Department of Electrical and Computer Engineering
Concordia University

Communication Networks and Protocols - COEN 445
Fall 2018

Project: Auction System

Designed by: Ferhat Khendek

1. Introduction

The project consists of designing and implementing in, Java, C++, C#, C or Python, an Auction System (AS), over UDP and TCP (see description below for when to use what). The requirements are stated in Section 3. The logics and the behavior of this application layer protocol is described in Section 2.

2. Auction System Protocol Description

The Auction System (AS) consists of several clients/users and a server communicating through the Internet. The AS allows for the clients to register with the Auction server. Only registered clients are allowed to offer items for auction and bid. A registered client can advertise many items for sale at a time. Auctions for an item are open during 5 minutes after its advertising and the broadcasting of the information to all registered clients. An item is sold to the highest bid at the end of the period of 5 minutes. For every item, the AS keeps informing the clients about the current highest bid. A client can bid as many times as he wishes for an item on auction.

2.1. Registering with the AS

A client has a unique Name. In order to receive information, offer items for auction, or bid, it has to register with the server. The server has to keep this information in a table, which also has to be saved in a file, in order to reload it when needed; if server crashes for instance.

In order to communicate with the server, the client has to provide its IP address where he can be reached, and the Port# it is listening to. The REGISTER message can be sent from a computer with IP address X but you may want to use another computer with address Y to interact with the server.
[Assume clients have distinct names, and everyone is really what he is pretending to be, .i.e. no authentication is required]

REGISTER	RQ#	Name	IP Address	Port#

REGISTER Message

To acknowledge this registration request and save the actual registration in the file, the server responds to the client with the following message.

REGISTERED	RQ#	Name	IP Address	Port#

REGISTERED Message

This message carries back all the information sent by the client for confirmation purpose.

If for some reasons the client cannot be registered because of damaged information (e.g. damaged IP address, duplicate name, etc.), server responds with message:

UNREGISTERED	RQ#	Reason

UNREGISTERED Message

After receiving such a message the client will try to resend the REGISTER message again.

A client can always try to leave the Auction System. For this purpose it has to send the message DEREGISTER

DEREGISTER	RQ#	Name	IP Address

DEREGISTER Message

The AS confirms the de-registration by sending the DEREG-CONF message.

DEREG-CONF	RQ#

DEREG-CONF Message

As for the registration if for some reasons the server cannot deregister the client, the following DEREG-DENIED message is sent back to the client.

DEREG-DENIED	RQ#	Reason

DEREG-DENIED Message

A client can be denied deregistration if he is currently offering an item for auction or active in bidding for at least one item (currently leading with the highest bid for at least one item), or the information provided in the DEREGISTER message is damaged or wrong.

All these messages are sent using UDP.

2.2. Offering items for auction

Every registered user can offer items for sale by sending the OFFER message to the server.

OFFER	RQ#	Name	IP Address	Description	Minimum

OFFER Message

RQ#, Name and IP Address carry the same information as in REGISTER message, the item for sale is described in the Description field, while the starting bid (acceptable) is given by Minimum. Any bid below that minimum will be just ignored by the server.

The server accepts the item and responds to the client with OFFER-CONF message.

OFFER-CONF	RQ#	Item#	Description	Minimum

OFFER-CONF Message

In this message, Item# is the number used to refer to the item. It is generated by the server.

If for some reasons the OFFER-CONF message does not arrive to the client or arrives damaged, the client will try to send the OFFER message again.

At this point the server will inform all the registered clients using NEW-ITEM message. This message will contain a TCP socket number through which it will handle the bidding for this item.

NEW -ITEM	**Item#**	**Description**	**Minimum**	**Port#**

NEW-ITEM Message

If a client is not registered and tries to offer items for auction, the server should reply using message OFFER-DENIED referring to the RQ# and give the Reason ("client not registered").

OFFER - DENIED	**RQ#**	**Reason**

OFFER-DENIED Message

For fairness a client cannot have more than 3 items offered for bidding simultaneously. If a client tries to have more than 3 items simultaneously then the message OFFER-DENIED will be sent to this client with the appropriate reason of "exceeding the number of items offered simulataneously!".

All these messages are sent using UDP.

2.3 Bidding for items

Before bidding for a given item, **a registered client has to establish a TCP connection to the TCP socket associated with the item of interest at the server side**. After this connection, a client can bid on the item by sending a BID message.

BID	RQ#	Item#	Amount

BID Message

To keep the clients informed on every item, any change in the current highest bid is sent to all the registered clients using HIGHEST message, where Amount stands for the current highest bid amount.

HIGHEST	Item#	Amount

HIGHEST Message

As mentioned earlier a client can submit as many bids as he wishes. When a bid is lower than the current highest bid for a given item, it is just ignored by the server.

When the bidding period of 5 minutes is over, the server informs the winning client using WIN Message. If there is more than one client with the highest bid, the first bid to reach the server wins. The WIN message carries the information about the identity (Name, IP Address, Port#) of the client selling the item.

WIN	Item#	Name	IP Address	Port#	Amount

WIN Message

All the other clients involved in this item auction (who have made at least a bid on this item) will be informed by the server using BID-OVER message. This message will carry the winning bid in the amount field.

BID-OVER	Item#	Amount

BID-OVER Message

At this point all the TCP connections related to this item should be ended for every client.

The server has to inform now the client who is selling the item about the winner by sending SOLDTO message informing the client about the item sold, and the identity (Name, IP Address, Port#) of the winning client, and the winning bid (amount).

SOLDTO	Item#	Name	IP Address	Port#	Amount

SOLDTO Message

When an item has not attracted a single valid bid, the server returns the item to the client using the message NOT-SOLD indicating the Item# and the reason ("no valid bids"). The client may decide to re-offer the item for bidding and lower it required minimum…

NOT - SOLD	Item#	Reason

NOT-SOLD Message

3. Requirements

Project should be done in groups of 2 students. You should send, by September 25th, 2018, your group list including student names, ID numbers and **ECE** email addresses to khendek@ece.concordia.ca.

Design and Implement the client(s) and the server that follow the protocol aforementioned. The coding of the protocol messages is part of your design, i.e. you have to come up with the appropriate coding of the messages. You can decide to use simple text message, etc.

Clients and Server should be multi-threaded as it is generally the case for communication protocol entities.

Reporting: The server and the clients should be reporting their communications (i.e. the exchanged PDUs) using a log file or printing directly into the screen. In other words, during the demonstration, I would like to see the messages sent and received by these entities, progress and failures.

Assumptions/Error/Exception Handling
You should be aware that the description as it is does not state everything. For instance what happens if a client receives a response with a RQ# that does not correspond to any of its (pending) requests?

State and document clearly any assumption you make beyond the assumptions made by the instructors.

You should hand in a report, by Week 13, where you document clearly your assumptions, design decisions, code and experiments. You should also state clearly the contributions of every member of the group. Every student has to contribute **technically** (designing and implementing the protocol) to the project. - You also have to submit a signed Expectation of Originality Form, which you can download from.

A demo will be held during Week 13 of this fall term. During the demo all the members of the group should be ready to answer questions of the instructor. We will be running at least 6 clients and 1 server on different machines and tests all the messages in this protocol.

During the demo we may also go through the code and the report.

The project will be discussed further in class.

Bonuses

Bonus marks will be given to students for going beyond the requirements stated in this document. For instance designing and implementing a GUI for using this system, or using a client on a hand held device like a Smartphone… **Before you start working on such features please consult with the instructor**.