Telemedicine Final Project – December 2024

Multiple Sclerosis Server Application

ADMINISTRATOR MANUAL

Table of contents

1. Protocol used

We used TCP protocol with multithreading in the communications between our applications.

WHY DID WE CHOOSE **TCP PROTOCOL?**

In this project, the decision to use a TCP protocol (Transmission Control Protocol) was based on its principal characteristics that align with the requirements of the system for monitoring physiological parameters using a BITalino device.

TCP is a connection-oriented protocol that establishes a connection between devices before transmitting data, providing reliable communication. This is critical for an application where data integrity and accuracy are essential, particularly in this case when handling sensitive and precise physiological measurements of ECG and EMG signals. Unlike UDP, which is a connectionless protocol and does not ensure reliable delivery of packets, TCP is designed to identify transmission errors and ensure that data is transmitted accurately. This guarantees that the transmitted data is received without corruption or loss, which is particularly important for monitoring patients in real time. By using TCP, the system ensures that no critical physiological data is lost, a factor that is essential for maintaining the accuracy of patient records and analysis.

The application uses a BITalino device to acquire physiological parameters, which are transmitted to the server for processing. Since these data packets represent important medical information, it is essential to prevent issues such as data duplication, out-of-order delivery, or packet loss. TCP manages these issues through mechanisms such as ACK (acknowledgments), retransmissions, and sequence control. Furthermore, TCP provides flow control, allowing the system to manage variable network conditions effectively, ensuring a stable connection.

Finally, TCP's connection-oriented characteristic makes it easier to manage multiple connections between clients and servers. This is especially important when multiple patients are sending data simultaneously, as it ensures that each connection remains distinct and secure.

WHY **MULTITHREADS**?

We used multithreading to ensure that multiple clients (patients and doctors) could connect to the server at the same time. This is crucial for the functionality of the whole application itself. It is a remote monitoring app, which means it is intended for the use of more than one person at the same time and from all around the world.

USE OF **TEXT (TXT) FILES**?

The decision to use TXT files for storing the physiological parameters received on the server stems from the simplicity, portability, and human readability offered by this format. A plain text file ensures that the stored data, which includes the patient's name, the date and time of monitoring, and the corresponding physiological parameters, is easily accessible and comprehensible without requiring specialized software for viewing or editing. This approach supports the integration of the generated files into a broader workflow or external systems where straightforward access to the information is essential. Furthermore, the flexibility of TXT files allows for seamless sharing and compatibility across platforms and applications.

The choice of TXT aligns with the requirements of the project to store critical patient data in a format that is both reliable and easy to manage. Using plain text avoids potential issues related to proprietary formats, ensuring that the data remains universally accessible regardless of technological constraints. Additionally, the structured yet simple format of a TXT file facilitates the generation of detailed reports or summaries, which can later be used by healthcare professionals for analysis, diagnosis, or decision-making processes.

In conjunction with the TCP protocol, which guarantees the reliable delivery of the data to the server without corruption or loss, the use of TXT files ensures that the received physiological parameters are securely stored in an organized manner. This combination of TCP's reliability and TXT's simplicity provides a robust solution for managing critical monitoring data while maintaining the focus on accessibility and practicality for all stakeholders involved in the healthcare process.

2. Communication scheme

The communication scheme can be seen through this link:

LINK TO UML

3. Program Structure

3.1 ER Diagram

The diagram can be seen through this link:

LINK TO ER DIAGRAM

3.2 USE Cases