Name: Ben Mohr Matr. Nr. 1605057

Übung 3

Vergleich der UDP- und TCP-Programme

Die Programme binden beide ihre Server-Sockets an 0.0.0.0, um Nachrichten über alle Netzwerkschnittstellen hinweg empfangen zu können. Dies ermöglicht sowohl lokalen als auch externen Zugriff (z. B. im LAN).

TCP: Verbindung und Threading

Beim TCP-Programm wird mit dem Aufruf listen() der Server in einen passiven Zustand versetzt, in dem er auf eingehende Verbindungsanfragen wartet. Sobald ein Client eine Verbindung herstellt, wird ein neuer Socket durch accept() erzeugt und in einem separaten **Thread verarbeitet**. Jeder Thread verarbeitet unabhängig genau **einen Client**, wodurch **mehrere parallele Verbindungen** möglich sind. TCP stellt eine **zuverlässige, verbindungsorientierte Kommunikation** sicher: Pakete kommen in der richtigen Reihenfolge an, verloren gegangene Pakete werden neu übertragen.

UDP: Verbindungslos und einfach

Das UDP-Programm verwendet keinen Verbindungsaufbau. Der Server empfängt Datenpakete direkt über den recvfrom()-Aufruf, bei dem die Absenderadresse mitgeliefert wird. Es wird **kein Thread** pro Client benötigt, da jede Nachricht einzeln und unabhängig verarbeitet wird. UDP ist **verbindungslos**, das heißt es gibt keine Garantie für Reihenfolge oder Zustellung der Pakete – es ist dafür aber schneller und ressourcenschonender.

Unterschiede in den PCAP-Dateien (Wireshark-Analyse)

In den aufgezeichneten PCAP-Dateien zeigen sich die Unterschiede deutlich:

TCP:

- Es findet ein **3-Wege-Handshake** statt (SYN, SYN-ACK, ACK), bevor Daten übertragen werden.
- Jede Datenübertragung wird bestätigt (ACK).
- Bei stop oder Programmende sieht man das **Beenden der Verbindung** (FIN, FIN-ACK).

UDP:

- Keine Verbindungspakete (kein Handshake)
- Jedes Datenpaket steht für sich – keine Quittung, keine Sicherung

- Es gibt **nur die Nutzdatenpakete**, meist im Format UDP → <Zielport> sichtbar

4)

Folge ist: 0110101