1. **Specifying a network protocol - an example**

This exercise is relevant to Project - A Simple File Transfer Protocol.

A network protocol must specify all three of the following:

* 1. the format of messages exchanged over the network
  2. the content of the messages exchanged over the network
  3. the sequence of messages exchanged over the network

Let's use the example of sending a file from the client to the server (which is a part of Project - A Simple File Transfer Protocol).

* 1. Client sends one message to the server. The message format is given below:
     + one byte opcode which is ASCII character A followed by
     + two-byte integer value in two's compliment and in network byte order, which represents the length of the filename of the file to be sent to the server followed by
     + the sequence of ASCII characters representing the filename
  2. Once the server receives a message with opcode A, it responds with the following message:
     + one byte opcode which is ASCII character A followed by
     + one byte acknowledgment code which is one of the following ASCII characters:
       - 0 - the server is ready to accept the named file
       - 1 - the server cannot accept the file as there is a filename clash
       - 2 - the server cannot accept the file because it cannot create the named file
       - 3 - the server cannot accept the file due to other reasons.
  3. Once the client receives the A message from the server and if the acknowledgment code is 0, it responds by sending the following message:
     + one byte opcode which is the ASCII character B followed by
     + four-byte integer (let's call it N) in two's compliment and in network byte order which represents the length of the file followed by
     + a sequence of N bytes which is the content of the file

Note in the above specification, it is preferable to use diagrams to help illustrate the message format (I have not drawn the diagrams because it is difficult to do it using HTML). Also note: the above specified the message format (such as 1 byte, 2 bytes, ...), the message content (such as opcode A, B, length of filename, filename, length of file, content of file etc), and the sequence: from client to server, from server to client, possible from client to server again.

Note also in the above specification, we did not say how the client should report the error message to its user. The latter is not part of this protocol. Here it is an implementation issue which should be left to the implementor of the client to decide. For example, the client implementation may report the error as a text message in English, or in Chinese, or as an image on the desktop, or does not report the error at all.

In the above specification, words like "ASCII character A" are necessary. Imagine if the words were instead "character A", it would open up for many possible interpretations (such as EBCDIC character, or UTF-16 character etc). The same applies to words like "four-byte integer in two's compliment and in network byte order". If you miss any one word, the meaning will be ambiguous.