Protokoll

Versuch:	Linux ans Netz
Datum:	01.11.2007
Gruppe:	1
Studiengang:	Informationstechnik / Technische Informatik
Praktikanten:	Linda Fröck Karsten Wiedmann Maik Gotzmann
Ort:	Universität Rostock, Institut für Informatik, Raum D004

Benutze Geräte:

- Computer-Arbeitsplatz Nr. 1
- Netgear Netzwerk-Hub, 4-Port
- drei cat5-Netzwerkkabel, beschriftet
- BS: Fedora Core

Ziele:

Durch die Durchführung des Versuches sollen die Praktikanten folgende Fähigkeiten erwerben:

1. Integration eines Linux-Rechners in ein bestehendes TCP/IP-Netzwerk

2. Erkennen fehlerhafter Netzwerkkonfigurationen und deren Fehlerbehebung

Aufgabe 1: Verbindung - Hardware

Mit Hilfe des 4-Port-Netzwerkhubes sind drei durchnummerierte Netzwerk-Testkabel auf ihre Eigenschaften hin (defekt, patch, crossed-patch) zu überprüfen.

Durchführung:

Zur Inbetriebnahme des 4-Port Hubes wird dessen Netzteil am Stromversorgungsnetz angeschlossen.
Der Hub besitzt für den Port Nr. 4 einen Umschalter (Normal/Uplink), mit deren Hilfe sich weitere Hubs in Reihe schalten lassen.

Nun werden alle drei Testkabel der Reihe nach überprüft. Dies geschieht, indem der Stecker an einem Ende des Kabels in Port 1 des Hubs eingesteckt wird, während der Stecker des anderen Endes mit Port 4 verbunden wird. Eine Leuchtdiode (Anzeige: Link) an jedem Port des Hubs signalisiert dabei eine bestehende Verbindung. Der Umschalter des Port 4 ist dabei bei jedem Testkabel einmal auf Normal und Uplink zu schalten. Um eine Fehlfunktion des Hubs in einem gewissem Maße auszuschließen, wird jedes Testkabel gleichfalls an den Ports 2-3 getestet.

Meßreihe:

Kabel Nr. 1:

Port 1-4	"Link"-Signalisierung auf Port 4 im Uplink-Mode, keine Signalisierung im Normal-Mode
Port 2-3	Keine Signalisierung

Kabel Nr. 2:

Port 1-4	"Link"-Signalisierung auf beiden Ports im Uplink-Mode, keine Signalisierung im Normal-Mode
Port 2-3	Keine Signalisierung

Kabel Nr. 3:

Port 1-4	"Link"-Signalisierung auf beiden Ports im Normal-Mode, keine Signalisierung im Uplink-Mode
Port 2-3	Keine Signalisierung

Ergebnis und Auswertung:

Kabel Nr. 1: defektes Patchkabel

Bei einem Patchkabel erfolgt im Normal-Mode keine Signalisierung "Link" (und damit auch keine Verbindung), da hier durch die ungekreuzte Verbindung der TX-Kanal des Einen auf den TX-Kanal des anderen Ports zeigt (Anm.: TX - senderseitig, RX - empfangsseitig). Mit dem RX-Kanal der beiden Ports verhält es sich ebenso. So versuchen beide Ports zu senden, und Keiner kann etwas empfangen. Im Uplink-Mode zeigt der TX-Kanal des 1. Ports auf den RX-Kanal des Vierten: Die Verbindung steht (signalisiert durch "Link" an Port 4). Die fehlende "Link"-Signalisierung an Port 1 führt zu oben genannter Diagnose: Die Verbindung des TX-Kanals von Port 4 auf den RX-Kanal des Port 1 ist unterbrochen, was auf durchgetrennte Adern im Netzwerkkabel zurückzuführen ist.

Kabel Nr. 2: intaktes Patchkabel

Die jeweilige "Link"-Signalisierung an Port 1 und 4 im Uplink-Mode läßt darauf deuten, dass die beiden Ports ordnungsgemäß verbunden sind, d.h., der TX-Kanal des Port 1 zeigt auf den RX-Kanal des Port 4, und der TX-Kanal des Port 4 zeigt auf den RX-Kanal von Port 1. Es erfolgt keine Signalisierung im Normal-Mode (oder auch zwischen Port 2-3), somit ist auszuschließen, dass Adern gekreuzt oder kurzgeschlossen sind !

Kabel Nr. 3: intaktes Cross-Patchkabel

Durch die Signalisierung "Link" an Port 1 und 4 im Normal Mode (ebenso bei Verbindung Port 2 zu 3) ist davon auszugehen, dass die Adern in diesem Netzwerkkabel gekreuzt sind. Dafür spricht auch die fehlende Signalisierung im Uplink-Mode.

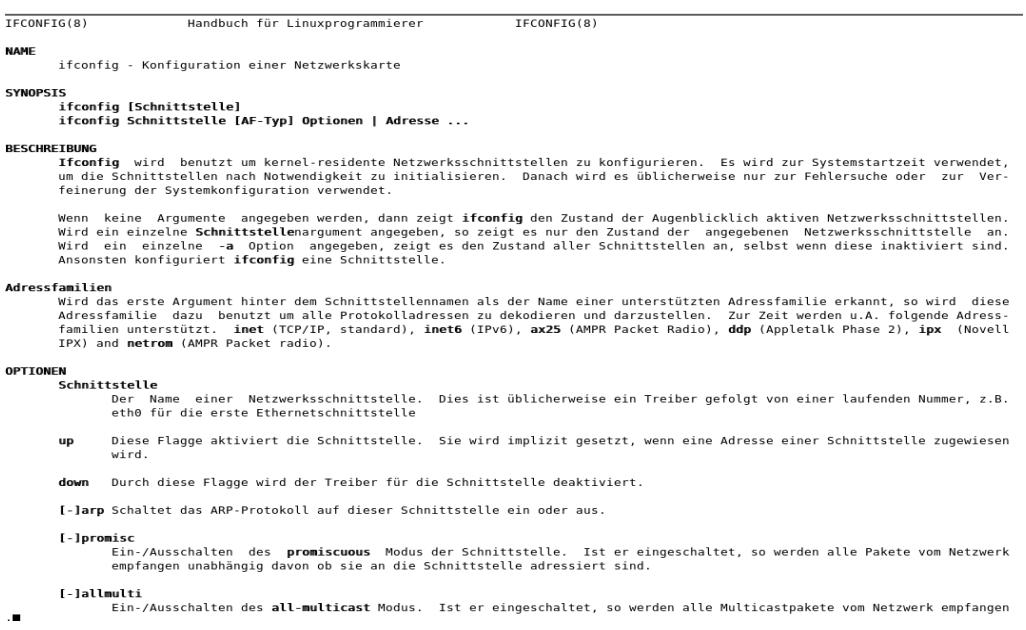
Der Port 4 des Hubs erwies sich beim Testen der Kabel als besonders gut geeignet, da er sich durch manuelles Umschalten in zwei Moden betreiben lässt: Im Normal-Mode verhält er sich wie Port 1,2 und 3 (gleiche Beschaltung der Kanäle), während im Uplink-Mode RX und TX miteinander vertauscht sind. Erst dadurch wird das Testen eines normalen Patchkabels (inklusive Funktionstest) möglich.

Aufgabe 2: Verbindung Software

Der Rechner wird nun softwareseitig manuell konfiguriert, um ihn in das bestehende Netzwerk zu integrieren.

Mittels des Befehls *man* lassen sich jederzeit aus der Konsole heraus, Erklärungen und Hilfen zu den einzelnen Linux-Befehlen anzeigen.

Mit *man ifconfig* wird der Befehl/Programm ifconfig ausführlich erklärt:



```
IFCONFIG(8)            Handbuch für Linuxprogrammierer            IFCONFIG(8)

NAME
       ifconfig - Konfiguration einer Netzwerkskarte

SYNOPSIS
       ifconfig [Schnittstelle]
       ifconfig Schnittstelle [AF-Typ] Optionen | Adresse ...

BESCHREIBUNG
       Ifconfig  wird  benutzt um kernel-residente Netzwerksschnittstellen zu konfigurieren.  Es wird zur Systemstartzeit verwendet,
       um die Schnittstellen nach Notwendigkeit zu initialisieren.  Danach wird es üblicherweise nur zur Fehlersuche oder  zur  Ver-
       feinerung der Systemkonfiguration verwendet.

       Wenn  keine  Argumente  angegeben werden, dann zeigt ifconfig den Zustand der Augenblicklich aktiven Netzwerksschnittstellen.
       Wird ein einzelne Schnittstellenargument angegeben, so zeigt es nur den Zustand der  angegebenen  Netzwerksschnittstelle  an.
       Wird  ein  einzelne  -a  Option  angegeben, zeigt es den Zustand aller Schnittstellen an, selbst wenn diese inaktiviert sind.
       Ansonsten konfiguriert ifconfig eine Schnittstelle.

Adressfamilien
       Wird das erste Argument hinter dem Schnittstellennamen als der Name einer unterstützten Adressfamilie erkannt, so wird  diese
       Adressfamilie  dazu  benutzt um alle Protokolladressen zu dekodieren und darzustellen.  Zur Zeit werden u.A. folgende Adress-
       familien unterstützt.  inet (TCP/IP, standard), inet6 (IPv6), ax25 (AMPR Packet Radio), ddp (Appletalk Phase 2), ipx  (Novell
       IPX) and netrom (AMPR Packet radio).

OPTIONEN
       Schnittstelle
              Der  Name  einer  Netzwerksschnittstelle.  Dies ist üblicherweise ein Treiber gefolgt von einer laufenden Nummer, z.B.
              eth0 für die erste Ethernetschnittstelle

       up     Diese Flagge aktiviert die Schnittstelle.  Sie wird implizit gesetzt, wenn eine Adresse einer Schnittstelle zugewiesen
              wird.

       down   Durch diese Flagge wird der Treiber für die Schnittstelle deaktiviert.

       [-]arp Schaltet das ARP-Protokoll auf dieser Schnittstelle ein oder aus.

       [-]promisc
              Ein-/Ausschalten  des  promiscuous  Modus der Schnittstelle.  Ist er eingeschaltet, so werden alle Pakete vom Netzwerk
              empfangen unabhängig davon ob sie an die Schnittstelle adressiert sind.

       [-]allmulti
              Ein-/Ausschalten des all-multicast Modus.  Ist er eingeschaltet, so werden alle Multicastpakete vom Netzwerk empfangen
.■
```

Mit dem Befehl *ifconfig -a* informieren wir uns nun über alle installierten Netzwerkschnittstellen des Rechners (läßt man die Option -a weg, so werden hier nur die zur Zeit aktiven Schnittstellen angezeigt):

```
[root@eins ~]# ifconfig -a
eth0      Link encap:Ethernet  Hardware Adresse 00:30:05:4F:EE:75
          inet Adresse:192.168.0.11  Bcast:192.168.0.255  Maske:255.255.255.0
          inet6 Adresse: fe80::230:5ff:fe4f:ee75/64 Gültigkeitsbereich:Verbindung
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:461 errors:0 dropped:0 overruns:0 frame:0
          TX packets:280 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:100
          RX bytes:98918 (96.5 KiB)  TX bytes:30599 (29.8 KiB)
          Basisadresse:0x3000 Speicher:e9000000-e9020000

lo        Link encap:Lokale Schleife
          inet Adresse:127.0.0.1  Maske:255.0.0.0
          inet6 Adresse: ::1/128 Gültigkeitsbereich:Maschine
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:34 errors:0 dropped:0 overruns:0 frame:0
          TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:0
          RX bytes:2360 (2.3 KiB)  TX bytes:2360 (2.3 KiB)

[root@eins ~]# ■
```

Man sieht, dass zwei Schnittstellen installiert und aktiviert sind: Die Eine ist eine lokale Schleife, die andere (eth0) stellt die Netzwerkkarte dar, deren augenblickliche Konfiguration ebenfalls angezeigt wird.

Ob der Gerätetreiber der PCI-Ethernetkarte ordnungsgemäß geladen und installiert ist, erkennt man nach Eingabe des Befehls *lspci*.

```
[root@eins ~]# lspci
00:00.0 Host bridge: Intel Corporation 82865G/PE/P DRAM Controller/Host-Hub Interface (rev 02)
00:01.0 PCI bridge: Intel Corporation 82865G/PE/P PCI to AGP Controller (rev 02)
00:03.0 PCI bridge: Intel Corporation 82865G/PE/P PCI to CSA Bridge (rev 02)
00:1d.0 USB Controller: Intel Corporation 82801EB/ER (ICH5/ICH5R) USB UHCI Controller #1 (rev 02)
00:1d.1 USB Controller: Intel Corporation 82801EB/ER (ICH5/ICH5R) USB UHCI Controller #2 (rev 02)
00:1d.2 USB Controller: Intel Corporation 82801EB/ER (ICH5/ICH5R) USB UHCI Controller #3 (rev 02)
00:1d.3 USB Controller: Intel Corporation 82801EB/ER (ICH5/ICH5R) USB UHCI Controller #4 (rev 02)
00:1d.7 USB Controller: Intel Corporation 82801EB/ER (ICH5/ICH5R) USB2 EHCI Controller (rev 02)
00:1e.0 PCI bridge: Intel Corporation 82801 PCI Bridge (rev c2)
00:1f.0 ISA bridge: Intel Corporation 82801EB/ER (ICH5/ICH5R) LPC Interface Bridge (rev 02)
00:1f.1 IDE interface: Intel Corporation 82801EB/ER (ICH5/ICH5R) IDE Controller (rev 02)
00:1f.3 SMBus: Intel Corporation 82801EB/ER (ICH5/ICH5R) SMBus Controller (rev 02)
00:1f.5 Multimedia audio controller: Intel Corporation 82801EB/ER (ICH5/ICH5R) AC'97 Audio Controller (rev 02)
01:00.0 VGA compatible controller: Matrox Graphics, Inc. MGA G400/G450 (rev 85)
02:01.0 Ethernet controller: Intel Corporation 82547EI Gigabit Ethernet Controller
[root@eins ~]#
```

An letzter Position ist dann auch unsere Netzwerkkarte zu finden: Es ist eine von Intel, das Modell wird angezeigt sowie eine allgemeine Gerätebezeichnung. Hier handelt es sich also um eine Ethernetkarte (kein On-Board Controller, dann müsste in der ersten Spalte "00" stehen) deren theoretisch maximale Datenübertragungsgeschwindigkeit 1 GBit/s beträgt. Ob sie auch in diesem Modus betrieben wird, ist hieraus nicht ersichtlich (hängt von den Einstellungen und dem übrigen Netzwerk ab !).

Ob und welches Kernel-Modul zum Betreiben der Netzwerkschnittstelle beim Booten von Linux geladen wurde, wird nach Eingabe des Befehls *lsmod* ersichtlich:

```
e1000                 118401  0
iTCO_wdt               14693  0
snd_page_alloc         14281  2 snd_intel8x0,snd_pcm
i2c_core               27841  1 i2c_i801
iTCO_vendor_support     7877  1 iTCO_wdt
ide_cd                 40545  0
cdrom                  37089  1 ide_cd
parport_pc             30821  1
parport                38281  2 lp,parport_pc
serio_raw              10821  0
ata_piix               18757  0
libata                120881  1 ata_piix
sd_mod                 31297  0
scsi_mod              140621  2 libata,sd_mod
ext3                  125641  1
jbd                    59881  1 ext3
mbcache                12485  1 ext3
ehci_hcd               35405  0
ohci_hcd               23877  0
uhci_hcd               27089  0
[root@eins ~]#
```

An erster Position steht hier das Modul "e1000", welches genau das entsprechende Modul für unsere Netzwerkschnittstelle ist. Sollte das entsprechende Modul in dieser recht langen Liste nicht auftauchen, so ist es jederzeit möglich, manuell weitere Module zu laden. Dies geschieht mit dem Befehl *modprobe e1000* zum Beispiel, um das Gigabit-Übertragungsmodul zu laden.

Mit dem Befehl netstat -r können wir uns nun die Routingtabelle anzeigen lassen:

```
[root@eins ~]# netstat -r
Kernel IP Routentabelle
Ziel            Router          Genmask         Flags   MSS Fenster irtt Iface
192.168.0.0     *               255.255.255.0   U         0 0          0 eth0
169.254.0.0     *               255.255.0.0     U         0 0          0 eth0
default         192.168.0.1     0.0.0.0         UG        0 0          0 eth0
```

Damit gehen alle Datenpakete, die an die ersten beiden Empfänger (Ziel) adressiert sind, direkt an die Rechner oder Netze, die als Empfänger deklariert sind. Datenpakete, deren Empfänger nicht den ersten beiden Einträgen dieser Tabelle entspricht, werden an den Rechner 192.168.0.1 weitergeleitet, der hier als Router fungiert und damit "hoffentlich" weiß, an wen er sie weiterleiten muß. Der letzte Eintrag definiert damit, über welchen weiteren Knoten/Rechner die Kommunikation mit externen Netzen (z.B. Internet) abläuft.

In der Spalte Iface wird noch darüber informiert, über welche Netzwerkschnittstelle die Weiterleitung der Datenpakete realisiert wird. Hier ist es unsere Intel Netzwerkkarte.

Nun wird es Zeit, die eingebaute Netzwerkkarte (Device eth0) manuell zu konfigurieren. Dazu geben wir folgenden Befehl ein:

ifconfig eth0 192.168.0.21 netmask 255.255.255.0 broadcast 192.168.0.255 up

Mittels der Eingabe von *ifconfig* und *netstat -r* überzeugen wir uns vom Resultat:

```
[root@eins ~]# ifconfig eth0 192.168.0.21 netmask 255.255.255.0 broadcast 192.168.0.255 up
[root@eins ~]# ifconfig
eth0      Link encap:Ethernet  Hardware Adresse 00:30:05:4F:EE:75
          inet Adresse:192.168.0.21  Bcast:192.168.0.255  Maske:255.255.255.0
          inet6 Adresse: fe80::230:5ff:fe4f:ee75/64 Gültigkeitsbereich:Verbindung
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:527 errors:0 dropped:0 overruns:0 frame:0
          TX packets:304 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:100
          RX bytes:118622 (115.8 KiB)  TX bytes:35077 (34.2 KiB)
          Basisadresse:0x3000 Speicher:e9000000-e9020000

lo        Link encap:Lokale Schleife
          inet Adresse:127.0.0.1  Maske:255.0.0.0
          inet6 Adresse: ::1/128 Gültigkeitsbereich:Maschine
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:34 errors:0 dropped:0 overruns:0 frame:0
          TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:0
          RX bytes:2360 (2.3 KiB)  TX bytes:2360 (2.3 KiB)

[root@eins ~]# netstat -r
Kernel IP Routentabelle
Ziel            Router          Genmask         Flags   MSS Fenster irtt Iface
192.168.0.0     *               255.255.255.0   U         0 0          0 eth0
[root@eins ~]#
```

Wir haben unsere Netzwerkkarte (eth0) also wie folgt konfiguriert:

- Zuordnung der IP-Adresse für diesen Rechner mit *192.168.0.21*
- mit der Netzmaske *255.255.255.0* haben wir uns genau diesem Klasse C-Netz zugeordnet
- unsere Datenpakete werden nun über das Netz *192.168.0.255* veröffentlicht
- zu guter Letzt haben wir unsere Netzwerkkarte mit genau diesen Einstellungen aktiviert, durch den Zusatz *up*

Die Routing-Tabelle enthält jedoch nur einen einzigen Eintrag, womit zwar die Kommunikation innerhalb des C-Netzes gewährleistet ist, aber in externe Netze (z.B. Internet) gelangt man so nicht. Erst durch Hinzufügen des Default-Gateways mit dem Befehl *route add default gw 192.168.0.1* sind wir in der Lage, das eigene Subnetz zu verlassen.

```
[root@eins ~]# route add default gw 192.168.0.1
[root@eins ~]# netstat -r
Kernel IP Routentabelle
Ziel            Router          Genmask         Flags   MSS Fenster irtt Iface
192.168.0.0     *               255.255.255.0   U         0 0          0 eth0
default         192.168.0.1     0.0.0.0         UG        0 0          0 eth0
[root@eins ~]#
```

Nun testen wir eine Verbindung zu einem anderen Rechner mit den Befehl:

ping -c 5 192.168.0.20

```
[root@eins ~]# ping 192.168.0.20 -c 5
PING 192.168.0.20 (192.168.0.20) 56(84) bytes of data.
64 bytes from 192.168.0.20: icmp_seq=1 ttl=64 time=0.417 ms
64 bytes from 192.168.0.20: icmp_seq=2 ttl=64 time=0.499 ms
64 bytes from 192.168.0.20: icmp_seq=3 ttl=64 time=0.290 ms
64 bytes from 192.168.0.20: icmp_seq=4 ttl=64 time=0.316 ms
64 bytes from 192.168.0.20: icmp_seq=5 ttl=64 time=0.352 ms

--- 192.168.0.20 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.290/0.374/0.499/0.079 ms
[root@eins ~]#
```

Und wer hätte es gedacht, der Rechner mit der IP-Adresse *192.168.0.20* hat auf die von uns gesendeten 64 Byte Datenpakete geantwortet, womit sichergestellt ist, dass eine Verbindung besteht. Weiterhin wird uns angezeigt, wieviel Zeit verging, bis die Antwort zu unserem Datenpaket den Absender erreichte. Die Option *-c 5* bestimmt dabei lediglich, wie oft das Ziel zu Testzwecken angepingt wird: in diesem Falle 5-mal.

Mit dem Befehl *traceroute 192.168.0.20* läßt sich der Weg zurückverfolgen, der nötig war, um diesen Empfänger zu erreichen:

```
[root@eins ~]# traceroute 192.168.0.20
traceroute to 192.168.0.20 (192.168.0.20), 30 hops max, 40 byte packets
 1   (192.168.0.20)  0.501 ms  0.435 ms  0.407 ms
[root@eins ~]#
```

Und hier gibt es nur einen einzigen Eintrag, was vermuten lässt, dass das Ziel direkt angesprochen wurde (es befindet sich im eigenen Subnetz). Das zuvor definierte Standard-Gateway wurde jedenfalls nicht benutzt, und Andere sind z.Zt. nicht definiert.

Aufgabe 3: Dekonfiguration

Zunächst wurde der Default-Router aus der Routing-Tabelle gelöscht. Dies macht man in diesem Fall mit dem Befehl *route del default gw 192.168.0.1*.Dieser Befehl wird in die Kommandozeile eingegeben.

```
[root@eins ~]# route del default gw 192.168.0.1
[root@eins ~]# netstat -r
Kernel IP Routentabelle
Ziel            Router          Genmask         Flags   MSS Fenster irtt Iface
192.168.0.0     *               255.255.255.0   U         0 0          0 eth0
```

Wie in der Abbildung zu sehen ist, fehlt nun die Adresse 192.168.0.1 in der Routing-Tabelle.

Weiterhin sollte die Netzwerkkarte deaktiviert werden. Dazu nutzt man den Befehl *ifconfig eth0.0.0.0 down.*

```
[root@eins ~]# ifconfig eth0 0.0.0.0 down
[root@eins ~]# ifconfig
lo        Link encap:Lokale Schleife
          inet Adresse:127.0.0.1  Maske:255.0.0.0
          inet6 Adresse: ::1/128 Gültigkeitsbereich:Maschine
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:34 errors:0 dropped:0 overruns:0 frame:0
          TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:0
          RX bytes:2360 (2.3 KiB)  TX bytes:2360 (2.3 KiB)
```

```
[root@eins ~]# ifconfig
eth0      Link encap:Ethernet  Hardware Adresse 00:30:05:4F:EE:75
          inet Adresse:192.168.0.21  Bcast:192.168.0.255  Maske:255.255.255.0
          inet6 Adresse: fe80::230:5ff:fe4f:ee75/64 Gültigkeitsbereich:Verbindung
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:673 errors:0 dropped:0 overruns:0 frame:0
          TX packets:495 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:100
          RX bytes:140138 (136.8 KiB)  TX bytes:52578 (51.3 KiB)
          Basisadresse:0x3000 Speicher:e9000000-e9020000

lo        Link encap:Lokale Schleife
          inet Adresse:127.0.0.1  Maske:255.0.0.0
          inet6 Adresse: ::1/128 Gültigkeitsbereich:Maschine
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:34 errors:0 dropped:0 overruns:0 frame:0
          TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:0
          RX bytes:2360 (2.3 KiB)  TX bytes:2360 (2.3 KiB)
```

In der Abbildung ist zu sehen, dass die Netzwerkkarte nicht mehr aktiviert ist. In der nächsten Abbildung ist zum Vergleich zu sehen, wie es vor der Eingabe des Befehls aussah.

Es ist ganz klar zu sehen, dass es vorher eine Verbindung zum Ethernet gab, die nun nicht mehr existiert.

Als letztes sollte in dieses Aufgabe auch der Treiber entfernt werden. Zu diesem Zweck wird der Befehl modprobe -r <<TREIBER>>.verwendet. In diesem Fall ist der Treiber e1000. So ergibt sich der Befehl *modpobe -r e1000*. In der folgenden Abbildung ist zu sehen, dass der Treiber e1000 nicht mehr in der Liste auftaucht.

```
[root@eins ~]# modprobe -r e1000
[root@eins ~]# lsmod
Module                  Size  Used by
autofs4                24773  2
hidp                   26689  2
l2cap                  30401  5 hidp
bluetooth              57893  2 hidp,l2cap
sunrpc                161981  1
nf_conntrack_netbios_ns     7105  0
ipt_REJECT              8641  1
nf_conntrack_ipv4      21837  2
xt_state                6593  2
nf_conntrack           64713  3 nf_conntrack_netbios_ns,nf_conntrack_ipv4,xt_state
nfnetlink               9945  2 nf_conntrack_ipv4,nf_conntrack
iptable_filter          7105  1
ip_tables              16517  1 iptable_filter
ip6t_REJECT             9537  1
xt_tcpudp               7233  9
ip6table_filter         6849  1
ip6_tables             17669  1 ip6table_filter
x_tables               18629  6 ipt_REJECT,xt_state,ip_tables,ip6t_REJECT,xt_tcpudp,ip6_tables
ipv6                  277957  25 ip6t_REJECT
vfat                   16193  1
fat                    53341  1 vfat
dm_mirror              25153  0
dm_multipath           21961  0
dm_mod                 57089  2 dm_mirror,dm_multipath
video                  20937  0
sbs                    22729  0
button                 12113  0
dock                   13921  0
battery                14149  0
ac                      9285  0
lp                     16105  0
snd_intel8x0           36061  0
snd_ac97_codec         96613  1 snd_intel8x0
ac97_bus                6465  1 snd_ac97_codec
snd_seq_dummy           7877  0
snd_seq_oss            33473  0
snd_seq_midi_event     11073  1 snd_seq_oss
snd_seq                50609  5 snd_seq_dummy,snd_seq_oss,snd_seq_midi_event
snd_seq_device         11981  3 snd_seq_dummy,snd_seq_oss,snd_seq
snd_pcm_oss            43457  0
snd mixer oss          19521  1 snd pcm oss
```

Aufgabe 4: Konfiguration über Startskripte

Um den Netzwerkkartentreiber automatisch beim Zugriff auf eth0 zu laden sollten wir in dieser Aufgabe als erstes *alias eth0 e1000* mit einem Texteditor in die Datei /etc/modprobe.conf einfügen.

```
alias snd-card-0 snd-intel8x0
options snd-card-0 index=0
options snd-intel8x0 index=0
remove snd-intel8x0 { /usr/sbin/alsactl store 0 >/dev/null 2>&1 || : ; }; /sbin/modprobe -r --ignore-remove snd-intel8x0
alias eth0 e1000
```

Als nächstes haben wir dann das Verzeichnis gewechselt. In dem Verzeichnis /etc/sysconfig/network-scripts haben wir die Datei ifcfg-eth0 mit folgenden Inhalt erstellt.

DEVICE=eth0
BOOTPROTO=static
BROADCAST=192.168.0.1
IPADDR=192.168.0.21
NETMASK=255.255.255.0
NETWORK=192.168.0.0
ONBOOT=yes

Dieser Eintrag bewirkt, dass dem Rechner nun manuell die Daten zugewiesen werden. Deswegen wird BOOTPROTP auf staic gesetzt. Anderenfalls wenn BOOTPROTO auf dynamic gesetzt wird, dann werden die Daten dem Rechner automatisch zugewiesen.

```
# Intel Corporation 82547EI Gigabit Ethernet Controller
DEVICE=eth0
BOOTPROTO=static
BROADCAST=192.168.0.255
IPADDR=192.168.0.21
NETMASK=255.255.255.0
NETWORK=192.168.0.0
ONBOOT=yes
```

Weiterhin haben wir den Eintrag GATEWAY in dem Verzeichnis auf 192.168.0.1 gesetzt.

```
NETWORKING=yes
NETWORKING_IPV6=yes
HOSTNAME=localhost.localdomain
GATEWAY=192.168.0.1
```

Wie in der nachstehenden Abbildung zu sehen ist, ist jetzt der Default-Router mit der Gateway-Adresse 192.168.0.1 wieder aktiviert.

```
[root@eins ~]# netstat -r
Kernel IP Routentabelle
Ziel            Router          Genmask         Flags   MSS Fenster irtt Iface
192.168.0.0     *               255.255.255.0   U         0 0          0 eth0
169.254.0.0     *               255.255.0.0     U         0 0          0 eth0
default         192.168.0.1     0.0.0.0         UG        0 0          0 eth0
```

```
[root@eins ~]# ifup eth0
[root@eins ~]# ifconfig
eth0      Link encap:Ethernet  Hardware Adresse 00:30:05:4F:EE:75
          inet Adresse:192.168.0.21  Bcast:192.168.0.255  Maske:255.255.255.0
          inet6 Adresse: fe80::230:5ff:fe4f:ee75/64 Gültigkeitsbereich:Verbindung
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:39 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:100
          RX bytes:64 (64.0 b)  TX bytes:8604 (8.4 KiB)
          Basisadresse:0x3000 Speicher:e9000000-e9020000

lo        Link encap:Lokale Schleife
          inet Adresse:127.0.0.1  Maske:255.0.0.0
          inet6 Adresse: ::1/128 Gültigkeitsbereich:Maschine
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:34 errors:0 dropped:0 overruns:0 frame:0
          TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:0
          RX bytes:2360 (2.3 KiB)  TX bytes:2360 (2.3 KiB)
```

Zuletzt haben wir dann die Netzwerkkarte wieder aktiviert. Dazu haben wir den Befehl *ifup eth0* in die Kommandozeile eingegeben. Wie man sieht ist jetzt wieder eine Verbindung zum Ethernet hergestellt.

Aufgabe 5: DNS-Unterstützung

5.1. Datei „/etc/resolv.conf“

Zu Beginn der Aufgabe haben wir in der Datei /etc/resolv.conf dem DNS-Server seine IP-Adresse zugewiesen:

*resolv.conf(Wird bearbeitet)

```
; generated by /sbin/dhclient-script
search netprak.iuk.test
nameserver 192.168.0.1
```

Mit dem Befehl „nameserver 192.168.0.1“ ordnen wir die IP unseres DNS-Servers zu (konkret 192.168.0.1). Somit erhält der „Resolver“ (Sammlung von C Routinen, die für den Zugriff auf das Namenssystem im Internet (DNS) verantwortlich sind) die Adresse des zuständigen DNS-Servers und kann diesen zur Namensauflösung verwenden.

Die Zeile „search netprak.iuk.test“ definiert den Suchpfad für die Domainauflösung. Somit wird der Resolver versuchen alle Domains des Suchpfades abzuarbeiten, bis eine Übereinstimmung mit Name und Domain gegeben ist.

Weiterhin könnten in dieser Datei:
- Domainname (domain),
- Sortierungsart (sortlist) und
- innere Beeinflussung (options / debug)

für den Resolver defineirt werden.

5.2. Datei „/etc/nsswitch.conf“

In der Datei „nsswitch.conf“ werden, unter anderem, Einstellungen für die Namensauflösung (Namenservice) definiert, wie z.B. der DNS (Domain Name Service) oder NIS (Network Information Service).
Unsere Aufgabe ist es nun zu definieren, dass zuerst lokale Konfigurationsdateien und dann erst der DNS verwendet wird:

```
hosts:      files dns
aliases:    files nisplus
```

- „Files“ ist dabei ein Alias und beinhaltet „nisplus“:

Somit wird die Priorität von „nisplus“ vor „dns“ gestetzt.

5.3. Datei „/etc/hosts“

In dieser Datei werden die Auflösung der Rechnernamen konfiguriert:

hosts(Wird bearbeitet)

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       .       localhost.localdomain localhost
::1     .       localhost.localdomain localhost
```

Zeile: „127.0.0.1 localhost.localdomain localhost“. Hier wird also der IP 127.0.0.1 der Name „localhost“ zugewiesen. Die IP 127.0.0.1 bezeichnet nun also die eigene Adresse mit „localhost“. Dies ist sinnvoll, da der Computer somit eine Abtrennung von einem Netzwerk unterscheiden kann.

5.4. nslookup

Nslookup dient zu Ermittlung von Servernamen bzw. IP-Adressen:

Ermittlung der IP-Adresse des Rechner „zehn“:

```
[root@eins ~]# nslookup zehn
Server:         192.168.0.1
Address:        192.168.0.1#53

Name:   zehn.netprak.iuk.test
Address: 192.168.0.20
```

Nslookup ermittelte die IP-Adresse 192.168.0.20.

5.5. dig

Dig dient zur Ermittlung der zuständigen DNS-Servernamen von Rechnernamen.

DNS-Servernamen von „www.informatik.uni-rostock.de“:

```
[root@eins ~]# dig +noall +authority www.informatik.uni-rostock.de
informatik.uni-rostock.de. 172682 IN    NS      haegar.informatik.uni-rostock.de.
informatik.uni-rostock.de. 172682 IN    NS      eric.informatik.uni-rostock.de.
informatik.uni-rostock.de. 172682 IN    NS      lara.informatik.uni-rostock.de.
```

DNS-Servernamen von „www.uni-rostock.de“:

```
[root@eins ~]# dig +noall +authority www.uni-rostock.de
uni-rostock.de.         86260   IN      NS      hp1.uni-rostock.de.
uni-rostock.de.         86260   IN      NS      top.uni-rostock.de.
uni-rostock.de.         86260   IN      NS      deneb.dfn.de.
uni-rostock.de.         86260   IN      NS      ws-kar1.win-ip.dfn.de.
```

DNS-Servernamen von „www.denic.de“:

```
[root@eins ~]# dig +noall +authority www.denic.de
denic.de.               84740   IN      NS      ns3.denic.de.
denic.de.               84740   IN      NS      ns4.denic.net.
denic.de.               84740   IN      NS      ns5.denic.net.
denic.de.               84740   IN      NS      ns1.denic.de.
denic.de.               84740   IN      NS      ns2.denic.de.
```

DNS-Servernamen von „www.fedora.com:

```
[root@eins ~]# dig +noall +authority www.fedora.com
fedora.com.             1942    IN      NS      b.ns.ultsearch.com.
fedora.com.             1942    IN      NS      a.ns.ultsearch.com.
```

Anschliesssend wurden zu den Rechnernamen die Namensauflösung der zugehörigen DNS-Server ermittelt:

Namensauflösung von „www.informatik.uni-rostock.de“:

```
[root@eins ~]# dig +trace www.informatik.uni-rostock.de

; <<>> DiG 9.3.4-P1 <<>> +trace www.informatik.uni-rostock.de
;; global options:  printcmd
.                       335327  IN      NS      k.root-servers.net.
.                       335327  IN      NS      l.root-servers.net.
.                       335327  IN      NS      m.root-servers.net.
.                       335327  IN      NS      a.root-servers.net.
.                       335327  IN      NS      b.root-servers.net.
.                       335327  IN      NS      c.root-servers.net.
.                       335327  IN      NS      d.root-servers.net.
.                       335327  IN      NS      e.root-servers.net.
.                       335327  IN      NS      f.root-servers.net.
.                       335327  IN      NS      g.root-servers.net.
.                       335327  IN      NS      h.root-servers.net.
.                       335327  IN      NS      i.root-servers.net.
.                       335327  IN      NS      j.root-servers.net.
;; Received 436 bytes from 192.168.0.1#53(192.168.0.1) in 3 ms
```

Namensauflösung von „www.uni-rostock.de“:

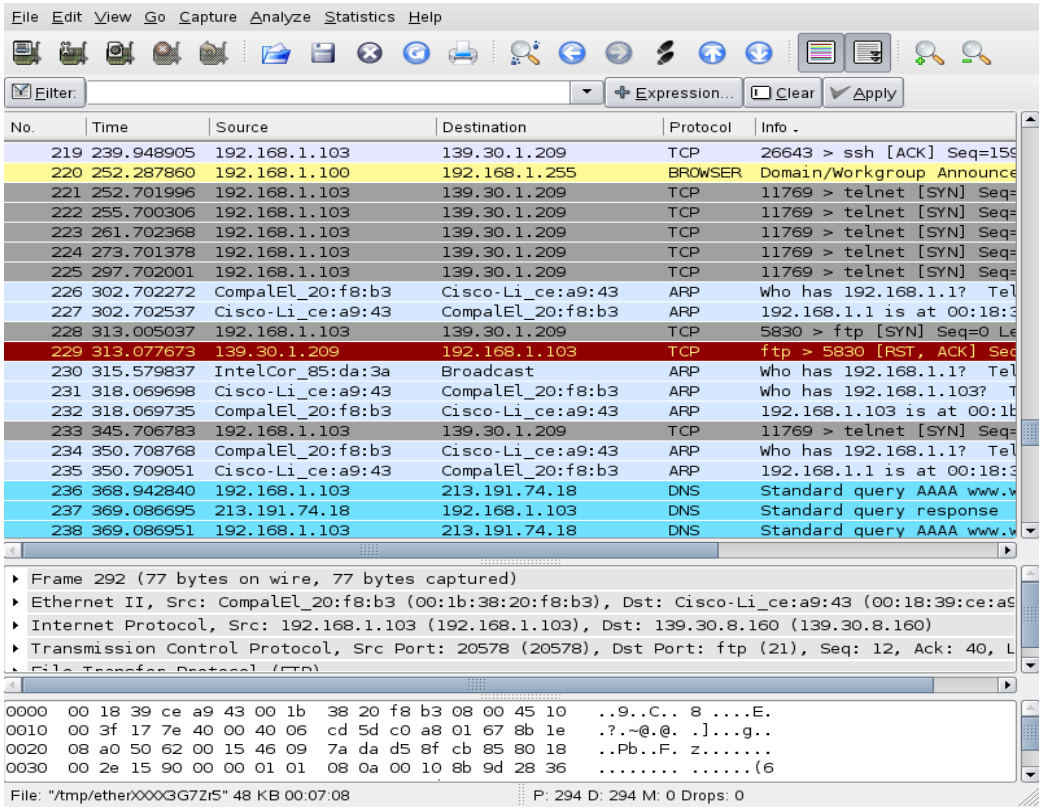
```
; <<>> DiG 9.3.4-P1 <<>> +trace www.uni-rostock.de
;; global options:  printcmd
.                       335254  IN      NS      l.root-servers.net.
.                       335254  IN      NS      m.root-servers.net.
.                       335254  IN      NS      a.root-servers.net.
.                       335254  IN      NS      b.root-servers.net.
.                       335254  IN      NS      c.root-servers.net.
.                       335254  IN      NS      d.root-servers.net.
.                       335254  IN      NS      e.root-servers.net.
.                       335254  IN      NS      f.root-servers.net.
.                       335254  IN      NS      g.root-servers.net.
.                       335254  IN      NS      h.root-servers.net.
.                       335254  IN      NS      i.root-servers.net.
.                       335254  IN      NS      j.root-servers.net.
.                       335254  IN      NS      k.root-servers.net.
;; Received 436 bytes from 192.168.0.1#53(192.168.0.1) in 2 ms
```

Namensauflösung von „www.denic.de“:

```
; <<>> DiG 9.3.4-P1 <<>> +trace www.denic.de
;; global options:  printcmd
.                       335190  IN      NS      b.root-servers.net.
.                       335190  IN      NS      c.root-servers.net.
.                       335190  IN      NS      d.root-servers.net.
.                       335190  IN      NS      e.root-servers.net.
.                       335190  IN      NS      f.root-servers.net.
.                       335190  IN      NS      g.root-servers.net.
.                       335190  IN      NS      h.root-servers.net.
.                       335190  IN      NS      i.root-servers.net.
.                       335190  IN      NS      j.root-servers.net.
.                       335190  IN      NS      k.root-servers.net.
.                       335190  IN      NS      l.root-servers.net.
.                       335190  IN      NS      m.root-servers.net.
.                       335190  IN      NS      a.root-servers.net.
;; Received 436 bytes from 192.168.0.1#53(192.168.0.1) in 2 ms
```

Aufagbe 6: Wireshark

Im folgenden Versuch haben wir den Datenverkehr über das Netzwerk mit Hilfe des Programmes „wireshark“ mitgeschnitten:



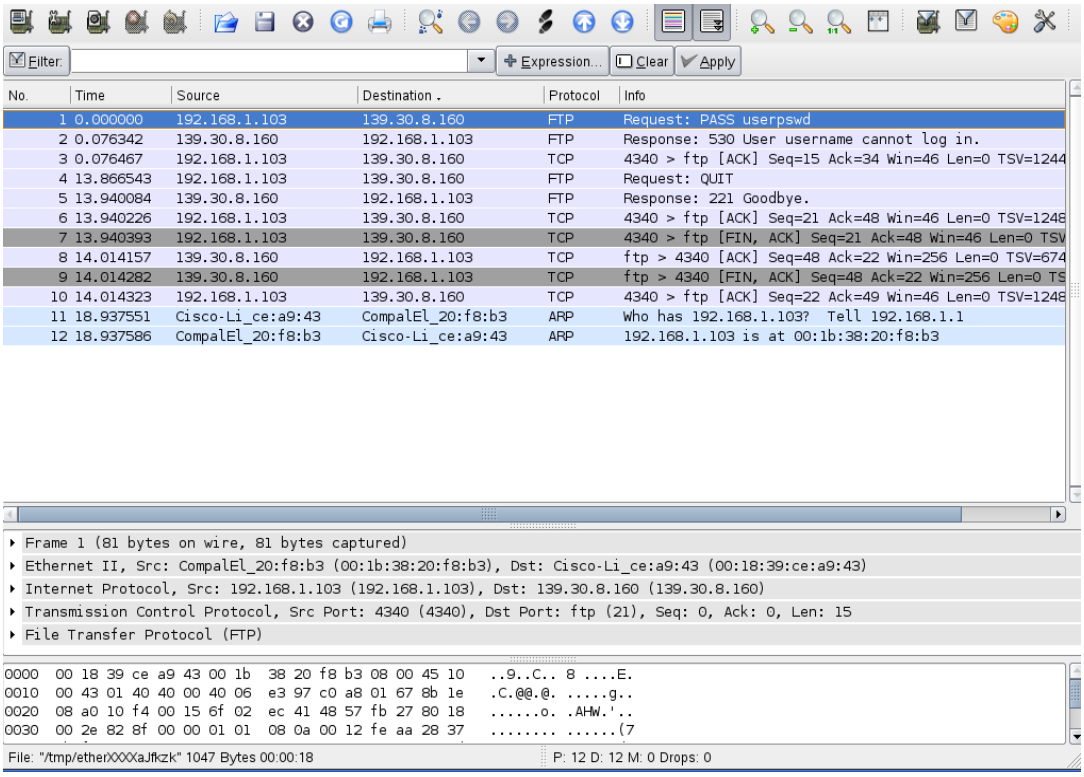
„Wireshark“ liefert den gesmanten Datenverkehr über die Ethernetschnittstelle.

Im Folgenden wird der Datenverkehr einer ftp-Verbindung analysiert:

Mittels ftp-Programm (FileTransferProtocol) haben wir versucht, uns mit einem Benutzernamen („username“) und einem Passwort („userpswd“) auf den Rechner „home.uni-rostock.de“ anzumelden:

```
linux-3dsh:~ # ftp
ftp> open
(to) home.uni-rostock.de
Connected to longhorn.uni-rostock.de.
220
Name (home.uni-rostock.de:root): username
331 Password required for username.
Password:
530 User username cannot log in.
ftp: Login failed.
ftp> exit
221 Goodbye.
```

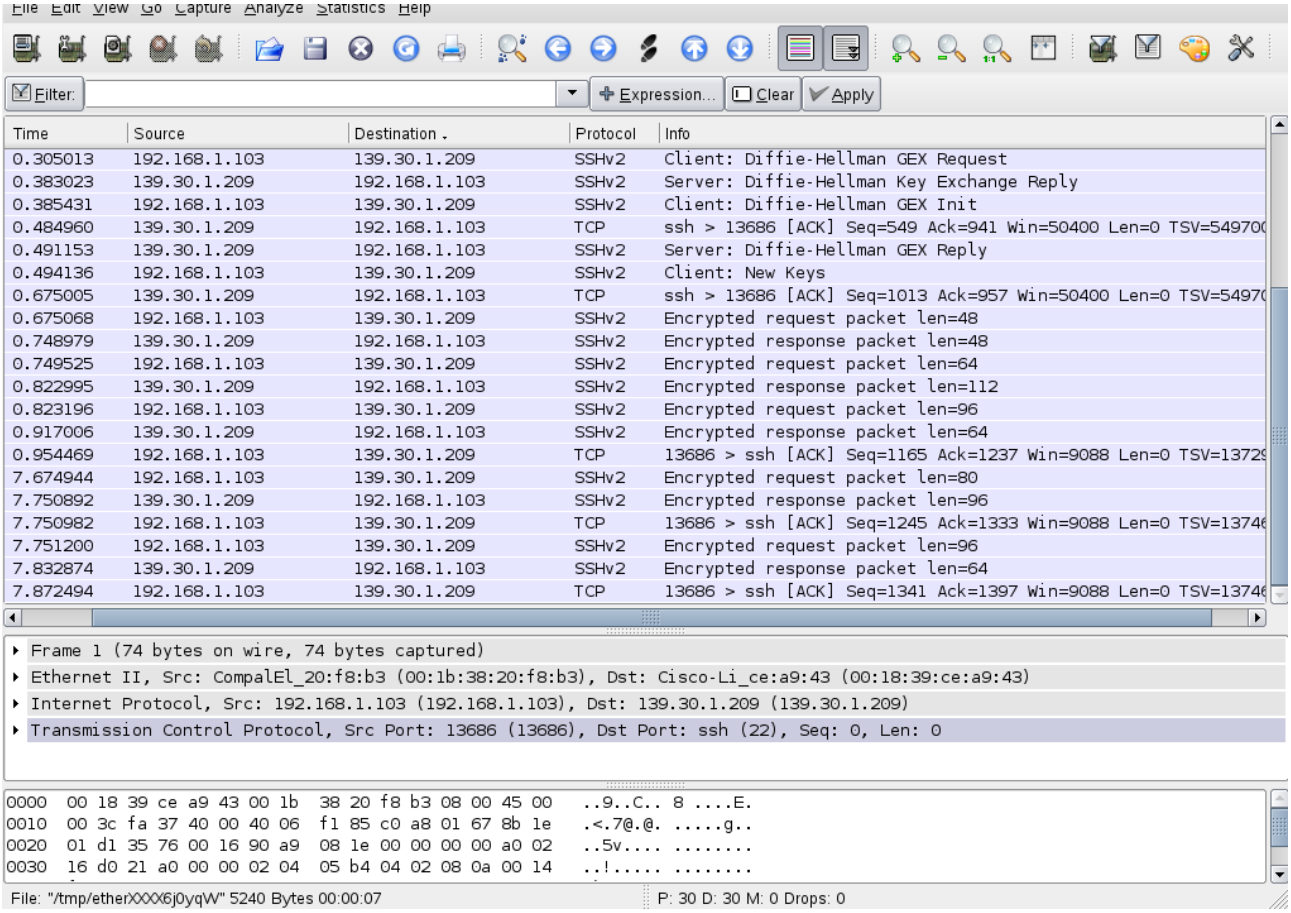
Betrachtet man nun über „wireshark“ den Datenverkehr, so sind die übermittelten Daten unverschlüsselt zu finden (Passwort „userpswd“ in blau makierter Zeile):



Anschliessend haben versucht uns per SSH-Programm (SecureShell) auf vesuv.informatik.uni-rostock.de anzumelden:

```
linux-3dsh:~ # ssh vesuv.informatik.uni-rostock.de
Password:
Password:
```

Analysiert man nun den Datenverkehr, so ist das Passwort verschlüsselt übertragen worden und somit nicht mehr lesbar:



Zusatzaufgabe 1: DNS (Domain Name System):

Mit Hilfe des DNS kann zu einem gegebenen Hostnamen die IP-Adresse ermittelt werden, bzw. der zugehörige Hostname einer IP-Adresse ermittelt werden. Dies ermöglicht z.B. das Programm „host“.

Mittels „host“ haben wir den Servernamen zur IP: 209.85.135.99 ermittelt:

```
alf@linux-3dsh:~> host 209.85.135.99
99.135.85.209.in-addr.arpa domain name pointer mu-in-f99.google.com.
alf@linux-3dsh:~>
```

Die IP-Adresse gehört somit zum Rechner: www.google.de.

Analog kann man sich alle IP-Adressen von einem Rechner anzeigen lassen: z.B.: von www.google.de:

```
alf@linux-3dsh:~> host www.google.de
www.google.de is an alias for www.google.com.
www.google.com is an alias for www.l.google.com.
www.l.google.com has address 209.85.135.104
www.l.google.com has address 209.85.135.99
www.l.google.com has address 209.85.135.147
www.l.google.com has address 209.85.135.103
www.google.de is an alias for www.google.com.
www.google.com is an alias for www.l.google.com.
www.google.de is an alias for www.google.com.
www.google.com is an alias for www.l.google.com.
```

Zusatzaufgabe 2: Fehler

Der Kontakt zu Rechnern außerhalb des eigenen Subnetzes bleibt verwehrt, wenn ein Standardgateway zu einem Ziel definiert wird, welches nicht als Router oder Routerserver fungiert. Mit dem Befehl *route add default gw IP-Adresse eines Clients* werden nun alle Datenpakete, adressiert an externe Rechner, an den normalen Teilnehmer gesendet, der jedoch damit nichts anfangen kann.

Aufgabe 7: Abschluß

Um die manuell konfigurierten Netzwerkeinstellungen rückgängig zu machen, schlagen wir zwei Methoden vor:

1. Editieren der Datei ifcfg-et01 im Ordner /etc/sysconfig/network/

   Im Startskript ersetzen wir den Eintrag BOOTPROTO=satic durch BOOTPROTO=dhcp, mit den Auswirkungen, dass der Netzwerkkarte nun alle benötigten Adressen vom für das eigene Netz zuständigen DHCP-Server (Router, Server) zugewiesen werden.

2. Mittels dem Kontrollzentrum der Grafischen Benutzeroberfläche

   Im Kontrollzentrum von KDE besteht die Möglichkeit unter Internet & Netzwerk/Netzwerkeinstellungen nach Auswahl des entsprechenden Adapters, diesen auf automatische Adressvergabe einzustellen. Das Ergebnis ist dasselbe, wie bei der Ersten Methode. Weiterhin hat man hier die Möglichkeit, die Routing-Tabelle zu bearbeiten.