling, and the miscellaneous metal rods laid on the floor.

Figure 7-19: Mine interior using DIT

Humans in the field of view were well resolved against the background when the lighting is appropriate. However, this identification can only be made at a certain distance, observed to be around 5 metres. The silhouette effect allowed the outline of the human to be resolved against the black background at 1:50 and 3:06 (SMAPLE A). The door shown at 91 shows that the adjustable method was able to adjust the image to attain the maximum image quality. One interesting effect that was observed was when the human figure turned and the headlamp caused the camera to adjust to this. The adjustable threshold was used to tone down this lamp to better resolve the cave walls and the human figure. The bright-spot effect can be shown in figure 7-20, left. This figure shows the headlamp with an oval light streak on the ground in the right side of the image. The right figure shows the human figure on the left side against the metal door.

Figure 7-20: Human identification using DIT compression

Sample B examines how the DIT method handles objects whilst inside the mine. This experiment shows that context and familiarity increases the saliency of the resultant video stream. The beginning of Sample B shows a supermarket trolley inside the mine, the adjustable threshold was able to distinguish the item and its background, albeit separately. The archways in the mine from 0:15 onwards shows that in a mine environment with context, the video can be easily deciphered by a human operator. The lighting gradient and linear positioning of the archways allowed the tunnel and the surrounding environment to be identified. In addition to inferring the position of the far end of the tunnel, the presence of these archways allowed the ground to be mapped and hence the relative size of the tunnel was apparent as shown in figure 7-21.

Figure 7-21: Archway tunnel

Sample C shows that in a mine position with insufficient context and object identification, the field of view can be difficult for an operator to comprehend. The effectiveness of the adjustable threshold was decreased due to the lack of information about the mine environment. The optimal setting was difficult for the operator to find. In this sample, the ground, ceiling and walls are not easily resolved and the pathway of this mine environment is not easily resolved. The lack of shadowing prevented the silhouette effect from highlighting most of the darker obstacle features. Visual difficulty in identifying the mine environment can be shown in figure 7-22.

Figure 7-22: Obscured visual field

Sample D shows the advantage of the DIT compression at close range with high contrast. The beginning of sample D shows that the rocks on the wall are easily defined. The shadowing effect of the rocks at 2:15 in figure 7-23 shows that good contrast and lighting can allow the mine tunnel to be well defined.

Figure 7-23: Rockfall silhouette

7.6 Evaluation of HERMES performance

7.6.1 RF range

The range attained by the WizFi220 was observed to be variable depending on the conditions of the environment. The maximum range the WizFi220 was capable of transmitting data was observed to be up to 130 metres in the Wright's hill tunnels as shown in the extended range tests discussed in section 7.4.3. This provides positive evidence of the absolute maximum range that can be attained using 2.4 GHz devices with a bandwidth of 22 MHz inside a mine environment. However, this performance was specific to the case when the transmitter is elevated. Experiments conducted in the hallway environment at ground level with perfect line-of sight only attained a viable data transmission range

of up to 100 m. Experiments on ground level with uneven terrain as shown in the Wright's Hill tests reduced the effective range of the devices further to around 80 m as shown in section 7.4.2. This was due to the uneven ground impeding the line of sight condition between the two nodes when the nodes were sufficiently displaced. This result seems to suggest that the HERMES node performance was primarily limited by ground proximity and not tunnel attenuation as initially predicted. The node is capable of reaching the target 80 m as specified in section 2.6.2. Longer ranges were attained when the node was raised in an elevated position but this range advantage is difficult to use in practice. This is because elevated positions are difficult to identify and deploy on in a mine environment.

The exploration of the consequences of non-line of sight shows that performance is greatly dependant on the topology of the surrounding mine environment. As predicted, the loss of line of sight between nodes resulted in a significantly degraded transmission range. The two non-line of sight experiments conducted in a 90 degree bend and a 30 degree bend shows that loss of line-of sight causes a significant reduction in range and receptivity. A right turn angle outside of line of sight is shown to limit range to roughly 15 m around the bend. Non-line of sight losses in a 30 degree bend is observed to be lower than losses in a straight 90 degree bend as expected. A 200 kbps data-rate is maintained over 30 metres of this bend as shown in section 7.5.7. Total maximum non-line of sight range can be estimated to around 10 m as shown in section 7.5.5.

7.6.2 RSSI

The RSSI readings obtained over the course of experimentation showed several irregularities. The WizFi220 datasheet states that the minimum RSSI reading for 1 Mbps coding rate under 802.11b is -94 dBm. The WizFi220 internal RSSI indicator is used to determine the RSSI of a received signal in the experiments conducted in sections 7.5.5, 7.5.6 and 7.5.4. Despite the consistent application of the WizFi220 hardware in the data acquisition and RSSI readings, the minimum RSSI for practical data transmission varied. Data transmission is observed at 130 metres with an observed RSSI of -88 dBm. This is consistent with the stated data-sheet limit. However, in the case of the non-line of sight tests, the data transmission was observed to be lost at an RSSI of -85 dBm for the 30 degree test at non-line of sight. This reading technically denotes a stronger RSSI than at a range of 130 metres, yet the data-rate dropped to zero at this stronger signal strength. A similar observation is made during the test at 90 degrees outside of line of sight. During this test, an RSSI of -84 dBm was observed at 7 m in section 7.5.5 yet this resulted in zero data-rate during the data-rate tests. It is observed that the RSSI readings do not predict signal loss in the case of non-line of sight RSSI readings. Furthermore, a large variance in RSSI readings are observed in all RSSI observations, especially in figure 7-17, where the RSSI variance does not accurately map to distance reliably. The minimum RSSI indicator for data transfer is accurate in the cases where line of sight was drawn, but inaccurate in the cases where line of sight was lost. Figure 7-11, 7-14 and 7-17 all show that the RSSI indicator does not

accurately map distance, and hence other methods for determining the position for new node placement will be required. As the nodes will only reliably transfer data at 80 m, another method of detecting distance (such as using odometer methods on HADES) would be desired to avoid range-related network disconnections. The implication of this result is that an operator receiving RSSI data should not rely on it in order to determine the placement of nodes to get around obstacles in the mine, nor should the operator completely rely on it as a measure of distance. The work of Parameswaran et al. corroborates this result with a conclusion that the RSSI is not suitable for localization [52].

7.6.3 Application level data-rate

The single-hop RF experiments showed that 250 kbps data transmission at the application layer was observed and this was consistent across all experiments conducted in section 7.4 and 7.5. The pilot test showed the viability of the HERMES node transmitting 320 kbps at 80 to 90 m. This high data rate transmission resulted in occasional disconnect errors that would cause the WizFi220 to lose association with the network. While transmission at this speed was viable, the transmission speed was throttled down to 250 kbps for the remainder of the experimentation. It is recommended that video transmission data-rates are maintained at half of the maximum overall data-rate, and hence the drop to 250 kbps at 80 metres still maintained the application level transmission rate as stated in section 2.6.2. This is important as it was recognised that the data-stream is bi-directional in real-time. Due to the need for the video to be transmitted in sync with the control signals, a margin is afforded to the control signals (operator teleoperation commands) that must be transmitted in the other direction. An effective data-rate of 160 kbps was observed due to the delay required after each data-transmission to prevent buffer overruns on the WizFi220. This data-rate was used for the video data-rate limit.

7.6.5 HERMES video performance

When the DIT method is applied to video of mine interiors, three important factors to consider are revealed that are not evident in the office environment tests (shown in appendix A.2). The performance of the adjustable threshold is observed to be variable depending on the mine conditions. The context of the mine is observed to be highly important and known shapes and positioning of landmarks can improve the identifiability of the video feed. Finally, lighting is observed to be a highly important factor in the performance of the DIT compression system. Direction, intensity, and stability were observed to significantly affect the video quality.

The adjustable threshold was initially implemented as a means of allowing the human operator to change the threshold in order to improve the video quality. Over the video examinations of a mine environment, it was shown that in each of the video samples that this adjustability was vital to the function of the DIT compression scheme. Two distinct effects are observed in the compressed video samples in the mine. The first effect is that of the camera auto-focus. This effect changes the light entering the aperture of the camera, and this automatically dims or brightens the image depending on

the camera control algorithm. This means that the gradient field of the tunnels can change from a useful gradient to a field of view that is less useful to an operator. The adjustable threshold is required to allow the user adjust the final threshold, and hence the view of the final image. The second effect that necessitates the inclusion of the adjustable threshold is that the DIT method is observed to be heavily affected by small changes in the gradient. Due to the binary nature of the black-white thresholding process, small adjustments in the camera angle can cause large changes in the resultant image. This is especially prevalent if many pixels are close to the threshold boundary. The adjustable threshold is a vital core mechanic that allows a stable frame-view using the DIT method.

The DIT method is observed to primarily rely on the interactions between gradients in order for a user to infer the position of the HADES robot and external conditions. The compressed view of sample C shows that this method has significant problems when the environment that is being pictured has no immediately discernible features that stands out to a human operator. The changing gradients observed in this sample clip showed that while the end of the tunnel was easily identified (through the black patch indicating far distance) the shape of the tunnel and the walls were not. The similarity between the gradients of the rock and the ineffective lighting in the case of sample C prevented effective identification of the walls of the tunnel. This would in practice prevent HADES from being driven as the tunnel walls would be difficult to identify. Furthermore, identification capabilities of objects in the field of view would be significantly degraded. As a mine disaster could potentially cause silt to be suspended into the air and all objects in a mine, the lack of identifiability could potentially cause items of interest to be missed when using the DIT compression method. In contrast, the compressed view of sample B showed that even minor visual cues in form of the archways in the tunnel would improve the field of view obtained using the DIT method. The spacing of the archways provides a visual effect of distance and spatial reference not obtained in sample C. The archways also provide a sense of distance between the camera and the walls. The location of the archways on the ground provides a significant indicator of the wall position as well as the width of the tunnel walls.

The lighting conditions observed in all four samples heavily affected the video image. In all four samples, the lighting was provided by a human with a hand-held torch. Several points of interest are identified. The torch is used in Sample A and is a constant beam to the front, which easily identifies the topology of the tunnel. When the torch is shined elsewhere, the field of view changes quickly (as shown in Sample A around 0:50). The adjustable threshold method is again used to readjust the thresholds until the torch angle is adjusted. The image output was observed to be highly sensitive to this change. When the torch was moved, the tunnel walls seemed to merge with the darkness of the far tunnel dark-spot. Direct light in a dark tunnel as observed in Sample A 1.50 showed that high direct light would cause the camera to close the apertures, and this caused the human in the field of view to appear to be a bright-spot. The adjustable threshold was again used to reduce the threshold to allow the human figure to be visible again. A major advantage of this DIT compression showed that light

sources were easily identifiable and hence objects such as high-visibility vests, miner headlamps and other bright objects could potentially be easily identified. However, the presence of light was also observed to easily white-out the camera view. The presence of the light spot showed that an alternative diffuse lighting solution may provide better performance as focussed light sources have a potential to white-out areas of interest, as can be shown in figure 7-20.

The shifting images and the video obtained as part of this experiment shows that this method still has several limitations. It was observed during all samples that quickly changing and stilted motion of the human-held camera was able to cause a degree of nausea due to the amount of focus required to control the camera. The quickly changing image view in response to light changes and angle changes observed throughout all samples required the use of the adjustable thresholding technique. The management of this thresholding uses the analogue stick. The HADES motion and the camera settings need to be managed synchronously. This results in an awkward and difficult to manage control scheme with lots of control management responsibilities passed to an operator. This could make effective real-time control of the HADES robot challenging. A potential solution could be to split the teleoperation and camera control roles between two base station operators. Figure 7-18 demonstrates the limitations of the obtained video even with this better image quality offered by DIT. Facial recognition is still not possible with the DIT method as the thresholds are still spaced out such that most facial gradients still register as a single colour. While very basic features of this person can be resolved, such as the fact that the figure is wearing a cap, the exact facial features of the figure cannot be easily deciphered with this video feed. Furthermore, minor vertical and horizontal artefacts are observed in the images obtained using this method due to the interpolation process, and this can be further mitigated with a better recombination filter on the receiver side. The quality of DIT is vastly superior to the video quality of 1-Bit RLE as shown in Appendix A.2.3. However, the analysis in section 7.5.2 shows that this compression has limits when the effective range is considered. This is mainly a factor of the low image resolution of 320 x 240 in conjunction with the thresholding process. Far-off objects are not resolved well due to the lack of lighting and this results in a similar colour which is not well differentiated with the DIT method.

7.7 Summary

This chapter discussed the results of a full HERMES system test, including topics of the testing environment, method, and results discussion. Metrics for evaluating the performance of HERMES were discussed and this informed the design of testing procedures embedded into the HERMES MCU process. Tests that were performed in this chapter involved the examination of the HERMES node in a two-node pair to determine static characteristics, and this was extended to a system of multiple nodes. The range and data-rate performance of the HERMES node system was verified in a mine

environment. Video compression was discussed alongside the examination of the application-level performance. The conclusions of this work are presented in Chapter 8.

Chapter 8 Conclusions and Summary of HERMES performance

8.1 Overview

This chapter summarises the technical and network aspects of the HERMES wireless node system. Section 8.1 presents the hardware and network performance with respect to the specifications stated in chapter 2. Section 8.2 discusses the future work and improvements that can be made to the design to improve aspects of the HERMES node. Improvements mainly focus on the extension of the node hardware to account for user-operability and contingency management. Additional hardware improvements are briefly discussed in the event that future capabilities become available or in the construction of a further revision. This chapter is concluded with a summary of HERMES' overall performance.

8.1.2 Hardware performance

8.1.2.1 Power supply

Two similar power supply solutions are detailed in section 4.4, with a 3-cell 11.1 V lithium-polymer battery system chosen as the preferred power supply. Eight hours of operation are specified in section 2.6.5. The power supply is theoretically calculated in section 4.3.5 to operate under worst case expected power draw of 13.32 Wh over the lifetime of the node. A power supply test experiment (shown in section 7.3.4) shows that the voltage of the battery was able to remain within safe limits over 10 hours of continuous operation, exceeding the 8 hours of budgeted operation.

8.1.2.2 Physical characteristics

The full node as shown in figure 8-1 measures 15.5 cm long in its largest dimension including the antenna. The full node size excluding the antenna is 13 cm. The width of the node is 7 cm with a node thickness of 2.7 cm. The PCB system sits compactly inside the enclosure and measures 4.7 x 4 x 1.5 cm when stacked appropriately. The antenna measures 9 cm long from the base to the antenna mount to the top of the antenna. The node construction weighs a total of 225 grams which meets the weight specification of 250 grams per node. The HERMES node meets the specifications of weight and size as described in section 2.6.7.

Figure 8-1: HERMES node

8.1.2.3 Node peripheral support

Section 4.3.3 discusses how the HERMES node features both temperature and voltage detection circuitry. The power supply offers safety cut-off features inbuilt to the power packs and this provides under-voltage and over-current protection. This circuitry allows the MCU to detect anomalous conditions within the HERMES node enclosure. The software discussed in section 5.2.6 enables the HERMES node to make periodic message transmissions to the base station to report on the node condition, satisfying the monitoring specifications as stated in section 2.6.5.

The HERMES node contains indicator power LEDs as described in section 4.2.2. These two LEDs provide illumination to the node and immediate surroundings. An accelerometer is implemented to allow the node detect its own orientation in space to provide a deployment trigger. These features fulfil the requirements discussed in section 2.6.8.

8.1.2.4 RF Range performance

The RF range of the HERMES node is highly dependent on the operating conditions. Due to the difficult environment, placement of the HERMES node is a significant factor in the RF range performance. Experimentation on two different tunnels (environments described in section 7.1) reveals that the reliable RF range of HERMES is roughly 80 metres as shown in section 7.4.2. The result in 7.4.3 showed that extended range up to 130 metres is possible when the node is in an open position with elevation, although this is impractical in normal operation. The range requirement in section 2.6.1 for 80 m is therefore satisfied.

8.1.3 Network and software performance

A basic video compression method is explored to determine a viable compression for teleoperation of HADES over the network formed by the HERMES nodes. Due to system constraints, WizFi220 limitations and the need for a data-rate buffer, the effective data-rate was observed to be 160 kbps. A basic 1-bit compression technique is deemed unsuitable but provides the groundwork for a better compression scheme.

A decimation in time compression technique is implemented, extending the 1-bit run length encoding method. By decimating the frame into four sub-frames, more image quality is retained by using a different threshold for each image section. This allowed more hues to be added to each frame and allowed each sub-frame to be reconstructed without receiving the other sub-frames. This technique was able to quadruple the refresh rate by splitting the frame spatially instead of sequentially. This improved the frame-rate to 40 Hz and reduced the overall end to end delay for a finished frame as shown in section 7.5.1. A moving threshold set by the user allows for resolution of objects over the entire monochrome spectrum as shown in section 7.5.2 and this offered the operator significantly better depth perception and object resolution as discussed in section 7.6.3. This function was observed to be vital to the consistency of the video stream. While this basic video was able to identify features in a mine environment, certain conditions caused the performance of the video to be unusable. In particular, context was important when deciphering a video stream and this required identifiable landmarks or items on the video stream. When the image silhouette was not immediately recognisable as shown in section 7.5.2.2, the operator lost sense of the image stream.

A delay of roughly 800 ms is observed on video frames, satisfying the end to end delay constraint. The depth of the image is maintained at 320 x 240 pixels and hence video with at least 10 frames per second can be transmitted as specified in section 2.6.2.

8.1.3 Bill of materials

An abridged breakdown of the costs of the HADES node is provided in table 8.1, with exact component breakdowns available in appendix A.6.

Table 8-1: Bill of materials – HERMES node

Item	Cost Per Node
Main electronics	\$120
Node enclosure	\$50
Seals and node lid	\$11
Battery Cells	\$13
Antenna construction	\$11
PCB fabrication	\$7
Cost per unit:	\$212 NZD

Table 8.1 shows that the electronics components do not exceed \$250, with the total node design falling under \$212 per node, satisfying the cost requirement in section 2.6.6. The removal of the Teensy 3.2 development board (and simply using the MCU chip from factory) can save up to 50% on the MCU unit. Significant reductions in the cost of the node design could also be attained if these node components are mass-sourced. It is likely that these nodes can be reduced in cost to below \$150 during multiple-unit manufacture.

8.2 Future Work

8.2.1 User Operability

Aspects of user-operability and field application design are not considered in the design of this prototype HERMES node. The current HERMES hardware design features a start-up procedure that requires an involved process (described in appendix A.7). The node initialization process is both long, complex and requires technical knowledge of the system. This raises several issues that need to be addressed before the node can be deployed in the final use case.

The HERMES node forms a communications network that is a critical point of failure for the HADES scout robot. Therefore, the node reliability must be maximized. A major concern is the possibility of a node being assembled or initialized incorrectly by the user. The simplification of the node setup process will likely decrease the likelihood of operator error during a disaster situation. It is desired to reduce node initialization time by simplifying the process of node loading and setup. An intuitive and simplified node initialization process is advantageous as the nodes will generally be assembled into working order by non-specialist staff under time pressure.

Minor foibles are encountered in the existing hardware design and hence several improvements are identified that can improve user-operability. The power LEDs do not currently have status labels to indicate the meaning of the LED. Furthermore, there are only two LEDs present on the node and more may be required for a user to fully identify the state of the node at a glance. This is easily

implemented as the SPI bus to the top board is not used and hence these lines can be repurposed to perform this function.

Software initialization is also a concern. The current node design is loaded via Arduino code and a USB cable. This requires the operator to use the Arduino IDE and also requires the Teensy add-on. Similarly, the IP address is loaded by changing code variables. All of these factors introduce the potential for operator error. A better solution for node-operability would be to allow the HERMES node hardware to be configured via a dedicated HERMES configuration serial device (a separate piece of standalone hardware) to negate the possibility of an incorrect upload to the HERMES MCU firmware. This can also simplify and speed up the node configuration process.

Other improvements to the hardware include the change of the Teensy 3.2 to just an ARM-core microcontroller implemented onto the microcontroller board. Currently, a reset can only be initialized by reloading the HERMES operating system software via USB serial. A discrete reset switch which is easily accessible by the operator is desired. The HERMES operating electronics requires two PCB boards to be put together and then placed into the node enclosure. The antenna and power must be connected before the fitting of the node electronics in the node enclosure. The two PCB boards must be separated and then put back together, and this process is time consuming. Improvements in this area are desired, possibly through easily accessible power and RF connectors. This can potentially be facilitated by edge power and RF connectors or the possibility that both PCBs are merged into one. In either case, the HERMES electronics can be shipped within the node enclosure and hence do not need to be removed to be initialized.

8.2.2 Software Improvements

The HERMES node also requires additional features to allow it to be fully reliable in practice. The implemented software handles the main case data processing and deployment but does not have any software checks implemented in the case of a contingency. The HERMES node implements several monitoring features that allow it to report its status back to the user at the base station. However, the mechanisms that allow the user to shutdown nodes and set network parameters are not implemented. A node malfunctioning necessitates its removal from the network. However, the network setup as described in section 5.2.2 requires a strict ordering of IP addresses in order for routing to function. A process of attaching a node to a position within the network after node failure is not implemented. This would involve the changing of an IP address after the deployment of the node, something that is not currently possible with the network setup. Furthermore, the network system as described in 5.2.5 shows a network consisting of chained nodes arranged in an ad-hoc fashion while in the mine. This simplification assumes that each node only sees two nodes and this allows a simple routing algorithm that is currently implemented into the nodes. However, in mines, tunnels that consist of a tree or maze topology mean that forks in the network where the node network branches are likely in practice if

HADES backtracks. This means that nodes need the ability to route data to the correct branches under these conditions. The current software as discussed in section 5.2.5 does not currently reflect this and more advanced addressing and routing is desired.

8.2.3 Hardware Improvements

Improvements in the capabilities of RF modules in the future can also potentially result in an improvement in the HERMES node. The WizFi220 module may not be available in the future. Due to the ubiquity of 2.4 GHz devices, a suitable replacement module with equal or better radio characteristics is likely to be found. The WizFi220 is also shown to have several limitations. Section 3.8.1 describes the need for an MCU unit to provide instructions to the unit. This is recognised to be a fundamental impediment as the need for an MCU unit limits the packet transfer rate, no matter how fast the MCU unit. Section 7.3.3 discusses how the ratio of delay on the HERMES node approaches 3:1 in terms of delay on the MCU, with the MCU unit constituting the majority of the node hop delay. A better solution that does not use an MCU with the same or similar features has the potential to further improve the data-rate and delay performance of the HERMES node. The implementation of the WizFi220 in the HERMES node further demonstrates the viability of 2.4 GHz in a mine environment up to 80 m. It is possible that a device with similar or better RF characteristics as the WizFi220 can be substituted in the node. As discussed in section 2.5.1, alternative technologies in the 900 MHz band (with theoretically better mine transmission characteristics) were not available. Further improvements to the fundamental design may be attained if radios offering better performance in the sub-GHz range would be available in the future.

8.2.4 Video system Improvements

The thesis explored the use of video compression methods to achieve a video stream that can be transmitted across the HERMES node network. A particular area of concern that is not addressed is the exact suitability of this video for teleoperation. While certain characteristics of the HADES video stream are quantified with video quality characteristics, the full usability of this video is unclear. The quality of 320 x 240 pixels is chosen as the screen resolution and this is easily attained with the hardware. Several issues are identified in the particular HADES use case that casts doubt on this video quality. Due to the proximity of the HADES chassis to the ground, the video stream will likely obtain a skewed view of the mine interior from ground level. The positioning of the illumination LED, dust and glare effects as well as the irregular movement of HADES (due to the wheels) will likely also degrade visual quality further, and these factors were not investigated. Other processing will likely be required to resolve this so that the HADES video stream can be effective for driving. More sophisticated compression methods is likely necessary to improve overall video quality.

8.3 Final Summary

A wireless network node named HERMES is designed as a complete wireless solution for HADES, a mine scout robot. HERMES provides a wireless network that allows the transmission of video data to the operator alongside crucial sensor measurements and teleoperation commands to aid mine rescue efforts. This wireless node design features a light, small-form stackable electronics package with a battery life of at least 10 hours. A wireless range of 80 m is obtained with a 2.4 GHz 802.11b ad-hoc radio system implemented with the WizFi220 wireless module. A raw data-rate of 320 kbps is observed over this network, and this allows the transfer of low-quality compressed video. A bespoke compression scheme is implemented to achieve sub-160 kbps data-rate performance using a decimation in time technique to attain a 40 Hz refresh rate that minimizes frame delay to under 800 milliseconds. A basic video stream with a resolution size of 320 x 240 is attained with this DIT compression technique.

Chapter 9 References

- [1] Molyneaux, L., *Development of an Underground Mine Scout Robot*. ME Thesis, Victoria University of Wellington, 2016.
- [2] Heesterman, P., *Tetherless Robotic Communication*, ENGR489 report, Victoria University of Wellington, 2015
- [3] “802.11b White Paper” by VOCAL technologies, Ltd., Internet page at URL: www.vocal.com/white_paper/802.11b_wp1.pdf last accessed 02/02/18
- [4] “The Bluetooth Wireless Technology White Paper” by Atmel, Internet page at URL: <http://educyclopedia.karadimov.info/library/DOC1991.PDF> last accessed 04/02/18
- [5] Anastasi, G., Borgia, E., Conti, M., Gregori, E., “IEEE 802.11b Ad Hoc Networks: Performance Measurements,” *Cluster Computing*, vol. 8, pp. 135-145, July 2005
- [6] “802.15.4-2015 – IEEE Standard for Low-Rate Wireless Networks”, Internet page at URL: <http://ieeexplore.ieee.org/document/7460875/?reload=true> last accessed 04/03/17
- [7] “Demystifying 802.15.4 and Zigbee” by Digi, Internet page at URL: https://www.mouser.com/pdfdocs/digi-wp_zigbee.pdf last accessed 19/03/17
- [8] Bandyopadhyay, L. K., Chaulya, S. K., Mishra, P. K., Choure, A. Baveja B. M., “Wireless information and safety system for mines,” *Journal of Scientific & Industrial Research*, vol. 68, pp. 107-117 , Feb 2009

- [9] Patri, A., Minaje, D., “Radio frequency propagation model and fading of wireless signal at 2.4 GHz in an underground coal mine,” *South African Institute of Mining and Metallurgy*, vol.115, pp 629-636, Aug 2015
- [10] Zhou C., Plass, T., Jacksha R., Waynert J. A., “RF Propagation in Mines and Tunnels,” *IEEE Antennas and Propagation Magazine*, vol. 57, pp 88-102, August 2015
- [11] Internet page at URL: <https://www.digi.com/products/xbee-rf-solutions/sub-1-ghz-modules/xbee-pro-900hp#specifications> last accessed 19/03/17
- [12] Zeta Miniature RF Transceiver by rfsolutions, Internet page at URL: https://media.digikey.com/pdf/Data%20Sheets/RF%20Solutions%20PDFs/ZETA_Series.pdf last accessed 19/03/17
- [13] eRIC9 Radio Transciever by LRPS, Internet page at URL: https://www.mouser.com/ds/2/743/lprs_eRIC9-FCC-1217491.pdf last accessed 19/03/17
- [14] Sun, W. Choi, M, Choi, S., “IEEE 802.11ah: A Long Range 802.11 WLAN at Sub 1 GHz,” ICT standard vol. 1 pp 83-107, 2013
- [15] “PT 7000 Heavy Equipment Management” by Orbcomm, Internet page at URL: <https://www.orbcomm.com/PDF/datasheet/PT-7000.pdf> last accessed 25/03/17
- [16] Internet page at URL: <https://mdt.ca/products/data-acquisition/wireless-mdt-rtu/> last accessed 25/03/17

[17] “Introducing the Crescendo Series UHF/VHF Half Duplex Radio Modem,” by RF Innovations, Internet page at URL: [http://www.rfinnovations.com.au/Uploads/Images/UHF-VHF%20Crescendo\(1\).pdf](http://www.rfinnovations.com.au/Uploads/Images/UHF-VHF%20Crescendo(1).pdf) last accessed 25/03/17

[18] “NS40 I.S. Wireless Network Switch User Guide,” by Mine Site Technologies, Internet page at URL https://support.mstglobal.com/releases/manuals/english/NS40/NS40_UG_EN_D.pdf last accessed 25/03/17

[19] Internet page at URL http://mstglobal.com/wp-content/uploads/2016/09/MST_WRN_Overview_en.pdf last accessed 26/03/17

[20] “All about batteries, Part 4: Alkaline Batteries,” Cowie, I., Internet page at URL https://www.eetimes.com/author.asp?section_id=36&doc_id=1320919 last accessed 29/03/17

[21] Internet page at URL <http://www.epectec.com/batteries/chemistry/> last accessed 29/03/17

[22] Internet page at URL <http://datasheet.octopart.com/5030852-Ansmann-datasheet-5400527.pdf> last accessed 29/03/17

[23] “Alkaline Manganese Dioxide Handbook and Application Manual,” by Energizer., Internet page at URL http://data.energizer.com/pdfs/alkaline_appman.pdf last accessed 29/03/17

[24] “Characteristics of Rechargeable Batteries,” by Texas Instruments., Internet page at URL <http://www.ti.com/lit/an/snva533/snva533.pdf> last accessed 29/03/17

[25] Internet page at URL <https://www.lsr.com/downloads/lsr/engineers-approach-antenna-selection-implementation-webinar.pdf> last accessed 02/04/17

[26] “AN4190 Application Note Antenna Selection Guidelines,” by De Vita, P., ST microelectronics, Internet page at URL

http://www.st.com/content/ccc/resource/technical/document/application_note/82/38/3f/9f/fd/4d/4f/ab/DM00068254.pdf/files/DM00068254.pdf/jcr:content/translations/en.DM00068254.pdf last accessed 02/04/17

[27] Chen, J., Haas, E., Barnes, M. “Human Performance Issues and User Interface Design for Teleoperated Robots,” *IEEE Transactions on systems, man, and cybernetics – part c: Applications and Reviews*, Vol. 37, no. 6, pp 1231-1245, November 2007

[28] Laine, J., *Low Latency High-Definition Video Streaming For Real-time Teleoperation Platform*, MSc. Thesis, Tampere University of Technology, 2016

[29] Apostolopoulos, J., Tan, W., Wee, S., Video Streaming: Concepts, Algorithms and Systems, white paper, Internet page at URL <http://www.hpl.hp.com/techreports/2002/HPL-2002-260.pdf> last accessed 18/03/18

[30] “HVEC Demystified, A primer on the H,265 Video Codec” by Elemental, Internet page at URL https://www.e-idsolutions.com/wp-content/uploads/2016/10/HEVC_Demystified.pdf last accessed 18/03/17

[31] Sullivan, G., Ohm, J-R., Han W-J., Wiegand, T., “Overview of the High Efficiency Video Coding (HVEC) Standard” in *IEEE Transactions on Circuits and systems for video technology*, vol. 22, no. 12, pp 1649-1668, December 2012

[32] “Recommended upload encoding settings,” Internet page at URL <https://support.google.com/youtube/answer/1722171?hl=en> last accessed 03/04/17

[33] Aghagolzadeh A., Meshgini, S., Nooshyar, M., Aghagolzadeh, M. “A novel video compression technique for very low bit-rate by combining H.264/AVC standard and the 2-D wavelet transform,” in *ICSP 2008 9th International Conference*, Beijing, China.

[34] Talluri, R., Oehler, K., Barmonm T., Courtney, J.D., Das, A., Liao, J., “A robust, scalable, object-based video compression technique for very low bit-rate coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol.7, no.1, pp 221-233, February 1997

[35] Thompson, N., - ME thesis in progress, Victoria University of Wellington, 2018

[36] Jones, H., *HERMES*, ECEN440 Report, Victoria University of Wellington, 2017

[37] Internet page at URL <https://www.digi.com/products/xbee-rf-solutions/embedded-cellular-modems/digi-xbee-cellular> last accessed 10/04/17

[38] Internet page at URL <https://cpb-us-west-2-juc1ugur1qwqqo4.stackpathdns.com/u.osu.edu/dist/0/890/files/2013/12/Friis-1ty893p.pdf> last accessed 10/04/17

[39] “Line of Sight Obstruction,” by Campbell Scientific, Internet page at URL <https://s.campbellsci.com/documents/au/technical-papers/line-of-sight-obstruction.pdf> last accessed 10/04/17

[40] “WGM110 Wizard Gecko Wi-Fi Module Data Sheet,” by Silicon Labs, Internet page at URL <https://www.silabs.com/documents/login/data-sheets/wgm110-datasheet.pdf> last accessed 25/04/17

[41] “WizFi220 Datasheet,” by WizNet, Internet page at URL http://www.wiznet.io/wp-content/uploads/wiznethome/WiFi%20Module/WizFi_210_220/Document/WizFi210_DS_V120E.pdf last accessed 22/04/17

[42] Internet page at URL http://www.wiznet.io/wp-content/uploads/wiznethome/WiFi%20Module/WizFi_210_220/App_Note/WizFi210_220_App_Note.zip last accessed 22/04/17

[43] Internet page at URL <https://www.allaboutcircuits.com/technical-articles/the-uart-baud-rate-clock-how-accurate-does-it-need-to-be/> last accessed 29/04/17

[44] “K20 Sub-Family: MK20Dx256VLH7,” by NXP, Internet page at URL http://cache.freescale.com/files/32bit/doc/data_sheet/K20P64M72SF1.pdf last accessed 29/04/17

[45] “1.5 A Step-Up/Down/Inverting Switching Regulators,” by ON semiconductor, Internet page at URL <https://www.onsemi.com/pub/Collateral/NCP3063-D.PDF> last accessed 15/05/17

[46] “MEMS digital output motion sensor: ultra-low-power high-performance three-axis “nano” accelerometer” by ST, Internet page at URL <http://www.st.com/content/ccc/resource/technical/document/datasheet/23/c3/ea/bf/8f/d9/41/df/DM00040962.pdf/files/DM00040962.pdf/jcr:content/translations/en.DM00040962.pdf> last accessed 16/05/17

[47] “Intel® RealSense™ Camera SR300,” by Intel, Internet page at URL https://www.mouser.com/pdfdocs/intel_realsense_camera_sr300.pdf last accessed 19/09/17

[50] “4100 Feet Underground in the Massive, Abandoned Admiral Aaron Ward Mine,” by Exploring Abandoned Mines and Unusual Places, Internet page at URL <https://www.youtube.com/watch?v=M1TAEff4CW0> last accessed 10/03/18

[51] “Creepy Sounds Captured in an Abandoned Mine While Reviewing the ThruNite TN12 Flashlight,” Exploring Abandoned Mines and Unusual Places, Internet page at URL <https://www.youtube.com/watch?v=UVEU6n2eKtg> last accessed 10/03/18

[52] Parameswaran, A-T., Husain, M-I., Upadhyaya S., “Is RSSI a Reliable Parameter in Sensor Localization Algorithms – An Experimental Study,” Internet page at URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.454.3357&rep=rep1&type=pdf> last accessed 01/03/18

Chapter 10 Appendix

A.1 1-bit RLE Compression

A.1.1 Video Processing Algorithms

Monochrome conversion

Section 6.1 shows that a significant savings factor of 3:1 can be attained if the colour is discarded for a monochrome stream. The loss of colour information reduces the frame size, but does not significantly impact the image features. Furthermore, mines are generally dark and of a single tone, and hence this loss of colour does not significantly impact on the saliency of the image. Most mine equipment is also painted or coloured in bright colours for safety reasons, and hence monochrome video streams are suitable.

The RealSense images arrive on the `hades_vid_proc` (see section A.5.5) with raw image messages. The image is transformed to monochrome grayscale 8 bit as described in section 5.5.1 using the BGR8 to Y'UV transformation mapping. The colour mapping matrix is shown in equation A.1. This process transforms the red/green/blue colour components to an alternate colour space. This colour space features the Y' channel which is used for monochrome.

$$\begin{pmatrix} Y' \\ U \\ V \end{pmatrix} = \begin{pmatrix} 77 & 150 & 29 \\ -43 & -84 & 127 \\ 127 & -106 & -21 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \dots (A.1)$$

$$Monochrome_{out} = (Y' + 128) \gg 8 \text{ (conversion to 8 bit unsigned output)} \dots (A.2)$$

Since monochrome only requires the black and white channel, the U and V outputs are not evaluated. The video processing node provides a buffer that allows the values of the conversion to be stored. As the number of pixels is 76800 (320 x 240) this is the size of the temporary buffer. The monochrome value is stored in a 32 bit integer format which is due to the overflow of the operation in equation A.2. Equation A.2 shows the method for transforming this data type back to an 8 bit format which reduces the overall space consumed by the monochrome process. A software implementation of the above process for each frame is shown in figure A.1.

```

//First send the packet to grayscale
//reset the data pointer
for (data_ptr = 0; data_ptr < frame_data_size; ++data_ptr){
    //do array access
    y_dash = B_coeff*msg.data[3*data_ptr] + G_coeff*msg.data[(3*data_ptr)+1] + R_coeff*msg.data[(3*data_ptr)+2];

    //fill the array.
    comp_data[data_ptr] = (unsigned char)((y_dash+128)>>8);
}
//comp data is 76800 array filled with grayscale data.

```

Figure A.1: This figure shows the C++ implementation of a grayscale conversion described in equations A.1 and A.2.

This algorithm is implemented in C++ within the ROS node HADES_vid_proc in appendix A.5.5. The function of this code loops through the raw sensor message consisting of 78600 24-bit blocks. The loop function increments in steps of three in to obtain all the RGB values during the monochrome conversion.

Low Pass Filter

A low-pass filter of an image averages a pixel with the neighbouring pixels. This is repeated for all pixels in the image. This process allows an image to be blurred and softens transitions in the image. A simple pixel averaging filter is employed here to allow an image to be deconstructed and processed. When an image is down-sampled (reducing the amount of information by discarding information) the low-pass filter prevents aliasing in the resultant image. A box blur kernel is applied as the low-pass filter. This kernel is shown in equation A.3. This applied kernel means that each pixel is now a non-weighted average of itself and its direct non-diagonal neighbours.

$$\frac{1}{5} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \dots (A.3)$$

This filter is applied to the monochrome image that takes the pixel and averages this pixel with the surrounding four pixels as shown in figure A.3. This is preferred to normalized box blur as this only filters the rows and columns, simplifying the process by only requiring four *if* statements to implement. Furthermore, processing is saved by only using additions and not multiplications that would be present in a weighted blurring method. The accumulated variable is simply added and divided out at the end. This simple method only involves four easily calculated array accesses with modulo and boundary checks to prevent illegal array accesses.

```

unsigned int buffer = 0;
char count = 1;

//pre-filter to smooth out thresholds
for (long int k = 0; k < 76800; ++k){

    //set the temp buffer
    buffer = comp_data[k];
    //set the count.
    count = 1;

    if ((k % 320) != 0) //not on left edge
    {
        buffer += comp_data[k-1];
        ++count;
    }
    if ((k % 320) != 1) //not on right edge
    {
        buffer += comp_data[k+1];
        ++count;
    }
    if (k > 320) //not on top edge
    {
        buffer += comp_data[k-320];
        ++count;
    }
    if (k < 76479) //not on bottom edge
    {
        buffer += comp_data[k+320];
        ++count;
    }

    filtered_data[k] = buffer/count; //filter out the pixel
}

```

Figure A.2: This figure shows the process of the low-pass filter that implements the filtering process shown in equation A.3.

Figure A.2 shows a loop through the entire pixel array. In order to filter the pixel, the data must first be accumulated in a buffer variable. This buffer variable is of type unsigned int to account for overflow that is liable to occur when five bytes of data-type char are summed. Four checks are performed on each boundary to ensure that the filter does not attempt to filter outside of the valid image. This is the function of the four *if* statements shown in figure A.2. For each value that is placed into the accumulator variable, the averaging divider increments. The filtered data is then the non-weighted average of the neighbouring pixels. The value of this operation will guarantee that the size of the variable will be from 0-255 and therefore an explicit cast is not necessary when placing the value into the filtered image array. This array is assigned to at the last step of the averaging process and the name of the buffer is filtered_data.

A.1.2 1-bit RLE system overview

A block diagram of the implemented 1-bit RLE process is shown in figure A.3.

Figure A.3: This figure depicts the video processing chain employed in generating encoding for 1-bit video compression.

The following sections describe the encoding and decoding process of this 1-bit video. The process for video encoding is implemented in the `hades_video_proc` node. The process for video decoding is implemented in the `hades_video_proc_base` node. The full C++ code implementing this process is found in the appendices A.5.3 and A.5.7.

A.1.3 Encoding process

The encoding process begins when a packet is processed by the monochrome transform as described in section 6.2. The data is stored in the `comp_data` array. Figure A.4 shows the process of starting the RLE compression for 1-bit monochrome. 1-bit encoding encodes the value of the pixel as either 1 or 0 to reduce the data from 8-bits per pixel to 1 bit per pixel. The RLE process is a lossless encoding scheme after the 1-bit transform. Table A.1 shows the encoding scheme of this RLE compression technique.

Value	Meaning
First Byte	Starting colour with value either 255 or 0. This value denotes the start colour which is either black or white. The following bytes then reference this starting colour.
0	A run length of value zero denotes a pixel length of 255 or greater. Since a run length of zero is meaningless, this zero value serves as a means of extending the current run length past 255.
1-255	Run length between 1 and 255.

Table A.1: This figure shows the encoding of the implemented RLE compression.

Table A.1 shows that the first value of every packet will be the starting value, either black or white, and this value will only encode one bit. A run length of 1 through 255 denotes that there are that many pixels of the current colour. Implied in the transition between bytes is the colour change between high and low. In order for a run length of greater than 255 to be represented, the unique zero value denotes a run length of 255 and also indicates that more pixels of the same colour are encountered later.

Figure A.4 shows the code implementation of the RLE process. The current run length is denoted by a variable named `run_length`. The first pixel at position (0,0) is compared to determine the starting byte and hence the first colour.

```
//set the run length default to unity.
//Zero is reserved for a 255 byte block.
run_length = 1;

//test_data[0] is the start colour reference
if (comp_data[0] > centroid){
    current_run = 255; //currently white px
    test_data[0] = 255; //set to white: 255 or 1
}
else{
    current_run = 0; //currently black px
    test_data[0] = 0; //set to black: Zero or 0
}
```

Figure A.4: This figure shows the code required to start the RLE process.

Figure A.5 shows the main loop encoding process. Each pixel is compared against the threshold to determine whether the current run continues or whether the colour is changed (either from black to white or white to black). An array named `test_data` is used to store the packet contents of the result of the encoding.

```
if (current_run == 0){
    //our run is zero: currently black

    //encode 1
    if (comp_data[data_ptr] >= centroid){
        ++ones;

        //stop the current run and swap.
        test_data[test_data_pos] = (unsigned char)run_length;
        ++test_data_pos;

        //reset run length
        run_length = 1;

        //now change to encode 255.
        current_run = 255;
    }
    //encode 0
    else{
        ++zeros;
        ++run_length; //continue the current run
    }
}
else{
    //our run is one: currently white

    //encode 1
    if (comp_data[data_ptr] >= centroid){
        ++ones;
        ++run_length; //continue the current run
    }
    //encode 0
    else{
        ++zeros;

        //stop the current run and swap.
        test_data[test_data_pos] = (unsigned char)run_length;
        ++test_data_pos;

        //reset run length
        run_length = 1;

        //now change to encode 0.
        current_run = 0;
    }
}
```

Figure A.5: This figure shows the process of data encoding.

The video comparison is made against a variable named `centroid`, which is used to compare the value of the pixel with the threshold. This variable can be adjusted to highlight lighter or darker objects when the user requires. The value of `centroid` is changed through the analogue stick as discussed in

section 6.4.1 and adjusted depending on user input. The control packet is received by the `hades_vid_proc` node and the centroid threshold value is changed as shown in figure A.6.

```
//call back to modify video system parameters.
void controlCallback(const hades_com_sys::control_packet control){

    //when a centroid packet is recieved, we simply set the centre threshold to that particular value.

    //TODO: MODIFY THE THRESHOLD and use this packet to do it.
    centroid = control.data[9];
}
```

Figure A.6: This figure shows the callback function used to adjust the threshold.

An additional check is conducted at the end of each loop that determines whether more than 255 values have been encountered. If this occurs, then the value of zero is substituted instead of the count, and the count continues from the next pixel. The check shown in figure A.7 determines whether the next value will cause a value of 255. If this is the case, a zero value is substituted and the count is reset with the current colour unaffected. This allows the encoding to encode run lengths of greater than 255 pixels long.

```
//overflow check.
//check we are not at the end so we do not over-access array.
if ((run_length == 255) && (data_ptr != 76799)){
    //if we are currently checking for zero (black)
    if (current_run == 0){

        if (comp_data[data_ptr+1] >= centroid){ //if white

            //put a zero into the item.
            test_data[test_data_pos] = 255; //no delta
            ++test_data_pos; //still raise the data pos

            run_length = 1; //reset the run length
            current_run = 255;
        }
        else{ //if black
            //put a zero into the item.
            test_data[test_data_pos] = 0; //delta
            ++test_data_pos; //still raise the data pos

            run_length = 1;
            current_run = 0;
        }
    }
    //if we are currently checking for one (white)
    else{
        if (comp_data[data_ptr+1] >= centroid){ //if white

            //put a zero into the item.
            test_data[test_data_pos] = 0; //no delta
            ++test_data_pos; //still raise the data pos

            run_length = 1; //reset the run length
            current_run = 255;
        }
        else{ //if black
            //put a zero into the item.
            test_data[test_data_pos] = 255; //delta
            ++test_data_pos; //still raise the data pos

            run_length = 1;
            current_run = 0;
        }
    }
}
```

Figure A.7: This figure shows the process of dealing with run lengths of longer than 255.

The packet is encoded with the packet type identifier. Three types of identifiers are provided in this field to delimit the frame to ensure that the proper start frame is known to the decoder process using the 3 byte header format discussed in section 5.2.4. Their values are shown in table A.2.

ID Value	Meaning
‘K’	‘K’ indicates the start of the video frame and the decoder waits for this frame.
‘V’	‘V’ indicates data that is neither the start nor the end of the frame.
‘X’	‘X’ indicates the end of a video frame. The reception of this frame resets the state machine.

Table A.2: This table shows the values of the packet ID field.

The start frame can be identified with a video packet of ID value ‘K’. The receiver can identify the start of a frame by the packet ID. Furthermore, the receiver can use this system to identify two distinct error conditions. This allows the system to tolerate some level of packet loss. The first error condition that can be detected is a loss of the start packet. As this packet provides the reference colour on the first byte, the loss of a ‘K’ packet means that subsequent run length packets cannot be decoded. The receiver state machine must therefore only attempt to decode once a ‘K’ start packet is received. The second error condition that can be detected occurs if an end-frame (ID ‘X’ or ‘K’) is not detected. This error condition is detected by proxy by the decoder process. When the current packet is not finished but another ‘K’ start packet is found, it is assumed that subsequent packets are lost. In this case, a partial reconstruction of a frame can be made and the partial frame (from the packets received so far) is shown to the user.

```

while (test_data_pos >= 497){
    //add 500 byte offset.
    arr_modifier = count_subpacket*497;
    //set the packet size.
    vid_data.pack_size = 50; //50*10;

    //fill out the length field of the packet.
    vid_data.data[0] = '0';
    vid_data.data[1] = '5';
    vid_data.data[2] = '0';
    vid_data.data[3] = '0';

    vid_data.data[4] = 50; //500 bytes

    if (count_subpacket == 0){
        //the first packet
        vid_data.data[5] = 'K'; //for video
    }
    else{
        //not the first packet
        vid_data.data[5] = 'V'; //for video
    }

    //do not retransmit: UDP
    vid_data.data[6] = 0; //UDP

    //data starts at vid_data position 7.

    //Set 500 bytes to send.
    for (data_ptr = 0; data_ptr < 497; ++data_ptr){
        //send data at appropriate place.
        vid_data.data[data_ptr + 7] = test_data[arr_modifier + data_ptr];
    }

    //publish the packet
    video_pub.publish(vid_data);

    test_data_pos -= 497; //take away 497 bytes.

    ++count_subpacket; //increment the packet position at 497.
}

```

Figure A.8: This figure shows the process of placing the data into individual data-packets.

Figure A.8 shows the construction of a WizFi220 packet in the ROS system by generating the length field of 500 bytes. Vid_data.data[4] denotes the position of the start of the packet. As three bytes are used for the custom header, the amount of data transferred in this packet is 497 bytes. The ID is set by determining whether the packet is the first chunk of data encountered, and the ID is set in vid_data.data[5]. While there is still data to be transmitted, the loop creates new packets of size 500 bytes until no more data is present. After this process, the frame is complete and sent to the hades_com_sys for communication to the WizFi220.

A.1.4 Decoding process

The decoding process of RLE is implemented on the base station on hades_base_vid_proc. A callback function is used to process data as soon as an image is received. The initialization step on hades_base_vid_proc first initializes several image packets and filter frames. The metadata is known in advance, and therefore these fields can be initialized first. The ROS system has dedicated memory management, and therefore the memory for the data of each of the packets needs to be pre-initialized. This is done with the resize function as shown in figure A.9.

```

//Initialize the processed video with metadata.
//the frame size is 320x240
processed_video.height = 240;
processed_video.width = 320;
processed_video.encoding = "bgr8"; //bgr 8 bit per colour 24 bit per px
processed_video.is_bigendian = 0; //little endian
processed_video.step = 960; //960 pixels per row.

//fill in metadata in the filter frames.
for (char i = 0; i < 2; ++i){
    filter_image[i].height = 240;
    filter_image[i].width = 320;
    filter_image[i].encoding = "bgr8"; //bgr 8 bit per colour 24 bit per px
    filter_image[i].is_bigendian = 0; //little endian
    filter_image[i].step = 960; //960 pixels per row.
    filter_image[i].data.resize(FULL_SIZE);
}

//fill in final frames
final_out.height = 240;
final_out.width = 320;
final_out.encoding = "bgr8"; //bgr 8 bit per colour 24 bit per px
final_out.is_bigendian = 0; //little endian
final_out.step = 960; //960 pixels per row.
final_out.data.resize(FULL_SIZE);

```

Figure A.9: This figure shows the frame initialization process.

The frame decoding process begins with a valid 'K' frame. The decoding process involves parsing the received packets and determining the first value of the colour.

```

//select the starting character
if (starting_char == 0) character_in = 0; //black character = 0
else character_in = 255; //white character = 1

```

Figure A.10: This figure shows the selection of the start colour.

When a video packet is received, the run length is decoded by iterating over all values of the packet. This is shown in figure A.11. When a new non-zero value is encountered, the loop process flips the value of the colour to the opposite. As frames may have different numbers of packets, a variable `data_ptr` is checked against the size of the image and this is used to determine whether the process is complete. If a zero value is encountered, the pixel hue is placed into the image array but the colour is not flipped. The zero value indicates that the run length of the colour is greater than 255, so more pixels of the same colour are expected.

```

//set the last count to 1, reset the image data pointer.
data_ptr = 0;
last_char = 1;

//loop through all of the packet.
for (int i = 4; i < 500; ++i){

    //loop through the array at 3 times the size and fill in the pixels.
    if (data_pack.data[i] == 255){

        //if last-char is nonzero
        if ((i != 4) && (last_char != 0)){
            //deswap the character.
            if (character_in == 0) character_in = 255;
            else character_in = 0;
        }

        for (int k = 0; k < 762; ++k){

            //now put the data into the array.
            processed_video.data[data_ptr] = character_in;

            ++data_ptr;

            //if we are finished.
            //send the frame out to the image viewer.
            if (data_ptr == FULL_SIZE){

                //do cleanup.
                state = 0;

                frame_finished = 1;

                process_stats();

                //frame_time = ros::Time::now().toSec() - frame_time

                video_image_pub.publish(processed_video);

                return;
            }

            last_char = 255;
        }
    }
    else if (data_pack.data[i] != 0){
        //non-zero number - swap after x px.
    }
}

```

Figure A.11: This figure shows the code that iterates through the received packet and encodes the video data.

To account for error states on non-‘K’ start packets, a check is performed to determine if a packet of type ‘V’ or type ‘X’ is lost. If a ‘K’ packet is received unexpectedly, the current packet contents are displayed. Figure A.12 shows that the video image is published in the event of a lost packet. Due to the nature of run-length encoding, lost ‘K’ packets and subsequent follow-up packets cannot be recovered as these packets contain base references to the position and colour in the ‘K’ packet.

```

if (data_pack.data[1] != 'V' && data_pack.data[1] != 'X'){
    //We return if we don't get an appropriate video packet.
    //State Machine starts again.
    state = 0;

    frame_finished = 1;

    frame_time = 0;

    video_image_pub.publish(processed_video);

    return;
}
//if we did not return, we have recieved a non-start frame.d

```

Figure A.12: This figure shows the process for handling packet loss. The video obtained so far is displayed.

A.1.5 Video Output

Figure 6-4 shows the result of the 1-bit compression scheme using run-length encoding. Several issues are found with this solution. The features as shown on figure 6-4 are not well defined apart from the outline of the certain items. The gradient of the wall to the ceiling is not well defined by this method.

While the shading on the wall does not fully reveal the wall in the background, the silhouette effect of the dark objects against the white background allows the robot legs shown in the middle of the image to be identified. The general shape of the desk is also well defined. Large features such as the drawers and the robot wheels extruding out of the box are easily visible. Items that are not immediately obvious includes the stack of laptops on the far right of the image. The 1 bit encoding process resolves these items as a pile of dark objects. The loss of colour is noticed in two major areas. The red box with the sheet of paper is easily identifiable as a box in the raw image, but when converted to 1-bit RLE, the image shows that the box becomes indistinguishable as the piece of paper appears to blend into the background.

A.1.6 1-bit RLE Video compression static results

The video decoder run for 500 frames and 500 frames are captured and processed. The result for the delay of each block component is shown in table A.3.

	Delay contribution	Average Delay length (ms)	Average Delay Variance (ms)
A	Monochrome Processing	3.37	0.88
B	RLE encoding process	1.03	0.20
C	Packet generation	0.17	0.14
D	Video packet parsing to transmitter	0.15	0.12
E	Video packet parsing to WizFi220	10.00	10.00
	Total	14.72	11.34

Table A.3: This table shows the cumulative delay for the 1-bit video RLE encoding.

Table A.3 shows that the average delay length per frame is around 15 ms per frame with a standard deviation of 11.34 ms. The packet delay incurred on the endpoints is observed. The receiver process decodes the packets received over the radio. 500 packets are captured on the receiver and the delay for each element of the decoding process is shown in table A.4.

	Delay contribution	Average Delay length (ms)	Average Delay Variance (ms)
A	WizFi220 to NUC	10.00	10.00
B	RLE decoding process	2.02	0.63
C	Packet wait time	83.46	45.30
	Total	95.48	55.93

Table A.4: This table shows the delay contribution of the receiver decoding process for video.

The total value of the delay in table x is substituted into the value of the processing delay in equation h. The sum of the processing delay is the total endpoint delay as shown in equation A.4.

$$Proc_{Endpoints} = Proc_{Rx} + Proc_{Tx} = 110.18 \text{ ms} \dots (A.4)$$

$$Proc_{Tx} = 14.72 \text{ ms from table A.3.}$$

$$Proc_{Rx} = 95.48 \text{ ms from table xA.4}$$

A.1.7 Adjustable threshold – 1-bit RLE

Figure A.13 shows the performance of the adjustable threshold system of the 1-bit encoding system. This figure shows that as this is user controlled and the thresholds are not relative, the depth information gain is abstract and hence less useful to a human operator. Figure A.13 does not allow the

back wall to be resolved and hence depth is still not well perceived. Even with the adjustment of the threshold, several objects do not become identifiable such as the papers, the robot body and boxes on the bench on the left side of the image. It is clear from the results shown here that only objects with a significant change in colour brightness are resolved with this method. Some objects such as the silhouette of the shelves and the robot wheels are easily identified due to the positioning of these darker items on a white background.

Figure A.13: This figure shows the effect of a rolling 1-bit RLE threshold. Thresholds are taken at 64 (top-left), 100 (top right), 128 (bottom left), 140, (bottom right)

The differing threshold allows the box over the shoulder of the hooded figure to be resolved. In the top-left frame, words and logos can clearly be resolved on the side. This shows the advantage of the adjustable threshold method, as these words were unable to be resolved with either higher or lower thresholds. This shows that there are certain features that can only be identified using an adjustable threshold system.

A.2 DIT compression scheme

A.2.1 Encoding Process

The process for encoding the video using the DIT system relies on acquiring the video data from the RealSense device as shown in section 6.1. This video is then transformed into a monochrome format

as in 6.4 and then filtered using the low-pass filter as described in section A.1.1. The DIT specific video processing system introduces the frame decimation step that separates the video data from 320 x 240 into four frames of 160 x 120 pixels. The spatial information is distributed through each of the four arrays in the two-dimensional array `decimation_frame`. A separate variable named `decimation_pos` allows the position of the data in the array to be maintained. This is shown in A.14.

```
for (long int k = 0; k < 76800; ++k){
    //change the row indicator
    if ((k % 320) == 0){
        //flip the row.
        if (even_not_odd == 1) even_not_odd = 0;
        else even_not_odd = 1;
    }

    //now put the pixels inside the frame.
    if ((k % 2) == 0){
        //even col
        if (even_not_odd == 1){
            //even col even row
            decimation_frame[decimation_pos[0]][0] = filtered_data[k];
            ++decimation_pos[0];
        }
        else {
            //odd row
            decimation_frame[decimation_pos[1]][1] = filtered_data[k];
            ++decimation_pos[1];
        }
    }
    else{
        //odd col
        if (even_not_odd == 1){
            //odd col even row
            decimation_frame[decimation_pos[2]][2] = filtered_data[k];
            ++decimation_pos[2];
        }
        else {
            //odd row
            decimation_frame[decimation_pos[3]][3] = filtered_data[k];
            ++decimation_pos[3];
        }
    }
}
```

Figure A.14: This figure shows the frame decimation process.

The encoding process of each of the four sub-frames uses the run-length encoding scheme described in A.1.2. Each frame is encoded with a different threshold in order to improve image quality.

A.2.2 Decoding Process

The only modification to the receiver process involves positioning each decimated sub-frame in the correct position in the larger frame. As there are four sub-positions in the frame, this means that each sub-frame persists for 4 frames or 100 ms before the process overwrites the data with new information. The decoding process is almost identical to that described in section A.1.4. The difference in the decoding process for the DIT scheme is to only change a quarter of the frame per received packet, as each packet can only decode a single sub-frame. Therefore, when a packet is received with a full sub-frame only a quarter of the image changes. The data pointer (*data_ptr*

variable) does not increment by one, but it instead increments by three. This skips every second pixel and the data pointer also skips every second row so each sub-frame is only decoded to a quarter of the overall pixels in the full frame displayed to the user. As each frame now arrives four times as fast, the overall refresh rate of the receiver frame can be quadrupled.

The decoding process first determines the position of the first data pointer, and then evaluates whether the data packets are extrapolated to even or odd pixel positions or even and odd rows. The starting data pointer positioning process is presented in figure A.15.

```
px_counter = 0;

//set the initial data pointer position.
if (frame_qtr == 0){
    data_ptr = 0;
}
else if (frame_qtr == 1){
    data_ptr = 3;
}
else if (frame_qtr == 2){
    data_ptr = 3;
}
else { // frame_qtr = 3
    data_ptr = 0;
}

//change the row select.
if (even_not_odd == 1){
    data_ptr += 960;
    even_not_odd = 0;
}
else{
    even_not_odd = 1;
}
```

Figure A.15: This figure shows the positioning of the data pointer offset in order to properly position frames in the decoder

The decoder process starts the data_ptr variable depending on which quadrant the frame is on. On even rows, the even_not_odd variable adds 960 (1 row offset) to the data_ptr every second frame. When the data_ptr is at position 3, this indicates that the data is being decoded to odd rows, and when the data_ptr is at position 0, this indicates that the data is being decoded to even rows.

The decoding process of the DIT encoding scheme adds additional filtering steps after the pixels have been decoded to the final image. Filtering is applied to interpolate the four sub-frames to reconstruct the image. A temporal filter is used to preserve motion by averaging the current frame with the preceding three frames. A temporal filter maintains each frame inside an array named filter_image as shown in figure A.16. The final output video is placed inside a ROS message named final_out and this is displayed on the screen. The temporal filter averages the current and past 4 frames in order to filter the equivalent of 1 full frame from the transmitter to blend four of the past frames together. A filter length of four is chosen as each frame is decimated into four sub-frames. In order to maintain full information, the filter process retains the last four frames to effectively allow a full frames' worth of

information to be displayed to the user. This process also moves the sliding window by adding the new data to the old data buffers. This is shown in the last 3 lines of the *for* loop in figure A.16.

```
//temporal filtering
for (long int j = 0; j < FULL_SIZE; ++j){

    //output the frame
    final_out.data[j] = (filter_image[0].data[j] +
        filter_image[1].data[j] +
        filter_image[2].data[j] +
        processed_video.data[j])/4;

    //roll the frame into the buffers.
    filter_image[2].data[j] = filter_image[1].data[j];
    filter_image[1].data[j] = filter_image[0].data[j];
    filter_image[0].data[j] = processed_video.data[j];
}
}
```

Figure A.16: This figure shows the temporal filter process.

A spatial filter is used as the final step before parsing the image data to the user. The spatial filter is identical to the filter described in section 6.2 in function. The spatial filter partially compensates for the reconstruction artefacts that exist when sub-frames of a different threshold are put together. By smoothing out the pixel discontinuities, a smoother image is attained. The effect of this spatial filter is shown in figure A.17.

Figure A.17: This figure shows the effect of the filtering process, without the temporal filter (left) and with the temporal filter (right)

A.2.3 Decimation in time compression performance

This section discusses the image quality attained by the decimation in time technique. Basic environments are examined with this method and a qualitative analysis is provided on the image quality. Figure A.18, A.19 and A.20 examines the performance of this video system in an office environment. These images explore the basic performance of the DIT method.

Figure A.18: This figure shows a view of around 10 metres to a bright window comparison view of the compressed video feed (left) and the raw image feed (right.)

Figure A.18 shows the effectiveness of the compression scheme in a known environment with sufficient context. The office with sunlight through the windows is an environment with contrasting light and dark. The compressed image clearly resolves these windows with the light gradient easily noticeable. Despite the blinds in the left side of the window being not visible, the overall shape of the room is revealed. Items such as the office chair are able to be identified, with the shelves and the dark computer screens also visible in the compressed feed. As is evident in figure A.19 and figure A.20 below, the colour difference of the boxes on top of the centre desk of brown against dark grey means that the boxes are visible in the uncompressed feed. However, due to the colour loss, the boxes appear to be merged with the table as the gradient between these items are similar. Of particular note in this figure is the fact that light sources are easily identifiable. This is a distinct feature that is highly important in a mine environment. The lamps and sunlight gradients are easily visible in this figure.

Figure A.19: This figure shows the comparison between the raw feed (right) and the compressed video feed (left) of a set of close up objects and a background.

This figure shows how two different objects of similar colour can be indistinguishable from one another even at close range. The two boxes on the left side of figure A.19 appear to be completely grey as they do not feature significant gradient differences. The only distinguishing marks between

the two boxes are the presence of the darker letters on the far box. Furthermore, the flat box behind the wireless nodes appears to be the same colour as the table. The shadowing effect on the edges allows the box to be partially visible through the silhouette effect. While this does resolve the box, this is not immediately obvious. The black cables on the bottom of the image are easily resolved against the whiter table, and the grey door is also visible. The fire extinguisher beside the door is easily identified. However, the darkening on the compressed video feed does not make the fire extinguisher immediately evident.

Figure A.20: This figure shows the comparison between the raw feed on a human object (right) with the compressed feed (left).

The gradient effect of the decimation in time technique adds additional detail to the background. The gradient of the table is easily identified. While improvements to the background wall are visible, the gradient here is less useful due to the non-recognisable nature of the pattern. When identifying close range objects, the decimation in time method still relies on contrasting colours to identify the overall shape of the object. When considering a human-sized object at around 50 cm as shown in figure A.20, the overall outline of the hooded figure is shown. The exact facial features are obscured due to the similarity in the gradient of the colours. Other disadvantages are found in that the boxes over the right shoulder of the figure are also unable to be resolved fully. The glare on the screen of the glass on the boxes is the only object that is easily resolved.

A.2.4 DIT vs. 1-bit RLE performance comparison

This section discusses the comparison between the 1-bit RLE and the DIT compression schemes examined in this chapter and chapter 6. Several comparison frames are provided to compare the images attained with these compression schemes. This section provides a qualitative analysis and comparison of the two compression systems. The following figures show the comparison of images obtained with both the 1-bit RLE and the DIT compression methods. Figure A.21 shows a bright doorway with a collection of objects. The distance from the camera to the door is roughly 5 metres.

Figure A.21: This figure shows the doorway comparison frame.

Figure A.22 shows the raw RLE and DIT video compression performance. The left image shows the RLE video stream with a threshold set at 127. This stream shows that the door and the whiteboard are easily resolved through their outlines. The image is crisp when hard discontinuities exist such as the resolution of the light switches on the wall as shown in the centre of the image. Furthermore, other details are also well resolved, such as the fire extinguisher beside the door and the fire extinguisher indicator sign above the door. These objects are easily visible as they contain elements that have colours that fall on either side of the threshold. An area of this image that is not so clear is the ceiling which appears to blend into the walls. It is clear that the darkness appears to connect the ceiling with the walls and this breaks up the outline of the walls and hence apparent the structure of the room. The light source in the corner booth is easily distinguished. However, the shading pattern on the drawers and the corner computer equipment makes resolving these objects difficult. The DIT stream shows a marked improvement.

Figure A.22: This figure shows the doorway when viewed using the 1 bit RLE (shown left) and the decimation in time method (shown right.)

The DIT algorithm with multiple thresholds in the sub-frames reveals a shadow gradient. This shadow gradient is shaped to the contours of the wall and hence the corners and edges of the room are better defined compared to the 1-bit RLE system. Also more evident is the positioning of the lamps as the light sources' also produces a light gradient as shown on the ceiling. The DIT method easily allows

the drawers and computer equipment to be resolved, as opposed to the RLE method which makes this equipment difficult to identify. In contrast with the RLE method, the DIT method does have some drawbacks. The light-switches and the well-defined outline of the fire extinguisher become less clear due to the gradients on the walls. The lines on the door and the fire extinguisher sign are less well defined due to the filtering process smoothing the image.

Figure A.23 shows that when observing a human entity, the outline of the silhouette can be easily identified with both the RLE and DIT methods. The RLE method shows a significantly clearer image of the silhouette as the DIT method results in vertical filter artefacts due to the interpolation process. It is clear that the additional depth and luminescence information is obtained at the cost of pixel resolution. The 1 bit thresholding attained using the basic RLE algorithm offers sharper and more accurate images, despite the loss of depth information.

Figure A.23: This figure shows the video performance of a human silhouette using the 1-bit system (left) and the DIT system (right).

Figure A.24 shows a field of view of an office room with additional windows in the windowed view to evaluate the image quality at ranges further away than 5 metres. This image demonstrates the performance of the respective video compression schemes with predominantly light areas with some dark areas such as the computer screens and the sections under the desks.

Figure A.24: This figure shows the office space comparison frame.

It is clear from figure A.25 that the DIT compression scheme offers a significantly improved video quality, allowing the windowed section to be visible with some depth perception offered by the

gradient shading. This additional quality also enables some other areas of the image to be resolved, such as the plastic bag that is on the floor beside the computer monitors. When viewed with the 1-bit RLE system, this object is difficult to identify. The added depth and colour resolution allows this to be easily identified. It is clear that in both images in figure A.25, the low frame resolution of 320 x 240 pixels makes identification of far-away objects difficult. One area that is equally difficult for both methods to identify is the person sitting at the computer on the left side of the image. Due to the obstructions and the overall changing nature of the video field, it is difficult to identify the objects in this area of the frame.

Figure A.25: This figure shows the office space when viewed using 1 bit RLE (left) and DIT (right).

The results shown in sections A.2.3 through A.2.4 shows a marked improvement using the decimation in time compression scheme over standard run-length 1-bit encoding. Several advantages are attained by the decimation in time compression. Improvements are observed in practically all areas including perceived frame-rate, delay, image quality, and jitter. However, some minor advantages are still present when using the 1-bit encoding scheme.

Frame-rate improvements are observed with the decimation in time compression technique. Because each frame is decimated into 4 smaller sub-frames, the refresh rate of the frames on the receiver side is increased by a factor of four. On a 10 Hz image generation rate, the frames appear to be refreshing at 40 frames per second. This is far above the visual motion threshold of 10 Hz which improves the overall cohesiveness of the video feed, even if no more information is provided by the transmitter. A frame-rate smoothing improvement is made possible by the use of the frame temporal filter that averages the last three frames to attain hues that are not restricted to discrete states. This filter also provides motion blur which allows motion to be visible through an after-image effect. When a frame changes, the last frames and the newer frames are blended which allows the old frames and the new frames to be shown simultaneously. This method provides the user with an indicator of motion and allows the user to potentially react to fast-moving changes.

When a start frame is lost in run length 1 bit encoding, the entire frame cannot be referenced to the start frame and 100% of the frame is lost. Even if a non-start frame is lost, full information about the image is entirely contained within that frame is lost. Spatial divisions in the decimation in time compression divides the frame into 4-sub frames. This allows the general image to be recovered even if one frame is lost in transmission. As opposed to losing the whole frame, losses in a frame will only cause 25% of the image to be lost. This means that successfully received sub-frames can always be decoded and this increases the overall good-put of video through the network. As the image is filtered with prior images on the receiver end, sub-frame loss becomes negligible over the long run. As images are no longer reliant on a single 'K' frame from reaching the receiver, the decimation-in time technique is more loss-tolerant than the 1-bit run length encoding system. The decimation in time technique divides a frame evenly across its sub-frames, as opposed to the run length encoding which divides frames sequentially. Therefore, a dropped frame divides the lost information evenly across the frame as opposed to in sequential blocks. The effect of this loss is then filtered by the spatial filter on the receiver.

Frame-rate jitter is improved by this process by dividing up the information both spatially and temporally. A full frame loss in the prior 1 bit run length encoding reduces the frame-rate to 9 Hz. This constitutes a 10% drop in frame-rate. When a full sub-frame is lost in the decimation in time compression, the refresh rate reduces from 40 to 39 Hz, which is a drop of 2.5%. Frame loss does not constitute a major drop in frame-rate which is a major advantage in a wireless environment with packet loss. A vastly improved jitter tolerance is attained, mitigating the effect of frame loss on the smoothness of the video stream. This provides a vital frame-rate stability function. Because the RLE process employed in both methods cannot tell whether the packet will go over the 2000 kbps limit until it performs the process, the total packet size is unknown until the packet is actually transmitted. When a frame exceeds this limit, the frame takes longer to transmit, reducing the frame-rate. Overall, the entire packet takes longer to view on the receiver side. The decimation in time method improves on this as if a frame increases to over the limit, the refresh rate of the frames is not heavily affected.

Image quality is vastly improved over the 1-bit system. The decimation in time technique allows the frame to be more detailed as each sub-frame is encoded with a different threshold. This thresholding increases the overall spatial resolution of the frame. As the numerical thresholds for the 4 frames divides the frame into thresholds of four, system technically has 4 states per sub-frame and hence 16 states per 2 x 2 block of pixels. This is effectively a 2-bit colour system. The DIT technique combined with the moving threshold allowed the image quality to be vastly improved. This allowed a user to resolve many more features compared to the moving threshold technique implemented with just 1-bit RLE. The increased gradients and number of colour stats vastly improves the depth resolution and object resolution rate.

There are several areas where the DIT method does not improve on the RLE method. The image spatial quality is improved, but the DIT method does add some vertical image artefacts due to the interpolation process. A better filter can be added on the receiver side to further mitigate this. However, due to the filters on the receiver side, some smaller features can end up smoothed out such that they are unresolvable. This is more clearly shown in figure A.19 where the fire extinguisher is more difficult to identify in the DIT method than the RLE method. The RLE 1-bit system does have an advantage in that the image is sharper without the filter and smaller features are more readily identified in cases where vital features fall on either side of the threshold as in the case of figure A.22. A level of quality that is attained by neither the DIT nor the RLE method is facial and human recognition. While the silhouette of a human can be identified, facial and other features are difficult to observe. It is clear that features such as eye colour and the shape of a human's face are features that are not easily deciphered, even with a moving threshold as shown in figure 7-18. These features are highly vital in identifying survivors, and future work to improve video such that identities can be resolved would significantly improve the video system.

The comparative delay characteristics between the raw 1-bit RLE and DIT system is shown in figure A.26. This bar-graph shows the relative delay incurred by each endpoint of the 1-bit RLE and implemented DIT systems.

Figure A.26: This figure shows the endpoint process delay comparison between 1-Bit RLE and DIT.

Figure A.26 shows that the DIT method's combined delay characteristic is vastly improved over the 1-Bit RLE delay characteristic. The total (encoder + decoder) delay for the DIT system is roughly half of the combined delay of the 1-Bit RLE system. The DIT system encoder process costs roughly 12% more time as it requires additional steps such as the pre-frame filtering and the decimation process. However, due to the nature of sub-frame decoding, the time required to decode the frame on the

receiver process is reduced. A delay savings of approximately 60% is attained by the DIT method over the RLE method.