



HOCHSCHULE KONSTANZ TECHNIK, WIRTSCHAFT UND GESTALTUNG (HTWG)
Fakultät Informatik
Rechner- und Kommunikationsnetze
Prof. Dr. Dirk Staehle

Vorlesung Rechnernetze

Laborübung

Go-Back-N

Prof. Dr. Dirk Staehle

Die Abgabe erfolgt durch Hochladen der bearbeiteten Word-Datei in Moodle.

Bearbeitung in Zweier-Teams

Team-Mitglied 1:

Team-Mitglied 2:

1 Aufgabe

In dieser Aufgabe soll eine Klasse „GoBackNSocket“ implementiert werden, die eine Datenübertragung mit Go-Back-N oberhalb von UDP realisiert. Ein Socket wird über den lokalen Port, den Remote-Port sowie die Remote-Adresse initialisiert. Analog zu einem TCP-Socket soll die Klasse einer Anwendung die Schnittstellen „send(msg)“ und „msg=recv(bytecount)“ bieten. Ein Socket wird über die Schnittstelle „stop()“ beendet. Die tatsächliche Paketübertragung soll über UDP erfolgen, genauer sollen Sie dazu die Klasse „Lossy UDP Socket“ nutzen, um eine Übertragung über UDP mit Paketverlusten zu erzeugen.

2 Lossy UDP Socket

Der Lossy UDP Socket ist ein UDP Socket, bei dem Pakete am Empfänger mit einer konfigurierbaren Wahrscheinlichkeit verworfen werden. Sie müssen den Socket nicht selbst implementieren sondern können die bereitgestellte Klasse „lossy_udp_socket“ verwenden.

„lossy_udp_socket“ Objekte werden mit einem „Handler für empfangene Pakete“, lokalem Port, Remote-Port, und Remote-Adresse sowie einer Paketverlustwahrscheinlichkeit erzeugt. Die Klasse „lossy_udp_socket“ bietet die Schnittstellen „send(packet)“ und „stop()“, um Pakete zu versenden bzw. den Socket zu schließen. Der „Handler für empfangene Pakete“ muss eine Funktion „receive(packet)“ bieten, die vom „lossy_udp_socket“ aufgerufen wird, um empfangene Pakete zu übergeben, d.h. Sie sollen diesen „Handler“ in der Klasse „GoBackNSocket“ implementieren.

Beim Versenden und Empfangen von Paketen wird eine kurze Ausgabe erzeugt, das die Länge des Pakets ausgibt. Gegebenenfalls sollte diese Ausgabe erweitert werden, um Pakete eindeutig zu identifizieren.

3 Test

Testen Sie ihr Protokoll indem Sie eine Datei erfolgreich über einen Link mit einer Paketfehlerwahrscheinlichkeit von 10% übertragen. Stellen Sie aber bei der Implementierung zunächst eine Paketfehlerwahrscheinlichkeit von 0% ein, um ein funktionierendes Protokoll zu erstellen. Testen Sie auch

- Nachrichten, die kleiner als die eingestellte Segmentgröße sind
- Nachrichten, deren Größe kein Vielfaches der eingestellten Segmentgröße ist
- die Übertragung von mehreren Nachrichten

4 Statistische Auswertung

Bestimmen Sie für eine Nachricht von 257500 Bytes, einer Segmentgröße von 1000 Bytes und einer Paketverlustwahrscheinlichkeit von 10%, für welche Sendfenstergröße N die Nachricht am schnellsten zwischen zwei Laborrechnern übertragen wird.

