

# Linux im Netzwerk - ein "Mini"-Tutorial (Teil 3)

[Jana Jaeger](#)

Eine Art Postscriptum zum zweiten Teil des Netzwerktutorials nachgereicht: Was sind die Grundwerkzeuge zur Netzwerkanalyse? Welche Aussagen lassen sich mit ihrer Hilfe über den Zustand des Netzes machen? Wie sind sie sinnvoll zu kombinieren?

Teil 3 des Netzwerktutorials will erste Hilfen für den Umgang mit Netzwerkanalysertools geben, so daß die in [&content/server/fire2.html">Teil 2](#) gemachten Konfigurationsschritte überprüft werden können. Der Testablauf geht von einem voll eingerichteten Netzwerk (samt DNS auf `nick`) aus. Wie genau DNS in unserem Beispiel eingerichtet werden könnte, wird Thema des vierten Teils sein.

## Wichtige Tools zur Netzwerkanalyse

Ist das Netzwerk dann fertig zusammengestellt und konfiguriert, sollten die grundlegenden Funktionen getestet werden. Der Testablauf grob skizziert:

- I. [ping](#) - "lebt" das Netz?
- II. [nslookup](#) - funktioniert der Nameservice?
- III. [ifconfig](#) - Ist die Netzwerkkarte korrekt konfiguriert?
- IV. [traceroute](#) - Welchen Weg nehmen meine Daten?
- V. [netstat](#) - Funktioniert die Netzwerkanbindung?

### *ping*

*ping* dient dazu, zu testen, ob der angewählte Host überhaupt über das Netzwerk erreichbar ist. Testpakete werden an den Host und wieder zurück geschickt und über ihre Reise durchs Netz wird Protokoll geführt. Wurde die Konfiguration aller Bestandteile ordnungsgemäß abgeschlossen, sollten die einzelnen Rechner alle ein erfolgreiches *ping* zurückliefern. Eine typische *ping* Ausgabe sieht so aus:

```
tux@sean:~ > ping nick.home.nil
PING nick.home.nil (192.168.17.1): 56 data bytes
64 bytes from 192.168.17.1: icmp_seq=0 ttl=252 time=6.724 ms
64 bytes from 192.168.17.1: icmp_seq=1 ttl=252 time=6.370 ms
64 bytes from 192.168.17.1: icmp_seq=2 ttl=252 time=13.519 ms
64 bytes from 192.168.17.1: icmp_seq=3 ttl=252 time=9.733 ms
64 bytes from 192.168.17.1: icmp_seq=4 ttl=252 time=198.315 ms
64 bytes from 192.168.17.1: icmp_seq=5 ttl=252 time=128.995 ms
64 bytes from 192.168.17.1: icmp_seq=6 ttl=252 time=6.350 ms
64 bytes from 192.168.17.1: icmp_seq=7 ttl=252 time=6.339 ms
64 bytes from 192.168.17.1: icmp_seq=8 ttl=252 time=6.346 ms
--- nick.home.nil ping statistics ---
9 packets transmitted, 9 packets received, 0% packet loss
round-trip min/avg/max = 6.339/42.521/198.315 ms
```

Die Informationen, die der konkrete *ping* jetzt liefert (auf den Gateway unseres Beispielnetzes) im Einzelnen:

- `nick.home.nil` nimmt die Testpakete in der richtigen Reihenfolge an, wie an der `icmp_sec=` Angabe nachzuvollziehen ist.
- die Zeitangaben an den einzelnen Paketen (`time=`) besagt, daß eine Rundreise zu `nick.home.nil` vom lokalen Rechner und zurück mindestens 6.339, durchschnittlich 42.521 und maximal 198.315 ms dauert.
- Die Statistik über die übermittelten bzw. verlorengegangenen Pakete besagt, daß kein Paket auf dem Weg "fehlgegangen" ist.

Paketverluste oder übermäßig lange Antwortzeiten deuten meistens auf Hardwareprobleme hin, das heißt man ist als Administrator in die Pflicht genommen, die einzelnen Kabelverbindungen, etc. peinlich genau zu überprüfen.

Bekommt *ping* den expliziten Rechnernamen mit und liefert dann als Rückmeldung `unknown host`, liegt dies daran, daß entweder der Name Service dieses Netzes noch nicht aktiviert oder nicht korrekt konfiguriert ist, der Name falsch ist oder ein Hardwareproblem vorliegt. Als nächste Überprüfung kann *ping* mit dem Argument `-n <IP-Adresse>` gestartet werden. Wird der Rechner dann immer noch nicht gefunden, liegt ein Hardwareproblem vor oder die Konfiguration der Netzwerkkarte ist fehlerhaft. Ist der Rechner erreichbar, liegt das Problem beim Nameserver. Mit dem Befehl *nslookup* werden die Funktionen des Nameservers getestet. Mit seinen Funktionen wird sich der nächste Abschnitt befassen.

### ***nslookup***

Wird trotz mehrerer Versuche, den gewünschten Host über das Netz (per *ping*) zu erreichen, immer nur die Fehlermeldung zurückgegeben, daß er nicht im Netz bekannt ist (`unknown host`), sollten dringend die Funktionen des netzinternen Nameservers überprüft werden.

Im Beispiel wird jetzt versucht, von `sean` Auskünfte über `nicks` Namen und IP zu bekommen. Hierzu kann man den vollen Namen eines Rechners oder nur seine IP-Adresse angeben. Die Ausgabe bei erfolgreichem *nslookup* gliedert sich in zwei Abschnitte. Im oberen Abschnitt ist der Rechner angegeben, auf dem der Nameserver betrieben wird (Name und IP-Adresse), hier in diesem Fall eben `nick`, der Gateway. Der untere Abschnitt gibt Name und Adresse des gesuchten Hosts an, hier ebenfalls `nick`.

```
tux@sean:~ > nslookup 192.168.17.1
Server:  nick
Address:  192.168.17.1

Name:    nick.home.nil
Address:  192.168.17.1
```

oder

```
tux@sean:~ > nslookup nick
Server:  nick
Address:  192.168.17.1

Name:    nick.home.nil
Address:  192.168.17.1
```

Kommt diese Meldung zurück, kann man getrost davon ausgehen, daß die interne Namensauflösung reibungslos funktioniert. Sollte ein *nslookup* nicht erfolgreich gewesen sein, empfiehlt es sich, die betroffenen Konfigurationsdateien noch einmal mit "Röntgenblick" darauf durchzugehen, ob die korrekte IP-Adresse dem richtigen (und real existierenden) Namen zugeordnet wird.

### ***ifconfig***

*ifconfig* dient dazu, die Konfiguration der Netzwerkkarte zu überprüfen. Lieferte zum Beispiel der vorangegangene *ping* ein Ergebnis der Art, daß der angefragte Host nicht zu erreichen war, sollte die Konfiguration der Netzwerkkarten sowohl beim anfragenden Rechner als auch beim angefragten Host überprüft werden. Häufig werden bei der Konfiguration kleine Fehler bei der Angabe der einzelnen IP-Adressen oder Subnetzmasken gemacht. Mit *ifconfig* erhält man die genauen Konfigurationsangaben der einzelnen Netzwerkgeräte angegeben.

Ein Beispiel: Die Konfiguration von seans Netzwerkkarte soll überprüft werden.

Der Benutzer tux muß zuerst einmal root werden, bevor er *ifconfig* starten kann.

```
sean:~ # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:80:C8:95:A5:1B
          inet addr:192.168.17.4  Bcast:192.168.17.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:9636349  errors:42  dropped:2  overruns:0  frame:42
          TX packets:12226360  errors:4  dropped:0  overruns:4  carrier:0
          collisions:0  txqueuelen:100
          Interrupt:15  Base address:0xe400
```

Diese Ausgabe meldet, daß mit seans Netzwerkkarte alles in bester Ordnung ist. Die Adressangaben und Subnetzmaske sind richtig gesetzt und der Status dieses Geräts wird mit "UP" und "RUNNING" beschrieben. "UP" heißt, daß es aktiviert und zum Gebrauch freigegeben ist (root kann es auch deaktivieren, mit: *ifconfig eth0 down*). "RUNNING" besagt, daß mit Netzwerktreiber o.ä. alles in Ordnung ist und das Gerät betriebsbereit ist.

### ***traceroute***

Die Verfolgung einer Route durch's Netz mit *traceroute* (als root ausgeführt) ist der nächste sich anbietende Schritt wenn ein Host per *ping* nicht erreichbar ist. *traceroute* liefert gleich zwei aufschlußreiche Aukünfte zurück:

- Gelangen Datenpakete zum gewünschten Empfänger?
- Wenn sie unterwegs aufgehalten werden, dann wo?

*traceroute* verfolgt, Zwischenstation für Zwischenstation, welchen Weg seine zu Testzwecken versandten UDP-Pakete nehmen. Für jede Zwischenstation wird, so eine Zuordnung möglich ist, angegeben, wie der entsprechende Gateway heißt (Name und IP-Adresse), an wievielter Stelle er in der Route sitzt und welche Zeit die Testpakete für den Weg Hin- und zurück benötigen.

Ein Beispiel:

```
sean:~ # traceroute www.uni-kl.de
traceroute to www.uni-kl.de (131.246.119.51), 30 hops max, 40 byte packets
 1 dialin3f.kdt.de (195.8.240.230)  51 ms  37 ms  36 ms
 2 cisco-f.kdt.de (195.8.224.197)  56 ms  268 ms  215 ms
 3 seldon.isq.de (195.8.224.198)  53 ms  58 ms  40 ms
 4 212.162.48.241 (212.162.48.241)  54 ms  43 ms  46 ms
 5 212.162.34.193 (212.162.34.193)  60 ms  43 ms  58 ms
 6 loopback0.decix1.Frankfurt1.Level3.net (212.162.32.8)  46 ms  41 ms  43 ms
 7 * * *
 8 cr-frankfurt1.g-win.dfn.de (188.1.80.37)  57 ms  46 ms  53 ms
 9 cr-stuttgart1.g-win.dfn.de (188.1.18.73)  63 ms  47 ms  50 ms
10 ar-kaiserslautern1.g-win.dfn.de (188.1.76.38)  58 ms  60 ms  59 ms
11 gwin.rhrk.uni-kl.de (131.246.1.1)  400 ms  64 ms  63 ms
12 vgw190.rhrk.uni-kl.de (131.246.190.254)  220 ms  127 ms  138 ms
13 www-admin.uni-kl.de (131.246.119.50)  242 ms  109 ms *
```

Konkret läuft ein *traceroute* Aufruf folgendermaßen ab:

*traceroute* sendet UDP-Pakete mit einem kleinen ttl(time-to-live)-Wert in Richtung des gewünschten Rechners. Time-to-live gibt an, über wieviele Zwischenstationen ein Paket im Internet geroutet werden kann, bis es an seinem Ziel angekommen ist. Standardmäßig muß jedes Paket nach 30 "Hops" an seinem Ziel sein - jeder Router, den es auf seinem Weg passiert, setzt den Zähler um eins herunter. Der ttl-Wert verhindert, daß Datenpakete für alle Ewigkeit im Internet herumschwirren und über unzählige Ecken umgeleitet werden. Dadurch, daß *traceroute* von kleinen ttl-Werten aufsteigend hochzählt, werden die zwischen Start und Ziel liegenden Stationen erkannt. Jeder passierte Gateway setzt den ttl-Wert wieder um einen Zähler zurück, wenn der ttl danach bei 0 angekommen ist, meldet der entsprechende Gateway dies an *traceroute* zurück ("time exceeded"). Der Gateway, an dem der 0 Zähler erreicht wurde, erscheint dann in der Auflistung der Zwischenstationen. Sind die Pakete dann endlich bis zum Ziel vorgedrungen, sendet der Empfänger an *traceroute* eine ICMP-Meldung mit dem Inhalt "Unreachable Port". Für *traceroute* ist dies das Signal aufzuhören, und für den Benutzer besteht auch kein Grund zur Panik. Der ungültige Port auf der Zielmaschine wurde mit Bedacht gewählt, da die Pakete nicht wirklich vom Empfänger entgegengenommen und verarbeitet werden sollen, sondern nur seine Erreichbarkeit getestet wird.

Wenn wie in der beispielhaften *traceroute* Ausgabe oben, anstelle des Gateways nur drei Sternchen auftauchen, bedeutet dies, daß drei Versuche, von dem Rechner eine Antwort zu bekommen (ICMP "Time Exceeded"), fehlgeschlagen sind. Der Paketverlust kann zum einen in einem wirklichen Defekt des Netzwerks begründet sein, oder aber vom Administrator des entsprechenden Rechners bewußt hervorgerufen werden, wenn er beispielsweise einen Paketfilter aufgesetzt hat, der ICMP-Meldungen nicht zuläßt.

Kann ein (externer) Host nicht erreicht werden, liefert letztendlich nur die Kombination aus *ping* und *traceroute* einen Hinweis darauf, ob innerhalb des eigenen Netzes schon etwas im Argen liegt oder ob der Fehler bei den Betreibern eines anderen Netzes (z.B. dem Provider) liegt. Ein erfolgloser *ping* auf eine externe Maschine mag tatsächlich bedeuten, daß mit dieser Maschine oder dem Netz, in das sie eingebunden ist, etwas nicht stimmt. Allerdings kann es genauso gut sein, daß die Daten - aus welchem Grund auch immer - den eigenen Gateway nicht korrekt passieren konnten. Und genau hier greift *traceroute*, indem es die Zwischenstationen testet :-)


### ***netstat***

Ist genau der oben genannte Fehler ("Pakete erreichen externen Host nicht") aufgetreten, und *traceroute* konnte schon den eigenen Gateway nicht ausmachen, sollte schleunigst das Routing des Gateways überprüft werden. Dies erreicht man mit dem Aufruf `netstat -r` (als root ausgeführt) oder, wenn man die IP-Adressen anstelle von Rechnernamen sehen möchte, mit `netstat -rn`. Mehr zu den Fähigkeiten von *netstat* später, wenn es um die Erstellung von Filterregeln für Firewalls geht...

## **Ausblick**

Dies war eine Art Schnelldurchgang durch die Möglichkeiten der Netzwerkanalyse. Die genannten Tools sind noch lange nicht das gesamte Rüstzeug, mit dem ein Linux-Netzwerker sich ins Getümmel stürzen kann. Zum Beispiel wird er auch lernen (müssen), Kernelmeldungen unter `/var/log/messages` zu deuten - zum Beispiel wenn der Kernel eine nicht erkannte Netzwerkkarte meldet. Das ist allerdings wieder für sich gesehen eine eigene Geschichte. Für ein kleines überschaubares Netz (siehe unser Beispiel) sind die Grundlagen aber in diesem Artikel weitgehend gelegt ;-)

## **Weiterführende Literatur und Links**

- Ein Klassiker, den der Linux Netzwerker gelesen haben sollte, ist Olaf Kirchs [Linux Network Administrator's Guide](#), die erste Ausgabe (von 1996) gibt es auch in einer deutschen Version als [Linux - Wegweiser für Netzwerker](#). Auch für Anfänger eine richtige Fundgrube zum Nachschlagen oder Schmökern. 

**Linux auf dem Server 27.04.2001**