

Category	Description	Reviewers Comments
Build	Could you clone from Git and build using the README file?	The README was easy to understand and follow. If I had the required hardware I suspect that the raspberry pi's would be able to be built successfully.
Legability	Was the flow sane and were variable names and methods easy to follow? Does the code adhere to general guidelines and code style?	The flow was sane and variable names were understandable. Methods were somewhat not easy to follow. There were large functions that could be easily split up to be more readable. There also needs to be a lot more comments. Reading the code I had no idea what certain parts were doing, so having comments would help me understand.
Implementation	is it shorter/easier/faster/cleaner/ safer to write functionally equivalent code? Do you see useful abstractions?	Most code was contained in the "main" function. As described previously there are a lot of functions that are too large. All of the functions need to be split up into smaller and cleaner functions. It doesn't seem like it I would be hard to split them up unless it causes problems with multi-threading.
Maintainability	Are there unit tests? Should there be? Are the test covering interesing cases? Are they readable?	There are no unit tests. There defenity should at least be unit test to make sure the communcation within the ethernet is valid.
Requirements	Does the code fulfill the requirements?	Yes the code fulfills all the requirements for the capstone class.
Other	Are there other things that stand out that can be improved?	

**Build:** No revision necessary

**Legibility:** Comments were added to the code base explaining what functions and code blocks are doing.

**Implementation:** I disagree with this comment. We have our code in a variety of functions and executed across multiple threads. I think I could place some of the code which populates protobuf packets into a function but I do not think this would lend greatly to readability or maintainability. Generally decomposing a large function is best to do if the function can be reused across the code base. The main() function may be slightly large but much of the code cannot be reused in other parts of the program. I would argue that adding function to just reduce the size of another method without improving code re-usability is not necessary.

**Maintainability:** I created unit testing suites for both the comparison and corruption unit using the C++ unit testing framework Catch2. Unit tests can be easily run in the pre-configured Raspberry Pi VM image. Details are in the "Unit Testing" section of the README.

**Requirements:** No revision necessary.

Revised By: Kamron Ebrahimi