



Mini-Howto für mehrere NICs & Crossover-Kabeln als Hub-Ersatz

Vorbemerkungen

Teil I: Konfiguration mit ISDN

Netzwerkarten Treiber

Netzwerke definieren

Masquerading

Domain Konfiguration

Teil II: Unterschiede bei T-DSL

Teil III: Anpassen der Clients

Teil IV: Besonderheiten in Windows Netzwerken

Vorbemerkungen:

Dieser Aufbau lohnt sich von vornherein nur für kleine Netze, da Steckplätze und die zu vergebenden Interrupts im Fli4l Router schließlich begrenzt ist. Wer also plant, ein größeres Netz aufzubauen, oder für eine spätere Erweiterung gewappnet sein will, sollte sich überwinden, und das Geld für einen Hub ausgeben, besser noch einen Switch, damit man 10 & 100MBit Clients mischen kann (mittlerweile gar nicht mehr so teuer). Die Konfiguration von Fli4l ist damit wesentlich einfacher (man braucht dies Howto nicht) und man sitzt nachher nicht auf einem Berg von NICs. Schließlich ist zu bedenken, daß die Clients auch noch alle eigene NICs brauchen. Ein weiterer Grund für ein solches Setup kann das Mischen von Twisted Pair mit BNC sein, hier bietet sich Fli4l als günstiger Medienkonverter an.

Nun zu den Sachen, die wir brauchen werden:

1. so viele Netzwerkkarten wie Clients im Netz (+1 bei DSL), vorzugsweise baugleich, da man dann lediglich einen Treiber laden muss.
2. Fli4l Host mit genügend freien Steckplätzen
3. so viele Crossover Kabel wie Clients im Netz

Ich stelle das hier am Beispiel der günstigen Realtek 8139 PCI vor. Diese kommt als Chip in vielen billigen Noname Karten vor und ist mit 25,-/Stück eine erschwingliche 10/100MBit Lösung. Sie wird im professionellen Einsatz zwar gemieden, sollte aber für den Heimgebrauch ausreichend sein. Anmerkungen für Setups mit verschiedenen Karten werden aber auch gemacht. Desweiteren stelle ich das hier für 3 Rechner dar, d.h. den Fli4l Router und 2 Clients.

[nach oben](#)

Teil I: Konfiguration mit ISDN

Ich werde hier nicht die generelle Einrichtung von ISDN darstellen, dafür gibt's das Quick-ISDN-Howto.

Also auf zum Einrichten des Netzwerks:

```
#-----  
# Ethernet card drivers:  
# uncomment your ethernet card  
#-----  
ETH_DRV_N='1'                # number of ethernet drivers to load, usually 1
```

```
ETH_DRV_1='rtl8139'           # PCI: RealTek 8129/8139 (not 8019/8029!) PCI
ETH_DRV_1_OPTION=''          # additional option, e.g. 'io=0x340' for ne
```

Wir haben zwar 2 Netzwerkkarten, aber da alle baugleich sind, brauchen wir nur **einen** Treiber zu laden. Als PCI Karte braucht die Realtek 8139 idR. keine Options, da die Adressen vom Bios zugeteilt werden.

Für den Fall, daß doch verschiedene Karten verwendet werden, sind hier entsprechend fuer alle **verschiedenen** Kartentypen separate Treiber anzugeben. Beispiel:

```
ETH_DRV_N='2'                 # number of ethernet drivers to load, usually 1
ETH_DRV_1='rtl8139'          # PCI: RealTek 8129/8139 (not 8019/8029!) PCI
ETH_DRV_1_OPTION=''         # additional option, e.g. 'io=0x340' for ne
ETH_DRV_2='8390 ne'         # ISA: NE2000 ISA clone
ETH_DRV_2_OPTION='io=0x220' # additional option, e.g. 'io=0x340' for ne
```

In diesem Beispiel Clients benutzen wir eine oder mehrere baugleiche Realtek 8139 und einen ISA NE2000 Clone. Wieder werden lediglich für unterschiedliche Kartentypen Treiber geladen. Mehr zu verschiedenen Karten und Treibern findet sich in der [Dokumentation](#) von Fli4l selbst, Abschnitt 2.3.

Bei verschiedenen Karten werden diese in der angegebenen Reihenfolge eingebunden, bei gleichen Karten hingegen ordnet der Treiber die Karten in ihrer Reihenfolge nach eigenem Gutdünken. Hier kann man lokal vom Router aus mittels pingn der Clients und dem Durchprobieren der Kabel die richtige Zuordnung finden. Einfach beispielsweise das Kabel zu 192.168.1.2 nacheinander an die verschiedenen Netzwerkkarten anschliessen und jeweils versuchen, 192.168.1.2 zu pingn, bis man eine Antwort bekommt. Dann entsprechend das Gleiche für die anderen Clients mit den verbleibenden Karten.

Wenn wir die soweit haben geht's weiter zu den Netzwerken:

```
#-----
# Ether networks used with IP protocol:
#-----
IP_ETH_N='2'                 # number of ip ethernet networks, usually 1
IP_ETH_1_NAME=''            # optional: other device name than ethX
IP_ETH_1_IPADDR='192.168.1.1' # IP Address der Karte im Router im ersten
Netzwerk
IP_ETH_1_NETWORK='192.168.1.0' # erstes Netzwerk
IP_ETH_1_NETMASK='255.255.255.0' # Subnetzmaske, bei allen gleich
IP_ETH_2_NAME=''
IP_ETH_2_IPADDR='192.168.2.1' # IP Adresse der Karte im Router im zweiten
Netzwerk
IP_ETH_2_NETWORK='192.168.2.0' # zweites Netzwerk
IP_ETH_2_NETMASK='255.255.255.0'
# wuerde so weitergehen:
#IP_ETH_3_NAME=''
#IP_ETH_3_IPADDR='192.168.3.1'
#IP_ETH_3_NETWORK='192.168.3.0' # drittes Netzwerk
#IP_ETH_3_NETMASK='255.255.255.0'
#IP_ETH_4_NAME=''
#IP_ETH_4_IPADDR='192.168.4.1'
#IP_ETH_4_NETWORK='192.168.4.0' # viertes Netzwerk
#IP_ETH_4_NETMASK='255.255.255.0'
```

Wenn dies soweit geschehen ist, weiter zum Masquerading:

```
#-----  
# Masquerading:  
#-----  
MASQ_NETWORK='192.168.0.0/16'          # networks to masquerade (e.g. our LAN)  
# die module je nachdem welche man braucht...
```

Auf diese Weise werden die gesamten Netzwerke maskiert und die Firewall Regel bleiben einfach. Wer will, kann hier aber auch die einzelnen Netze wie im nächsten Beispiel getrennt angeben:

```
MASQ_NETWORK='192.168.1.0/24 192.168.2.0/24'
```

Wenn jedoch über ISDN auf ein weiteres 192.168.er Netz zugegriffen wird (wegen zusätzlichem Anschluß an Firma, o.ä.) müssen die Netze zwingend getrennt werden.

Weiter zur Domain Konfiguration:

```
HOSTS_N='3'                            # number of hosts in your domain  
HOST_1='192.168.1.1 fli41'              # fli41 (erste NIC)  
HOST_2='192.168.1.2 client1'           # erster client  
HOST_3='192.168.2.2 client2'           # zweiter client  
# wuerde so weitergehen:  
HOST_4='192.168.3.2 client3'  
HOST_5='192.168.4.2 client4'
```

[nach oben](#)

Teil II: Unterschiede bei T-DSL

Bei der Verwendung von T-DSL fällt die ISDN Karte weg, und hinzu kommt statt dessen eine weitere Netzwerkkarte. Dieser muss jedoch kein Netzwerk und keine IP Adresse zugewiesen werden, etc. weil sie mit dem pppoe Protokoll läuft.

Die Konfiguration sieht dann bei 2 Clients und dem Router wie folgt aus:

Einstellungen für die Netzwerkkartentreiber bleiben gleich, sofern keine anderen Kartentypen und damit andere Treiber erforderlich sind.

Netzwerkangaben, Subnetze etc. sind nicht erforderlich.

Hier kann man höchstens die Variable PPPOE_ETH ändern:

```
#-----  
# Optional package: PPPoE  
#-----  
OPT_PPPOE='yes'                        # use ppp over ethernet: yes or no  
PPPOE_NAME='T-DSL'                      # circuit name, don't write spaces  
PPPOE_ETH='eth2'                        # eth card connected to ADSL modem, e.g. 2nd
```

Die Variable eth bezeichnet die Ethernet interfaces und beginnt mit 0. Ich persönlich habe hier die letzte NIC genommen, weil dies das einfachste ist. Welcher Wert hier genommen wird, ist bei baugleichen Karten eigentlich egal. Sie wird lediglich dann relevant, wenn man zB für den DSL Adapter eine bestimmte NIC verwenden will, so z.B. wenn man generell 100MBit Karten für die Clients verwendet, für die Verbindung zum DSL Adapter aber eine völlig ausreichende 10MBit Karte. Dann muß jedoch oben bei der Angabe der Netzwerke ETH_x_NAME explizit angegeben werden.

[nach oben](#)

Teil III: Anpassen der Clients

Client1:

IP: 192.168.1.2
Subnetz: 255.255.255.0
Hostname: client1
DNS-Server: 192.168.1.1
Gateway: 192.168.1.1

Client2:

IP: 192.168.2.2
Subnetz: 255.255.255.0
Hostname: client2
DNS-Server: 192.168.1.1
Gateway: 192.168.2.1

[nach oben](#)

Teil IV: Besonderheiten bei Windows Netzwerken:

Bei der aktuellen Fli4l Version 2.0.x kann zusätzlich das forwarden der NetBIOS Ports zwischen bestimmten Netzen explizit zugelassen werden. So können zwischen den internen Rechnern Windows Freigaben genutzt werden, gleichzeitig aber die Ressourcen gegenüber dem Internet vor Zugriffen geschützt werden.

```
#-----  
# Routing: ports to reject/deny forwarding (from inside and outside!)  
#-----  
FORWARD_DENY_PORT_N='1' # no. of ports to reject/deny forwarding  
FORWARD_DENY_PORT_1='137:139 REJECT' # deny/reject forwarding of netbios  
FORWARD_TRUSTED_NETS='192.168.1.0/24 192.168.2.0/24' # but allow forwarding between  
LANs
```

Hier sollte auf jeden Fall die aktuelle Version von Fli4l verwendet werden, da die Vorversion 1.6.x die Option FORWARD_TRUSTED_NETS nicht zu Verfügung stellte und daher beim forwarden von NetBIOS die internen Freigaben auch dem Internet zur Verfügung gestellt hätte - ein untragbares Sicherheitsrisiko.

Da bei einem Router ferner die vom Windows Netzwerk/NetBIOS verwendeten Broadcasts nicht zwischen den Netzen weitergeleitet werden, tauchen die fremden Rechner nicht in der Netzwerkkumgebung auf.

Hier kann über 2 Wege Abhilfe geschaffen werden:

1) Auf Host die Datei "LMHOSTS" anlegen. Hierfür bringt jedes Windows auf der Festplatte eine Beispieldatei "lmhosts.sam" mit, die im Windows Ordner oder in dessen Unterverzeichnissen zu finden ist. Sie enthält ein ausführlich kommentierte Beispielkonfiguration, an der man sich orientieren kann. Danach wird sie als "LMHOSTS" abgespeichert. Wenn danach in der Netzwerkkumgebung immer noch nichts zu sehen ist, sollte das Protokoll NetBEUI installiert werden.

oder:

2) Fli4l als DHCP und WINS Server einrichten. Durch den im OPT_DHCP enthaltenen DHCP Server wird die Konfiguration der Clients mit IP, etc. entbehrlich. Der WINS Server (NMBD) ist in OPT_SAMBA enthalten. Genaueres zur Einrichtung

findet sich hierzu in der Dokumentation der entsprechenden OPT Pakete. Aus Sicherheitsgründen kann man jedoch Bedenken gegen die Einrichtung von Samba auf einem Internet Router haben.

Referenz für den Windows Netzwerk-Teil ist [Arno Behrends](#).

[nach oben](#)

So, nun funzt's hoffentlich ;-). Viel Spass.

Falls Fehler auftauchen, bitte Mail an mich, [Karsten Winkelmann](#). Wenn **nach ausgiebiger Lektüre der Dokumentation, des Archivs der Mailingliste und dieses Howtos** noch Fragen offen sind, helfe ich gern.

[Klicken Sie hier, um die Seite auszudrucken.](#)