



Lösung 06: IP Adressen

Aufgabe 1: Longest Prefix Matching

- 135.46.63.10 → 1
- 135.46.57.14 → 0
- 192.53.56.7 → sonst

Tipp: Schauen Sie sich in Binärnotation das Oktet genau an, bei dem der Präfix endet bzw. der Netzanteil beginnt.

Beispiel: 135.46.56.0/22

135.46.(00111xxx)₂.(xxxxxxx)₂. Die mit „x“ gekennzeichneten Position gehören zum Netzanteil. Der Rest gehört zum Präfix. Ein Match liegt dann vor, wenn das ankommende Paket bzgl. des Präfix mit dem Eintrag in der Routingtabelle übereinstimmt.

Aufgabe 2: IPv4 Broadcast- und Netzadresse

Subnetz	Netzmaske	Broadcast IP	Anzahl nutzbarer IPs
10.1.0.0/16	255.255.0.0	10.1.255.255	$2^{16} - 2 = 65534$
12.0.0.0/8	255.0.0.0	12.255.255.255	16777214
200.13.12.0/25	255.255.255.128	200.13.12.127	126
200.13.0.0/23	255.255.254.0	200.13.1.255	510
200.13.128.0/23	255.255.254.0	200.13.129.255	510

Hinweis: Sowohl die Broadcast IP Adresse als auch die „Netzadresse“ werden nicht an Hosts vergeben. Letzteres hat auch historische Gründe. Deshalb zieht man von der theoretischen Anzahl an IP Adressen immer 2 ab.

Aufgabe 3: IP Fragmentation

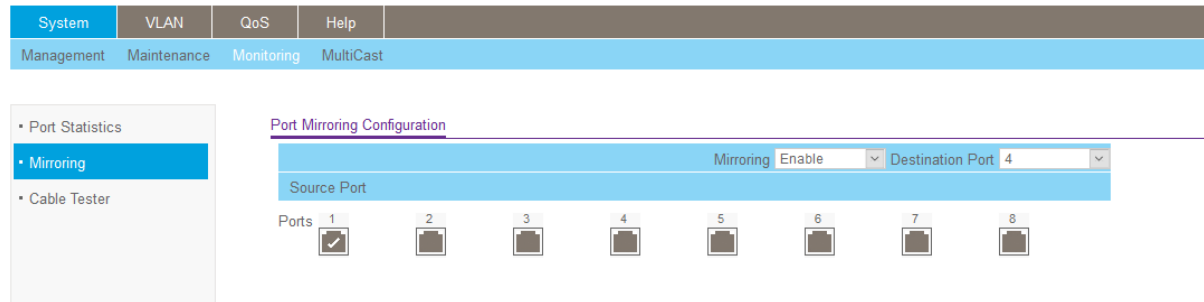
- a) Es werden 4 Fragmente benötigt. In jedem Fragment können wegen des 20 Byte großen IP Headers nur $500 - 20 = 480$ Bytes an Nutzdaten übertragen werden. Insgesamt müssen $1600 - 20 \text{ Bytes} = 1580$ Bytes an Nutzdaten übertragen werden. $1580 / 480 = 3,29$ ergibt aufgerundet 4 Fragmente.
- b)
1. Fragment: 480 Bytes an Nutzdaten groß, Offset = 0, fragflag = 1
 2. Fragment: 480 Bytes an Nutzdaten groß, Offset = 60, fragflag = 1
 3. Fragment: 480 Bytes an Nutzdaten groß, Offset = 120, fragflag = 1
 4. Fragment: 140 Bytes an Nutzdaten groß, Offset = 180, fragflag = 0
- Hinweis: Um Platz zu sparen, wird der Offset nicht absolut angegeben (z.B. 480 Byte), sondern als Vielfaches von 180.
- c) Die Fragmente werden am Empfänger des IP Datagramms wieder zusammengebaut, nicht am nächsten Router, der am anderen Ende des Links sitzt.
- d) Dazu gibt es eine 16-Bit Identifier (ID), die sich alle Fragmente eines Datagramms teilen.

Aufgabe 4: Monitoring Port (Kurzdemo)

Der Screenshot deutet an, wo Mirroring konfiguriert.

NETGEAR

GS108Ev3 - 8-Port Gigabit ProSAFE Plus Switch



Aufgabe 5: Linux – Statische IP Adresse und Routingtabelle

a) Kommando: `ip addr` (oder wahlweise / veraltet `ifconfig`)

```
dev@OS-Dev ~ $ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b4:04:6b brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 86251sec preferred_lft 86251sec
    inet6 fe80::9924:b4d5:9bd:d456/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Im konkreten Fall ist die IP Adresse von `enp0s3` – das ist die Ethernet-Schnittstelle – 10.0.2.15. Die Netzmaske ist 255.255.255.0 (also /24). Die VM bildet zusammen mit dem Virtual Box Host ein LAN.

Die erste Schnittstelle `lo` ist die Loopback-Schnittstelle. Dieses Interface wird verwendet, um z.B. mit einem Server auf dem gleichen Computer zu kommunizieren.

b) Kommando: `ip addr add 10.0.2.32/24 dev enp0s3`

Mit `ip addr` überzeugen, dass die Adresse übernommen wurde. `ifconfig` zeigt normalerweise nur die primäre IP an.

```
dev@OS-Dev ~ $ sudo ip addr add 10.0.2.32/24 dev enp0s3
[sudo] password for dev:
dev@OS-Dev ~ $ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b4:04:6b brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 86019sec preferred_lft 86019sec
    inet 10.0.2.32/24 scope global secondary enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::9924:b4d5:9bd:d456/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

c)

Die Routingtabelle enthält 3 Einträge. Der erste Eintrag ist die Defaultroute (0.0.0.0/0). Diese wird verwendet, wenn es sonst keinen Match gibt. Pakete werden in diesem Fall an die IP Adresse 10.0.2.2 weitergeleitet.

Wie 10.0.2.2 erreichbar ist, regelt der 2. Eintrag: Durch die zugewiesene IP Adresse 10.0.2.15 und der konfigurierten Netzmaske 255.255.255.0 (automatisch durch Oracle VirtualBox zugewiesen), hat der Kernel implizit ein Routingeintrag für 10.0.2.0/24 hinzugefügt. Alle Pakete für dieses Subnetz / LAN werden über enp0s3 ausgesendet.

Der letzte Eintrag wurde automatisch generiert. Rechner (hier die VM) können sich selbst eine freie IP Adresse automatisch zuweisen, diesen liegen immer im Bereich 169.254.0.0/16). Weitere Details hierzu: <https://de.wikipedia.org/wiki/Zeroconf>

```
dev@OS-Dev ~ $ ip route
default via 10.0.2.2 dev enp0s3 proto dhcp metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
169.254.0.0/16 dev enp0s3 scope link metric 1000
dev@OS-Dev ~ $
```

Die Angabe „metric“ gibt an, welcher Eintrag bei zwei sonst gleich guten Einträgen (= Matches) bevorzugt wird.