

**WORK IN PROGRESS** Die folgende Seite ist noch nicht fertig. Sie enthält nur Fragmente und wird in Kürze neu strukturiert.

## ToDoS

- erste Schritte auf der Terminal-Console
- Konfig zurücksetzen / sichern / einspielen
- IP setzen / ändern / löschen
- Kurzer Blick auf das Webfront-End mit seinen Möglichkeiten

# Mikrotik Einführung

Die Mikrotik-Router sind mit IP-Adressen beschriftet. Über diese IP-Adresse können sie über `telnet` oder `ssh` von einem Linux-Client aus angesprochen werden.

Folgende Werte sind per default gesetzt:

Parameter	Wert	Bedeutung
IP-Adresse	Raum G216 192.168.33.xxx	IP auf Port 1 (ether1)
	Raum W610 192.168.65.xxx	
Benutzername	schueler	Login über <code>telnet</code> oder <code>ssh</code> <b>OHNE</b> Passwort

## Konfiguration sichern

Die Befehle, mit denen die Konfiguration angelegt wurde, kann auf jeder Befehlsebene mit dem Kommando `export` im Terminal ausgegeben werden.

```
/ip address export
```

Den ausgegebenen Text kann man nun mit dem Mauszeiger markieren und mit der mittleren Maustaste (Rollrad) in einen Texteditor (z.B. `gedit`) einfügen.

## Konfiguration auf Default zurücksetzen

Default-Reset auf voreingestellte Werte durchführen:

```
/system reset-configuration
```

Damit werden alle Werte auf die default-Werte zurückgesetzt. Auf Port1 (ether1) ist dann die auf dem Gerät aufgedruckte IP-Adresse konfiguriert und der Benutzername `schueler` (ohne Passwort) kann für den Login verwendet werden.

## HINWEIS:

## Konfiguration Importieren

Mit `export file=MYFILE.rsc` kann die aktuelle Konfiguration in der Datei `MYFILE.rsc` gespeichert werden. Dabei werden alle Parameter gespeichert. Falls nur Teile gespeichert werden sollen, dann kann in den Unterbereich gewechselt werden und z.B. mit `routing rip export file=rip.rsc` nur der RIP-Konfigurationsteil gespeichert werden.

**HINWEIS:** Bei Wiedereinspielen der Konfiguration wird der Inhalt additiv importiert. Dh. es sollte die Default-IP-Adresskonfiguration auskommentiert werden, da es ansonsten zu Fehlermeldungen kommt.

Die Konfiguration kann per Files→ Datei Upload auf den Router gebracht werden.

## Mikrotik als virtuelle Maschine

Man kann eine Mikrotik-Router als sogenannten Cloud Hosted Router innerhalb einer Virtualisierungssoftware wie `VirtualBox` betreiben. Dabei verfügen diese Router über alle Funktionen, die ein Hardware Mikrotik-Router besitzt. Lediglich die Übertragungsraten auf den Schnittstellen ist auf 1Mbit/s gedrosselt.<sup>1)</sup>

Auf der [Downloadseite von Mikrotik](#) kann man unter `Cloud Hosted Router`<sup>2)</sup> eine sogenanntes VDI-Image<sup>3)</sup> herunterladen. Empfohlen wird die `stable`-Version. Dieses VDI-Image kann in `VirtualBox` als Festplatte eingebunden werden, soll dass die Neuinstallation komplett entfällt.

Der grundlegende [Umgang mit VirtualBox](#) wird einer eigenen Seite erklärt.

**HINWEIS:** Anders als dort beschrieben muss keine neue Festplatte generiert werden, sondern das CHR-Image wird als bestehende Festplatte eingebunden.

Als Parameter für die neue virtuelle Maschine wählt man:

- Typ: Linux
- Version: Other Linux (64-bit)
- 512MB RAM
- Verhandene Festplatte verwenden → die oben heruntergeladene Datei auswählen

Anschließend sollte man im Netzwerkbereich noch mindestens eine weitere Schnittstelle hinzufügen, damit man sinnvolle Routing-Szenarien konfigurieren kann.

Die Netzschnittstellen sollten auf Brücke und mit „einfach“ unterscheidbaren MAC-Adresse konfiguriert werden.

Nun kann die virtuelle Maschine gestartet werden. Als Benutzername ist `admin` ohne Passwort eingestellt. Danach muss noch einmal die EULA akzeptiert werden.

Ab hier können alle weiteren Übungen mit der virtuellen Maschinen durchgeführt werden.

# Übungen

## VLAN

```
/interface bridge
add name=bridge1
/interface vlan
add interface=ether2 name=Vlan5_1 vlan-id=5
add interface=ether3 name=Vlan5_2 vlan-id=5
add interface=ether4 name=Vlan7_1 vlan-id=7
add interface=ether5 name=Vlan7_2 vlan-id=7

/interface bridge port
add bridge=bridge1 interface=ether2
add bridge=bridge1 interface=ether3
add bridge=bridge1 interface=ether4
add bridge=bridge1 interface=ether5
add bridge=bridge1 interface=ether6
```

## RIP

IP-Adressen setzen

LAN und öffentlicher Bereich (Switch an dem alle Router hängen)

```
/ip address add address=192.168.40.1/24 comment=LAN interface=ether5
network=192.168.40.0
/ip address add address=10.0.0.40/8 comment=switch interface=ether2
network=10.0.0.0
```

RIP aktivieren:

```
# Ausgabe der rip-Konfig
/routing rip export

/routing rip set redistribute-connected=yes
/routing rip network add network=10.0.0.0/8
```

## OSPF Übung

Im Folgenden ist ein Trace mit einer Area in der fünf Router konfiguriert wurden.

Trace OSPF Ring mit fünf Routern (RING2)

## mikrotik\_schulung\_2018.pdf

```
# IP-Adressen setzen letztes Oktett entspricht der Router Nummer. Hier R2
add address=10.0.1.2/24 interface=ether2 network=10.0.1.0
add address=10.0.2.2/24 interface=ether3 network=10.0.2.0

# Aktiv den router-id setzen. Hier: router-id=2.2.2.2 das R2
/routing ospf instance
set [ find default=yes ] router-id=2.2.2.2

# OSPF auf den neuen Netzen aktivieren.
/routing ospf network
add area=backbone network=10.0.1.0/24
add area=backbone network=10.0.2.0/24

# Überprüfen, welche Nachbarn existieren
/routing ospf neighbor print
```

## Kosten auf den Interfaces ändern

Zunächst sollte man sich einen Überblick über die aktuelle OSPF-Interface-Konfiguration machen. Folgender Befehl zeigt diese an.

```
# Interface Konfiguration anzeigen
/routing ospf interface print

# Ausgabe:
Flags: X - disabled, I - inactive, D - dynamic, P - passive
# INTERFACE COST PRIORITY NETWORK-TYPE
AUTHENTICATION AUTHENTICATION-KEY
0 ether3 100 1 broadcast none
1 D ether2 10 1 broadcast none
```

Am D kann man erkennen, dass das Interface mit der Nr 1 dynamisch ist. Hier sind die Kosten auf dem Default-Wert von 10. Diese können auch bei dynamischen Interfaces nicht geändert werden. Erst nachdem man aus dem dynamischen Interface ein statisches (ohne D) konfiguriert, kann man aktiv Kosten setzen.

## Routenaggregation in OSPF

Auf Router 3 des Backbones werden die Routen für 10.0.0.0/16 aggregiert, statt fünf einzelner /24er Netze. Das selbe wird für das 10.10.0.0/16 (Ring2) durchgeführt.

```
/routing ospf area range add area=Ring2 range=10.10.0.0/16
/routing ospf area range add area=backbone range=10.0.0.0/16
```

## Interface statisch machen und ändern der Kosten

Zunächst muss das bereits unter IP angelegte interface unter OSPF bekannt gemacht werden. HINWEIS: Dynamischen Interfaces (mit D gekennzeichnet) können unter OSPF keine Kosten zugewiesen werden.

```
# Aus dynamischen interface statische im OSPF machen
/routing ospf interface add copy-from=0
/routing ospf interface add copy-from=1

# Kosten setzen hier auf den oben übernommen interfaces 0 und 1
/routing ospf interface set 0 cost=100
/routing ospf interface set 1 cost=100
```

## Bridging / Switch

Man kann auf dem Mikrotik mehrere Ports zu einem Switch zusammenfassen. Dies wird über folgenden Befehl erreicht.

```
# Anlegen der Bridge mit dem Namen "lan"
/interface bridge add name=lan

# Hinzufügen der einzelnen Ports zur Bridge. Hier ether6-ether10
/interface bridge port add bridge=lan interface=ether6
/interface bridge port add bridge=lan interface=ether7
/interface bridge port add bridge=lan interface=ether8
/interface bridge port add bridge=lan interface=ether9
/interface bridge port add bridge=lan interface=ether10

# Setzen der IP-Adresse für die Bridge
/ip address add interface=lan address=192.168.34.1/24
```

## Masquerading / NAT

Die NAT-Funktionalität befindet sich im Bereich firewall.

```
/ip firewall nat
add action=masquerade chain=srcnat src-address=192.168.34.0/24
```

Im Webinterface können unterhalb weiter Bedingung festgelegt werden, die die Ausführung der Regel beeinflussen. Z.B. Tageszeiteinschränkungen oder Port oder Zielnetze.

## IP-Adressen entfernen

```
/ip address print

# Ausgabe
Flags: X - disabled, I - invalid, D - dynamic
#   ADDRESS           NETWORK           INTERFACE
0   ;;; Experimentalnetz 1
    192.168.33.90/24   192.168.33.0     ether1
1   10.0.1.2/24       10.0.1.0         ether2
2   10.0.2.1/24       10.0.2.0         ether3
3   192.168.34.1/32   192.168.34.1     lan

# Löschen der letzten IP-Adresse
/ip address remove numbers=3

/ip address print

# Ausgabe
Flags: X - disabled, I - invalid, D - dynamic
#   ADDRESS           NETWORK           INTERFACE
0   ;;; Experimentalnetz 1
    192.168.33.90/24   192.168.33.0     ether1
1   10.0.1.2/24       10.0.1.0         ether2
2   10.0.2.1/24       10.0.2.0         ether3
```

## Tools - Packet Sniffer

Mitschneiden von Traffic kann per Tools → Packet Sniffer stattfinden. Hierzu werden folgende Einstellungen vorgenommen.

Unter File name kann ein Dateiname für die Capture-Datei festgelegt werden. Hier nat\_cap.pcap  
Unetr Interfaces können die zu beobachtenden Schnittstellen angegeben werden. Hier ether2  
und ether9

Danach mit Apply die Parameter übernehmen und mit Start die Aufnahme beginnen.



## MPLS

MPLS-Package muss aktiviert sein.

Protokolle:

- Label Distribution Protocol<sup>4)</sup> (RFC 3036, RFC 5036) for IPv4
- Label Switched Path (LSP)
- Label Switched Router (LSR)
- Label Switched Core Router (MPLS interne Router)
- Label Switched Edge Router (Router Endpunkte zum normalen Routing)

Zunächst benötigt der Router eine eindeutige ID. Diese kann unter MPLS→MPLS→ LDP Settings → LSR ID und Transport Address vergeben werden. Hier sind IP-Adressen zu vergeben, die im MPLS-Netz eindeutig sein müssen. Zunächst wird ein Bridge-Interface erstellt und diesem ein IP vergeben

```
# Pseudo Bridge Interface für LSR-ID im LDP Settings Dialog
/interface bridge
add fast-forward=no name=mpls

# IP-Adresse als LSR-ID anlegen
/ip address
add address=10.255.33.90 interface=mpls network=10.255.33.90

# OSPF Basisnetz konfigurieren
/routing ospf instance
set [ find default=yes ] router-id=2.2.2.2
/routing ospf interface
add cost=100 interface=ether3 network-type=broadcast
add interface=ether2 network-type=broadcast
/routing ospf network
add area=backbone network=10.0.1.0/24
add area=backbone network=10.0.2.0/24

# MPLS Konfiguration
/mpls ldp
set enabled=yes lsr-id=10.255.33.90 transport-address=10.255.33.90
/mpls ldp interface
add interface=ether2
add interface=ether3
```

## VPLS in MPLS

Ein VPLS-Layer-2-Tunnel soll zwischen zwei LANs an R2 und R4 konfiguriert werden. Diese beiden Netze sollen transparent über ein MPLS-basiertes Netz mit einander kommunizieren.

Konfiguration des VPLS-Tunnels

```
# Die Grundkonfiguration wird aus dem MPLS Teil übernommen. Hier wird lediglich der VPLS-Tunnel hinzugefügt.

# Neues Bridge-Interface vpls_r4 anlegen.
/interface bridge
add fast-forward=no name=mpls
add name=vpls_r4

# Neues vpls-Interface einrichten mit der vpls-id=2:4
# Hier: R2->R4 auf Router R4 muss die identische VPLS-ID verwendet werden.
/interface vpls
add name=vpls2_4 remote-peer=10.255.33.82 vpls-id=2:4
```

```
# Neu erstellter VPLS-Tunnel vpls_r4 wird auf den LAN-Port ether5
geschaltet, an dem das lokale LAN,
# dass mit dem LAN an Router 4 über Layer-2 verbunden werden soll.
/interface bridge port
add bridge=vpls_r4 interface=ether5
```

Im Wireshark erkennt man bei funktionierendem Tunnel „von oben nach unten (außen nach innen)“ erst die MPLS-Basisverbindung und dann die darin eingebettete VPLS-Verbindung.

Weitere Möglichkeit zum Überprüfen findet man im Webfrontend im Bereich Tools→Traceroute

```
/tool traceroute address=10.255.33.82 src-address=10.255.33.90

# Ausgabe:
# ADDRESS                                LOSS SENT    LAST    AVG    BEST    WORST
STD-DEV STATUS
  1 10.0.1.1                               0%    8    0.3ms    1    0.3
5.5    1.7 <MPLS:L=80,E=0>
  2 10.0.5.1                               0%    8    0.3ms    0.3    0.2
0.3    0 <MPLS:L=29,E=0>
  3 10.255.33.82                           0%    8    0.2ms    0.2    0.2
0.4    0.1
```

## BGP

mikrotik\_schulung\_bgp\_basis.pdf

```
# IP-Adressen setzen.
# 2.2.0.2/8 öffentliche Adresse zu Peer AS200.
# 10.0.0.0/16 lokaler Bereich (AS300) hinter Router 3
/interface ether2
/ip address
add address=2.2.0.2/8 interface=ether2 network=2.0.0.0
add address=10.0.0.1/16 interface=ether9 network=10.0.0.0

# BGP Peer einrichten AS300 eigenes AS und AS200 Peer AS.
# 10.0.0.0/16 LAN-Bereich hinter R3/AS300
/routing bgp instance
set default as=300

# zu verbreitendes Netz aus dem lokalen Bereich AS300
/routing bgp network
add network=10.0.0.0/16 synchronize=no

# Partner Peer AS200 einrichten
/routing bgp peer
add name=AS200 remote-address=2.2.0.1 remote-as=200
```

Erweiterung um eine zweite direkte Verbindung von AS300 (R3) nach AS100 (R1)

```
# Neue IP-Adresse für die Verbindung zu Router1 AS100
/ip address
add address=3.2.0.2/8 interface=ether5 network=3.0.0.0

# Peer nach AS100
/routing bgp peer
add name=AS100 remote-address=3.2.0.1 remote-as=100
```

## IPv6 Router Advertisement

IPv6 Router Advertisement auf einer Schnittstelle aktivieren. **ACHTUNG:** Der Mikrotik-Router hat ein ungewöhnliches Verhalten bei IPv6. Die ND <sup>5)</sup> muss nach setzen einer ersten neue IPv6-Adresse aktiv neu eingeladen werden.

```
# IPv6 Adresse setzen
/ipv6 address
add address=2007::1 interface=ether9

# IPv6 ND neuladen, da sonst das Advertisement nicht ausgeführt wird.
/ipv6 nd
add interface=ether9 ra-lifetime=none

# Alternativ zum letzten Schritt kann auch ein reboot durchgeführt werden.
```

## STP

```
# Anlegen von STP auf einer Bridge mit Namen "stp_range" und der Priority
0x1000 (hex) im Modus STP.
/interface bridge
add name=stp_range priority=0x1000 protocol-mode=stp

# Hinzufügen von Ports zur obigen Bridge
# ether5 mit Port-Kosten von 50 und ether7 mit Port-ID von 0x70
/interface bridge port add bridge=stp_range interface=ether5 path-cost=50
/interface bridge port add bridge=stp_range interface=ether6
/interface bridge port add bridge=stp_range interface=ether7 priority=0x70

#
/interface bridge port
add bridge=stp_range hw=no interface=ether8
```

Anzeigen der Port-Rollen (root-port, designated port,...)

```
[admin@Router_90] /interface bridge> port monitor 0,1,2
                interface: ether5                ether6                ether7
```

status:	in-bridge	in-bridge	in-
bridge			
port-number:	1	2	3
role:	backup-port	designated-port	
designated-port			
edge-port:	no	no	no
edge-port-discovery:	yes	yes	yes
point-to-point-port:	yes	yes	yes
external-fdb:	no	no	no
sending-rstp:	no	no	no
learning:	no	yes	yes
forwarding:	no	yes	yes
root-path-cost:	50		
designated-bridge:	0x1000.CC:2D:E0:AC:4E:DF		
designated-cost:	0		
designated-port-number:	3		
hw-offload-group:	switch1	switch2	switch2

1)  
um einen Dauerbetrieb im professionellen Umfeld zu verhindern

2)  
CHR-Image

3)  
VDI: Virtual Disk Image

4)  
LDP

5)  
Neighbor Discovery

From:  
<https://www.kopfload.de/> - **kopfload - Lad Dein Hirn auf!**

Permanent link:  
[https://www.kopfload.de/doku.php?id=network:mikrotik\\_basics&rev=1569520357](https://www.kopfload.de/doku.php?id=network:mikrotik_basics&rev=1569520357)

Last update: **2025/11/19 16:12**

