

Emule und FLI4L



Wie bekomme ich den Emule mit FLI4L zum Laufen?

Diese Frage wird immer wieder gestellt, darum habe ich mir die Mühe gemacht, dieses HowTo zu verfassen. Ich selbst hatte Mühe, die ganzen Stolperfallen dabei zu finden und zu beseitigen.

Emule, Edonkey und Gleichgesinnte sind Sicherheitsrisiken, die man im Grunde genommen eigentlich vermeiden sollte. Wenn man nun unbedingt glaubt einen Emule oder Edonkey haben zu müssen, so muss man aber auch mit dem Umstand leben, dass der Schutz des FLI4L nicht mehr 100%ig gegeben ist.

Ein Vorwort möchte ich noch loswerden: ich gebe hier nur Anweisungen, wie man Emule und ggf. auch Edonkey nutzen kann mit FLI4L. Mir ist bewusst, dass mit Emule nicht zu 100% legale Sachen gemacht werden. Darauf habe ich aber keinen Einfluss.

Angenommen, ich erstelle eine Anleitung zum Bau eines Hammers, würde der Großteil der Leser sicherlich den Hammer nutzen, um völlig harmlos und legal einfach nur Nägel in die Wand zu hauen. Was kann ich aber dafür, wenn manche den Hammer nehmen, um damit auf einen Mitmenschen einzuschlagen?

Der Hammer ist nur ein Werkzeug, genau wie Emule nur ein Tool ist.

Es kommt daher letztendlich immer auf die jeweilige Nutzung und Vernunft des Menschen an. Verteilt keine illegalen Sachen damit, es gibt genügend legale Dinge, die man damit verbreiten kann.

Einrichtung des FLI4L für Emule

Zuerst müssen wir sicherstellen, dass der HOST (meist ein Windows Rechner) korrekt in FLI4L angegeben wurde.

Dafür kontrollieren wir folgende Zeilen in der base.txt.

DIE HOSTS

```
HOSTS_N=' 2 '  
HOST_1='192.168.0.1 fli4l '  
HOST_2='192.168.0.2 windows 11:22:33:44:55:66 '
```

Wir geben die IP Adresse und den Namen unseres Hosts an, gefolgt von seiner MAC-ID. Die MAC-ID ist nur nötig, wenn wir DHCP benutzen. Bei einer festen IP-Vergabe ist sie hier unnötig.

Der Clientname muß zu 100% übereinstimmen. Für eine korrekte Einstellung deines Betriebssystems schau dir das jeweilige HowTo auf www.fli4l.de an.

Vor allem ist es elementar wichtig, daß die DNS Einstellungen korrekt sind!

Ich wiederhole mich womöglich, aber ich sage es immer wieder:

Die Tatsache, daß das Internet funktioniert ist noch lange kein Beweis für eine absolut korrekte Konfiguration des Client und des FLI4L.

Näheres hierzu in der Dokumentation zu FLI4L. Wichtig ist nur, dass dies korrekt eingerichtet wurde.

DAS MASKIERTE NETZWERK

Der nächste Punkt ist das Maskierte Netzwerk

```
MASQ_NETWORK='192.168.0.0/16 '  
oder  
MASQ_NETWORK='192.168.0.0/24 '
```

Wichtig ist hier nur, dass das jeweilige Netzwerk maskiert ist. Auch hierzu steht näheres in der Dokumentation von FLI4L.

Die internen, vorgegebenen MASQ_MODULE sind für Emule selbst nicht erforderlich.

DAS PORTFORWARDING

Nun richten wir das Portforwarding ein:

```
PORTFW_5='4661-4662 192.168.0.2 tcp'  
PORTFW_6='4665 192.168.0.2 udp'
```

Der UDP Port ist in neuen EMule Versionen nun nicht mehr 4665, sondern 4672.
Sicherheitshalber forwardet man eben beide UDP Ports

```
PORTFW_6='4665 192.168.0.2 udp'  
PORTFW_7='4672 192.168.0.2 udp'
```

Frei nach dem Motto: wenn wir schon ein Loch in die Mauer schlagen, können wir auch zwei Löcher vertragen.

Der UDP Port ist in den neuen EMule Versionen frei einstellbar.
So könnte man u.U. sogar völlig andere Ports verwenden.

Nur Port 4662? NEIN!

Noch ein Hinweis für diejenigen die behaupten, dass ein Forwarding des Port 4662 reichen würde. Ich bekomme **definitiv** Anfragen auf UDP 4665 und TCP 4661 bei mir herein. Wenn der Router diese Ports nicht an meinen EMule weiterleitet, kann die Anfrage auch nicht vom EMule beantwortet werden. Folglich muss irgendjemand, entweder du oder der Anfragende, darunter leiden.

2 EMule Clients hinter einem Router? JA!

Von einem User bekam ich den Tipp, dass es durch geschicktes Portforwarding auch möglich sein soll zwei EMule-Client-Rechner mit hohen IPs versorgen zu können. Dazu müssen dann andere Ports an die jeweilige andere IP geforwardet werden.

Beispiel zusätzlich zu den oberen beiden Forwardings:

```
PORTFW_7='4671-4672 192.168.0.3 tcp'  
PORTFW_8='4675 192.168.0.3 udp'
```

Natürlich muss der EMule auf dem zweiten Client mit der IP 192.168.0.3 in der Konfiguration eine andere Portnummer bekommen. In dem Beispiel hätte 4671 funktioniert.

Laut Aussagen von fleissigen Testern soll es mit 2 oder sogar 3 EMulen gleichzeitig funktionieren. Da ich persönlich keinen Sinn darin sehe, zwei EMule am selben Anschluss zu betreiben, habe ich auf einen Test und die dafür nötigen Mühen verzichtet.

Einzelne Portforwardings? Warum nicht!

Ein Leser berichtete mir, dass er nur mit Einzel-Portforwarding Erfolg hatte. Statt also den Bereich von Ports zu forwarden, tat er es mit einzelnen Forwardings:

```
OPT_PORTFW='yes'  
PORTFW_N='4'  
PORTFW_1='4661 192.168.6.2:4661 tcp'  
PORTFW_2='4662 192.168.6.2:4662 tcp'  
PORTFW_3='4665 192.168.6.2:4665 udp'  
PORTFW_4='4672 192.168.6.2:4672 udp'
```

Wichtig ist laut Meinung des Lesers auch die Portangabe hinter der IP Adresse, obwohl dies laut Dokumentation eigentlich nicht nötig sein sollte. Aber wenn es denn so funktioniert, warum nicht!

FORWARD_DENY_PORT

Bei FORWARD_DENY_PORT dürfen natürlich die oben genannten Ports nicht blockiert werden.

FIREWALL_DENY_PORT

Auch bei FIREWALL_DENY_PORT dürfen die oben genannten Ports natürlich nicht blockiert werden.

Die Variable FIREWALL_DENY_ICMP='yes' darf ohne Probleme gesetzt sein, das stört den Emule meines Wissens nach nicht.

DNS_FORBIDDEN

Hier sollten keine Seiten oder Server liegen, die mit Edonkey oder Emule zu tun haben. Am Besten, man geht alle Einträge mal gezielt durch.

Ich kann nicht sagen was passieren würde, wenn ein EMule Server durch das DNS_FORBIDDEN gesperrt würde. Ob EMule den DNS-Namen benötigt oder die IP ausreicht ist mir nicht bekannt. Ob überhaupt eine Sperrung eines EMule Servers über die DNS_FORBIDDEN möglich wäre, kann ich deshalb noch nicht sagen. Wer möchte, kann sich darüber ja mal ein paar Gedanken machen.

Demgegenüber steht die Tatsache, dass es hunderte oder gar tausende EMule Server gibt. Da ist es nicht mehr so schlimm, wenn einer davon blockiert wurde.

So, das waren auch schon die Einstellungen von FLI4L. Nun kommen noch die Einstellungen von Emule dran.

Die Einstellungen in Emule

Nach dem Download des neuesten Emule (der wird zeitweise sehr oft durch eine höhere Version erneuert) installieren wir ihn zuerst und editieren danach die vorher leere Datei ADRESSES.DAT im Emule Verzeichnis wie folgt, damit diese Zeile drin steht:

<http://www.mrhussein.net/edonkey/serverliste/maurice/max/server.met>

oder die:

<http://ocbMaurice.dyns.net/pl/slist.pl?download/server-good.met>

oder die:

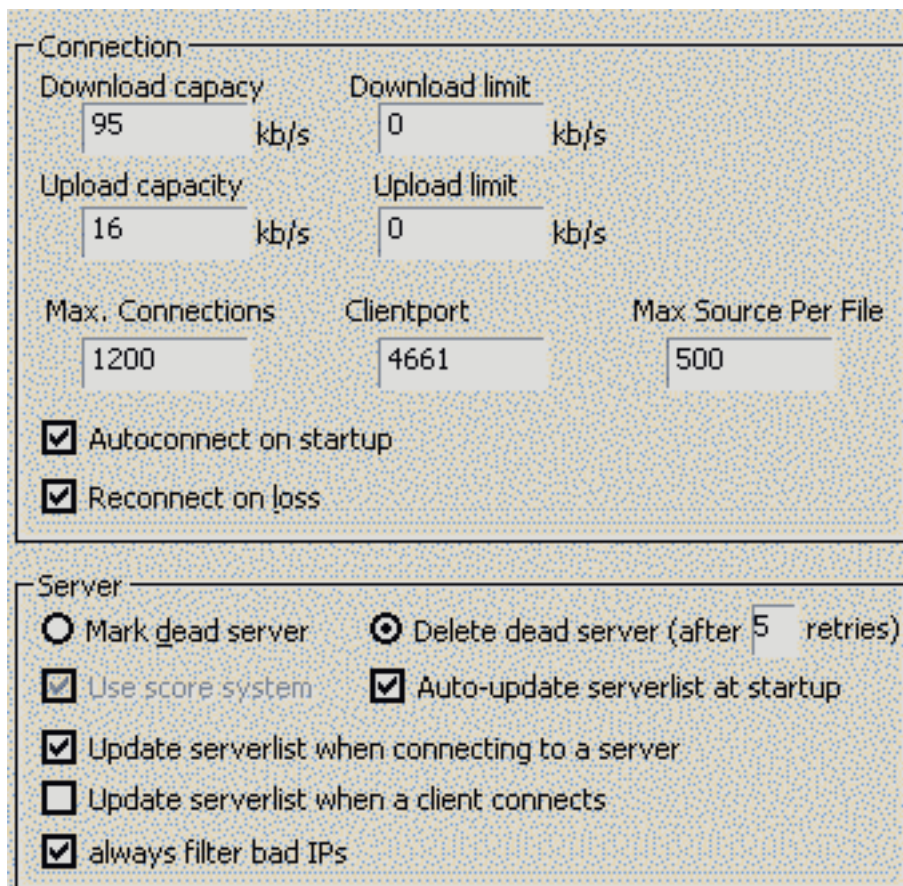
<http://ed2k.2x4u.de/dnp1fa/server.met>

Nun findet der Emule hoffentlich auch schön viele Server.

Ich persönlich halte nicht viel von gigantisch großen Server-Dateien. Ein paar Einträge darin reichen aus, da beim Connect sowieso eine ganze Latte neuer Server mit übertragen wird.

Sollten alle diese oben genannte Server nicht mehr erreichbar sein, tut es im Prinzip jeder andere Server auch, der eine server.met Datei führt. Google wird bei der Suche gerne helfen.

Die Preferences stellen wir ein wie folgt:



Das Bild ist in der Farbtiefe reduziert, damit es nicht so viel Speicherplatz benötigt. Man kann auch mit nur 16 Farben alles erkennen, was nötig ist.

Die 4661 beim Clientport gilt nur für den ersten Rechner. Wenn es gewünscht wird, dass EMule auch auf dem zweiten Rechner mit einer hohen ID laufen kann, dann sollte hier beim zweiten PC der Port 4671 eingetragen werden, analog zum Portforwarding weiter oben. Andere berichten davon, daß der Clientport der höhere der beiden TCP Ports sei. Wenn dein EMule mit dem 4661 nicht so richtig in die Gänge kommt, stelle als Clientport mal testweise 4662 ein, während das Portforwarding am FLI41 gleich bleibt.

Der hohe Wert in der Spalte Max Connections wird u.U. von deinem Betriebssystem nicht unterstützt. Bei einer neueren Version des EMule soll dann automatisch eine Warnung kommen.

Dadurch, dass ich bei Download limit Upload- und Downloadrate auf 0 setze, passt sie sich automatisch an.

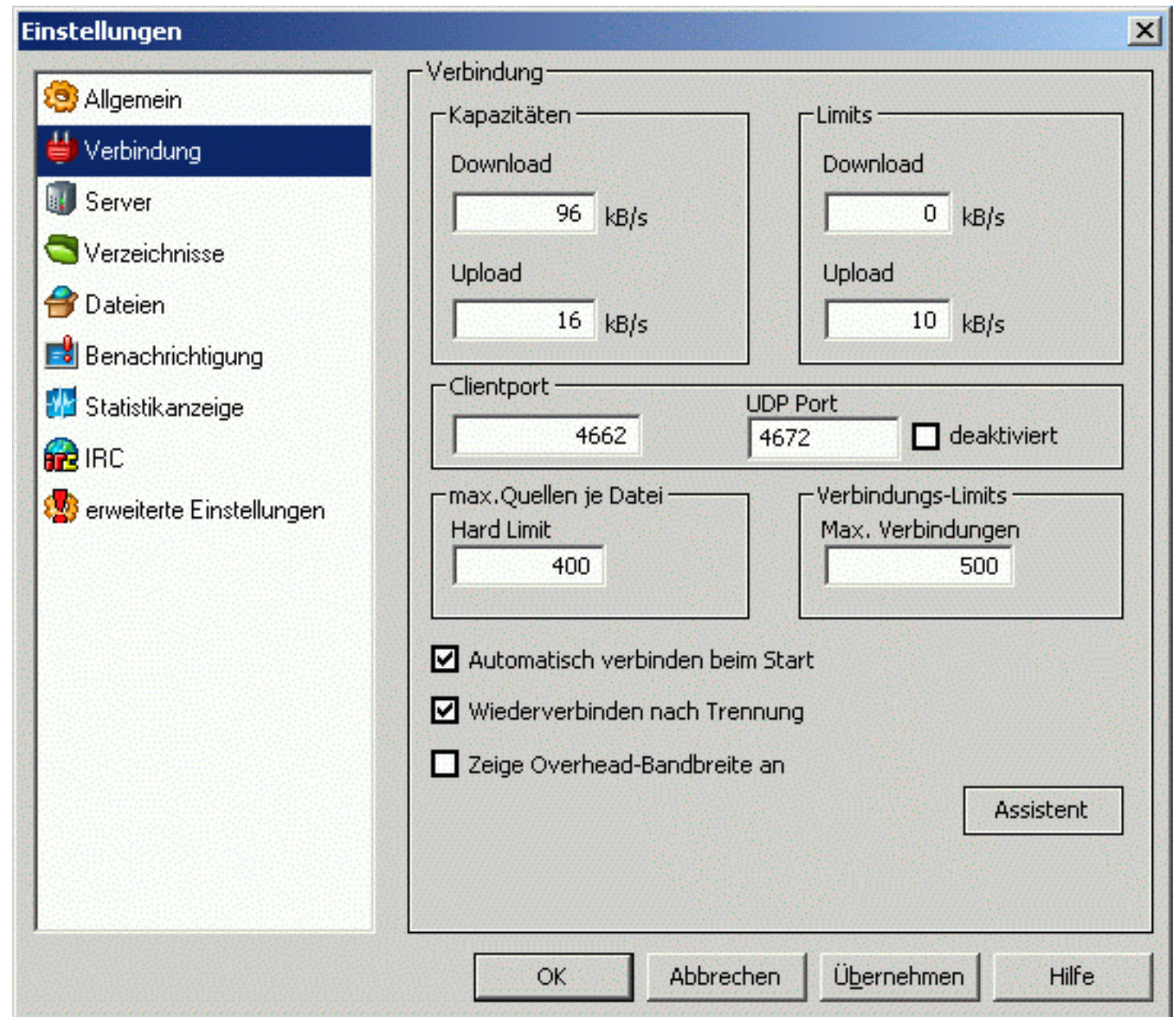
Mit diesen Einstellungen kann man kurzzeitig durchaus Downloads bis zum Anschlag bekommen, sofern die Community zur Zeit gerade so viel Leistung hergibt.

Obwohl ich von einigen Leuten darauf hingewiesen wurde, bleibe ich bei meiner Meinung: in die Spalte Upload Limit darf die 0 hineingeschrieben werden.

Bei verschiedenen Tests auf Systemen kam der Download zeitweilig bis zum Anschlag.

Neuere EMule scheinen da aber nicht sehr kulant darauf zu reagieren. Deshalb gilt die 0 bei den neuen EMules wohl nicht mehr. Wie mir gesagt wurde sollte der Upload bei neueren EMules auf etwa 10-14 kB beschränkt werden.

Bei neuen EMule sehen die Einstellungen in etwa aus wie folgt:



Reale Werte beim Download sind etwa 20 bis 30 kB/sec, wenn der Imonc dies anzeigt. Den angezeigten Werten von EMule kann man nicht trauen; ich verlasse mich ausschließlich auf die Angaben des Imonc.

Der erreichte Speed hängt natürlich maßgeblich von den anderen Quellen ab. Man kann nicht schneller empfangen, als andere senden können!

Die Werte links davon, Download capacity und Upload capacity sagen aus, was über diese Leitung möglich sein müsste. Dieser Wert ist total unerheblich für den Download. Er dient lediglich dazu, einen Anhaltspunkt für die Tabelle und Grafiken bei der Statistik zu bieten. Hier kann man einstellen, was immer einen beliebt. Sinnvoll ist natürlich der echte Wert der Leitung.

Andererseits kann es auch Sinn machen, die Leistung des EMule zu beschneiden, damit man noch Ressourcen frei hat für andere Tätigkeiten. Wenn man den EMule bei Download limit und Upload

limit beschneidet, dann kann man natürlich auch die Werte bei Download capacity und Upload capacity verringern, damit man sieht, wie weit man an sein selbst gesetztes Limit ran kommt.

Nun sollte nach einem Neustart vom FLI4L und des Windows Client einem Saugvergnügen mit Emule nichts mehr im Wege stehen.

Die Desktop Firewall

Viele Mitmenschen haben zusätzlich zum FLI4L noch eine Desktop Firewall laufen. Ich persönlich bin bei Anfängern eigentlich dagegen, eine Firewall einzusetzen. Der fli4l reicht bei richtiger Konfiguration völlig aus.






Von Pseudo-Firewalls (ich nenne sie ZANF, Insider wissen auf was ich anspiele...) rate ich ab und empfehle vernünftige Firewall Lösungen, wie beispielsweise die Tiny Firewall (als deren Freeware-Variante: "Kerio") oder Outpost.

Die Tiny Firewall gibt es an einigen Orten im Internet als laufzeitbeschränkte Testversion. Ältere Versionen der Tiny Firewall erfüllen ebenso ihren Zweck; es muss nicht die neueste sein!

Ambitionierte, fortgeschrittene User können sich hingegen mit einer ordentlich konfigurierten Firewall recht schnell einen Überblick verschaffen, was gerade auf der Maschine rein und raus geht. Ich nutze gerne die Kerio Firewall als Kontrollmodus. Es ist sehr interessant, welche Tools so alles raus ins Internet wollen.

Ich möchte nochmals deutlich betonen: eine Software-Firewall auf dem Windows Rechner ist nichts, aber auch wirklich GAR NICHTS für Anfänger! Damit kann man sich sehr schnell mehr Probleme schaffen, als sie überhaupt Nutzen bringen kann.

Was machen wir mit der Firewall? Leider reicht es für die Desktop Firewall nicht aus, nur die oben genannten Ports und Protokolle freizuschalten. Da müssen noch mehr Sachen freigeschaltet werden. Am Beispiel von Tiny Firewall habe ich es hier mal verdeutlicht:

1	<input checked="" type="checkbox"/>		←	eMule	TCP (In)	[Any port]	[Any address]:[4661]	C:\PROGRAMME\EMUL
2	<input checked="" type="checkbox"/>		←	eMule	TCP (In)	[4661]	[Any address]:[Any port]	C:\PROGRAMME\EMUL
3	<input checked="" type="checkbox"/>		→	eMule	TCP (Out)	[Any port]	[Any address]:[4661,4662]	C:\PROGRAMME\EMUL
4	<input checked="" type="checkbox"/>		↔	eMule	UDP (Both)	[Any port]	[Any address]:[4665]	C:\PROGRAMME\EMUL
5	<input checked="" type="checkbox"/>		→	eMule	TCP (Out)	[Any port]	[Any address]:[Any port]	C:\PROGRAMME\EMUL

Die Regel Nummer 3 ist eigentlich enthalten in Regel Nummer 5. Wem die Regel Nummer 5 zu heikel ist, weil sie seiner Meinung nach zu viel erlaubt, der sollte sie deaktivieren, oder weglassen.


Dann ist ja immernoch Regel Nummer 3 da.

Allerdings reicht die Regel 3 bei weitem nicht aus, das weite Spektrum an Ports abzudecken, die von Emule benutzt werden. Ich habe testweise mal die Firewall im Kontrollmodus mitlaufen lassen, um festzustellen welche Ports hier alles angesprochen werden. Nach nur etwa 10 Minuten hatte ich schon eine ganze Reihe von lokalen Ports gefunden.

Als da gewesen sind: 80, 4551, 4565, 4580, 4589, 4662, 4663, 4668, 4670, 4797, 4838, 4865, 4868, 4946, 6662, 7022

Das Schlimme dabei ist, sie werden wahllos ohne jede Regel ausgesucht. So haben wir mit der Desktop Firewall keine andere Wahl, als Regel 5 zu nehmen, wenn wir den vollen Umfang von Emule nutzen wollen.

Für all diejenigen, denen diese 5 Einzel-Regeln zu mühselig sind, können es sich auch einfacher machen:

 eMule Any (Both) [Any port] [Any address]:[Any port] C:\PROGRAMME\EMULE

EMule mit QoS in seine Schranken verweisen

Wenn EMule richtig schön läuft, dann belegt er einen DSL Anschluss so ziemlich zu 90%. Dass dies unangenehm ist wird jeder merken der normal surfen will, während EMule kräftig aktiv ist.

Die Lösung dagegen bietet uns der Dienst QoS, welchen fli4l selbstverständlich auch bietet und unterstützt. Dafür besorgen wir uns das OPT_qos und installieren es, wie sonst auch.

In meinem Beispiel bin ich von folgender Situation ausgegangen:

- 1 Surfstation mit der höchsten Priorität: 1
Beispielsweise für Ehefrau, Vater, Mitbewohner, usw.
Der Vorteil besteht darin, dass dieser Nutzer die höchste Priorität bekommt. Das heißt, wenn er surft, wird EMule beschneidenderweise gedrosselt. Erst wenn dieser Client nichts mehr zu Senden hat, geht EMule wieder auf Sendung.
- 1 Surfstation mit der zweithöchsten Priorität: 2
Dieser Client wird mein eigener Surf-Rechner. Um den Frieden im Haushalt zu wahren, gebe

ich mich mit der Bandbreite zufrieden, die der PC meiner Frau übrig lässt. In 99,9% aller Fälle ist das mehr als ausreichend.

- 1 EMule-Rechner mit der niedrigsten Priorität: 3

Dieser Rechner bekommt erst dann Bandbreite zugeteilt, wenn die beiden Surf-PCs welche frei haben. Als stromsparender Rechner ohne CD-ROM, Floppy und Lüfter kann er mal ein, zwei Tage durchrennen.

Der Vorteil ist, dass Downloads auf den Surf-Clients rasend schnell gehen, selbst Upload-FTP-Dateitransfers auf FTP-Server rasen mit voller Bandbreite. Sehr schön, wenn man auf seine Webseite Daten hochladen will. Bisher war es immer so, dass man den EMule manuell beenden musste, oder in den Einstellungen ihm zumindest den Upload Speed verringern musste.

Das geht nun alles vollautomatisch durch unseren fli4l. Der entscheidet anhand der QoS-Konfiguration, wer zur Zeit welche Bandbreite beanspruchen darf.

Verwendete Versionen: fli4l 2.0.7, QoS 0.4

Bitte einfach die Zeilen per Copy&Paste in die QoS.txt einfügen.

Die IP Adressen müssen selbstverständlich entsprechend angepasst werden.

In meinem Beispiel wären:

IP 192.168.0.1 die Surfstation mit der besten Priorität

IP 192.168.0.2 der zweite Surfrechner mit der zweitbesten Priorität, auch für FTP- oder ICQ-Dateiversand zu gebrauchen.

IP 192.168.0.3 der EMule Rechner mit der schlechtesten Priorität

```
OPT_QOS='yes'  
QOS_INTERNET_DEV='ppp0'  
QOS_INTERNET_BAND_DOWN='768kbit'  
QOS_INTERNET_BAND_UP='128kbit'  
QOS_INTERNET_DEFAULT_DOWN='0'  
QOS_INTERNET_DEFAULT_UP='2'  
QOS_LOCALNET_DEV='eth0'  
QOS_LOCALNET_BAND='100Mbit'  
QOS_CLASS_N='3'  
QOS_CLASS_1_PARENT='0'  
QOS_CLASS_1_MINBANDWIDTH='127kbit'  
QOS_CLASS_1_MAXBANDWIDTH='128kbit'  
QOS_CLASS_1_DIRECTION='up'  
QOS_CLASS_1_PRIO='1'  
QOS_CLASS_2_PARENT='0'  
QOS_CLASS_2_MINBANDWIDTH='1kbit'  
QOS_CLASS_2_MAXBANDWIDTH='128kbit'  
QOS_CLASS_2_DIRECTION='up'  
QOS_CLASS_2_PRIO='2'  
QOS_CLASS_3_PARENT='0'
```

```
QOS_CLASS_3_MINBANDWIDTH='1kbit'  
QOS_CLASS_3_MAXBANDWIDTH='90kbit'  
QOS_CLASS_3_DIRECTION='up'  
QOS_CLASS_3_PRIO='3'  
QOS_FILTER_N='3'  
QOS_FILTER_1_CLASS='1'  
QOS_FILTER_1_IP='192.168.0.1' # Hier die IP mit der besten Priorität  
QOS_FILTER_1_PORT=''  
QOS_FILTER_1_TYPE=''  
QOS_FILTER_1_OPTION=''  
QOS_FILTER_2_CLASS='2'  
QOS_FILTER_2_IP='192.168.0.2' # Hier die IP mit der zweitbesten  
Priorität  
QOS_FILTER_2_PORT=''  
QOS_FILTER_2_TYPE=''  
QOS_FILTER_2_OPTION=''  
QOS_FILTER_3_CLASS='3'  
QOS_FILTER_3_IP='192.168.0.3' # Hier die IP mit der schlechtesten  
Priorität: emule  
QOS_FILTER_3_PORT=''  
QOS_FILTER_3_TYPE=''  
QOS_FILTER_3_OPTION=''
```

Den Emule Upload Speed habe ich in Emule auf 10 kb/sec eingestellt. Da uns QoS nun 90 kbit zur Verfügung stellt, bleibt noch etwas Luft bis zur QoS Grenze für Bestätigungen und den Datei-Overhead.

Fehlersuche

Falls der EMule wider Erwarten keine hohe ID bekommt, ist der trivialste Tipp der schnellste: den Client und Router neu booten.

Wenn das nichts hilft, dann gehen wir mal systematisch vor und prüfen unser kleines Netzwerk.

Die Seite http://www.thedonkeynetwork.com/connection_test bietet einen Test an, um die Konfiguration online zu testen. Wenn sich hier heraus stellt, dass du keine HighID bekommen kannst und EMule dir auch keine HighID bietet, dann gehe vor wie folgt.

Öffne auf dem Client die Eingabeaufforderung (DOS-Box) und gib ein:

```
ping localhost
```

Als Ergebnis müssten folgende Zeilen erscheinen:

```
Ping windows.domain [127.0.0.1] mit 32 Bytes Daten:  
Antwort von 127.0.0.1: Bytes=32 Zeit<10ms TTL=128  
Antwort von 127.0.0.1: Bytes=32 Zeit<10ms TTL=128  
Antwort von 127.0.0.1: Bytes=32 Zeit<10ms TTL=128  
Antwort von 127.0.0.1: Bytes=32 Zeit<10ms TTL=128
```

Ping-Statistik für 127.0.0.1:

```
   Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),  
Ca. Zeitangaben in Millisek.:  
   Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms
```

Wichtig hierbei ist, dass keine Fehler angezeigt werden dürfen.

Die Zahl nach dem Wert "Zeit" beträgt bei Windows 2000 den Wert "<10ms", während bei Windows XP hier "0ms" stehen kann.

Danach pingen wir die IP Adresse unseres Clients an, wie sie in der base.txt bei dem Punkt

HOST_2='192.168.0.2 windows' definiert wurde:

```
ping -a 192.168.0.2
```

Als Ergebnis sollte das hier kommen:

```
Ping windows.domain [192.168.0.2] mit 32 Bytes Daten:  
Antwort von 192.168.0.2: Bytes=32 Zeit<10ms TTL=128  
Antwort von 192.168.0.2: Bytes=32 Zeit<10ms TTL=128  
Antwort von 192.168.0.2: Bytes=32 Zeit<10ms TTL=128  
Antwort von 192.168.0.2: Bytes=32 Zeit<10ms TTL=128
```

Ping-Statistik für 192.168.0.2:

```
   Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),  
Ca. Zeitangaben in Millisek.:  
   Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms
```

Nun pingen wir die IP von unserem Router, so wie sie in der base.txt bei dem Punkt

HOST_1='192.168.0.1 fli41' definiert wurde:

```
ping -a 192.168.0.1
```

Als Ergebnis sollte das hier kommen:

```
Ping fli41.domain [192.168.0.1] mit 32 Bytes Daten:  
Antwort von 192.168.0.1: Bytes=32 Zeit<10ms TTL=128  
Antwort von 192.168.0.1: Bytes=32 Zeit<10ms TTL=128  
Antwort von 192.168.0.1: Bytes=32 Zeit<10ms TTL=128
```

Antwort von 192.168.0.1: Bytes=32 Zeit<10ms TTL=128

Ping-Statistik für 192.168.0.1:

Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),
Ca. Zeitangaben in Millisek.:
Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms

Nun pingen wir die Rechnernamen, so wie sie in der base.txt bei dem Punkt
HOST_1='192.168.0.1 fli41' definiert wurde:

```
ping fli41
```

Als Ergebnis sollte das hier kommen:

```
Ping fli41.domain [192.168.0.1] mit 32 Bytes Daten:  
Antwort von 192.168.0.1: Bytes=32 Zeit<10ms TTL=128  
Antwort von 192.168.0.1: Bytes=32 Zeit<10ms TTL=128  
Antwort von 192.168.0.1: Bytes=32 Zeit<10ms TTL=128  
Antwort von 192.168.0.1: Bytes=32 Zeit<10ms TTL=128
```

Ping-Statistik für 192.168.0.1:

Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),
Ca. Zeitangaben in Millisek.:
Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms

Nun pingen wir die Rechnernamen, so wie sie in der base.txt bei dem Punkt
HOST_2='192.168.0.2 windows' definiert wurde:

```
ping windows
```

Als Ergebnis sollte das hier kommen:

```
Ping windows.domain [192.168.0.2] mit 32 Bytes Daten:  
Antwort von 192.168.0.2: Bytes=32 Zeit<10ms TTL=128  
Antwort von 192.168.0.2: Bytes=32 Zeit<10ms TTL=128  
Antwort von 192.168.0.2: Bytes=32 Zeit<10ms TTL=128  
Antwort von 192.168.0.2: Bytes=32 Zeit<10ms TTL=128
```

Ping-Statistik für 192.168.0.2:

Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),
Ca. Zeitangaben in Millisek.:
Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms

Nun pingen wir eine beliebige Internetadresse an:

```
ping www.irgendwas-irgendwo.de
```

Als Ergebnis sollte dann so etwas ähnliches wie das hier kommen:

```
Ping irgendwohin.de [vvv.xxx.yyy.zzz] mit 32 Bytes Daten:  
Antwort von vvv.xxx.yyy.zzz: Bytes=32 Zeit=131ms TTL=128  
Antwort von vvv.xxx.yyy.zzz: Bytes=32 Zeit=330ms TTL=128  
Antwort von vvv.xxx.yyy.zzz: Bytes=32 Zeit=321ms TTL=128  
Antwort von vvv.xxx.yyy.zzz: Bytes=32 Zeit=300ms TTL=128
```

Ping-Statistik für vvv.xxx.yyy.zzz:

```
   Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),  
Ca. Zeitangaben in Millisek.:  
   Minimum = 131ms, Maximum = 330ms, Mittelwert = 270ms
```

Nun pingen wir noch schnell die Internetadresse von eben als IP an:

```
ping -a vvv.xxx.yyy.zzz
```

Als Ergebnis sollte dann so etwas ähnliches wie das hier kommen:

```
Ping irgendwohin.de [vvv.xxx.yyy.zzz] mit 32 Bytes Daten:  
Antwort von vvv.xxx.yyy.zzz: Bytes=32 Zeit=131ms TTL=128  
Antwort von vvv.xxx.yyy.zzz: Bytes=32 Zeit=330ms TTL=128  
Antwort von vvv.xxx.yyy.zzz: Bytes=32 Zeit=321ms TTL=128  
Antwort von vvv.xxx.yyy.zzz: Bytes=32 Zeit=300ms TTL=128
```

Ping-Statistik für vvv.xxx.yyy.zzz:

```
   Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),  
Ca. Zeitangaben in Millisek.:  
   Minimum = 131ms, Maximum = 330ms, Mittelwert = 270ms
```

Wenn alle diese Tests erfolgreich waren, dann scheint deine Namensauflösung am Client und Router korrekt zu arbeiten. Wenn EMule nun immer noch nicht richtig arbeiten will, dann muss das Problem woanders liegen.

Eigene Tests und meine bisherigen Support-Erfahrungen lassen mich aber die Aussage treffen, dass LowIDs schätzungsweise zu 90% von fehlerkonfigurierter Namensauflösung herkommen. Wenn dein EMule nur LowIDs liefert, teste sie mal wie eben gezeigt.

Danach teste mal genau die Einträge PORTFW_5='4661-4662 192.168.0.2 tcp' und PORTFW_6='4665 192.168.0.2 udp' auf ihre Richtigkeit. Denn dies machte aus meinen Erfahrungen weitere 8% aller Fehlerquellen aus.

Ein weiteres Prozent der Fehlerquellen setzt sich zusammen aus falsch aktivierter Firewall, defekten Netzwerkkarten, zu geringer FLI41-Prozessor-Power und vollgelaufener RAM-Disk des FLI41.

Der RAM an sich ist auch eine mögliche Fehlerquelle. Aus gegebenem Anlass will ich darauf hinweisen, dass der Router mit defektem RAM überlastet sein kann. Mein Duron 600 konnte selbst mit 128 MB SD-RAM PC100 keine ordentlichen Geschwindigkeiten für EMule mehr bereitstellen. Kurios fand ich, dass es früher ging.

Meine Recherchen und Tests blieben soweit ergebnislos, als dass der EMule zwar immer schön eine HighID bekam, die Transferrate aber mit 10 kb/sec recht gering ausfiel. Ein zufälliger Austausch des RAMs aus anderen Gründen brachte dann die unverhoffte Besserung.

Deshalb möchte ich auf den (meiner Meinung nach) sehr guten RAMTest **MemTest 86** hinweisen.

Das allerletzte Prozent aller möglichen Fehlerquellen ist dann aber bestimmt das Problem, das DU gerade hast. Denn wenn man alle bekannten Fehlerquellen ausschliessen kann, muss es eine unbekannte Fehlerquelle geben. In diesem Fall - und NUR in diesem Fall - kannst du gerne mit mir Kontakt aufnehmen, sodass wir den Fehler gemeinsam suchen können.

Ich werde KEINESFALLS mehr Router konfigurieren, auch wenn man mich noch so herzlich darum anfleht. Bitte habt Verständnis dafür, dass ich auch anderen Usern in meiner beschränkten Zeit Support geben möchte. Je länger ich mich mit DIR aufhalten muss, desto weniger Zeit bleibt übrig für die ANDEREN. Seid also bitte so fair und versucht selbst, das Problem einzugrenzen.

Dieses How-To ist momentan noch in der Erprobung. Ich konnte es bisher erst auf sehr wenigen Rechnern testen. Für Verbesserungen, Optimierungen, Fehlermeldungen, Kommentare o.ä. bin ich zu erreichen unter Email: hoescheler@t-online.de

Mein Dank geht an die Leser, die mir bei dem HowTo mit Tipps halfen, um es zu verbessern. Zusätzlich zu den ungenannt bleibenden richtet sich mein Dank an:

- Bomat[Ion
- Johannes Binder
- Kuno Great
- Markus

Ferner geht mein Dank auch an die vielen Hilfen der FLI4L Fangemeinde und des FLI4L Teams, ohne die ich niemals einen ausgereiften Softwarerouter in vollem Funktionsumfang

hätte aufsetzen können.

'bye

Gebhard Höscheler, April 2003

Klicken Sie hier, um die Seite auszudrucken.