

# Die Route ins Nirvana

Von

duddits (Daniel Baier)

Aufgabe:

Erkläre die Funktionsweise des Routings im vorliegenden Szenario mit alter und neuer Routing-Tabelle auf dem Gateway-Router.

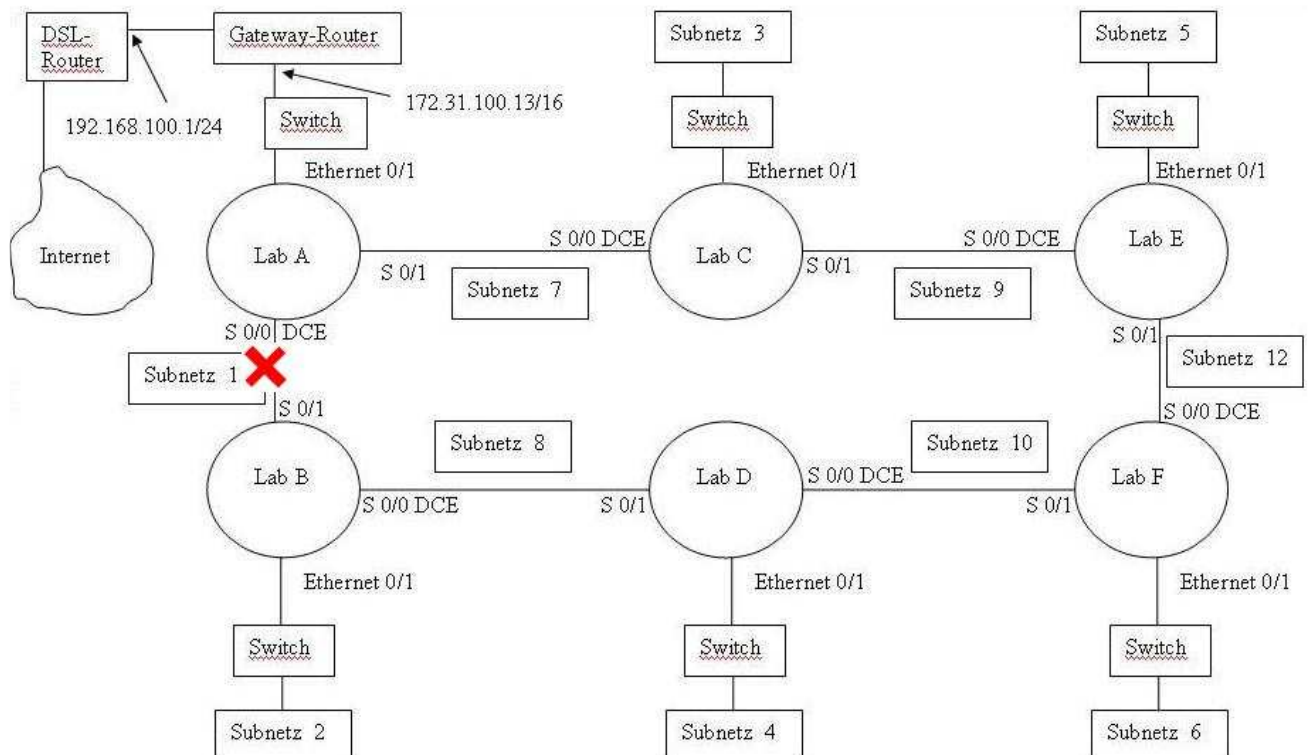
Lösung:

Der Proxy-Server bzw. Gateway-Router hatte in seiner ursprünglichen Form (siehe Abbildung 1), eine Fehlkonfiguration vorliegen. Dies hatte zur Folge, dass es nur dem Router LAB\_A (siehe Abbildung 2) möglich war, eine Verbindung mit dem Internet aufzubauen.

```
root@proxy:~ # route -n
Kernel IP routing table
Destination      Gateway         Genmask       Flags Metric Ref  Use Iface
192.168.100.0    0.0.0.0        255.255.255.0 U        0     0    0 eth1
172.31.0.0       0.0.0.0        255.255.0.0   U        0     0    0 eth0
0.0.0.0          192.168.100.1 0.0.0.0       UG       0     0    0 eth1
root@proxy:~ #
```

Abbildung 1: Alte Routing-Tabelle auf dem Gateway-Router

Im unten stehenden Netzplan sieht man die Struktur unseres Netzes. Dabei hält jeder Router bei seinen Seriellen Schnittstellen (genauer 2 Serielle Schnittstellen) eine private Klasse C Adresse, welche im RFC 1918 mit 192.168 spezifiziert ist.



## Abbildung 2: Netzplan

Das Problem in diesem Szenario ist, wie aus der Routing-Tabelle ersichtlich wird, das es eine Route ins 172.31.0.0/16 Netz und 192.168.100.0/16 Netz gibt, aber keine Route welche aus diesem Netz die Pakete wieder zurück zum LAB\_A Router schickt (genauer: um sie an die Netze des Szenarios weiterzuleiten), da dieser der einzige Router ist, der mit seiner Ethernet-Schnittstelle im 172.31.0.0/16 Netz ist.

Um das ganze genauer zu verdeutlichen beschreibe ich nun mal den Verlauf eines Ping-Paketes aus Sicht von LAB\_F auf Schicht 3 des ISO OSI-Referenzmodells.

LAB\_F schickt ein Ping-Paket auf google.de, dabei wird zunächst versucht eine Namensauflösung via DNS zumachen, damit der Router weiß wohin er das Paket schicken soll.

Da der DNS-Server mit der IP 172.31.100.11/16 angegeben wurde, wird zunächst versucht eine DNS-Abfrage an diesem zu schicken.

Dabei stellt der Router fest, das sich der DNS-Server nicht im selben Netz befindet und schickt nun sein DNS-Paket an seine Default-Route (0.0.0.0/0) bzw. Default-Gateway. Dieser ist hier für unsere Router LAB\_A.

Dieser besitzt eine Ethernet-Schnittstelle im 172.31.0.0/16 Netz mit der IP 172.31.100.15/16. Dieser Schickt dann sein Paket an den DNS-Server, welcher dann eine Namensauflösung für google.de durchführt und dann ein DNS-Reply zurück an die Absender-IP schicken möchte (192.168.12.2/24).

Da sich dieser nicht im Netz des DNS-Servers befindet, wird das Paket an seinen Standard-Gateway/Default-Gateway geschickt.

Dies ist der Gateway-Router/Proxy-Server mit der IP-Adresse 172.31.100.13/16. Nun sieht dieser in seiner Routing-Tabelle und stellt fest, dass dieser auch keine Route in 192.168.12.0/24 Netz hat.

Somit schickt dieser das Paket an seine Default-Route, welcher sich in Form des DSL-Routers repräsentiert mit der IP 192.168.100.1/24.

Da dieser auch keine Route zum Netz von LAB\_F hat, wird das Paket an dieser Stelle verworfen, da IP-Pakete aus dem privaten Netz nicht ins Internet geroutet werden.

Nun gilt es, eine Lösung dafür zu finden. Es bietet sich jetzt wohl an, am Gateway-Router mit der IP 172.31.100.13/16 eine Route zu erstellen, die alle Pakete von 192.168.12.0/24 Netz an LAB\_A zu schicken, da dies aber zu umständlich ist, wenn man das ganze für 15 verschiedene Netze machen muss, wäre es Sinnvoll hier eine Summen-Route (Supernetting) zu erstellen.

Diese wäre in diesem Fall 192.168.0.0/20 zum Router 172.31.100.15.

Mit folgendem Befehl wird diese Route auf dem Gateway-Router, welcher ein Linux-Rechner ist, hinzugefügt:

**Befehl:**

```
route add -net 192.168.0.0 netmask 255.255.240.0 gw 172.31.100.15
```

Alternativ geht auch folgendes, das man die Schnittstelle anstelle des Routers angibt:

**Befehl:**

```
#route add -net 192.168.0.0 netmask 255.255.255.0 dev eth0
```

Das Ergebnis der Routing-Tabelle ist nun folgendes und der Weg ins Internet ist nun für alle Router aus unserem Szenario möglich:

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.7.0	172.31.100.15	255.255.255.0	UG	0	0	0	eth0
192.168.100.0	0.0.0.0	255.255.255.0	U	0	0	0	eth1
192.168.8.0	172.31.100.15	255.255.255.0	UG	0	0	0	eth0
192.168.0.0	172.31.100.15	255.255.240.0	UG	0	0	0	eth0
172.31.0.0	0.0.0.0	255.255.0.0	U	0	0	0	eth0
0.0.0.0	192.168.100.1	0.0.0.0	UG	0	0	0	eth1

root@proxy:~ #

Abbildung 3: Neue Routing-Tabelle auf dem Gateway-Router