PR Rechnernetze und Netzwerkprogrammierung 621.801/621.802	WS 2024 / 2025
Institut für Informationstechnologie (ITEC)	Rainer/Timmerer
Übungsblatt 06	

Ü 6.1 Datenübertragung via TCP

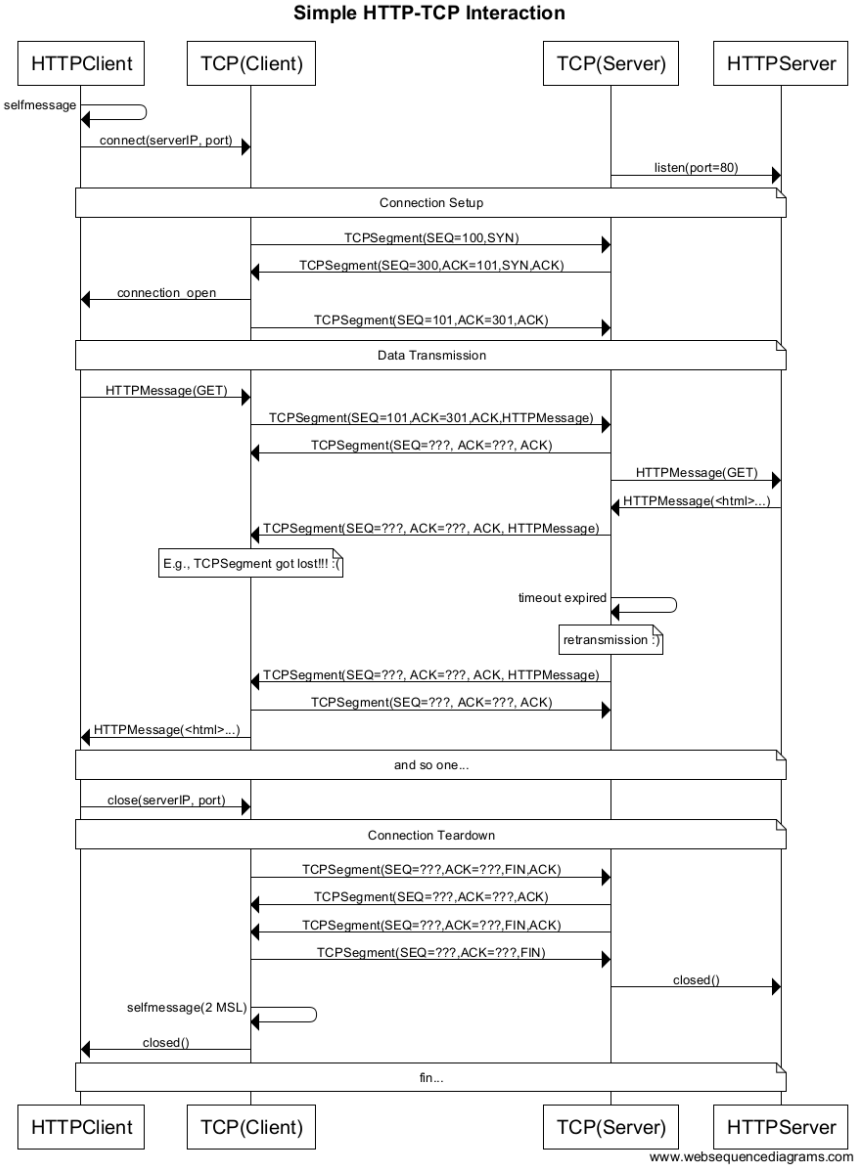
Zur eigentlichen Datenübertragung via TCP implementieren Sie ein *stop-and-wait*-Protokoll. Achten Sie dennoch auf das richtige Setzen der ACK- und SEQ-Nummer gemäß RFC. Implementieren Sie auch den Verlust eines TCP-Segments (z.B. 10% Segmentverlust) sowie das erneute Übertragen (eng. retransmission) des verlorengegangenen Segments.

Hinweis: Sie können persistente TCP-Verbindungen verwenden bzw. davon ausgehen.

Ü 6.2 TCP-Verbindungsabbau

Implementieren sie den TCP-**Verbindungsabbau** und verwenden Sie dazu Abbildung 13 im TCP-RFC.

Hinweis: Die nachfolgende Abbildung zeigt ein Beispielsequenzdiagramm für den Verbindungsabbau, Datenübertragung und Verbindungsabbau.



Ü 6.3 Vermittlungsschicht und Routing: Theorie

Welche Funktionalitäten stellt die Netzwerkschicht der Transportschicht zur Verfügung?

Ist es möglich, dass Pakete (aus der Sicht der Netzwerkschicht) unterschiedliche Routen zwischen gleichen Start- und Endknoten nehmen können? Begründen Sie Ihre Antwort.

Haben Router eigentlich IP-Adressen? Wenn ja, wie viele? Begründen Sie Ihre Antwort.

Erklären Sie den grundsätzlichen Aufbau einer IPv4/IPv6-Adresse.

Ü 6.4 Vermittlungsschicht und Routing: Link-State-Algorithmus

Gegeben sei ein Netzwerk wie in Abbildung 1 dargestellt. Berechnen Sie, unter Zuhilfenahme des Algorithmus von Dijkstra, den kürzesten Pfad ausgehend von Knoten A zu allen Netzwerkknoten. Benutzen Sie dazu eine ähnliche Tabelle wie Sie sie in der Vorlesung kennengelernt haben. Die Zahlen geben die Kosten für den jeweiligen Link an.

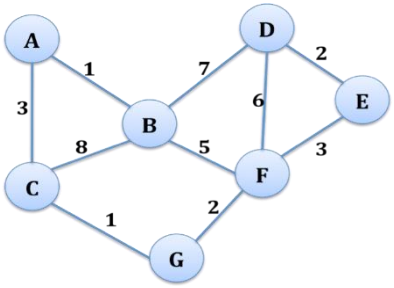


Abbildung 1: Netzwerk für Link-State-Algorithmus.

Ü 6.5 Vermittlungsschicht und Routing: Distance-Vector-Algorithmus

Gegeben sei ein Netzwerk wie in Abbildung 2 dargestellt. In diesem Netzwerk kennt jeder Knoten die Kosten der Verbindungen zu seinen unmittelbaren Nachbarn. Wenden Sie den Distance-Vector-Algorithmus an und geben Sie sowohl Distanz-Tabellen von Knoten y nach jedem Schritt als auch die initiale Tabelle an. Gehen Sie dabei davon aus, dass y zuerst den Distanzvektor von v geschickt bekommt, danach von x und zuletzt von z. Markieren Sie in den Distanztabellen die jeweiligen Änderungen und geben Sie auch den erhaltenen Distanzvektor in jedem Schritt an.

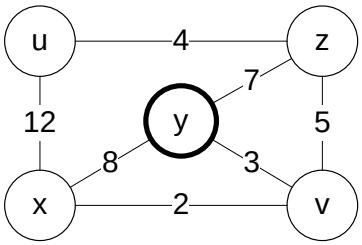


Abbildung 2: Netzwerk für Distance-Vector-Algorithmus.

Ü 6.6 OMNeT++-Netzwerkschicht

Gegeben ist das in Abbildung 3 gezeigte Netzwerk, in dem drei Clients und ein Server über ein Netzwerk bestehend aus 5 Routern verbunden sind. Router und Hosts sind unterschiedliche Arten von Netzwerkknoten. Während der Host (Client sowie Server) alle Schichten benötigen, operiert der Router nur auf den ersten drei Netzwerkschichten. Der Unterschied ist leicht in den jeweiligen Ned-Dateien zu erkennen (siehe IPHost.ned und Router.ned).

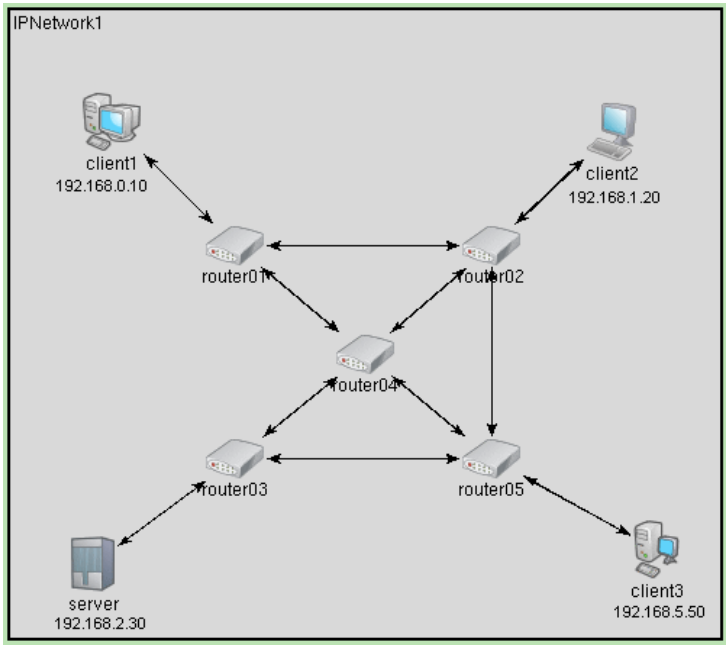


Abbildung 3: IPNetwork.

Die Clients (client1-3) schicken jeweils eine Anfrage an den Server. Die Anfrage enthält eine Zahl (int) und einen Befehl (int). Der Server addiert den Befehl zur Zahl und schickt das Ergebnis als Antwort zurück.

Ihre Aufgabe ist es die Netzwerkschicht zu implementieren. In den Stubs sind sowohl Anwendungsschicht als auch Transportschicht (auf Basis von UDP) implementiert. Alternativ dazu können Sie auch ihre Implementierungen von den letzten Übungsblättern verwenden.

Konzentrieren Sie sich auf die Folgenden Dateien:

IPDatagram.msg – Ergänzen Sie die Nachrichtendefinition um die benötigten Felder (siehe Vorlesungsunterlagen oder RFC).

IP.cc – Erweitern Sie die Implementierung der Netzwerkschicht um die Funktionalität des Routers (Forwarding anhand der zur Verfügung gestellten Tabelle) und um die Funktionalität des Hosts.

Hinweise: Lesen und Analysieren Sie die Implementierung der Transportschicht (UDP). Das hilft Ihnen bei der Implementierung der Netzwerkschicht (Abbildung 4).

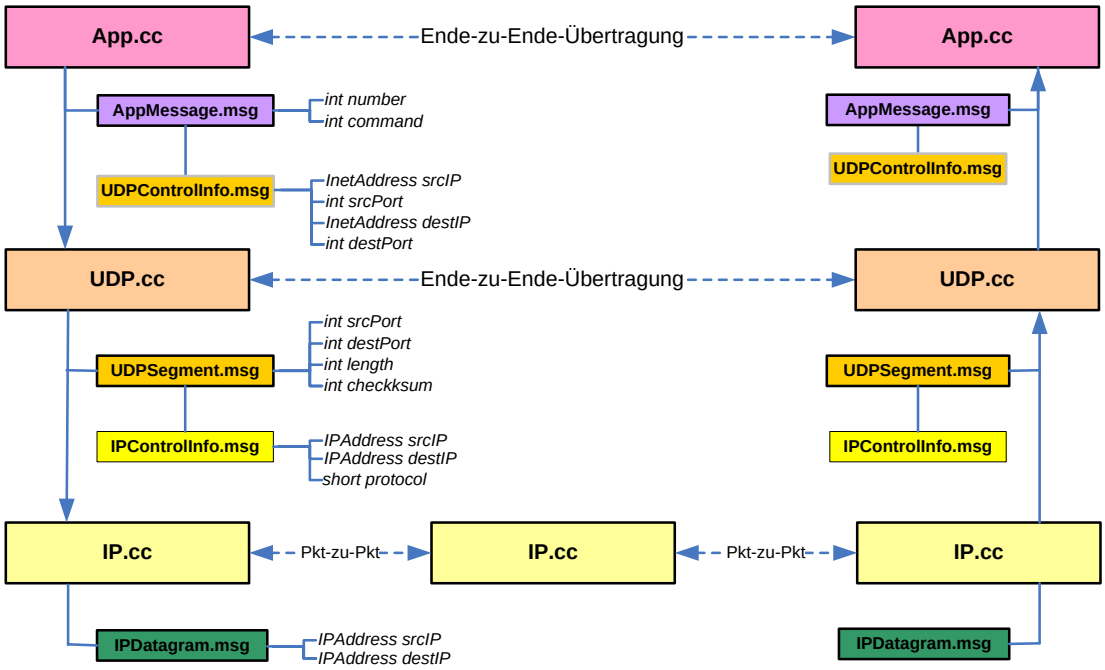


Abbildung 4: UDP-IP im aauTCP/IP-Referenzmodell.

Ü 6.7 OMNeT++-Netzwerkschicht und Routingalgorithmus

Nutzen Sie die in Ü6.6 fertig gestellte Implementierung und optimieren Sie die Forwarding-Tabelle in der Datei **IP.h**. Ihr Ziel ist es dabei alle Daten in möglichst kurzer Zeit an den Zielhost zu schicken. Die unterschiedlichen Latenzen (in Sekunden) sollen dabei wie in Abbildung 5 dargestellt angenommen werden.

Hinweis: Ermitteln Sie dazu eine ähnliche Tabelle wie in Ü6.6.

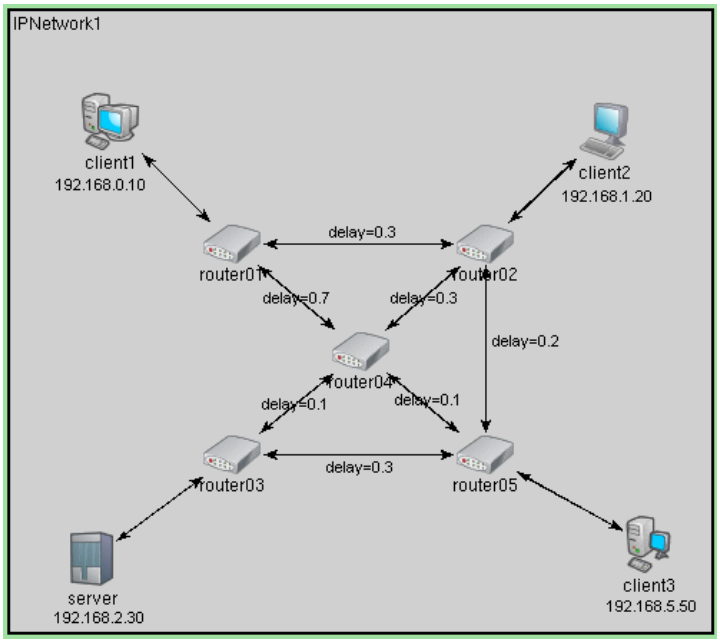


Abbildung 5: IPNetwork mit Verzögerungen.