Network Application Protocol Project

Grouping: A team could have 1 to 3 members. (2 people per group recommended.) Please email TA when you have a team and c.c. all your teammates. Only one email per team please.

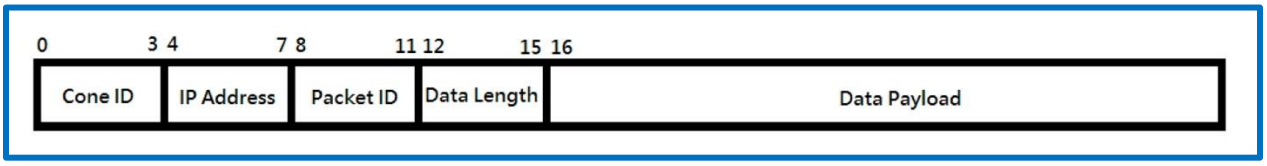
This project is for you to understand the meaning and usage of application layer protocol in communication networks. There will be 4 parts to turn in for this project. ***(1) Server program file(s)*** and ***(2) Client program(s) file***. You will choose your desired programming language to code these two files, they will perform a simple application you design in the Protocol documentation. ***(3) Protocol document*** will be a description of the network application protocol you design. In this document, you need to explain the following questions: What purpose does the protocol serve? (e.g., A file transport protocol or an online calculator or a chatting room…etc.) What is the protocol handshaking process? What commands/status codes are there for this protocol? What's the format of protocol messages? See below for detailed requirements. ***(4) ReadMe file*** for users to run the server and client programs on their devices.

**** Minimum Requirements for (3) Protocol Document ****

These are the minimum requirements for protocol designing and the protocol document.

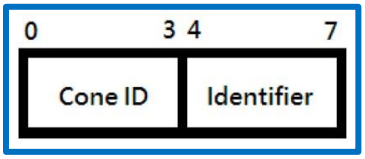
- (5 pts) A general introduction for the designed protocol. It should also mention whether it uses TCP or UDP as the transport layer service and why.
- (10 pts) Header fields should be included in message formatting. You need to explicitly list the formatting for the server, client, and handshaking process if they use different formats. You need to explain each field in the message format. Below are some examples.

  Data Packet



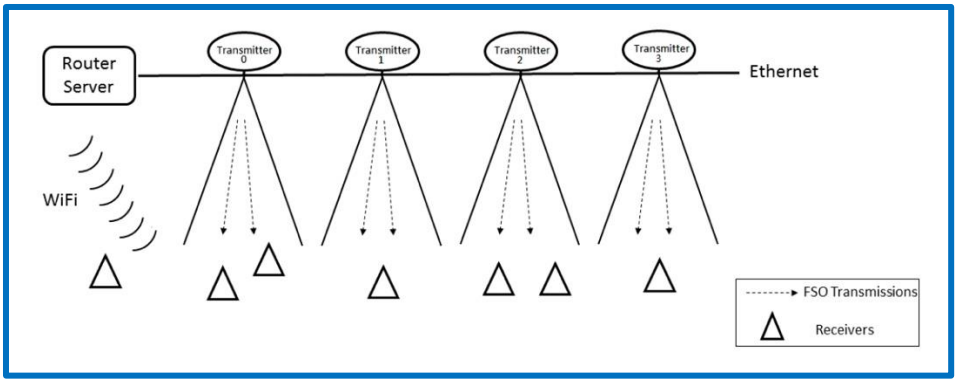
- Cone ID: 4 bytes. Each transmitter connected to the router server gets a unique ID assigned by the server. Negative value is invalid.
- IP Address: 4 bytes. This stands for the destination IP address of the packet. Negative value is invalid.
- Packet ID: 4 bytes. Each data packet has a unique packet ID that the server can track via the ack packet if the data is received by the receiver successfully. Negative value is invalid.
- Data Length: 4 bytes. This column indicates how many bytes there are in the data payload. Negative value is invalid.
- Data Payload: The plain data that has not applied any modulation and coding schemes. The length of data payload is indicated in the column of data length.

Beacon Packet



- Cone ID: 4 bytes. Same as the cone ID introduced in the data packets. Negative value is invalid.
- Identifier: 4 bytes. Here we use a specific negative number to distinguish the IP address in the data packets.

- (10 pts) An application layer hand-shaking process. (Note: This is NOT the TCP handshaking.) You should use both description and figures to help readers to understand the process.
- (10 pts) How this protocol works after hand-shaking process. Use both description and figures to help readers to understand the process. Below are some examples. Note: A good protocol should handle all unexpected/imperfect situations. For example, what happens after receiving a packet with the wrong format? Any mechanism to check/avoid corrupted packets? etc.



can move from one light cone to another. If a user moves out of a light cone, the pre-existing WiFi system will automatically continue transmitting data without packet loss. This feature is implemented by having each WiFO transmitter send beacons to all of the receivers within its light cone. Each beacon packet contains the beacon ID which corresponds with the specific transmitter. When a user receives a beacon packet, it sends an ACK with the beacon ID via WiFi to the router server. The server then updates a table recording the users IP address and which light cone the user belongs to. If the user moves out of a light cone, it stops receiving the beacons, stops sending ACKs to the router server, and after a specific time period, the router server removes the user/beacon ID table entry. When the data comes in to the router server, the server searches the table for the destination IP address. If it can be found, then the data is sent to the corresponding WIFO transmitter to broadcast to its users, otherwise, the data is sent via WiFi as normal. And there is a timeout mechanism to trigger retransmission avoiding packet loss.

What to submit:

1. (30 pts) Server program file(s)
2. (30 pts) Client program file(s)
3. (35 pts) Protocol document
4. (5 pts) ReadMe file

Grading Rubric:

	Requirement 1	**Requirement 2**	**Requirement 3**	**Score**
Client File	Successfully connect to server	Request in designed request format	Perform desired application functionality	30
Server File	Successfully connect to client	Respond to client requests with appropriate status codes	Perform desired application functionality	30
Protocol Documentation	See "Minimum Requirements for (3) Protocol Document" section above			35
ReadMe File	How to run the server, including the environment settings.	How to run the client, including the environment settings.	Anything else that the user should be aware of	5
Total				100