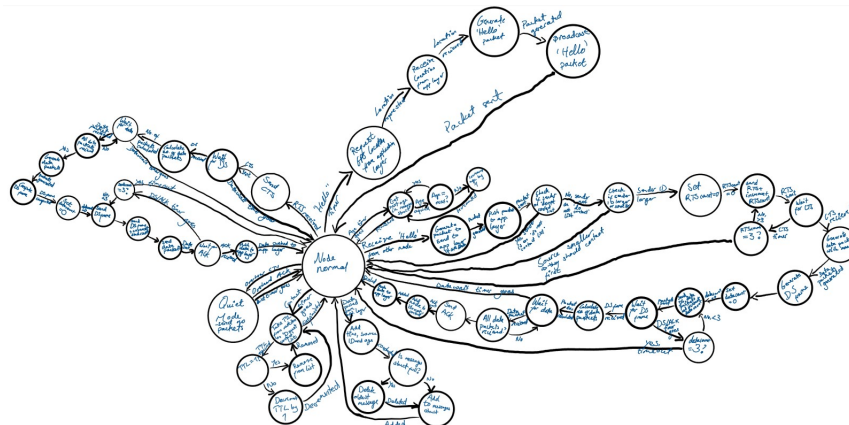


February 3, 2023

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.

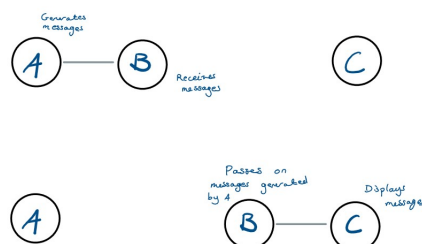
Figure 1: Network state machine



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/adafuit/Adafruit_Blinka. Adafruit Industries. GitHub. Accessed January 2023.