Engine System Proposal

I am Planning to make a networking system, which handles the network connection between multiple players. The users could use system to establish point to point connection between different clients.

Currently, my plan is to adopt a server-client pattern, where the clients submits user inputs to a server, the server do the logic of the game and return the graphics data need to render to all clients each frame.

The user needs to migrate the logic to a server application and add data exchanging codes using the networking system. The user will provide the ip address, a structure represents the players' inputs, and a structure represents data being rendered.

I will implement a common interface of a TCP IPv4 asynchronized parallel server, it's responding client, and their windows specific implementation. Then, I will implement some template classes to handle the encoding and decoding of the data sent by the user.

To accomplish this I will learn the windows socket api and other windows api. I will need write a multi-threaded system on the server side to manage communications with different clients. I will come up with certain formats for transforming the data.

If I have more times I would want to implements more features and give more controls to the player, if they wish to obtain more controls. For example, for what I had in mind, the server will run on multiple threads specified and allocated by me. It is easy to use and the user do not even need to know my system use multiple threads. However, on the other hand some user might complain about they do not have control over those threads, and the system does something other that the networking manipulations behind their back. Hence, maybe I would let the user provides the threads to be used by the system. I would also want to learn and use other protocols such as UDP, DNS, HTML, SSL and so on.