ABX Exchange Client

1. **Setting Up the Environment:**
   - Make sure you have the .NET SDK installed on your system.
   - Since the server runs with Node.js, ensure you have Node.js version 16.17.0 or higher installed.
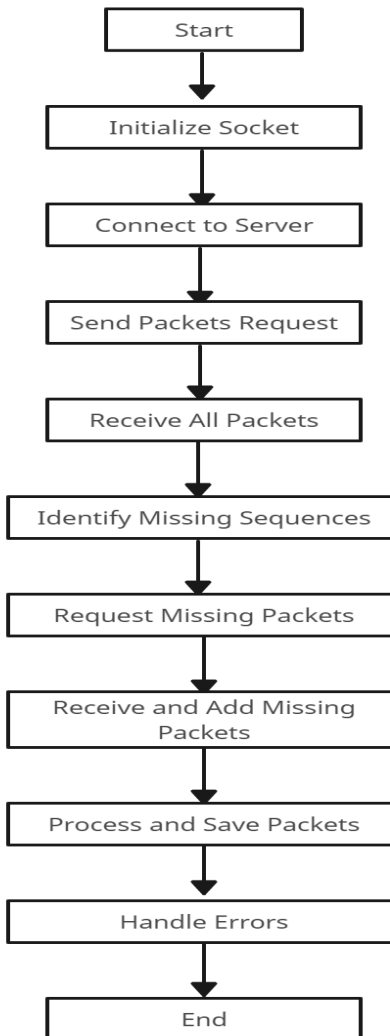2. **Run the ABX Exchange Server:**
   - Download the zip file and extract the zip file 'abx_exchange_server' .
   - Open a terminal or command prompt and run the command 'node main.js'. This will start the ABX exchange server.
3. **Add Necessary NuGet Packages:**
   - Newtonsoft.Json is required for JSON serialization.
4. **Code Flowchart:**

```
                    ┌─────────────┐
                    │    Start    │
                    └─────────────┘
                           │
                           ▼
                 ┌───────────────────┐
                 │  Initialize Socket │
                 └───────────────────┘
                           │
                           ▼
                 ┌───────────────────┐
                 │  Connect to Server │
                 └───────────────────┘
                           │
                           ▼
                 ┌───────────────────┐
                 │ Send Packets Request│
                 └───────────────────┘
                           │
                           ▼
                 ┌───────────────────┐
                 │ Receive All Packets│
                 └───────────────────┘
                           │
                           ▼
               ┌─────────────────────────┐
               │ Identify Missing Sequences│
               └─────────────────────────┘
                           │
                           ▼
               ┌─────────────────────────┐
               │  Request Missing Packets │
               └─────────────────────────┘
                           │
                           ▼
               ┌─────────────────────────┐
               │  Receive and Add Missing │
               │         Packets          │
               └─────────────────────────┘
                           │
                           ▼
               ┌─────────────────────────┐
               │ Process and Save Packets │
               └─────────────────────────┘
                           │
                           ▼
               ┌─────────────────────────┐
               │      Handle Errors       │
               └─────────────────────────┘
                           │
                           ▼
                    ┌─────────────┐
                    │     End     │
                    └─────────────┘
```

Code Flow Overview:
- The program starts, initializing and connecting a TCP socket to the server.
- It sends a request to stream all packets and receives the data packets asynchronously.
- Missing packet sequences are identified, and requests are made to the server to resend those specific packets.
- The received packets (including re-requested ones) are saved as a JSON file.
- The **Packet class** handles parsing byte arrays into readable packet data, ensuring correct byte order based on the system's endianness.
- **ReceiveAllPacketsAsync:** This method continuously reads packets from the server until the connection is closed and stores each valid packet in a list. It ensures that only complete packets of the expected size are processed.
- **ReceiveSinglePacketAsync:** This method is used to receive a single packet from the server, typically after requesting a specific sequence number. It ensures the packet is the correct size before returning it.
- **GetMissingSequences:** After all packets are received, this method checks for any gaps in the sequence numbers and returns a list of the missing sequences, which can then be requested from the server.