Project Requirements

# Aims

QUIC is a well thought-out and robust protocol with a clear RFC to describe it. It’s likely that protocols developed in the future will adopt qualities from QUIC, so it’s important that we are able to parse and generate code from the QUIC RFC to support automatic code generation for future protocols. The codebase already supports code generation in Rust, but it is limited to one language and doesn’t offer a simple way to add support for other languages. This project aims to extend the original codebase to parse QUIC-specific packet description structures and create a framework to aid in the addition of other languages.

# Initial Goals

* Parse QUIC structures and generate internal representation code (10 weeks)
* Create a more modular way to parse RFC sections so that the code is more readable and easier to modify in the future (2 weeks)
* Build a simpler framework for writing and organising formatted output code (4 weeks)

# Stretch Goals

* Write an output formatter to produce code in Python (4 weeks)
* Create custom error types for parsing to increase robustness (1 day)
* Refactor the ASCII diagram parser code to be more modular (2 weeks)
* Write code documentation (1 week)

# Outcomes

* Speed up the codebase by removing unnecessary code
* Make the code more modular and readable

# Timeline

## December

* Finish with the parsing code
* Reformat output formatter

## January

* Framework complete for output formatter

## February

* Work on stretch goals
* Mainly dissertation work