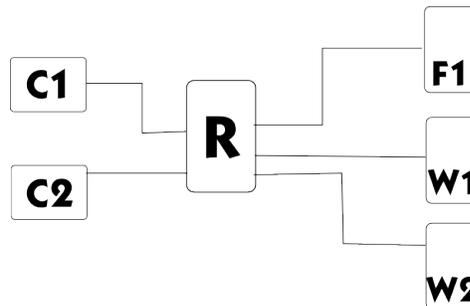


Übungen zur Vorlesung  
**Systeme II / Netzwerke I**  
 Sommer 2010  
 Blatt 10

**AUFGABE 1:**

1. Beschreiben Sie die Funktionsweise eines NAT bzw. eines NAPT und erläutern Sie deren Vor- und Nachteile.
2. In der unten abgebildeten Grafik ist ein Netzwerk dargestellt, wobei der Rechner R die NAT darstellt.



Rechner		IP	Port	Port (ausgehend)
C1		192.168.0.10		:1601
C2		192.168.0.11		:1701
R	if1:	192.168.0.1		
	if2:	192.168.1.1		
F1		192.168.1.10	:21	
W1		192.168.1.11	:80	
W2		192.168.1.12	:8080	

- Der Computer C1 möchte eine FTP Verbindung zu dem Rechner F1 herstellen.
- Der Computer C2 möchte eine HTTP-Verbindung zu W1 herstellen, wird aber vom NAT (R) nach Web2 umgeleitet.

Geben Sie die Routing-Tabelle des NAT(R) an, in ihr sollten die Source- und Destination-Adressen stehen, sowie die verwendeten Ports.

3. Ihr Computer befindet sich hinter einer Firewall, alle Ports unterhalb 5000, abgesehen von Port 80 (http), sind gesperrt. Wie ist es möglich mit Hilfe eines NAT trotzdem eine Verbindung zu einem entfernten Rechner unterhalb von Port 5000 herzustellen? Sie können dabei annehmen dass ihnen ein Server ausserhalb der Firewall zu Verfügung steht. Beschreiben Sie das Verfahren, um eine IMAP-Verbindung aufbauen zu können.

## AUFGABE 2:

In dieser Aufgaben sollen Sie einen Server programmieren der Folgende Aufgabe erledigt: Sendet ein Client eine Textnachricht an den Server, so sendet dieser die Textnachricht wieder zurück. In Java ließe sich diese Aufgabe in wenigen Zeilen mit den Klassen `ServerSocket` und `Socket`, die im standard JDK enthalten sind, lösen. Folgendes Codebeispiel soll Ihnen dabei helfen:

```
1 System.out.println("Warte_auf_Verbindung_auf_Port_xy...");
2
3 //Erselle einen neuen Socket, der auf Port xy hört...
4 ServerSocket echod= ...;
5
6 Socket socket = echod.accept();
7 System.out.println("Verbindung_hergestellt!");
8 InputStream in = socket.getInputStream();
9 OutputStream out = socket.getOutputStream();
10
11 int c;
12 while((c=in.read())!= -1){
13     //gib jedes Zeichen aus und sende es auch zurück
14 }
15 System.out.println("Verbindung_beenden");
16 socket.close();
17 echod.close();
```

1. Vervollständigen Sie den Code und starten Sie das Programm. Hinweis 1: Als Portnummer können Sie jede beliebige Ganzzahl von 0 bis 65535 wählen. Allerdings dürfen auf einigen Systemen aus Sicherheitsgründen Programme, die auf Ports kleiner als 1024 hören, nur mit Administratorrechten ausgeführt werden. Hinweis 2: Ein Java Programm starten Sie mit dem Befehl `java programm.class`, wobei `programm.class` die compilierte Version Ihres Programms ist.
2. Senden Sie mit dem Programm `telnet` einige Textnachrichten an Ihren Server, z.B. mit dem Aufruf `telnet localhost portxy`.
3. Versuchen Sie das Gleiche auch mit einem Webbrowser und notieren Sie sich die Ausgabe Ihres Servers.
4. Überlegen Sie sich einige Erweiterungen des obigen Codes und implementieren Sie diese, sodass Sie sie in der Übungsstunde demonstrieren können.