

Programmieraufgabe als APL

Realisierung eines Send-and-Wait Protokolls (SAW) auf der Basis von UDP-Sockets

Aufgabe für 4er Gruppen

Schreiben Sie ein Sender- und ein Empfänger-Programm in C (Konsolenprogramme), mit Hilfe derer eine Textdatei zeilenweise vom Sender an den Empfänger via UDPv6/IPv6- übertragen werden kann. Dem Sender werden der Dateiname, die IPv6- Adresse und der Port des Empfängers als *Programmargumente* übergeben, damit die Datei auf der Basis von ARQ und einer vorgegebenen Fenstergröße von 1 Paket (W=1) übertragen werden kann. Auch der Empfänger soll über Programmargumente konfigurierbar sein: Name der Ausgabedatei, und der Port, auf den der Empfänger hören soll.

Der Empfänger soll anhand von Sequenznummern in den Paketen die Sendereihenfolge der übertragenen Datagramme überprüfen und korrekt empfangene Pakete bestätigen. Erhält der Empfänger Pakete außerhalb der erwarteten Reihenfolge oder ist das empfangene Paket fehlerhaft (berechnete Prüfsumme != empfangener Prüfsumme) wird keine Quittung gesendet. Bei ausbleibender positiver Quittung wird nach sinnvoller Dauer das Paket vom Sender erneut übermittelt.

Die Prüfsumme ist im Einerkomplement zu berechnen. Nutzen Sie dazu den Algorithmus in <https://tools.ietf.org/html/rfc1071#section-4>.

Ein Verbindungsaufbau bzw. –abbau ist nicht zu realisieren. Lediglich die Datenübertragung soll wie oben beschrieben nach dem ARQ-Mechanismus mit positiven Quittungen erfolgen.

Die Protokollabläufe müssen anhand von Konsolenausgaben nachvollziehbar sein, um die Funktionalität überprüfen zu können. Dazu gehört ebenfalls die Produktion von Fehlerfällen wie Bitfehler, fehlende Pakete (Datenpakete/ Quittungspakete) etc. Sinnvollerweise ist die Fehlerart ebenfalls als Programmparameter zu übergeben.

Literaturhinweis: Der genaue Protokollablauf ist ebenfalls in Kurose/ Ross (rdt2.0) nachzulesen.

Abgabe:

Gliedern Sie Ihren Quellcode in die folgenden Schichten durch das Trennen der Funktionalitäten in verschiedene C-Module jeweils für Sender/Empfänger:

1. Anwendung
 - a. Sender: zeilenweises Einlesen der angegebenen Textdatei, Übergabe der Daten an send-Funktion Ihres Anwendungsprotokolls (SAW_send).
Hinweis: während des Sendens eines Paketes darf die Anwendung keine weiteren Pakete an das Anwendungsprotokolls reichen, erst wenn sichergestellt ist, dass das Paket korrekt empfangen wurde!
 - b. Empfänger: zeilenweiser Empfang der Textzeile durch Aufruf der Anwendungsfunktion (APP-recv) durch das Anwendungsprotokoll und Ausgabe in Textdatei
2. Anwendungsprotokoll (SAW-Protokoll)
 - a. Sender: Initialisieren des Senders durch Erstellung eines UDP-Sockets; Aufbau des zu sendenden Paketes (nutzen Sie eine Datenstruktur z.B. struct packet (char txtCol[...]; ... seqNr; ...checksum;}). Definieren Sie letztere in einem separaten Header-File.
Hinweis: Diese Struktur können Sie ohne Serialisierung direkt an die UDP-Socketfunktion sendto übergeben!
Setzen Sie unmittelbar nach dem Senden einen Timer. Nutzen Sie dazu die select() – Socketfunktion (<https://docs.microsoft.com/en-us/windows/win32/api/winsock2/nf-winsock2-select>). Achten Sie darauf, dass Sie dazu den Socket auf nichtblockierend setzen: `ioctlsocket(sock, FIONBIO, &iMode);` // <https://docs.microsoft.com/en-us/windows/win32/api/winsock2/nf-winsock2-ioctsocket>
 - b. Empfänger: Initialisieren des UDP-Sockets; blockierendes Empfangen (standardmäßig) mittels recvfrom (UDP-Socketfunktion); Überprüfen der SeqNr und der Prüfsumme; Senden der Quittung (nutzen Sie dazu eine weitere Datenstruktur z.B. struct ack (... seqNr; }). Definieren Sie letztere in einem Header-File.

Arbeiten Sie ohne Threads!

Die Lösungen sind mir in Form von gut strukturiertem und kommentiertem Quellcode per Email bis spätestens 31.1. zuzusenden. **Plagiate werden geahndet.**

Eine kurze Erläuterung den Aufruf von Client und Server betreffend, als auch die Erklärung auf die selbstständige Anfertigung unter Zuhilfenahme der angegebenen Quellen!!!! muss mit enthalten sein. Geben Sie des Weiteren alle Gruppenmitglieder an!

Die Lösungen müssen im Labor S311 übersetzbar und lauffähig sein. Als Entwicklungsumgebung steht Visual Studio zur Verfügung.

Die Abnahme erfolgt spätestens in der letzten Vorlesungswoche während der Praktikumszeiten am Rechner in Form einer Vorführung.