

IS 496 Computer Networks Project Report

Real-time Bidding System: A Reliable, Low-latency Communication Platform for Bidding Transactions

Authors: Ken Wu (shwu2), Jack Chuang (yzc2), Thomas Huang (yenshuo2)

1. Introduction

Our project's objective is to create a real-time bidding system that will enable buyers and sellers to make network-based real-time bids on goods. All parties engaged in the bidding process will have access to a dependable, low-latency communication platform via the system. Buyers and sellers searching for a quick and safe way to engage in auctions are the application's target users. Our program will enable customers to bid on numerous items simultaneously in addition to allowing merchants to list their products and launch the bidding. The application will make the bidding process more efficient, provide up-to-date information, and maintain data integrity. This is crucial in high-stakes auctions since every second counts.

2. Application Workflow and Implementation

The application is designed with a user-friendly command line interface and efficient workflow to support smooth bidding operations. The main components of the application include user login, a bidding dashboard, real-time notifications, and secure data transfer.

2.0 Getting Started

To start the application, the server have to be launched first. This can be done by running a command in the terminal `python3 server.py -hn hostname -p portnumber`. To start the client server, user will input `python3 server.py -hn hostname -p portnumber -un username` in terminal.

2.1 User Login

Once users start the client server, they will be prompt to enter their password. The returning user will be ask to enter their password, while new user will be ask to enter a password and create new account.

2.2 Bidding Interface

The bidding interface have several option. Client will sends command to server and then server responds accordingly.

- **BID:** This command allows the user to place a bid on a specific auction item. Once enter the command, server will send the biddable items to client. The user would provide the item name and the amount they wish to bid, and the

server would validate the bid and update the result accordingly. If the bid is successful, the server would respond with a success message. If the bid is invalid or unsuccessful, the server would respond with an error message.

- AUCT: This command allows the user to create a new auction item. The user would be asked to provide details about the item, such as item name and starting price. (Currently, the auction time by default is set to 180 seconds for testing purpose)
- GETALL: This command allows the user to retrieve information about all the auction items currently in the database. Each item has the information of
- EX: This command allows the user to exit or quit the bidding interface.

2.3 Socket Programming and Secure Data Transfer

Our team developed the key networking functions of the application using Socket programming. TCP is used for secure and reliable data transfer during the bidding process. This ensures that all bids are accurately received and recorded, maintaining data integrity. It also enabled us to create a reliable, low-latency communication platform for buyers and sellers. In addition to TCP, we also implemented secure data transfer protocols to protect data transmission. We used encryption and decryption methods to encode and decode user data during transmission. This ensures that all data transferred over the network is protected. This further enhances the security of the bidding process. Overall, the combination of TCP and secure data transfer protocols provides a robust and secure networking infrastructure for our application.

3. Technical Challenges

We faced various technological difficulties while we created our real-time bidding system.

The ability to manage several connections at once was the first issue we had to solve. We employed multithreading, which enables the program to process several connections simultaneously without blocking other connections, to ensure that the application can accommodate a huge number of users without any performance difficulties. This makes it easier to handle a big number of users, and users can bid or place different items at the same time.

Another key difficulty was ensuring data integrity. To prevent unwanted access to sensitive information, we deployed encryption and decryption techniques to encode and safeguard user data during transmission. As a result, data breaches were avoided and data integrity was preserved.

Finally, managing collisions that could happen when numerous users attempt to bid on the same item at the same time was one of the biggest problems. We created a concurrency management technique that limits the number of users that may alter an item at once in order to avoid collisions. The server locks an item when a user sets a bid on it, preventing other users from making changes to it until the bidding process

is through. In order to guarantee that bids are made on the right item and avoid collisions, we additionally employ the product name as a distinctive identifier.

4. Future Improvement

There are few points we would like to improve if we have extra time on designing the application.

First, adding front-end UI can significantly improve user experience with the app. User can bid items through graphical user interface, which will make their experience with the app more intuitive.

Second, give user a real-time notification when the user's bid is outbid by other users. This may give users an extra motivation to stay engaged in the bidding process and may lead to increased user engagement.

Third, the ability to customize auction time. Currently, the auction time is fixed for all items and cannot be changed by the user. By allowing users to customize the auction time, we can offer them more flexibility and control over their bidding experience. For instance, users can set the auction time to be shorter or longer based on their preferences. It can depend on how quickly they want to sell or purchase the item.

Fourth, giving the app a countdown timer function. A timer option can give customers a feeling of urgency, which will motivate them to place their bids as soon as possible and not lose out on their chosen things. For example, a countdown clock that shows how much time is left in the auction coupled with a warning or message that informs users when the sale is about to conclude. This can help increase user engagement and enhance their overall experience with the app.

Lastly, adding a database to store data will help improve data governance. Currently, we are only store bidding data, users data in txt files. With connecting the application with a database, retrieving data will be more efficient and reliable, as databases allow for faster search and retrieval of information, as well as easier data management and organization. Additionally, using a database can enable features such as data validation and concurrency control, ensuring the integrity and consistency of the data being stored.

5. Team Member Contributions

- Jack Chuang: Networking functions using Socket programming
- Ken Wu: Application function and implementation
- Thomas Huang: Report write-up, demo preparation, and testing

6. Conclusion

In conclusion, our team has successfully created a network-based real-time bidding system that enables buyers and sellers to put bids on products. The application provides a user-friendly command line interface, real-time notifications, secure data transfer, and efficient workflow to support smooth bidding operations. However, there is always room for improvement. We suggest including a front-end user interface (UI) that lets consumers place bids on things using a graphical user interface to improve the user experience. The application's database connection can provide data validation, concurrency management, and better data governance. Users can have greater freedom and control over their bidding experience, which can also boost user engagement, if the auction time can be customized and a countdown timer is included. Overall, we believe that implementing these improvements will make the application more efficient, user-friendly, and secure.