CS544 -- Computer Networks
Professor Brian Mitchell
Term Project Part # 2 – Network Protocol Design – Design Your own Application Protocol on top of QUIC
See Blackboard for DUE DATE

Introduction
Your assignment is to build on what you learned in the protocol analysis assignment and define a **NEW STATEFUL** application protocol, that will use QUIC as its transport protocol. Your protocol's goal is to provide a service for an application which could be useful in the real-world.

Some ideas:

- Chat protocol
- Game protocol (playing and or selecting)
- System update protocol
- Management protocol (server<->device)
- Streaming protocol (file, video, audio, etc)
- Application protocol between phone and watch
- Log aggregation protocol
- Load balancer health check protocol
- File transfer protocol
- Something you wanted to do – just let me know in advance so I can make sure you don't get yourself into too much trouble

This part of the project will involve writing a definition (think RFC) for you protocol. This definition should include:

- A description of the service (e.g. a file transfer service, peer to peer application, etc.) that your protocol and server will provide. Describe the basic service as well as any options / conditions that the protocol can do – and how they are negotiated.
- Define how the messages will be delineated, as well as the messages (both control and data) that your protocol will use. This should be detailed enough to show all the pieces of the message as well as data types, enumerations, endianness, etc. Think about implementation and think data types (what is an integer? What is a string?)
- Define and show the deterministic finite automata (DFA) that your protocol will use**. It must be a stateful protocol (not a stateful application).** We've reviewed in class the differences between protocol (conversation) state and application state.
- Describe how the protocol has been designed to allow extensions in the future (e.g. version information, handshake and option negotiation). You will define version 1 of this protocol, but your design should highlight how it should be able to be extended and updated over time.

- Describe the security of your service (how you will ensure that the right people can use it) – or do you allow everyone? What authentication mechanisms will your protocol use? **NOTE:** The implementation (in part 3) does not have to implement all authentication mechanisms; it can be hardcoded for the implementation to show the functionality at that time, like "bob" and "admin" are the only usercodes that work with fixed passwords). That said, the specification should say something about how security should work if that is important to a realistic implementation of your protocol.
- How do you ensure that the protocol is always working correctly?

IMPORTANT: To be a complete definition, it must contain enough information for someone to write a client program to talk to your version of the server using only the final specification (and vice versa – to write a server to talk to a client). It must also address security, performance, and extensibility. The biggest part of this definition is the DFA and the message definition to be COMPLETE – all data types, all fields, etc.

It is suggested that you include in your specification a C-style typedef to describe your PDU, message states, key constants, etc.

Please start early and use office hours to see that you're on the right track. I don't expect perfection for the first submission; but having a solid definition will make the final project deliverable significantly easier. The next assignment will be to implement this protocol (both a client and server) and will allow you to resubmit your design (for added points) with the implementation given you will have a lot more experience in this space.

Deliverable Format

Your design should be organized like part 1 of the project where you analyzed an existing protocol.

Section 1 – Service Description
Section 2 – Message Definition (PDUs)
Section 3 – DFA
Section 4 – Extensibility
Section 5 – Security implications

Grading Rubric

Here is a rough grading rubric for the project deliverable:

Definition of service:	15 points
Message definition (PDUs):	30 points
DFA:	15 points
Extensibility:	10 points
Security:	10 points
Degree of Difficulty:	10 points
Look & Feel:	10 points
Total:	100 points