Alexandra Riddell-Webster

A MANET to Facilitate Collision Avoidance in Rowing Boats

Computer Science Tripos – Part II

Murray Edwards College

2023

Declaration of Originality

I, Alexandra Riddell-Webster of Murray Edwards College, being a candidate for Part II of the Computer Science Tripos, hereby declare that this dissertation and the work described in it are my own work, unaided except as may be specified below, and that the dissertation does not contain material that has already been used to any substantial extent for a comparable purpose.

I am content for my dissertation to be made available to the students and staff of the University.

Signed: Alexandra Riddell-Webster
Date: April 4, 2023

Acknowledgements

This dissertation owes a huge amount to Matthew Ireland, for supervising me. My UTO, Jon Crowcroft was invaluable. Thanks to Cambridge University Boat Club, in particular Patrick Ryan, for allowing me to stick devices on rowing boats and for advising me on communication over water. I also thank Duncan Barnes for discussing GPS and electronics on rowing boats with me.

Proforma

Candidate Number:	TODO
Project Title:	A MANET to Facilitate Collision Avoidance in Rowing Boats
Examination:	Computer Science Tripos – Part II, May 2023
Word Count:	TODO [1]
Code Line Count:	TODO (inc. comments?)
Project Originator:	Alexandra Riddell-Webster
Supervisor:	Mr Matthew Ireland
University Teaching Officer:	Dr Jon Crowcroft

Original Aims of the Project

TODO

Work Completed

TODO

Special Difficulties

None.

[1] TODO

Contents

Introduction

This project built a mobile ad-hoc network (MANET) to facilitate collision avoidance amongst rowing boats. This MANET uses the geographic coordinates of boats and obstacles to determine if a boat is too close to an obstacle, then warns the user by buzzing or flashing an LED. A modified version of the Epidemic routing protocol has been implemented and nodes built to use it. The Evaluation chapter quantifies this implementation of Epidemic, then assesses the entire system. The Conclusion details further work that could be undertaken on the project. The project is open source, with a 'how to' guide [Appendix A] for construction of a node and the code available on the internet (TODO CITE) so any rowing club can use this as a collision avoidance tool.

1.1 Motivation

It is obvious that collisions between rowing boats and obstacles or other boats is unwanted, injuring rowers and causing equipment damage. While coxswains are responsible for steering a boat, they are fallible and may need extra assistance, particularly when steering in adverse conditions, such as fog. Coxless boats are at greater risk of collision, where rowers, facing away from the direction of travel, are responsible for steering boats. This project can be used in both coxed and coxless boats. This project has a very personal motivation, as a friend was hit by an eight while in a single three years ago, causing a severe concussion that resulted in two years of intermission from her studies.

This project produces devices designed to reduce the number and severity of collisions, warning users before they crash. While researching this project, I found a previous attempt to solve the problem, ROWCUS (TODO cite Rowcus). ROWCUS produced a device with radar to detect potential collisions. It seems to have proven commercially inviable, as the company states they "have decided to not pursue the commercial deployment of ROWCUS" (TODO cite). I am a proponent of open source, so the project is available entirely online. This includes a 'how to' guide to building a node [Appendix A], so others can build the solution.

1.2 Related Work

As mentioned above, ROWCUS (TODO cite) has attempted to solve this problem with a different technical solution – using radar rather than GPS to detect proximity to obstacles, and without networking. While ROWCUS has similar goals to my project, the technical methodologies are different. The main differences are use of GPS and networking – my project connects nodes together while ROWCUS has individual nodes.

The nodes in my project communicate known obstacles to each other. Vahdat and Becker's paper 'Epidemic Routing for Partially-Connected Ad Hoc Networks' is the key paper used to implement routing within the MANET. The term 'anti-entropy' as used in my dissertation, comes from this paper. An anti-entropy session occurs when two nodes come into communication range and exchange messages. While the implementation in my project differs from this paper in some areas, including medium access control (detailed in the Preparation and Implementation chapters), this paper forms the foundation for the project.

Preparation

This chapter is laid out in chronological order. It first details the work done before the start of term, then the analysis that lead me to decide on the Epidemic routing protocol, particularly analysis of the structure of the networks rowing boats generate. This chapter finishes with the state machines designed after the project was accepted.

2.1 Starting Point

Previous to this project, I had no hands-on experience with microcontrollers, although I had worked with Raspberry Pi single board computers. I had previously worked with AdaFruit components, however not with the radio and GPS boards used in this project. I therefore dedicated a small amount of time over the summer to learning about microprocessors and MicroPython, completing basic tasks such as flashing an onboard LED. In hindsight, while this was useful, my project is implemented in CircuitPython, so it would have been more beneficial to have explored this.
Before term started I felt it was important to verify the hardware I would use, particularly as there were chip shortages at the time. I therefore ordered and ran basic tests on the Raspberry Pi Picos and AdaFruit RFM69 radio and GPS boards. I chose to use AdaFruit and Raspberry Pi boards as there is a strong community surrounding the hardware, and the AdaFruit boards are supported by open source libraries that allow rudimentary operations to be performed.

I had some experience with networking and routing protocols prior to starting this project. This was composed of the Part IB networking module and a small amount of work during an internship. Throughout the course of my project I took the Part II principles of communications course, further expanding my knowledge. I also researched MANETs and their corresponding networking protocols during the summer vacation. Most of this research was centred around the Better Approach to Ad-Hoc Mobile Networking (BATMAN) protocol (TODO cite) and other networking protocols that are not delay-tolerant. My project implemented an extended version of the Epidemic routing protocol from scratch.

2.2 Networking

A MANET is characterised by wireless nodes, a frequently changing network topology and no reliance on pre-existing infrastructure. They are decentralised and therefore have no single point of failure (TODO cite D2). MANETs have a large range of uses, such as facilitating communication in military conflicts (TODO cite D3) to autonomous vehicles (TODO D4) and disaster relief scenarios where previously existing infrastructure is destroyed (TODO cite D5). This project constructs a MANET due to the lack of pre-existing infrastructure and the difficulties associated with setting up and maintaining a base station or similar. Requiring an infrastructure like this would also raise the barrier for entry for many clubs with limited funds and technical skills.

Routing protocols find a path from a source to one or more destinations destination within the network. Different routing protocols optimise different parameters, and are better suited for different network topologies and applications (TODO cite princomm lecture notes). Within MANETs, a routing protocol must allow the network topology to change over time. They tend to contain node discovery techniques to allow for this.

Before deciding on Epidemic as the routing protocol to implement for my project, I considered several other protocols. The the Better Approach to Ad-Hoc Mobile Networking (BATMAN) protocol (TODO cite) designed to route messages through MANETs, broadcasting originator messages (OGM) for node discovery. BATMAN has the interesting addition of a transmit quality (TQ) metric in the OGM packets, allowing the quality of connections between nodes to be factored into the route packets take through the network. While BATMAN does allow messages to be broadcast to all nodes, its primary focus is routing messages from one node to another. Additionally, while it allows for message mobility, it is not delay-tolerant.

Another routing protocol examined was Greedy Perimeter Stateless Routing (GPSR), a location based routing protocol (TODO cite). GPSR exploits the relation between geographic position and connectivity in a wireless network, where each node tells its immediate neighbours its current location. Greedy forwarding is predominantly used to send packets to nodes that are progressively closer to the destination, until the destination coordinates can be reached. Where greedy forwarding fails, GPSR uses perimeter forwarding (forwarding the packets around the perimeter of the region) until greedy forwarding can be used again. This protocol was ultimately deemed to be unsuitable for my project as, similarly to BATMAN, its primary focus is in sending messages between two nodes. Additionally, the high mobility of the nodes in the use case means that forwarding packets to a set of coordinates does not mean the message will reach the intended destination, as the node may have moved.

I chose to implement Epidemic in my project. Epidemic routing gains its names from its similarity to the spreading of infections. Each node replicates and transmits messages to neighbours that have not recently been contacted. These neighbours are discovered when each node broadcasts its existence to its neighbours. Epidemic was implemented in this project because it is a delay-tolerant routing protocol and best fits the likely network topology generated by rowing boats, as detailed in the Requirements section below. The networks generated by rowing boats have a high chance of partition, but the nodes are highly mobile. As Epidemic allows any node to carry network information it is best suited to the network. Also something about how Epidemic does not route messages node to node but rather everywhere which is better suited to the use case.

While these routing protocols often have mechanisms to minimise the probability of collisions, such as adding a jitter in the sending of discovery messages, none of them contain medium access control. Due to the probability of collisions between messages causing disruption to the sending of messages, particularly as all nodes are broadcasting on the same radio frequency (433 MHz). It was therefore prudent to include media access control in the system. Multiple Access with Collision Avoidance for Wireless (MACAW) is often used by ad-hoc networks. MACAW uses request to send (RTS) and clear to send (CTS) messages to minimise the probability of collision. As discussed in the System Design section, MACAW inspired the medium access control in my project.

2.3 Project Development

This project was structured using the waterfall model for software development. The project proposal [Appendix D] lays out the phases of the project and the output from each stage in a plan of work. The waterfall model of software development was used as it lends itself to the structure of a dissertation, with the five stages:

Requirements → System Design → Implementation → Testing → Maintenance

This project used CircuitPython, as it is compatible with the existing AdaFruit libraries I used and extended in the project, and can easily be run on the Raspberry Pi Pico.

GitHub was used to perform version control on the code and dissertation for this project. This also allowed me to fork repositories and submit a pull request to AdaFruit.

2.4 Requirements

My requirements were based around the use case for the project – as devices attached to rowing boats. This factored into my hardware decisions, as I wanted to use small and relatively lightweight items. Additionally, I chose to use the 433 MHz band as it is free to use without licencing, allowing me and other rowing clubs to use it without additional bureaucracy.

The potential topology of the network was then analysed. This was done by looking at example distributions of rowing boats on lakes and rivers. One potential use case for this network that I am familiar with is the river Thames. Below I have analysed Google Maps and Earth's satellite view of the Thames along a 5.5km stretch of the Championship Course, pinpointing rowing boats and coaching launches, then adding them to a map with potential connections between nodes, assuming the radios have a range of 500m, alongside obstacles. This process is shown in Figures 2.1 and 2.2. I then abstracted this to the network topology of these rowing boats, shown in Figure 2.3.

Figure 2.1: A rowing boat seen on Google Earth (TODO cite)

Figure 2.2: Rowing boats, potential obstacles and assumed connections marked on a Google Maps map of the river Thames

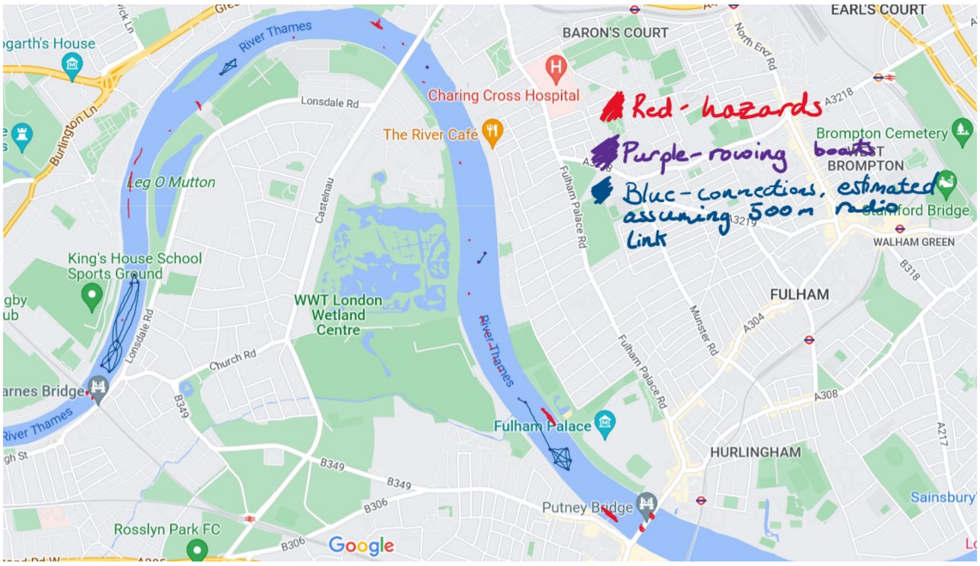
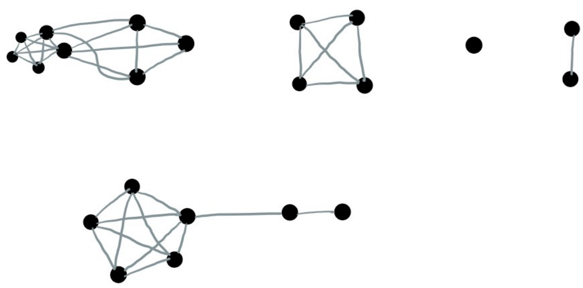


Figure 2.3: The network of rowing boats in an abstracted form



The final three requirements I identified as my success criteria were:

1. The Epidemic routing protocol is implemented on the network
2. An application layer to demonstrate the utility of the network has been implemented
3. An evaluation of the performance of the network has been carried out

All of these requirements have been implemented during this project. The finer details of this are in the Implementation chapter.

I laid out several extension tasks. Some of them were drawn from my research into networking protocols other than Epidemic. For instance, I wanted to use a metric for transmission quality – received strength signal indicator (RSSI) in radio communication – to influence whether an anti-entropy session was initiated. Additionally, the GPS location could be used, either by only contacting nearby nodes to increase the probability that an anti-entropy session is successful, or prioritising sending messages about new obstacles to nodes that are near these obstacles. While time constraints have not allowed me to implement these extensions, I have implemented an extension allowing messages to have two priorities – normal and urgent. This is also detailed in the Implementation chapter.

2.5 System Design

Having decided on the Epidemic routing protocol, I planned the structure of the software that would run on each node. I knew that the Raspberry Pi Pico uses the R TODO chip, containing two cores. To make best use of the hardware, I therefore decided to design an application and networking state threads that would run on each core, with global data structures and concurrency control to pass messages between the two layers.

The application thread is responsible for notifying the user when they are too close to an obstacle. TODO state machine.

The networking thread contains the implementation of Epidemic with medium access control. how the Epidemic one changes from the Epidemic proposed in the initial paper. Layering considered harmful and such here

The data structures designed to communicate messages between one thread and the other are

While these state machines were broadly implemented, the overall structure of the project changed, with only one thread running due to the limitations of CircuitPython. This decision and the reasoning behind it is detailed in the Implementation chapter.

Implementation

Evaluation

Conclusion

Appendices

A – Guide to Building a Node

B – Evaluation

Part II Project - Plan for Evaluation

March 2023

Overview

The evaluation for the mobile ad-hoc network (MANET) will be split into two parts – first the evaluation for the pure MANET and a whole system evaluation. The evaluation of the MANET will examine the implementation of Epidemic, the delay tolerant networking protocol will be checked for correctness and performance. The system evaluation will look at the MANET in the context of the use case, on the water.

All data will be logged on the node in a CSV file with columns
`Node Address, Time, Time Since Node Startup, Event Type, Event Information`
where the Event Information varies with the event being logged and may contain the GPS location and message keys. This data will then be analysed on my device using Python code.

Evaluating the MANET

These tests be conducted in a field, as this will give an outdoor environment similar to the use case, but I will be able to better control the conditions the node is in. They will use four nodes, as this is the maximum number of nodes we can accurately calculate time for. Two nodes will use GPS to find the current time and two will use the serial connection to laptops to calculate the time. These tests will be conducted first on a small scale, with short tests using a small number of nodes, to ensure the tests can be run. After this has been confirmed, the tests will be run for a longer time with the maximum number of nodes.

The tests can then be compared to the evaluation in the initial epidemic paper [3], which simulated the nodes with 50 mobile nodes in a 1500 x 300 m space using the Monarch extensions to the ns-2 packet-level simulator rather than hardware as I am doing. Evaluation metrics examined included message delivery latency, delivery rate, the average and maximum number of hops a message took to get to a node.

The first test will be the percentage of packets delivered in the four node network. This will be time limited (i.e. if a message is not received in x minutes, it is considered undelivered). Nodes will randomly generate a new message every second, for a total of 25 messages, mirroring the structure of testing used in Vahdat and Becker's paper [3]. The nodes will be in a box of area $20m^2$ with the range of the radios reduced to approximately $5m$ to simulate the environment in which the system will be used. The nodes will move constantly.

Next, the transfer delay will be measured. This will be the average time taken for each message to be delivered, and will use the same setup as percentage of packets delivered testing.

Finally, the time taken to propagate messages after partition will be measured. This will include one-sided, asymmetric partitions, where only one set of nodes has messages the other set has not seen. It will also include symmetric partitions, where both sets of nodes have messages the other set has not seen. The number of unseen messages will be increased and each test will be run five times.

Figure 1: The setup and axes for delivery rate and transfer delay

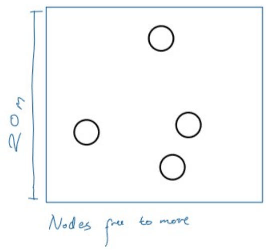
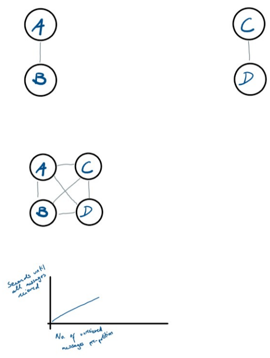


Figure 2: The setup and axes for partition testing



Evaluating the System

This will be performed on the water, the environment the MANET will be used in, after a 'dry run' on land to ensure there are no obvious flaws with the system. To examine the use of the system and how long it takes messages to arrive at other nodes, a well known obstacle has been selected. This is the red buoy at 51.48236626931181, -0.22641424521527762, shown below [1]. Using the time given from the GPS chips, we can then map the locations and time since the obstacle was added that other nodes receive messages about the buoy. This experiment will be run multiple times to gather a sufficient volume of data about the propagation of new obstacles.

Additionally, this will allow me to look at the behaviour of users when adding a new obstacle then tweak the MANET to fit. While there is a ground truth about the location of an obstacle, it is likely that most users will not be directly above the obstacle when they log it.

Figure 3: The location of the 'red buoy' obstacle [1]

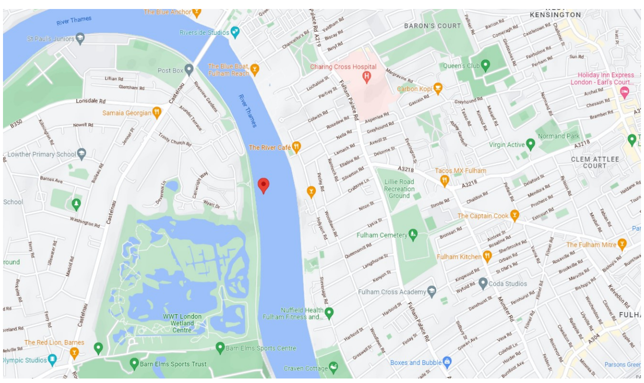


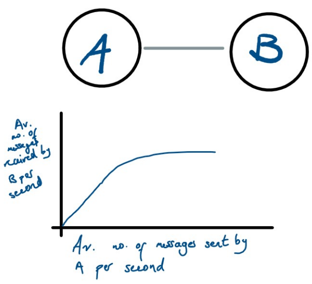
Figure 4: An image of the 'red buoy' obstacle [2]

Extensions

If there is sufficient time, further evaluation can be performed on the MANET. This will be structured in a similar way to the tests in the 'Evaluating the MANET'. Bandwidth will be tested in a fully connected network, where the number of messages per second transferred between two nodes may be found by generating random packets at set intervals at one node (node A shown below), and seeing how many are passed to another node (node B below). This test could then be performed with both nodes generating and receiving messages, to examine bandwidth on a two way connection.

Further extensions to the system evaluation will include a questionnaire for those who use the

Figure 5: The setup and expected graph for testing bandwidth



device, covering a range of users, including both coxes in coxed boats and rowers in coxless boats.

References

[1] Google Maps. 51.4823, -0.2264. [Online] Available at: https://www.google.co.uk/maps/place/51%C2%B028'56.5%22N+0%C2%B013'35.1%22W/@51.4820341,-0.2295313,15.5z/data=!4m4!3m3!8m2!3d51.4823663!4d-0.2264142!5m1!1e4 [Accessed March 2023]

[2] Riddell-Webster, A. Red buoy. Taken March 2023.

[3] Vahdat, A, Becker, D. Epidemic Routing for Partially-Connected Ad Hoc Networks. Published 2000. Duke University.

C – Progress Report

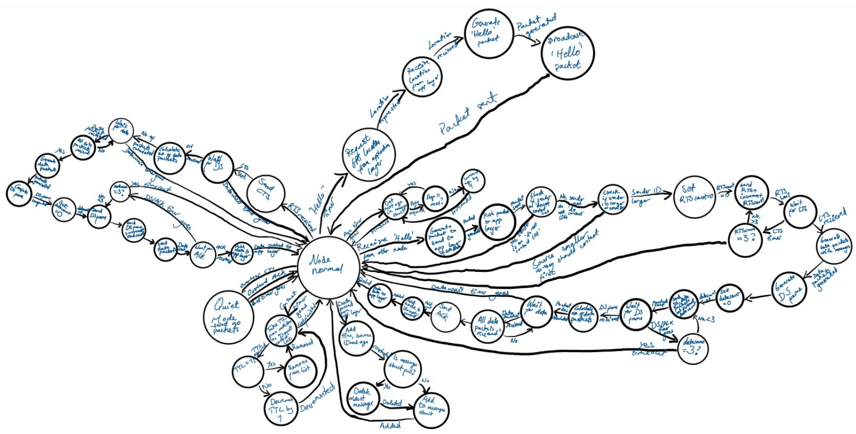
Part II Project - Progress Report

April 4, 2023

Name: Alex Riddell-Webster
College: Murray Edwards
Email Address: ahr38@cam.ac.uk
Director of Studies: Luana Bulat
Supervisor: Matthew Ireland
UTO: Professor Jon Crowcroft
Overseers: Ferenc Huszar and Andreas Vlachos
Title: A MANET to Facilitate Collision Avoidance in Rowing Boats

My project, a mobile ad-hoc network (MANET) to facilitate collision avoidance in rowing boats, attempts to reduce the frequency of rowing boat collisions. The technical core of the project is the Epidemic routing protocol [1], modified to include medium access control based on the Multiple Access with Collision Avoidance for Wireless (MACAW) protocol [2]. Both medium access control and Epidemic are implemented in the by the same state machine, to simplify the implementation and prevent work being repeated – a waste of the limited resources on the Raspberry Pi Pico [3]. The initial state machine is shown in Figure 1, although it has been changed during implementation. Most notably, data send (DS) packets have been removed and the information being put into the data packets to reduce the number of packets sent.

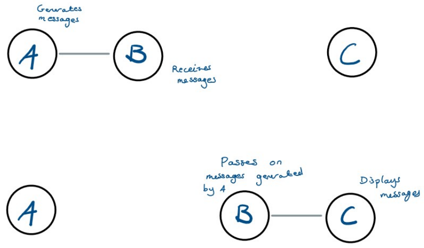
Figure 1: Network state machine



Construction of the MANET is going well. It has been constructed in hardware, working with the Raspberry Pi Pico, an ARM-based microcontroller without an operating system [3] and RFM69 radio. I have finished the networking machine so am currently tweaking and evaluating the network state machine while finishing the application machine. As the network has a physical implementation, I intend to test the network on rowing boats, the environment it would be used.

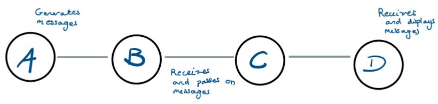
To ensure the network was delay tolerant, I ran a test with three nodes, A, B and C. At the start, nodes A and B were in range of each other and node C was out of range. I set node A up to generate random messages every 40 seconds, with a time to live (TTL) greater than 2 so the messages would survive for two 'hops' across the network. I then moved node B out of range of node A and into range of node C, which then displayed any messages it received so I could check that they were the same as those generated by A, with a reduced TTL. Figure 2 shows the setup.

Figure 2: Using three nodes to check the network is delay tolerant



Another test involved four connected nodes, A, B, C, and D, where the transmit power of each node was significantly reduced, so each node had at most two connections. Node A generated messages, and I checked to see if D received them. I will use a similar set up in the future to test the percentage delivery and latency of packets. The setup is shown in Figure 3.

Figure 3: Using four nodes to check the network can transfer packets over several links



The most significant obstacles have been in radio communication. The FIFO buffer on the RFM69 was occasionally being overwritten as the controlling library was not clearing the FIFO. I modified the library to clear the buffer and allow the sending of fixed length packets. Changing to fixed length packets (64 bytes, the maximum length of the FIFO) allowed for more reliable communication. Additionally, CircuitPython (the language in which AdaFruit's libraries are written) does not support interrupts. To work around this, I poll to see if a condition is met when a corresponding timer elapses.

Given the work completed so far, I am two weeks behind the timetable laid out in October. As I am working on the application machine and evaluation of the network in parallel, the project will likely be back on timetable by mid-February.

The remaining work is first to finish the application machine and evaluate the MANET. Evaluation metrics will include the percentage of received packets, transfer delay and variance, and the time taken to propagate messages in a previously segmented network. Finally, I will pull the application and network machines together, running them on the two cores in the Pico, with concurrency control over key data structures and the GPS chip. A concern here is the Adafruit Blinka libraries allowing interoperability between CircuitPython and MicroPython [4], given the errors and incompleteness found in other libraries.

References

[1] Epidemic Routing for Partially-Connected Ad Hoc Networks. Vahdat, A, Becker, D. Duke University. 2000.

[2] MACAW: A Media Access Protocol for Wireless LAN's. Bharghavan, V, Demers, A, Shenker, S, Zhang, L. ACM SIGCOMM Conference. 1994.

[3] RP2040 Datasheet A microcontroller by Raspberry Pi. Raspberry Pi Ltd. 2022.

[4] GitHub - adafuit/Adafruit_Blinka: Add CircuitPython hardware API and libraries to MicroPython & CPython devices. https://github.com/adafruit/Adafruit_Blinka. Adafruit Industries. GitHub. Accessed January 2023.

D – Project Proposal

Part II Project – Project Proposal

October 14, 2022

Preliminary Information

Name: Alex Riddell-Webster
College: Murray Edwards
Email Address: ahr38@cam.ac.uk
Director of Studies: Luana Bulat
Supervisor: Matthew Ireland
UTO: Professor Jon Crowcroft
Overseers: Ferenc Huszar and Andreas Vlachos
Title: A MANET to Facilitate Collision Avoidance in Rowing Boats

Introduction

My project will implement routing in a Mobile Ad-Hoc Network (MANET).

Routing protocols find a path from a source to one or more destinations destination within the network. Different routing protocols optimise different parameters, and are better suited for different network topologies and applications [1].

A MANET is characterised by wireless nodes, a frequently changing network topology and no reliance on pre-existing infrastructure. They are decentralised and therefore have no single point of failure [2]. MANETs have a large range of uses, from the military [3] to facilitate communication, to autonomous vehicles [4] or disaster relief scenarios [5] to gather and move data across locations where previously existing infrastructure has been destroyed.

My project wishes to use a MANET to share a set of locations throughout a set of rowing boats, in order to facilitate collision avoidance. Collisions in rowing can cause damage to both rowers and their equipment. This project's motivation is to avoid rowing boats colliding with each other and with other obstacles. A radar and AI based obstacle detection system exists [6]. However, to the best of my knowledge, collision avoidance has not been attempted by networking boats together. My project will represent each boat as a node in a network. Each node will store a set of locations the user should be warned about; passing location data throughout the network will be the technical core of the project.

My project will be implemented in hardware, in the real world. In general, networking protocols can be implemented in simulation or in hardware. Depending on the nature of the simulation environment, it might not be possible to use exactly the same code in the simulator as in the real hardware. Due to the time constraints on a Part II Project, I intend to implement my project only in hardware.

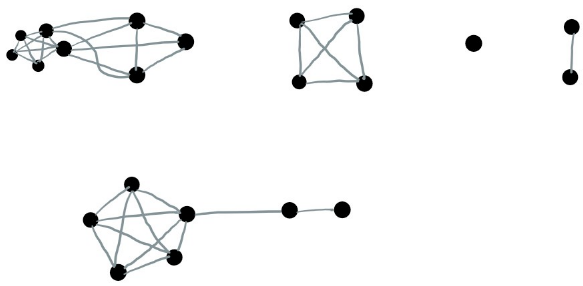
As stated in the cover sheet, Human Participants will be used to help test and evaluate the project. This will comprise a few volunteers to row boats, allowing the network to be run on the water. These volunteers will all be members of Cambridge University Boat Club, able to safely row a boat and navigate the river where the network is being tested.

Structure

The first part of the project will be dedicated to research, looking at the Epidemic routing protocol. Epidemic was chosen as it is a delay tolerant routing protocol, and best fits the likely topology of networks generated by rowing boats. There is a high chance of partitions in these networks. However, the nodes in the networks will be highly mobile, meaning that data can still be transferred through the network through exploiting this mobility. A potential topology for the network, generated from looking at satellite images of a 5.5km stretch of the river Thames [7], is shown in Figures 1 and 2, both on a map and as an abstracted topology.

Figure 1: Rowing boats, potential obstacles and assumed connections marked on a Google Maps map of the river Thames

Figure 2: The network of rowing boats in an abstracted form



The first research phase will also decide on any changes that need to be made to the routing protocol to make it better suit the application. Finally, the first research phase will refine the evaluation metrics needed for the project.

The second research phase will look at microcontrollers, particularly the Raspberry Pi Pico. It will also consider multi-threading on the Pico and in CircuitPython and how this can be exploited to most effectively implement my Part II project.

After the research phases, I will implement point to point communication between two neighbouring (in radio connection range) nodes in hardware.

The next part of the project will implement broadcast and controlled flooding routing between at least three nodes. A packet format will be defined as part of this, as flooding will form the start of the Epidemic routing protocol.

After a flooding protocol has been implemented, I will build on it to implement the Epidemic routing protocol. The code will be written in a two week block, then tested on hardware in a third week.

The application layer will then be implemented. This will ensure that the information passed through the network can be used. The application layer should warn the user when they are approaching an obstacle, and allow the user to add obstacles, which are then passed to the routing layer to be propagated through the network. My project aims to keep the routing and application layers separate for ease of construction, testing and evaluation.

The hardware will then be tested, tweaked and evaluated. Correctness will be evaluated first on land, likely in a field, where analysis of the network is easier and larger numbers of metrics can be examined than in the use environment. Performance will be evaluated in the use environment - on rowing boats on the water. I intend to evaluate the routing and application layers separately. Evaluation of the network will likely consider connected and partitioned instances of the network separately. Evaluation is likely to look at the time taken for the network state to be flooded through the nodes, and routing tables then updated, as well as a packet loss ratio [8]. Further evaluation would conduct some case studies, tracking a packet through the network and ensuring no unnecessary latency is added. The power consumption of each node could also be measured, giving a proxy measure for the traffic passing through each node. Due to time constraints, I consider these further evaluations to be extensions.

Success criteria

There are three success criteria I will hold for my project:

1. The Epidemic routing protocol is implemented on the network
2. An application layer to demonstrate the utility of the network has been implemented
3. An evaluation of the performance of the network has been carried out

Extensions

The success of my project will be defined by completion of the core criteria listed above. If there is time, I have set further challenges:

1. Case studies on the path and timing of individual packets are performed
2. The network is further evaluated by examining the power consumption of individual nodes as a proxy metric for traffic passing through a node
3. The User Interface of the device is evaluated
4. The application layer is further enhanced, using heuristics and extra data such as angle of attack from GPS and combining sensor data
5. The routing layer is further enhanced by passing additional data, such as location awareness, to the routing protocol

Plan of Work

Start of Block	End of Block	Block Length	Notes	Work to be Done	Milestones
14/10/2022	21/10/2022	7		Research - how to implement Epidemic protocol, evaluation methods used for ad-hoc networks	Develop a greater understanding of the Epidemic protocol and a plan for implementing it, create an evaluation plan
21/10/2022	28/10/2022	7		Learning how to use the microcontroller and boards	Ensure all the necessary hardware is available, develop a greater understanding of the Raspberry Pi Pico and CircuitPython
28/10/2022	04/11/2022	7		Start to work with the hardware - implement point to point communication between two nodes	Two nodes can send point to point messages
04/11/2022	18/11/2022	14	07/11 - Robotics Assignment 1	Implement controlled flood routing	Messages are flooded between at least three nodes
18/11/2022	02/12/2022	14	18/11 - 4s head; 28/11 - Robotics Assignment 2	Implement Epidemic routing protocol	Routing state information is shared between at least two nodes, Epidemic is implemented
02/12/2022	10/12/2022	8		Test, tweak and debug Epidemic implementation on hardware	Epidemic is implemented on hardware
10/12/2022	26/12/2022	16	14/11 - Trial 8s; Christmas	Time off	-
26/12/2022	02/01/2023	7	01/01 -> 11/01 - Camp	Implementing application layer - read location data from GPS and warn user when approaching known obstacle	The device warns the user when they are approaching a known obstacle
02/01/2023	16/01/2023	14	01/01 -> 11/01 - Camp	Implementing application layer - transfer data between the application and routing layers	The application layer is implemented on hardware
16/01/2023	23/01/2023	7		Tweaking the hardware, testing point to point links on land	The hardware runs on land, finish proof of concept
23/01/2023	30/01/2023	7		Water testing and tweaking	The hardware is implemented and run in the application environment (water)
30/01/2023	03/02/2023	4	03/02 - Cybercrime 1	Write progress report and presentation	Progress report and presentation
03/02/2023			**Progress report and presentation**		
03/02/2023	14/02/2023	11		Evaluation and tweaking on land	The hardware is evaluated for correctness on land
14/02/2023	21/02/2023	7	17/02 - Cybercrime 2	Evaluation and tweaking on water	The hardware is evaluated for performance on water
21/02/2023	07/03/2023	14	03/03 - Cybercrime 3	Dissertation - plan and bullet point what will be said	Dissertation bullet point form (first draft)
07/03/2023	21/03/2023	14	17/03 - Cybercrime 4	Dissertation - write out preparation and implementation	Dissertation has implementation and preparation written out
21/03/2023	04/04/2023	14	26/03 - Boat Race	Time off	-
04/04/2023	18/04/2023	14		Dissertation - write introduction, conclusion, evaluation	Dissertation fully written, sent to supervisor to proofread (second draft)
18/04/2023	02/05/2023	14		Dissertation - Take on criticism, add references and appendices	Dissertation - final draft
02/05/2023	12/05/2023	10		Contingency	-
12/05/2023			**Final deadline**		

Starting Point

I have a little experience with networking and routing protocols. My experience is limited to the Part IB networking module, although it is being expanded by the Part II Principles of Communications module and my research. I will need to add to my knowledge of networking and routing protocols.

I have previous experience using Raspberry Pi single-board computers with AdaFruit boards. I have no previous experience with microcontrollers. I will need to improve my knowledge of microcontrollers to complete this project, something I have set aside time for in my Plan of Work.

Resource Declaration

I plan to use my laptop to implement, evaluate and write up the project. It has a comprehensive system of backups through OneDrive and disk images. A backup of the project will exist with Git version control, hosted on GitHub. My own hardware, including Raspberry Pi Picos, breadboards and AdaFruit radio and GPS modules will be used to develop and implement the project.

Libraries to interface with the AdaFruit boards are written by AdaFruit in Circuit Python, and in my experience tend to be robust, although they occasionally contain bugs. If necessary, I can fork the code and implement bug fixes.

My project will partially rely on the correctness of routing protocols, work that others have already published. [9]

As the project has a real-world implementation, I have permission from Cambridge University Boat Club to test devices on their boats.

References

[1] 2022 Part II Lecture Notes on Principles of Communications. Crowcroft, J. 2022.

[2] Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. Corson, S. and Macker, J. Network Working Group. 1999.

[3] MILCOM'09: Proceedings of the 28th IEEE conference on Military Communications. Richardson, K, Jimenex, C, Stephens D. 2009.

[4] AD HOC Networks for the Autonomous Car. Davidescu, R, Negrus, E. IOP Conference Series Materials Science and Engineering. 2017.

[5] MANET for Disaster Relief based on NDN. Jin, Y, et all. IEEE. 2018.

[6] DEVICE | rowcus. https://www.rowcus.com/device. ROWCUS. Retrieved 07/10/2022.

[7] Hammersmith Bridge - Google Maps. https://www.google.com/maps/place/Hammersmith+Bridge,+London/@51.4883478,-0.2302753,17z/data=!3m1!4b1!4m5!3m4!1s0x48760fb3f5c78f85:0x932267a23 8m2!3d51.4883478!4d-0.2302753. Google. Retrieved 10/10/2022.

[8] SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS Internet protocol aspects – Quality of service and network performance. International Telecommunication Union. 2011.

[9] Epidemic Routing for Partially-Connected Ad Hoc Networks. Vahdat, A, Becker, D. Duke University. 2000.