

12. Übung

Ausgabe Abgabe
15.1.16 29.1.16

Bitte bei der Abgabe Name der Mitglieder einer Gruppe, Nummer der Übung/Teilaufgabe und Datum auf den Lösungsblättern nicht vergessen! Darauf achten, dass die Lösungen beim richtigen Tutor abgegeben werden. Achten Sie bei Programmieraufgaben außerdem darauf, dass diese im Linuxpool kompilierbar sind. Nutzen Sie dazu die Flags `-std=c99`, `-Wall` und `-pedantic`. Es sollten keine Warnungen auftauchen.

Zu spät abgegebene Lösungen werden nicht berücksichtigt!

Aufgabe 1: Klausurteilnahme (0 Punkte)

Überlegen Sie sich bis zum 30.1.2016, ob Sie an einer Klausur teilnehmen wollen. Überprüfen Sie bitte, den Status Ihrer An- bzw. Abmeldung zur Klausur im KVV. Melden Sie sich für die Klausur an oder ab.

Ohne Anmeldung im KVV zur Klausur ist es rechtlich nicht möglich an der Prüfung teilzunehmen. Bei nicht erscheinen zur Prüfung und vorheriger Anmeldung bzw. nicht Abmeldung zur Klausur im KVV, wird die Prüfung mit 5 gewertet und es wird einer von vier möglichen Versuchen als nicht bestanden vermerkt.

Weitere Informationen finden Sie unter: <http://www.inf.fu-berlin.de/w/Inf/Pruefungsausschuss#KlausurVerfahren>

Aufgabe 2: Layers (2 Punkte)

a) Sei 01111110 das Flag Bit Pattern, 11100000 das ESC Pattern.

Zeichen	Codierung
A	01000111
B	11100011

- 1) Es soll die Bitfolge A B FLAG ESC B B übermittelt werden. Es wird als Framing-Verfahren Flag Bytes mit Byte Stuffing verwendet. Geben Sie die zu übermittelnde Bitfolge und die entsprechende Zeichenkette an. Machen Sie hinzugefügte Bytes kenntlich.
 - 2) Es soll die Bitfolge A B FLAG ESC B B übermittelt werden. Es wird als Framing-Verfahren Flag Bytes mit Bit Stuffing verwendet. Geben Sie die zu übermittelnde Bitfolge und die entsprechende Zeichenkette an. Machen Sie hinzugefügte Bits und Bytes kenntlich.
 - 3) Es soll die Bitfolge 0111101111101110111110 übertragen werden. Wie sieht die Bitfolge nach Bitstuffing aus?
- b) Erklären Sie die Unterschiede zwischen 10Base-T, 100Base-TX und 1000Base-CX auf der Übertragungsschicht (Frequenzen, Bandbreiten, Modulationen, Codierung, etc.)
Erklären Sie die maximalen Kabellängen der Verfahren.
- c) Machen Sie sich mit dem "Hidden station problem" vertraut. Beantworten Sie dazu folgende Fragen:
- 1) Erklären Sie das Problem der „Versteckten Endgeräte“.

- 2) In was für Netzen tritt es auf?
- 3) Durch welches Verfahren kann man dieses Problem lösen?

Aufgabe 3: Traceroute (3 Punkte)

Programmieren sie das Tool Traceroute (tracert) nach. Traceroute liefert zu einem gegebenen Server alle Router zurück, die ein IP Paket auf dem Weg zum Ziel passiert. Recherchieren Sie zu nächst folgende Begriffe und erklären Sie diese:

- Traceroute
- Time to live (TTL)
- Internet Control Message Protocol (ICMP)
- RAW-Socket

Testen Sie anschließend einen Tracerouter. Unter Linux ist im Terminal der Befehl **traceroute Ziel** verfügbar.

Die Funktionsweise eines Tracerouters sollte folgende sein:

Schicken Sie ein UDP-Paket auf einen beliebigen Port des Ziel-Servers. Manipulieren Sie dabei die TTL so, dass der erste unbekannte Router das Paket verwirft und Sie per ICMP über die abgelaufene TTL benachrichtigt. Die ICMP Nachricht empfangen Sie und geben die Absender-IP (Der Router) aus. Optional können Sie auch noch über Reverse-DNS den Namen anzeigen. Danach erhöhen Sie die TTL um eins und machen das Selbe nochmal. Das machen Sie so lange, bis ihr Paket den Empfänger erreicht und eine ICMP-Nachricht über einen nicht geöffneten Port zurückkommt.

Implementierungshinweise:

Um bei Paketen die TTL zu manipulieren gibt es zwei Möglichkeiten:

1. Sie öffnen einen RAW-Socket und bauen den IP Header und den UDP Header selber. Für die Header gibt es includes mit passenden Structs.
2. Sie manipulieren auf einem normalen UDP-Socket `setsockopt()` den TTL.

Damit Ihr Programm die ICMP Pakete der jeweiligen Router auch empfangen kann müssen sie einen RAW-Socket (`SOCK_RAW`) öffnen und als Protokoll ICMP (`IPPROTO_ICMP`) nutzen. Dann bekommen Sie auf diesem Socket alle ICMP-Pakete des Systems.

Jetzt müssen Sie nur noch herausfinden, welche der ICMP-Pakete eine Antwort auf Ihre Anfragen sind. Dazu als Tipp: In den ICMP-Paketen steht der gesamte IP-Header und die ersten 8 Byte der Payload des gesendeten Paketes drinnen. Da unser UDP Header genau 8 Byte lang ist, ist es also der komplette UDP-Header. Jetzt können Sie einfach den Quellport der UDP-Anfrage mit dem von Ihnen genutzten Quellport vergleichen und so das gewünschte Paket identifizieren.

Diese Version von tracert wird mit fu-berlin.de funktionieren, mit den meisten anderen Servern jedoch nicht. Das liegt daran, dass einige Firewalls Ihren UDP-Port blocken werden und das Paket ohne ICMP-Nachricht einfach verwerfen. Wollen Sie also ein universelleres tracert implementieren, nehmen Sie stattdessen TCP statt UDP und Port80 für zu testende Webserver.

Nicht verzagen: Die Aufgabe programmiert man nicht mal so eben zwischen Tür und Angel, sondern muss die Konzepte verstanden haben. Bevor Sie einfach drauf los programmieren bemühen Sie lieber google und fragen Sie Ihre Tutoren.

Wichtig: Die in den Tutorien behandelte Lösung deckt nur das Implementieren unter Linux ab, unter Windows und BSD (Mac) funktionieren Raw-Sockets anders. Außerdem brauchen Sie für das Nutzen von Raw-Sockets root-Rechte. Daher müssen Sie wenn Sie keinen Linux Rechner besitzen, das Ganze in einer VM programmieren.