MikroTik CloudRouterSwitch

Features and configurations

Overview

Big picture

- Company Profile
- Introduction
- Hardware STP
- Switch Configurations

FMS Internetservice GmbH

Company Profile

About me

- Sebastian Inacker <inacker@fmsweb.de>
- FMS Internetservice GmbH, Germany
- MikroTik Trainer (TR0011, May 2007)
- MTCNA, MTCRE, MTCTCE, MTCUME, MTCWE, MTCIPv6E, MTCINE
- Own training center and on site

  (So far: Austria, Germany, Hungary, Luxembourg, Malta, Netherlands, Switzerland, Uganda)

FMS Internetservice GmbH

- Value Added Distributor
  - Distribution
  - Training
  - Consulting
  - Support

- Founded 1997
- 11 employees
- Southern Germany

Get in Touch

- Website: http://www.fmsweb.de
- MikroTik Mirror: http://www.mikrotik-software.de
- Shop: http://www.mikrotik-shop.de
- Wiki: http://wiki.fmsweb.de
- Twitter: https://twitter.com/fmsweb_de
- Facebook: https://www.facebook.com/fmsinternetservice

- Phone: +49 761 2926500
- Email: sales@fmsweb.de

Training Center

- Official MikroTik trainings
- All certification levels
- First German speaking partner
- Two trainers
- Own training facility
- Inquiries: sales@fmsweb.de

Sebastian Inacker: TR11

Patrik Schaub: TR23

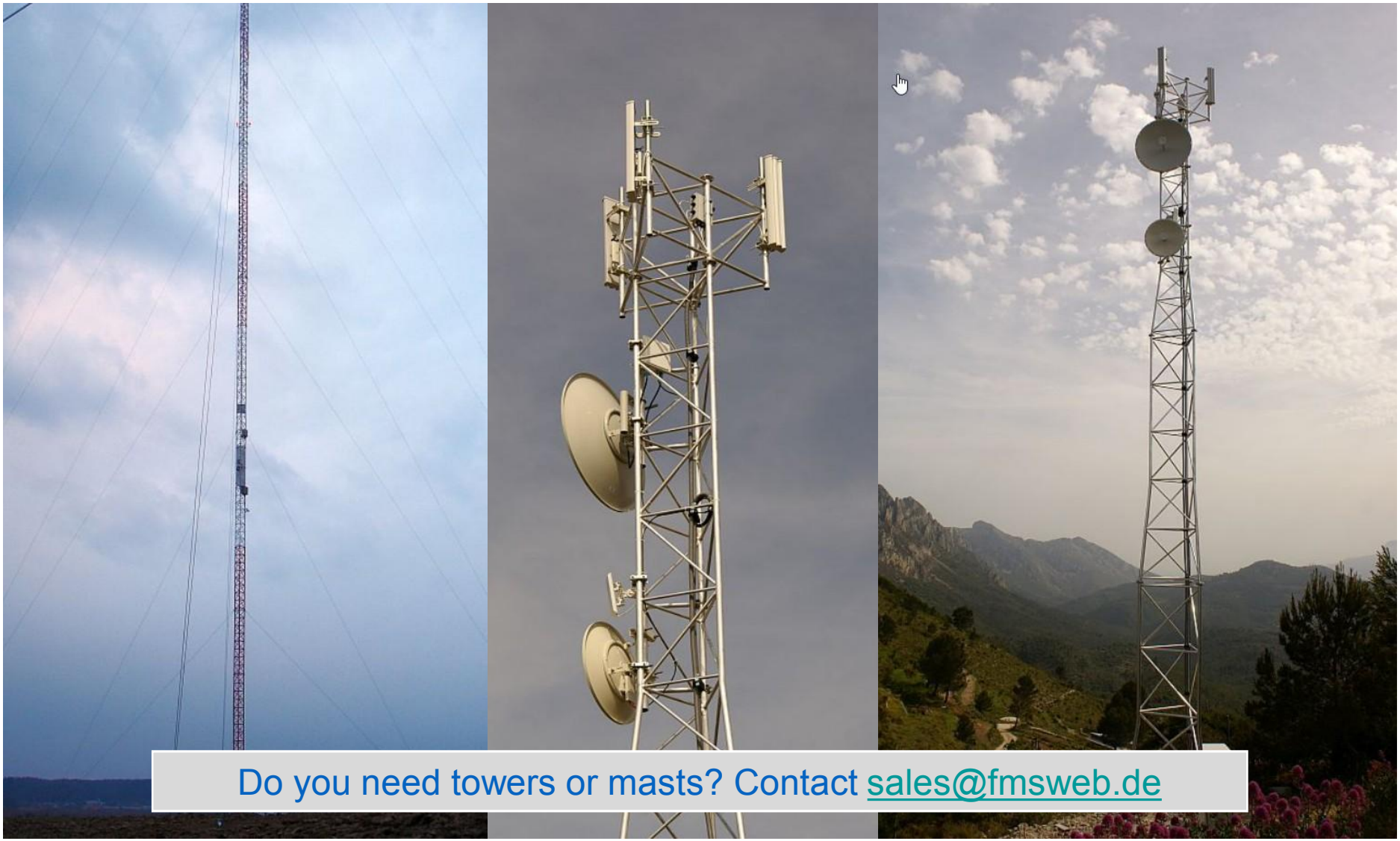
Distributor Table

Distributor Table

Live Demonstrations:

- Nokia Vplus setup
- Nokia AMS demonstration
- CRS 10G on 10 meter copper

Distributor Table



Do you need towers or masts? Contact sales@fmsweb.de

Introduction

About this talk

Topic:

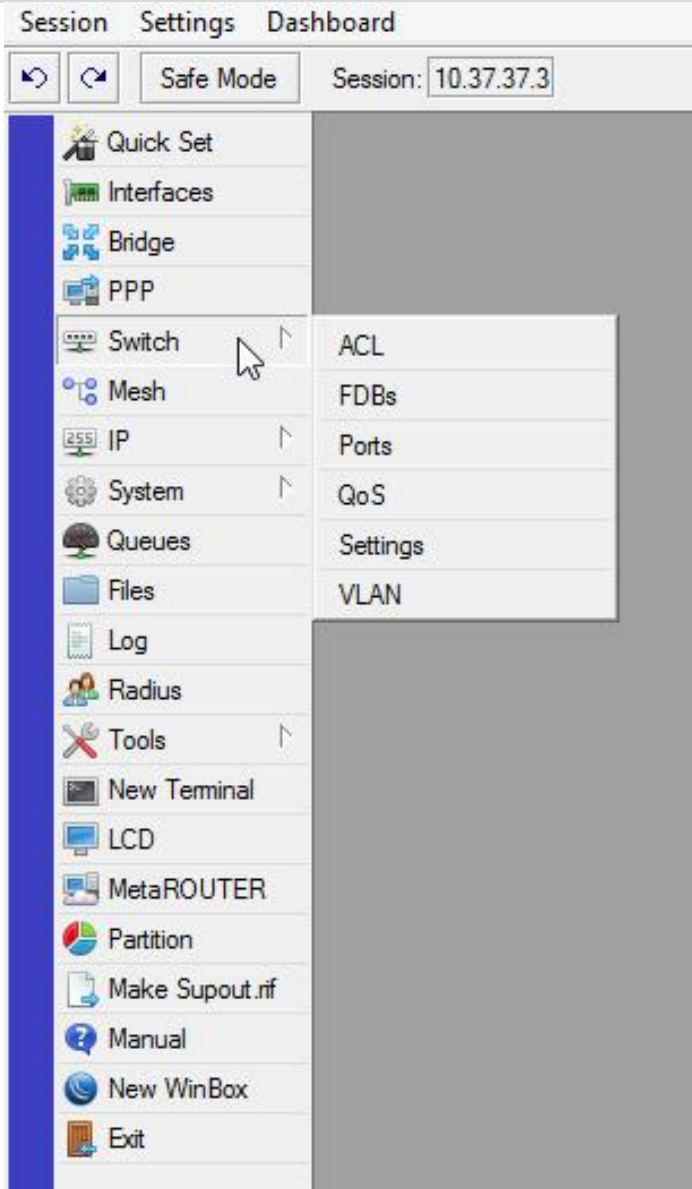
RouterOS on CRS

(Cloud Router Switch)

Not:

CSS (Cloud Smart Switch) with SWos

or switch chip on RB (RouterBOARD)



CRS or RB?

RouterBoard or CloudRouterSwitch?

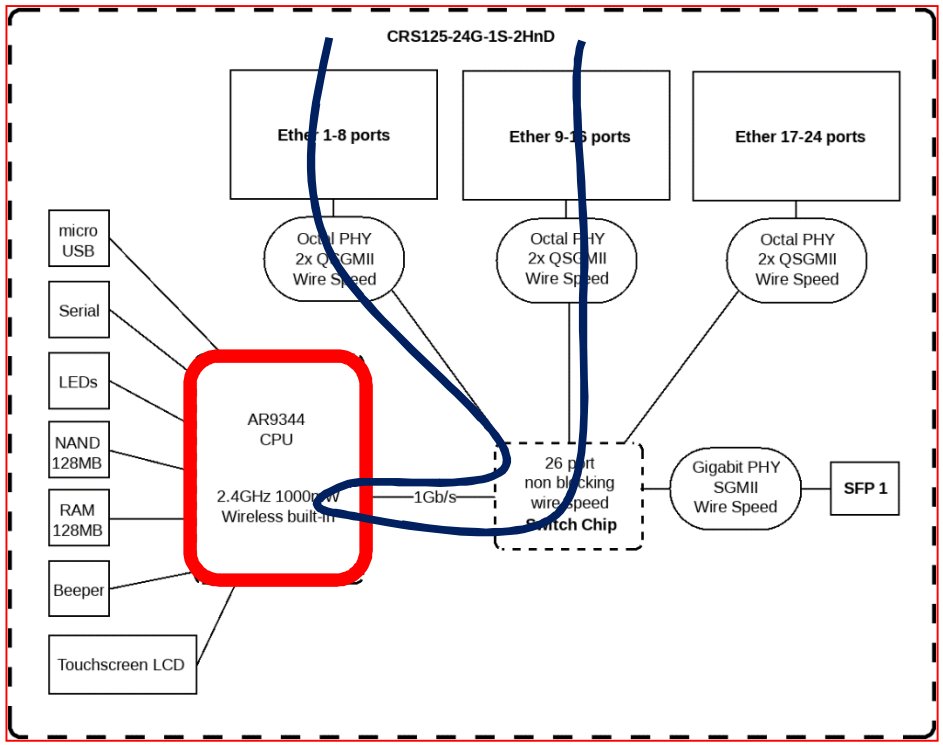
- RouterBOARD intended to be a router
- CloudRouterSwitch intended to be a switch

You can use them differently. Success depends on your needs

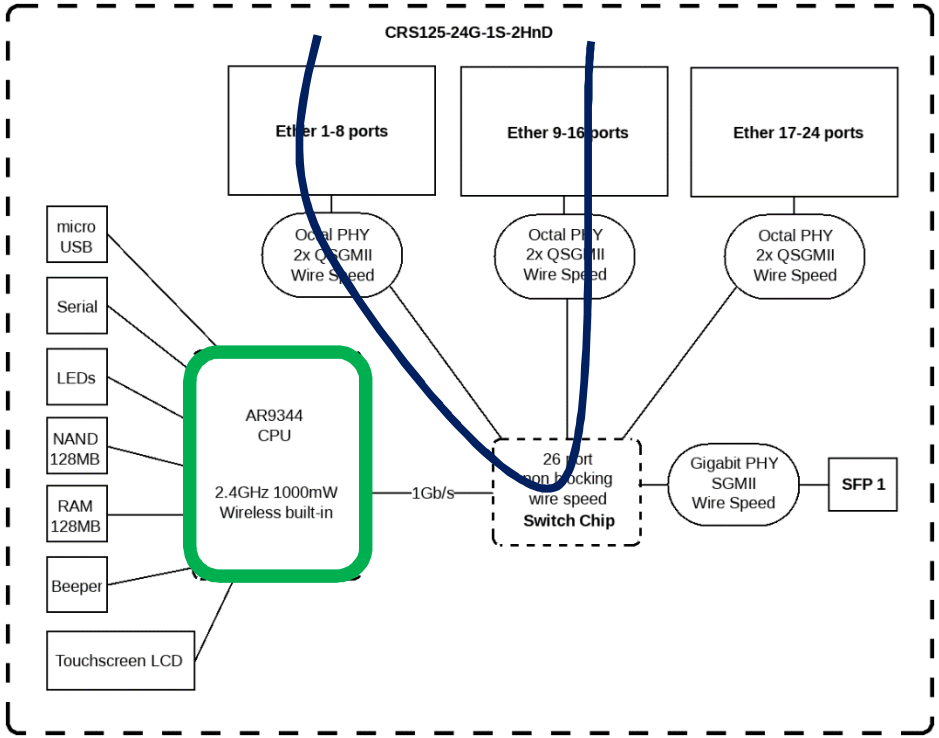
CRS125-24G-1S	Configuration	Mbps (1518 bytes)
Switching	Non blocking Layer 2 throughput	24,674.9
Bridging or Routing	none (fast path)	983.7

Bridge or switch chip (on CRS)?

Bridging (RouterOS)



Switching (switch chip)



Switch chip on RouterBOARD

Switch chip on RouterBOARD

- Wirespeed switching
- Different switch chips
- Different features (ACL, VLAN)

Wirespeed

Layer 1 throughput: Wirespeed at all packet sizes

		Packet sizes / Mbps		
CRS125-24G-1S	**Configuration**	**64 bytes**	**512 bytes**	**1518 bytes**
Switching	Non blocking Layer 1 throughput	25,000	25,000	25,000
Switching	Non blocking Layer 1 capacity	50,000	50,000	50,000

(Capacity only for comparision with other vendors)

CRS overview

Model	Ethernet	SFP	SFP+	ACL	CPU / RAM
CRS106-1C-5S	0-1	5-6	-	Yes	400 MHz / 128 MB
CRS212-1G-10S-1S+	1	10	1	Yes	400 MHz / 64 MB
CRS109-8G-1S-2HnD	8	1	-	No	600 MHz / 128 MB
CRS112-8G-4S	8	4	-	Yes	400 MHz / 128 MB
CRS210-8G-2S+	8	Up to 1 (sfp1)	1-2 (sfp1, sfp2)	Yes	400 MHz / 64 MB
CRS125-24G-1S(-2HnD)	24	1	-	No	600 MHz / 128 MB
CRS226-24G-2S+	24	Up to 1 (sfp1)	1-2 (sfp1, sfp2)	Yes	400 MHz / 64 MB

Switch, 16x ethernet

Switch > 16x ethernet

CRS125-24G-1S-RM

24x Gigabit Ethernet layer 3 Smart Switch, 1x SFP cage, LCD, 600MHz CPU, 128MB RAM, 1U rackmount, RouterOS L5

$199.00

CRS226-24G-2S+RM

24x Gigabit Ethernet Smart Switch, 2x SFP+ cages, LCD, 400MHz CPU, 64MB RAM, 1U rackmount case, RouterOS L5

$299.00

Model	Ethernet	SFP+	CPU / RAM	L1 Throughput	ACL
CRS125-24G-1S	24	-	**600 MHz / 128 MB**	25,000 Mbps	No
CRS226-24G-2S+	24	**1-2 (sfp1, sfp2)**	400 MHz / 64 MB	**44,000 Mbps**	**Yes**

Hardware STP

Hardware STP

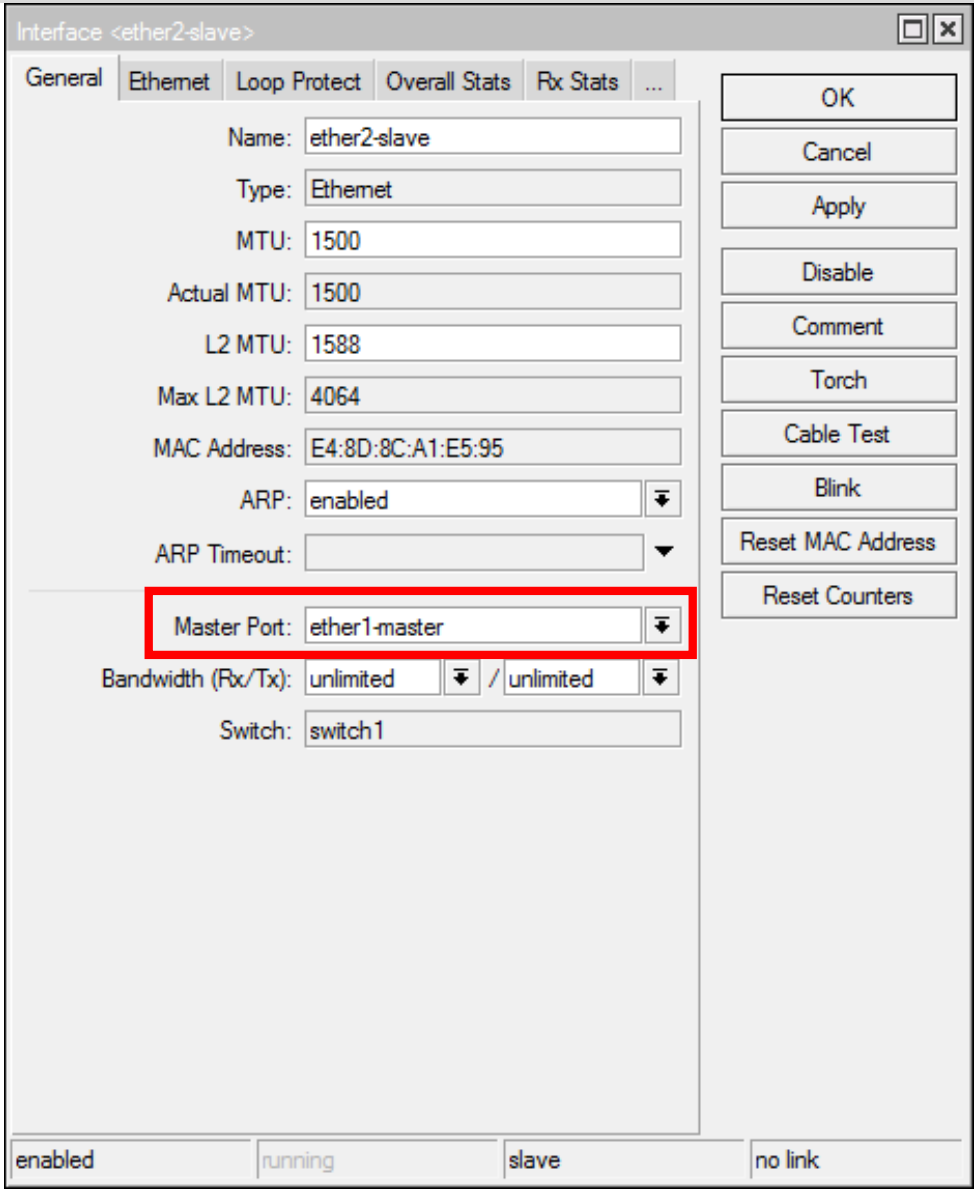
(R)STP = (Rapid) Spanning Tree Protocol:
Detect and prevent loops on your layer 2 network.

Hardware STP available since RouterOS v6.38rc2

Hardware STP

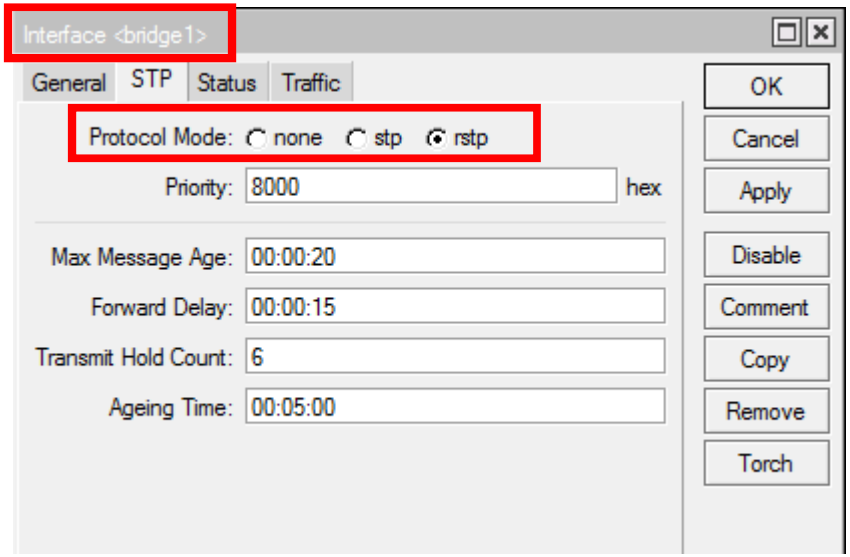
Simple setup:

- Define master port
- Create bridge(s) with RSTP
- Add (only) master port to bridge

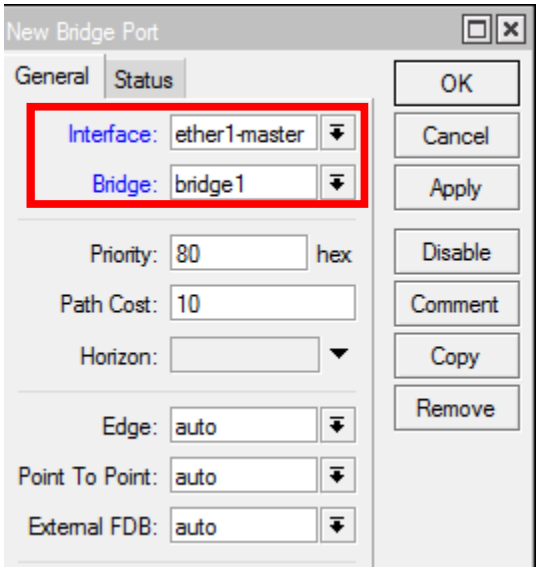


Hardware STP

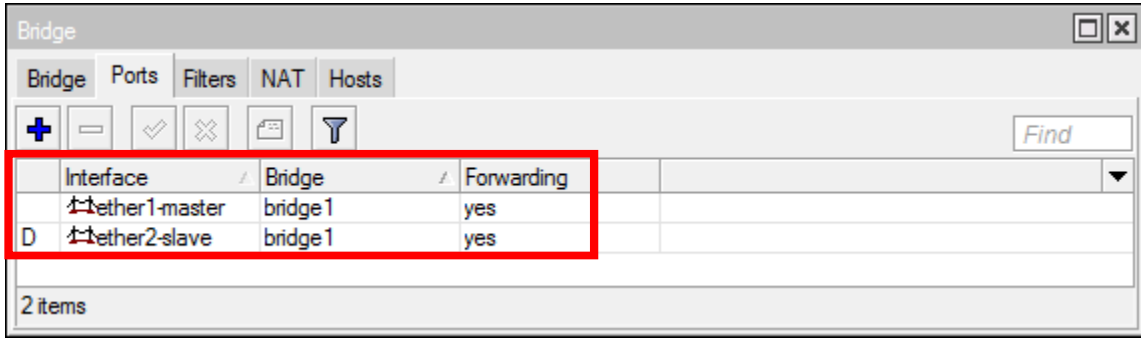
Create RSTP bridge



Add master port



Result: ether2 dynamic



Reference

```
/interface ethernet
set [ find default-name=ether1 ] name=ether1-master
set [ find default-name=ether2 ] master-port=ether1-master name=ether2-slave

/interface bridge
add name=bridge1 protocol-mode=rstp

/interface bridge port
add bridge=bridge1 interface=ether1-master
```

Hardware STP

Changelog: What's new in 6.38 (2016-Dec-30 11:33):

Important note!!!

RouterOS v6.38 contains STP/RSTP changes which makes bridges compatible with **IEEE 802.1Q-2014** by sending and processing **BPDU packets without VLAN tag**.

To avoid STP/RSTP compatibility issues with older RouterOS versions, upgrade RouterOS to v6.38 on all routers in Layer2 networks with VLAN and STP/RSTP configurations.

Hardware STP

What does this mean?

- Bridge Protocol Data Units (BPDUs, for STP loop detection) untagged.
- Loop detection: Untagged
- No dedicated loop detection per-VLAN (yet)

What could be a problem?

- More than one VLAN on interfaces: Loop on one VLAN will disable forwarding on **interface**, not VLAN

Hardware STP

No Problem (Loop detection as expected), if

- Only one VLAN on each interface
- No VLAN

Why did MikroTik do that?

- Switch chip hardware uses standard STP/RSTP protocol (IEEE 802.1Q-2014), which is not VLAN aware
- SW and HW Spanning Tree implementation compatible

Per-VLAN STP/RSTP: planned in future

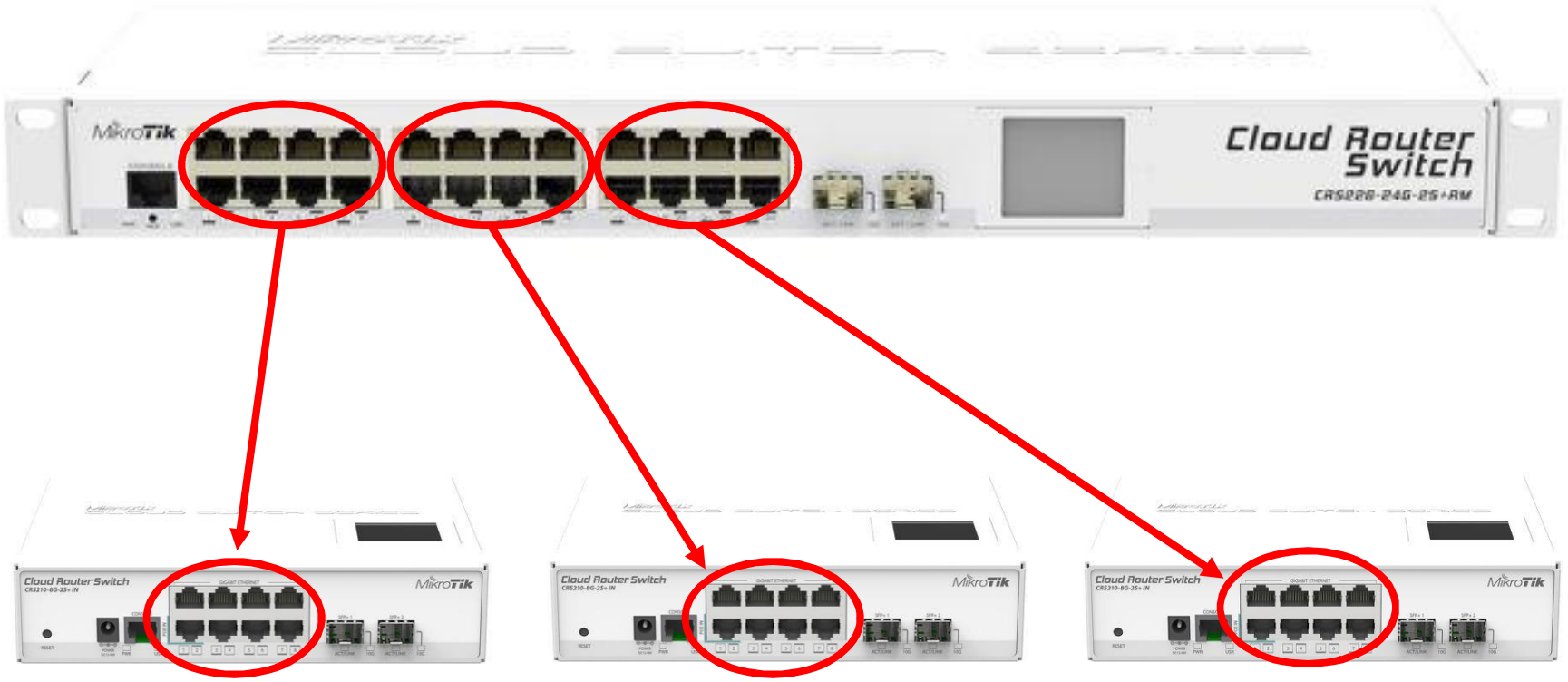
Wirespeed configurations

Common configurations

- One device, multiple switches
- Access Control Lists
- Split your trunk
- Dynamic VLAN definitions
  - MAC based
  - Protocol based
- Advanced traffic control example
- Client isolation
- Throughput control

One device, multiple switches

24 port CRS → 3x 8 port switches

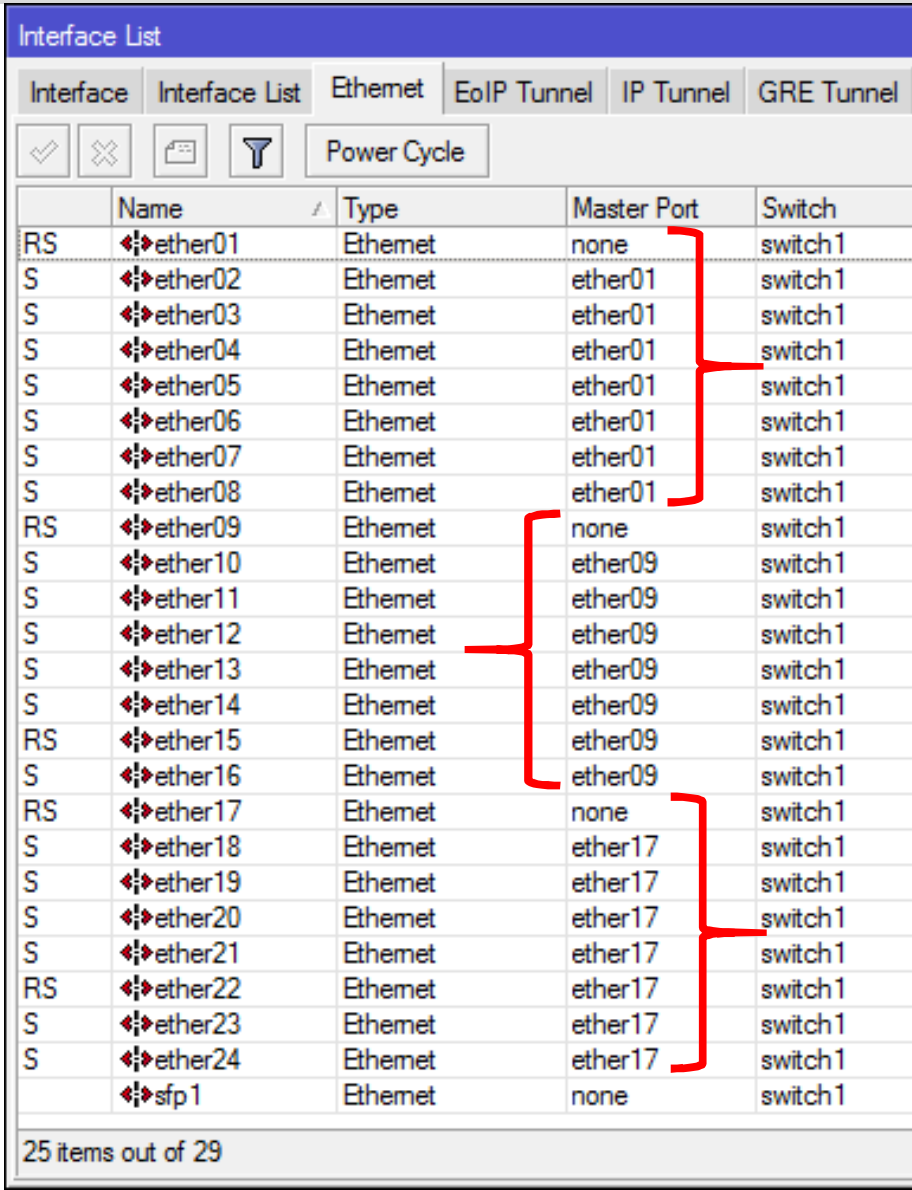


One device, multiple switches

Short, simple:
3 master ports:

Effective port isolation of connected devices

Switch chip on CRS:
Multiple master ports



Reference

```
/interface ethernet
set ether02,ether03,ether04,ether05,ether06,ether07,ether08 master-port=ether01
set ether10,ether11,ether12,ether13,ether14,ether15,ether16 master-port=ether09
set ether18,ether19,ether20,ether21,ether22,ether23,ether24 master-port=ether17
```

Access Control Lists

ACL

Access Control Lists:

- ACL tables: Ingress (incoming) and Egress (outgoing)
- Up to 128 ACL rules (RouterOS limitation)

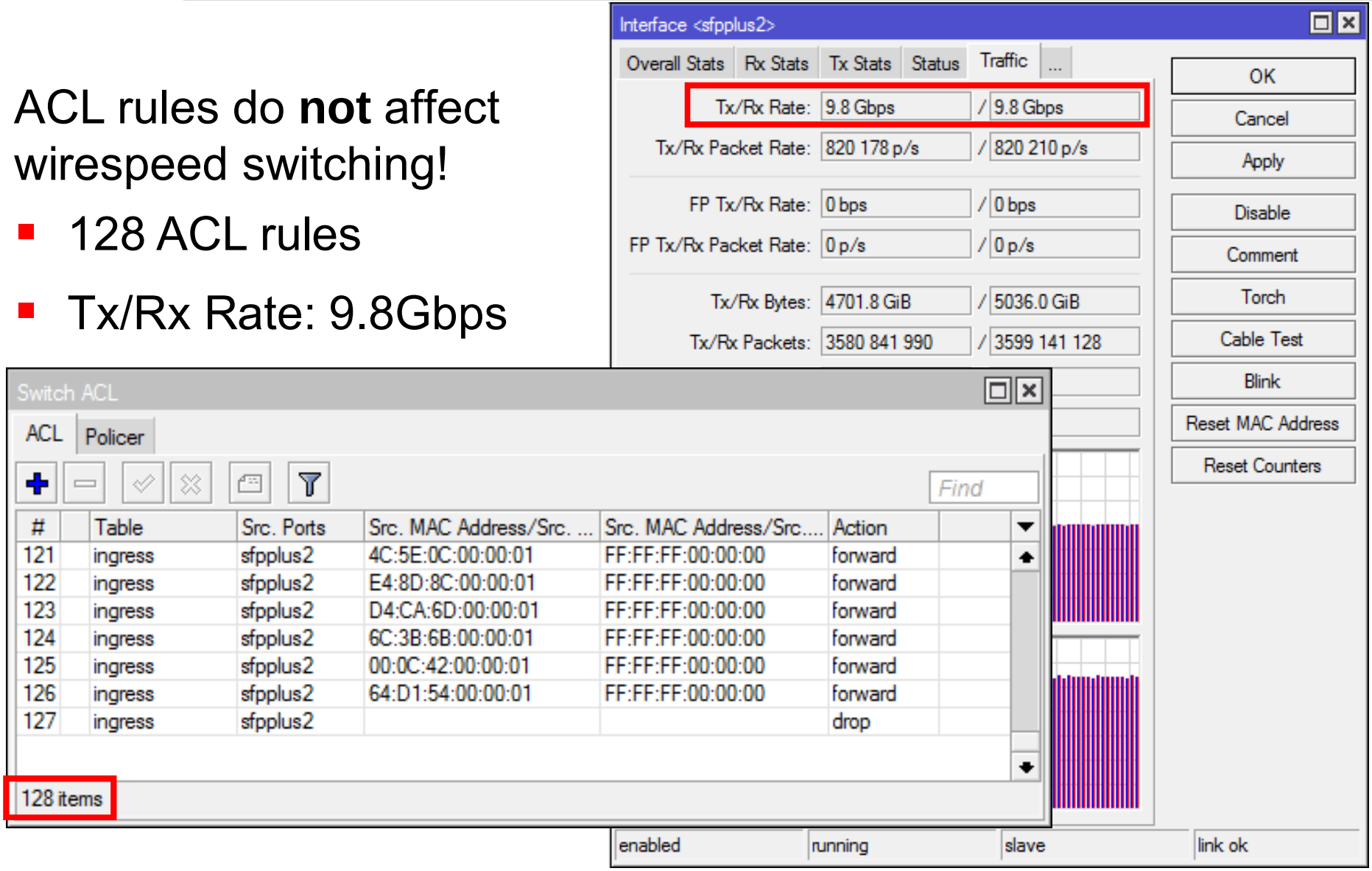
  Switch Chip CRS1xx / CRS2xx: 512 rules,

  CSS326 256 rules (SwitchOS)
- Classification based on ports, L2, L3, L4 protocol header fields

ACL and wirespeed

ACL rules do **not** affect wirespeed switching!

- 128 ACL rules
- Tx/Rx Rate: 9.8Gbps



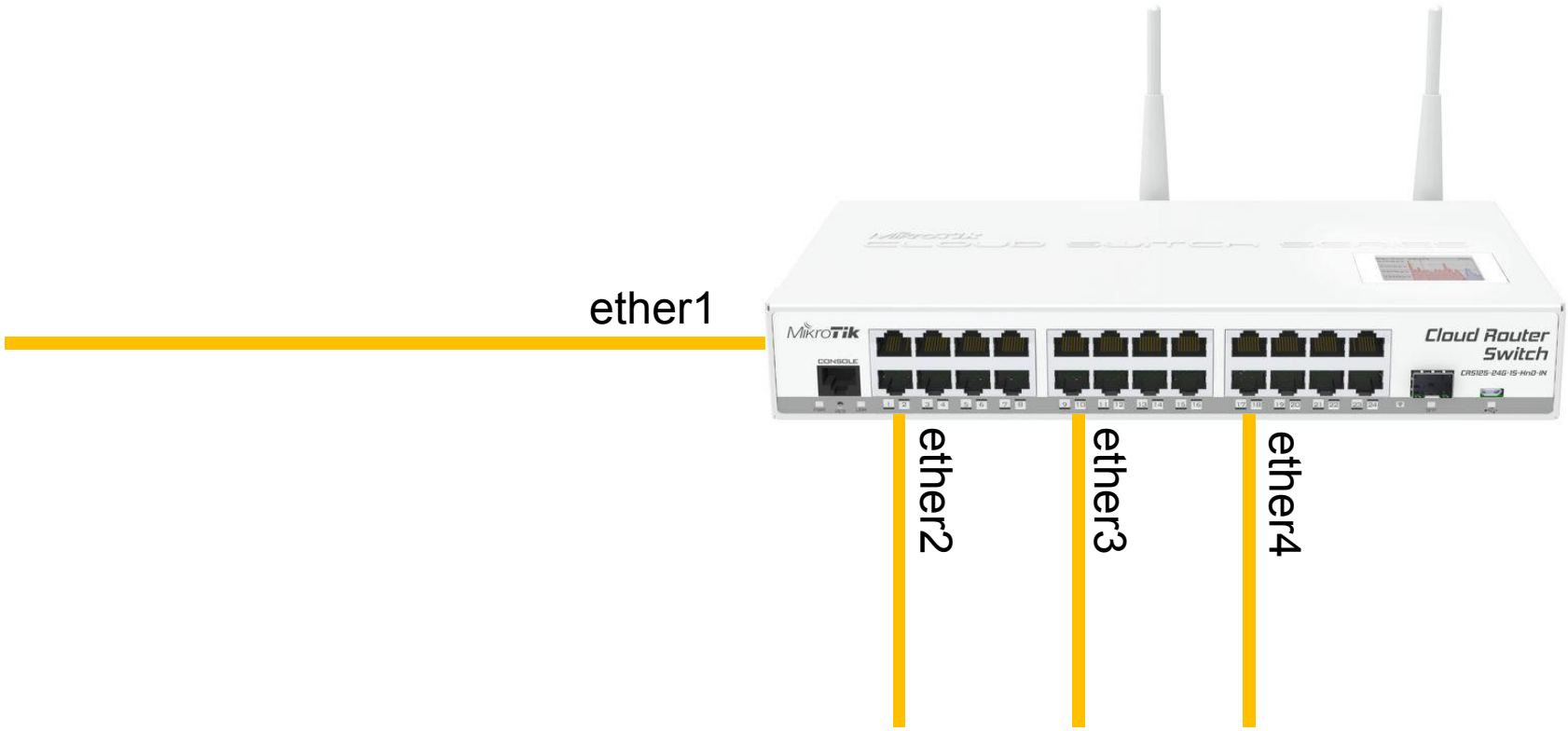
ACL support

Model	Switch Chip	Access Control List
CRS106-1C-5S	QCA-8511	Yes (128 rules available)
CRS112-8G-4S	QCA-8511	Yes (128 rules available)
CRS210-8G-2S+	QCA-8519	Yes (128 rules available)
CRS212-1G-10S-1S+	QCA-8519	Yes (128 rules available)
CRS226-24G-2S+	QCA-8519	Yes (128 rules available)
CRS125-24G-1S	QCA-8513L	No
CRS125-24G-1S-2HnD	QCA-8513L	No
CRS109-8G-1S-2HnD	QCA-8513L	No

ACL setups

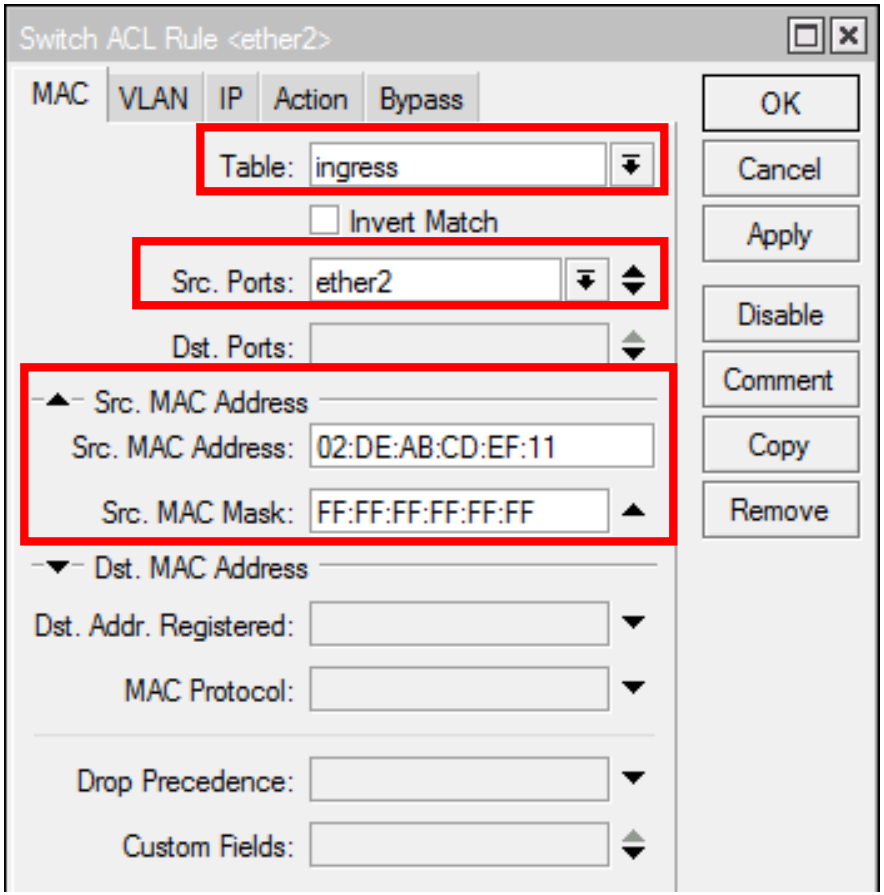
ACL – stay on your port!

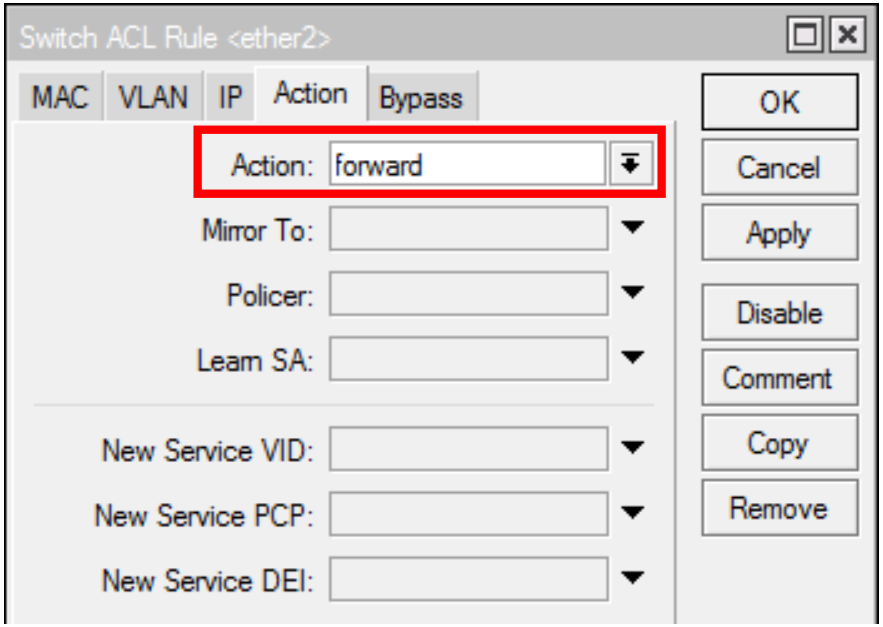
Bind 02:DE:AB:CD:EF:11 to ether2



ACL – stay on your port!

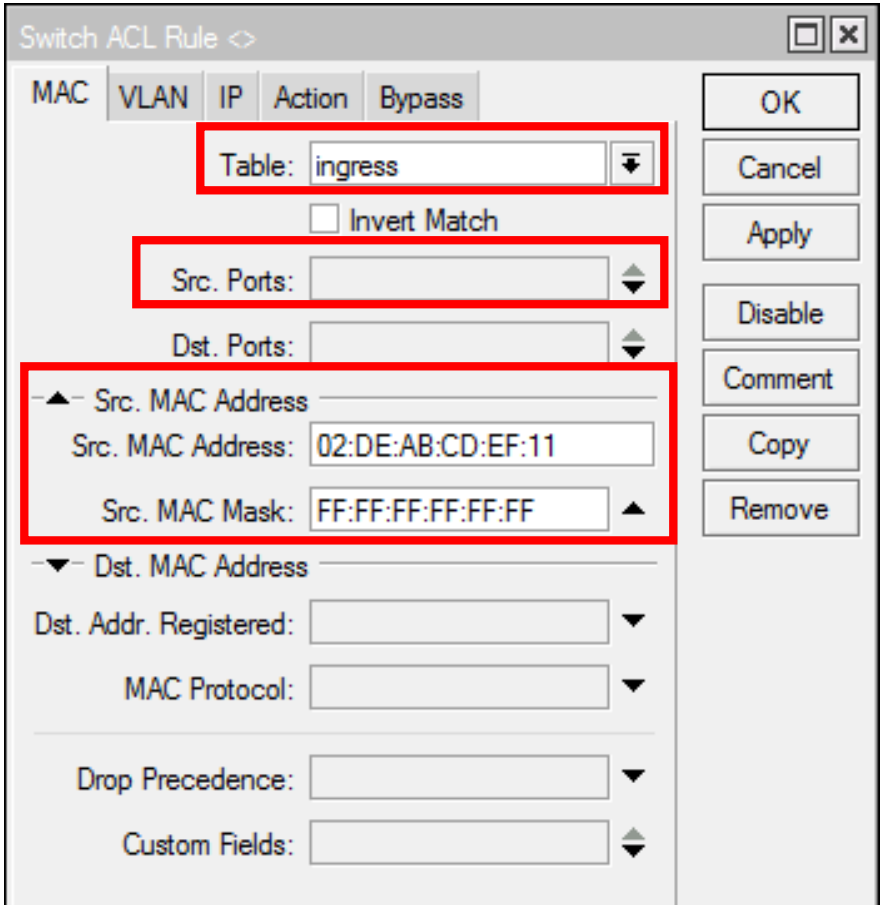
02:DE:AB:CD:EF:11 only at ether2:

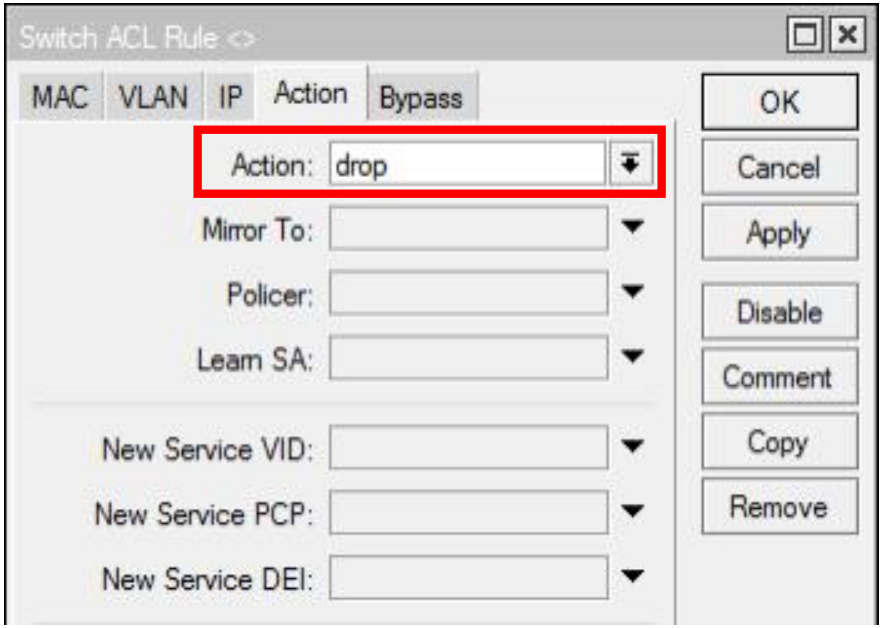




ACL – stay on your port!

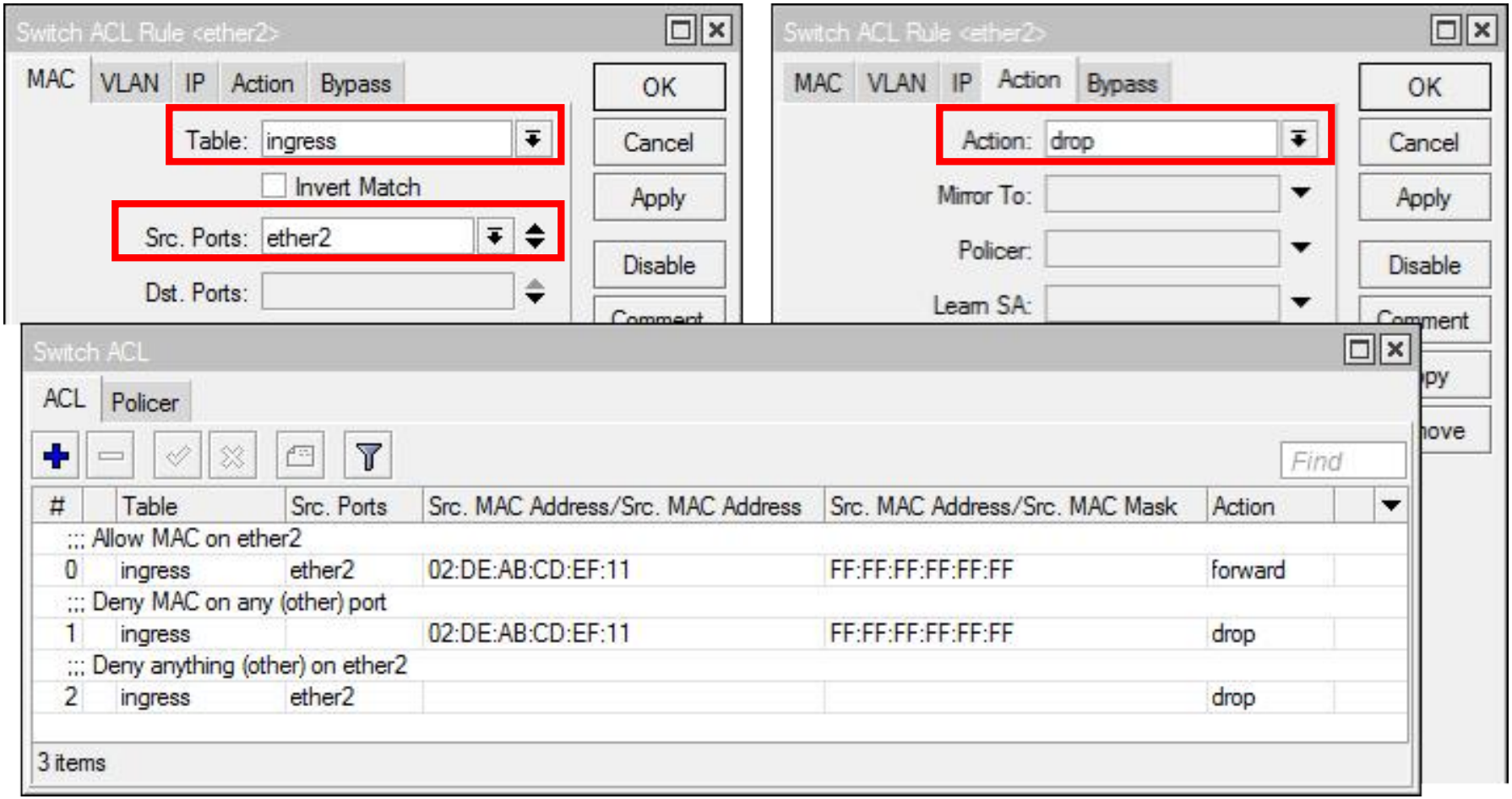
Deny 02:DE:AB:CD:EF:11 on other port:





ACL – stay on your port!

Drop anything (other) on ether2:

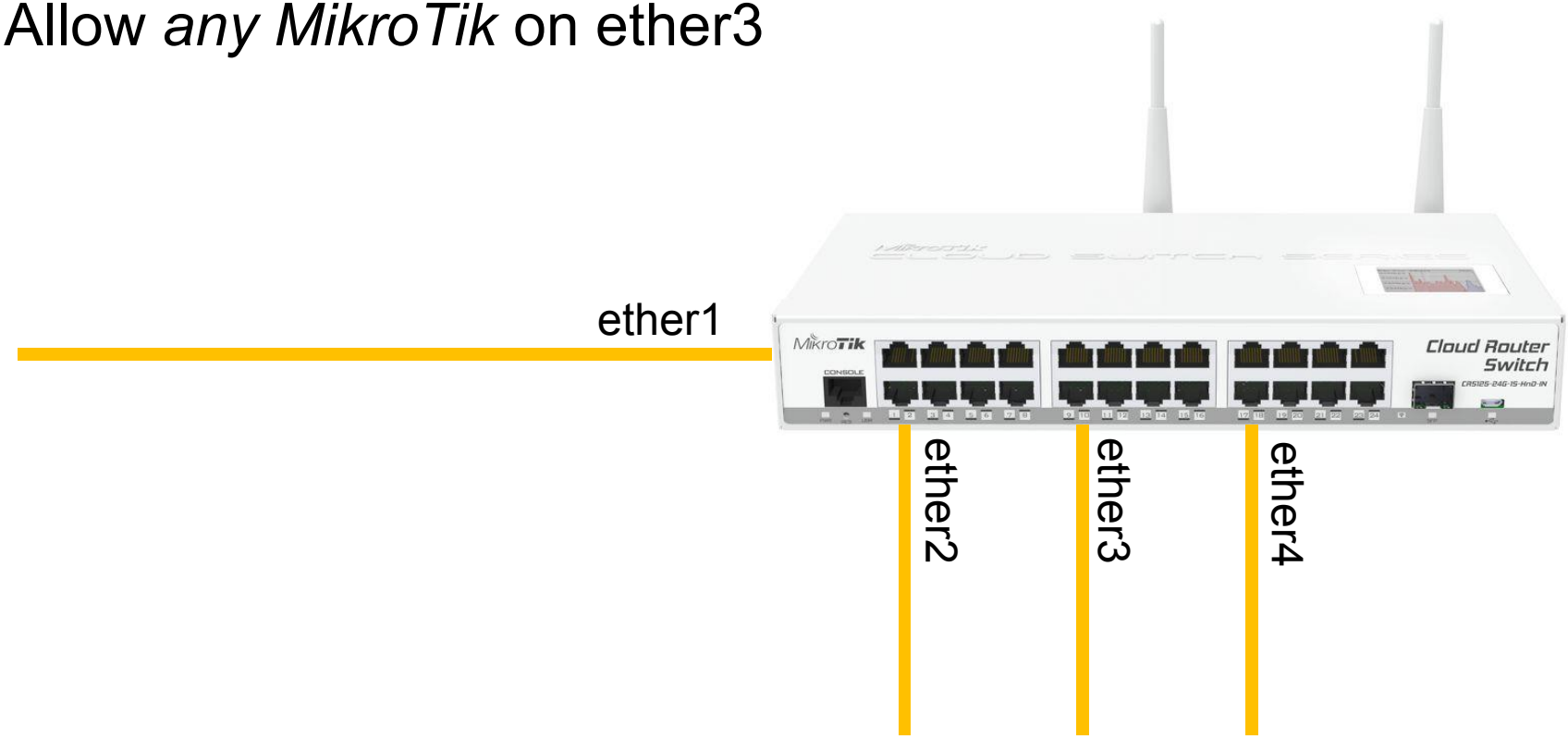


Reference

```
/interface ethernet
set ether2,ether3,ether4,ether5,ether6,ether7,ether8 master-port=ether1



# MAC 02:DE:AB:CD:EF:11 on ether2.

/interface ethernet switch acl

add table=ingress action=forward mac-src-address=02:DE:AB:CD:EF:11 \
        src-ports=ether2 comment="Allow MAC 02:DE:AB:CD:EF:11 on ether2"

add table=ingress action=drop mac-src-address=02:DE:AB:CD:EF:11 \
        comment="Deny MAC 02:DE:AB:CD:EF:11 on any (other) port"

add table=ingress action=drop src-ports=ether2 \
        comment="Deny anything (other) on ether2"
```

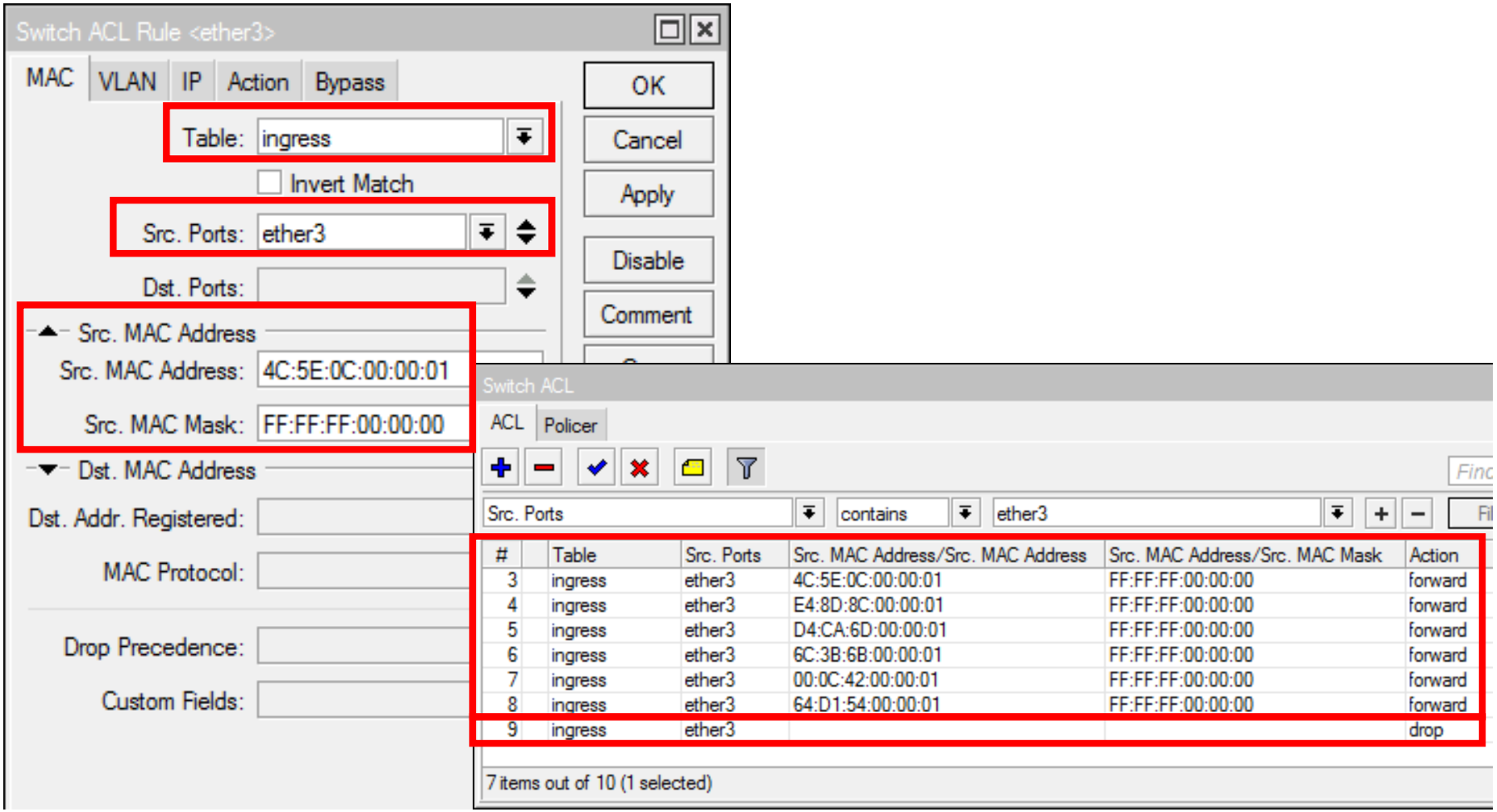
ACL – stay on your port!

- Bind 02:DE:AB:CD:EF:11 to ether2 (done)
- Allow *any MikroTik* on ether3



ACL – stay on your port!

Allow *any MikroTik* on ether3



Reference

```
# Allow any MikroTik on ether3 (table=ingress, action=forward)

/interface ethernet switch acl

add mac-src-address=4C:5E:0C:00:00:01/FF:FF:FF:00:00:00 src-ports=ether3
add mac-src-address=E4:8D:8C:00:00:01/FF:FF:FF:00:00:00 src-ports=ether3
add mac-src-address=D4:CA:6D:00:00:01/FF:FF:FF:00:00:00 src-ports=ether3
add mac-src-address=6C:3B:6B:00:00:01/FF:FF:FF:00:00:00 src-ports=ether3
add mac-src-address=00:0C:42:00:00:01/FF:FF:FF:00:00:00 src-ports=ether3
add mac-src-address=64:D1:54:00:00:01/FF:FF:FF:00:00:00 src-ports=ether3

add action=drop src-ports=ether3
```

ACL – stay on your port!

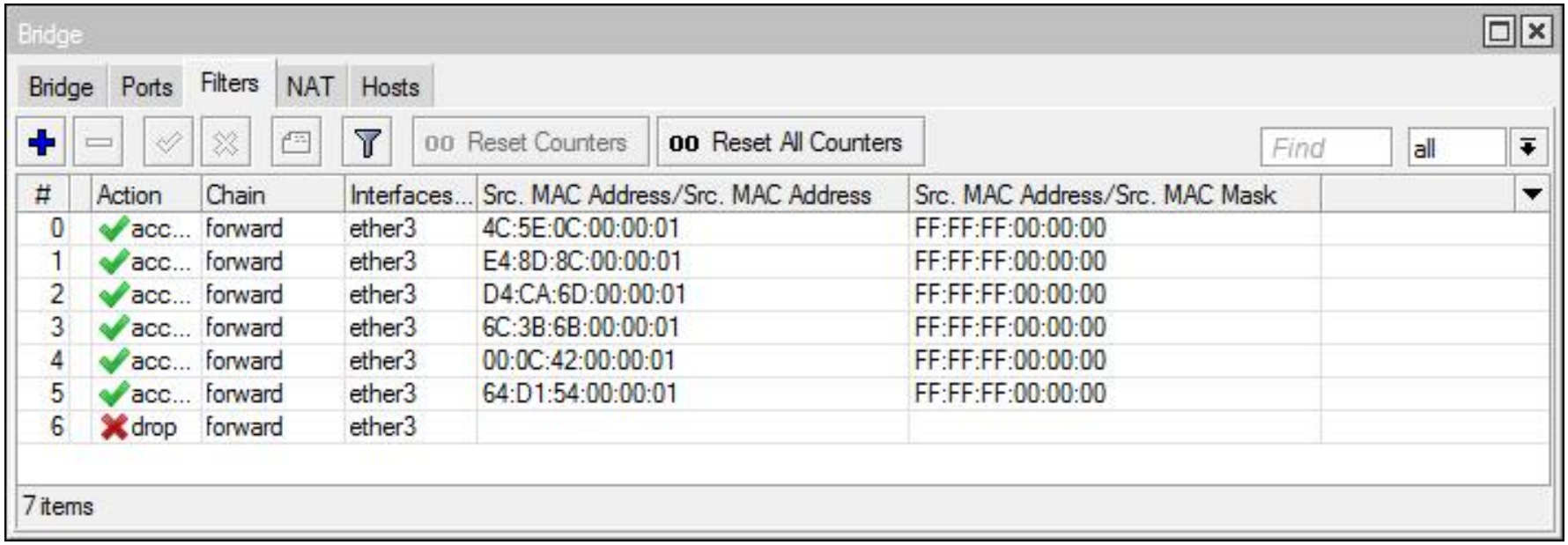
Caveat:

Default drop

```
/interface ethernet switch acl add action=drop
```

will disconnect you even on non-switch-chip-ports

ACL vs. Bridge filter



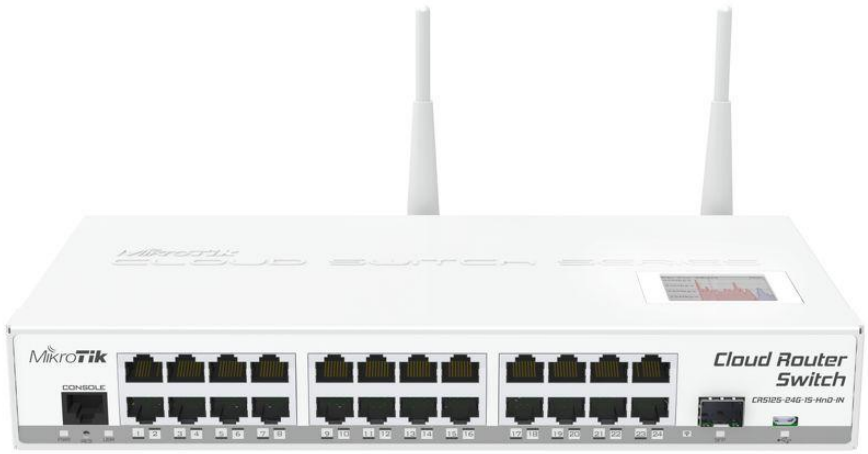
Mode (CRS125-24G-1S)	Configuration	Mbps (1518 bytes)
Switching	Non blocking Layer 2 throughput	24,674.9
Bridging	25 bridge filter rules	983.7

ACL = wirespeed

ACL

Many common setups possible without ACL

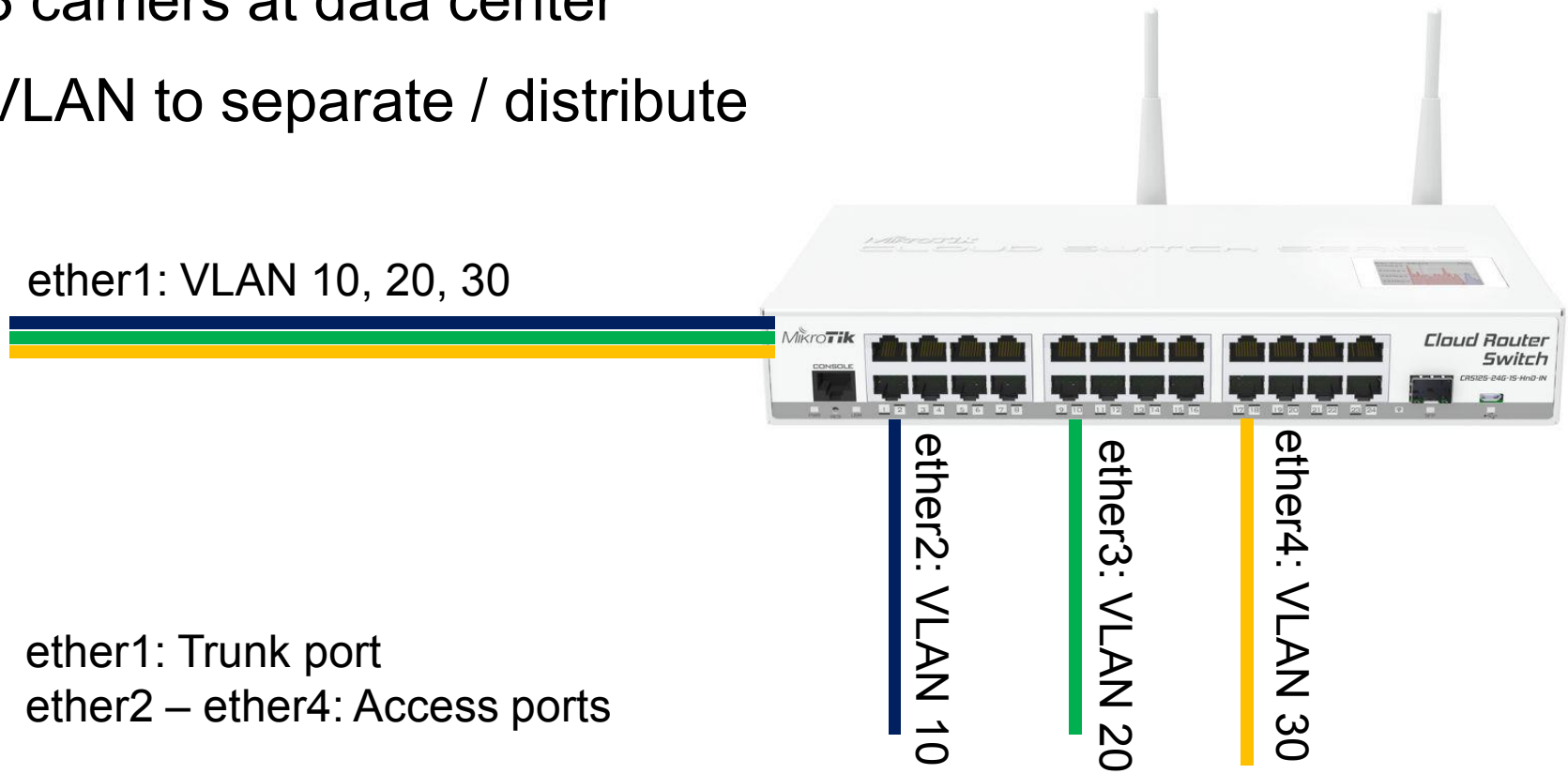
Model	Switch Chip	Access Control List
CRS125-24G-1S	QCA-8513L	No
CRS125-24G-1S-2HnD	QCA-8513L	No
CRS109-8G-1S-2HnD	QCA-8513L	No



Multiple possible setups

Split your trunk

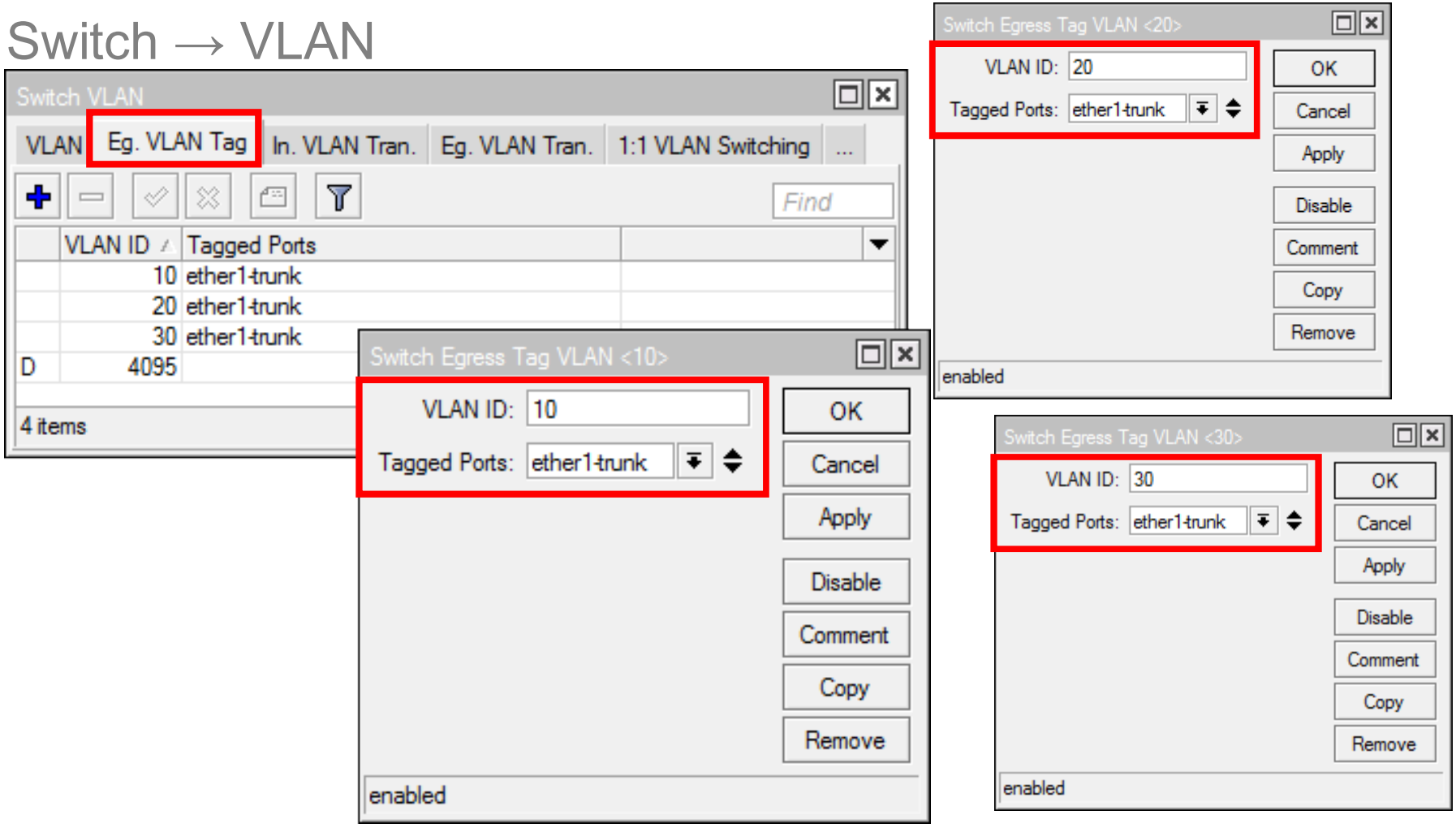
- One uplink to data center
- 3 carriers at data center
- VLAN to separate / distribute



ether1: Trunk port
ether2 – ether4: Access ports

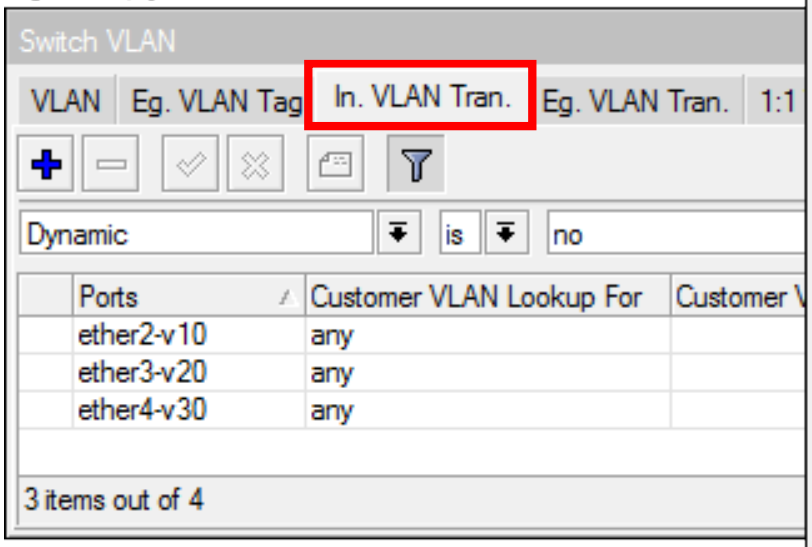
Egress, outgoing to trunk port

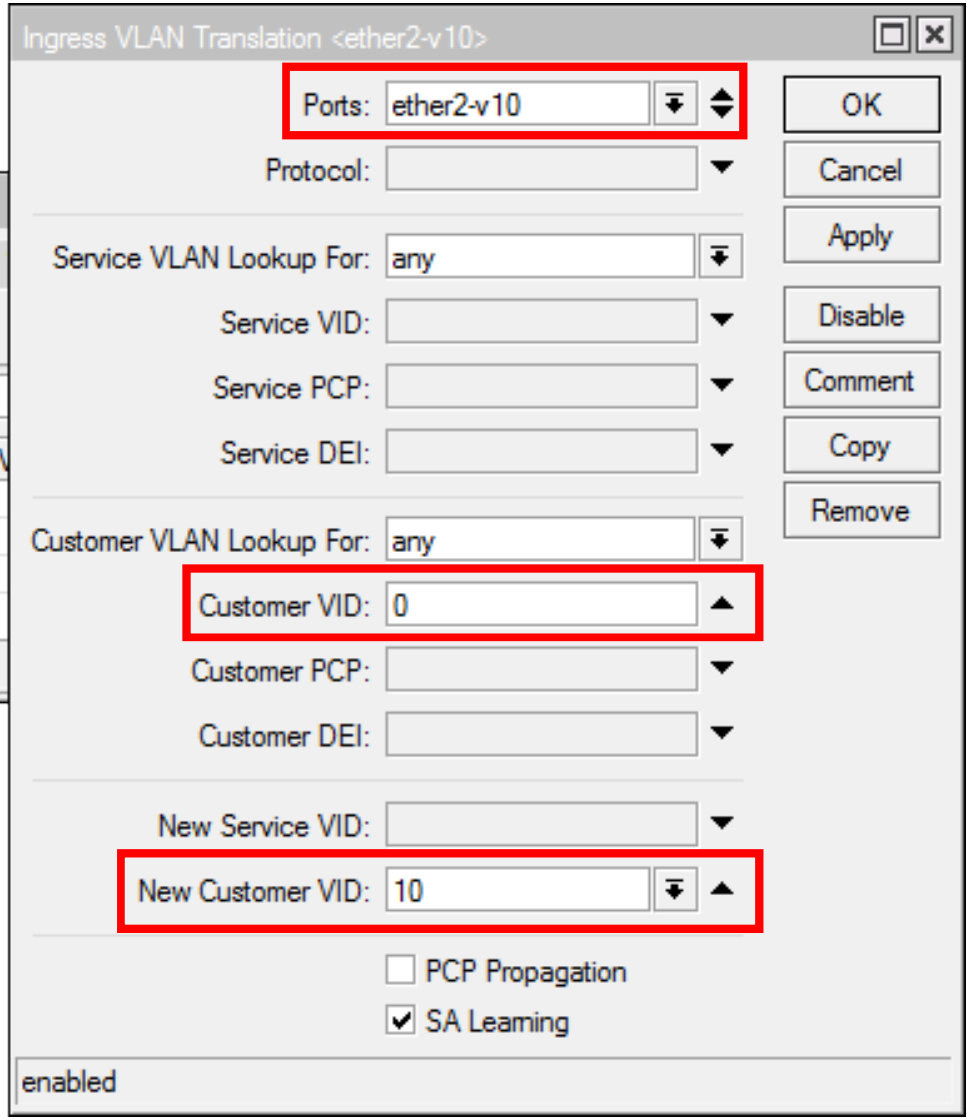
Switch → VLAN



Ingress, incomming from access port

Switch → VLAN



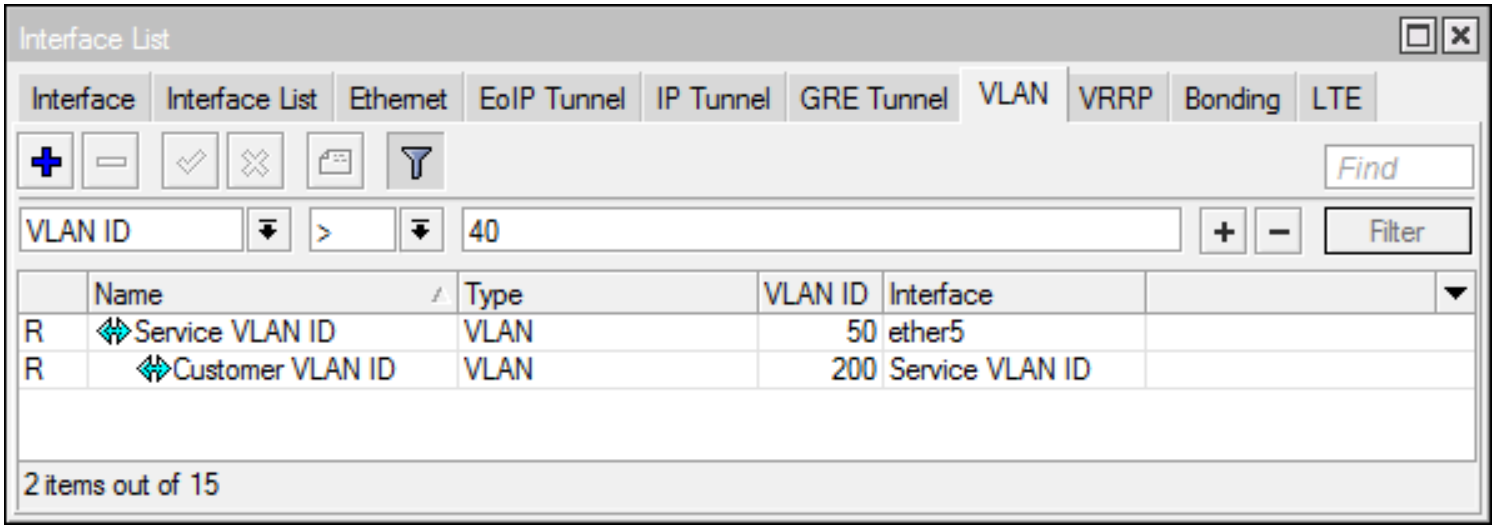


Reference

```
# Create switch
/interface ethernet
set [ find default-name=ether1 ] name=ether1-trunk
set [ find default-name=ether2 ] master-port=ether1-trunk name=ether2-v10
set [ find default-name=ether3 ] master-port=ether1-trunk name=ether3-v20
set [ find default-name=ether4 ] master-port=ether1-trunk name=ether4-v30

# Assign VLANs to trunk port
/interface ethernet switch egress-vlan-tag
add tagged-ports=ether1-trunk vlan-id=10
add tagged-ports=ether1-trunk vlan-id=20
add tagged-ports=ether1-trunk vlan-id=30

# Translate untagged traffic to specified VLAN
/interface ethernet switch ingress-vlan-translation
add customer-vid=0 new-customer-vid=10 ports=ether2-v10
add customer-vid=0 new-customer-vid=20 ports=ether3-v20
add customer-vid=0 new-customer-vid=30 ports=ether4-v30


# # to be continued…
```

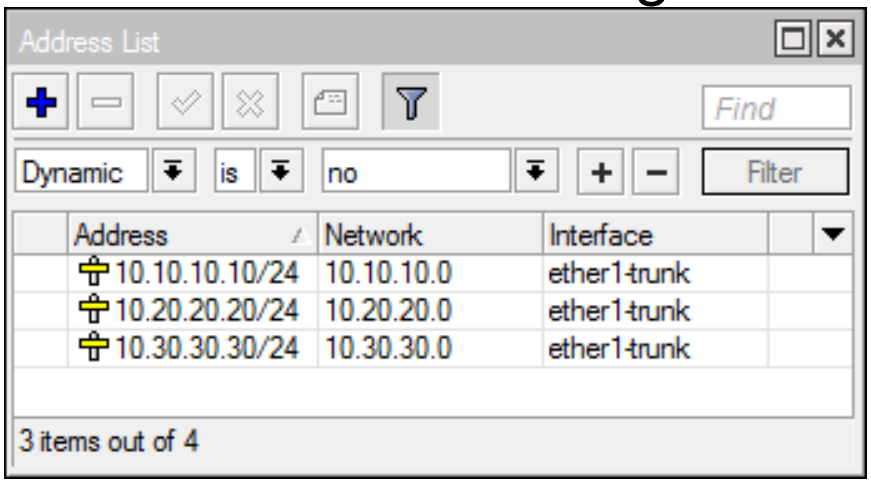
Reference

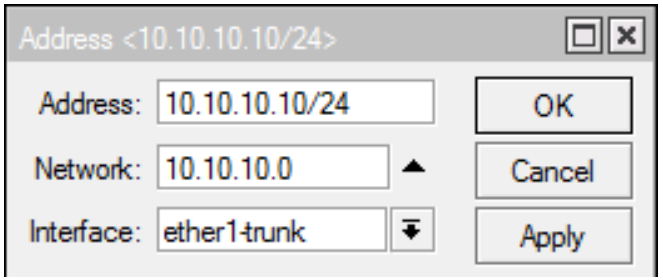
```
# CVID = Customer VLAN ID = inner VLAN tag id of the IEEE 802.1ad frame

# SVID = Service VLAN ID = outer VLAN tag id of the IEEE 802.1ad frame
```

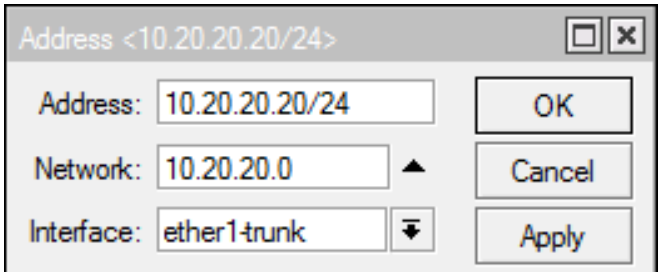


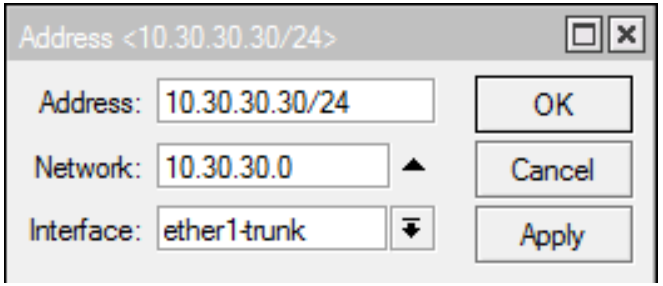
Split your trunk

Done! Wait… IP management?







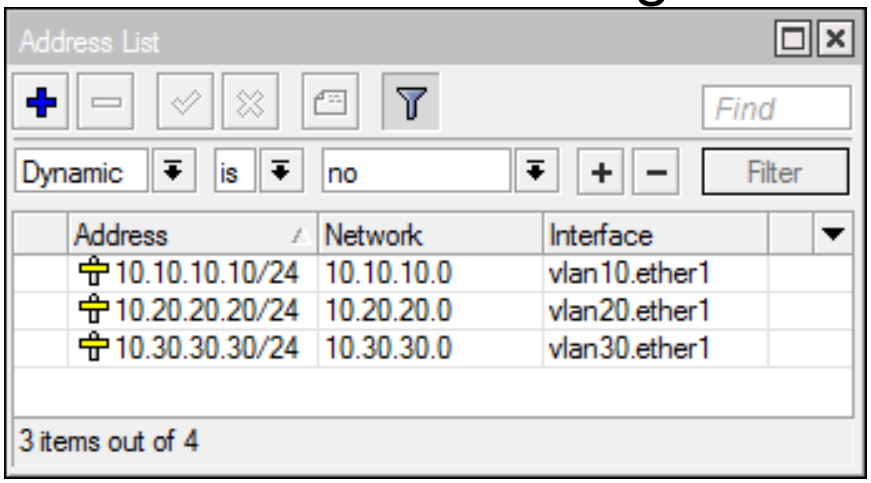


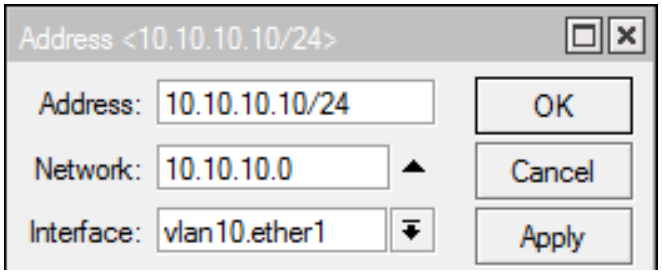
IP reachable from access port side.

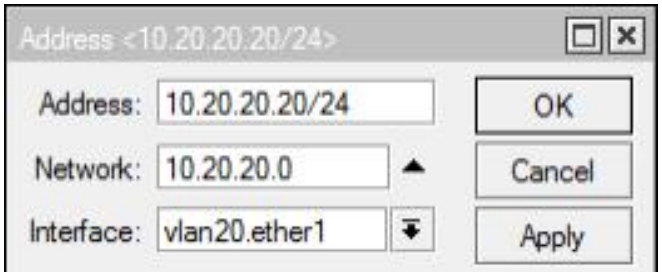
Not from trunk port side!

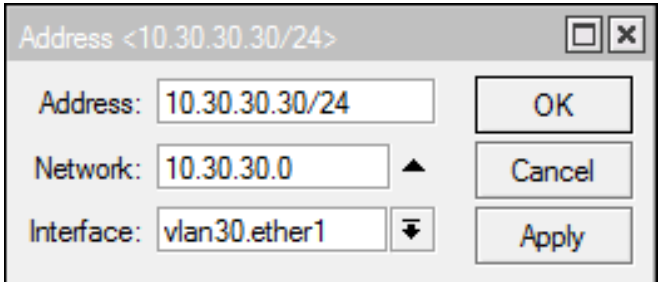
Split your trunk

Done! Wait… IP management?









IP **not reachable** from access port side
Not from trunk port side

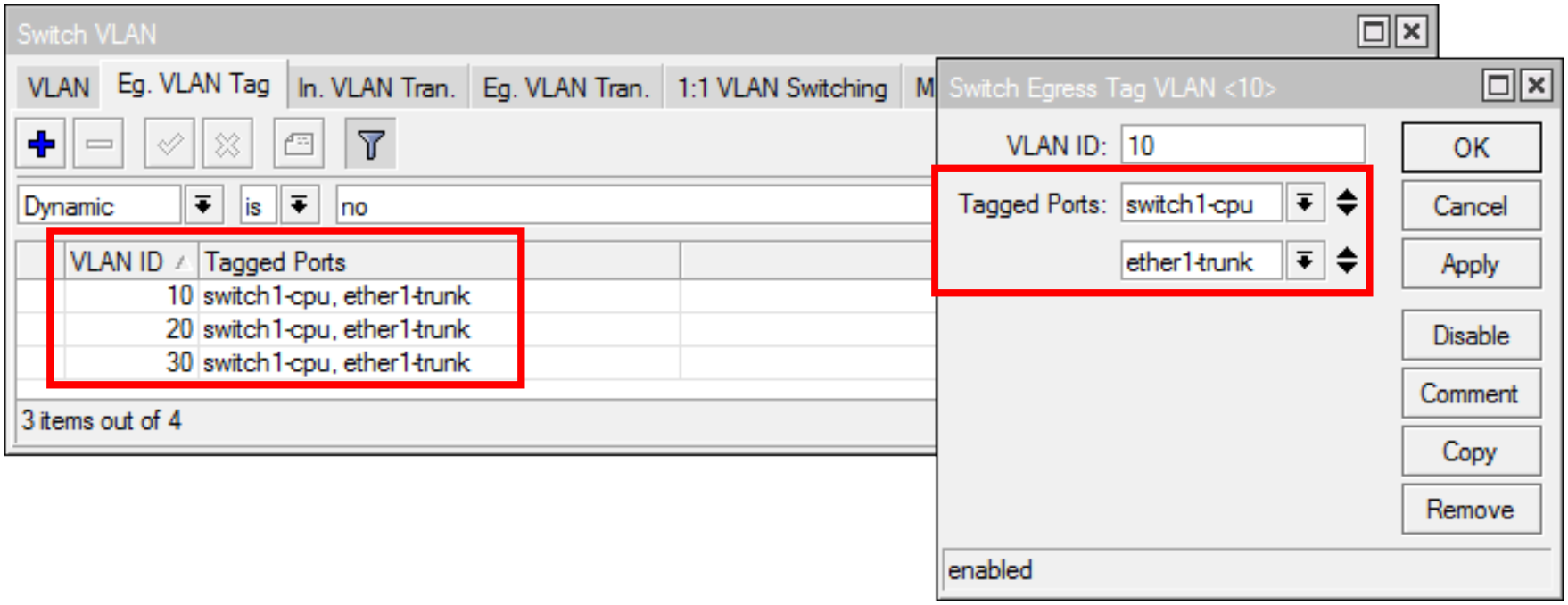
Switch (chip) does not know about VLAN / IP config (RouterOS part)

Management IP

Add “switch1-cpu” to switch egress-vlan-tag:

Understanding of VLAN tags also from CPU-port (RouterOS).

No performance issue



Reference

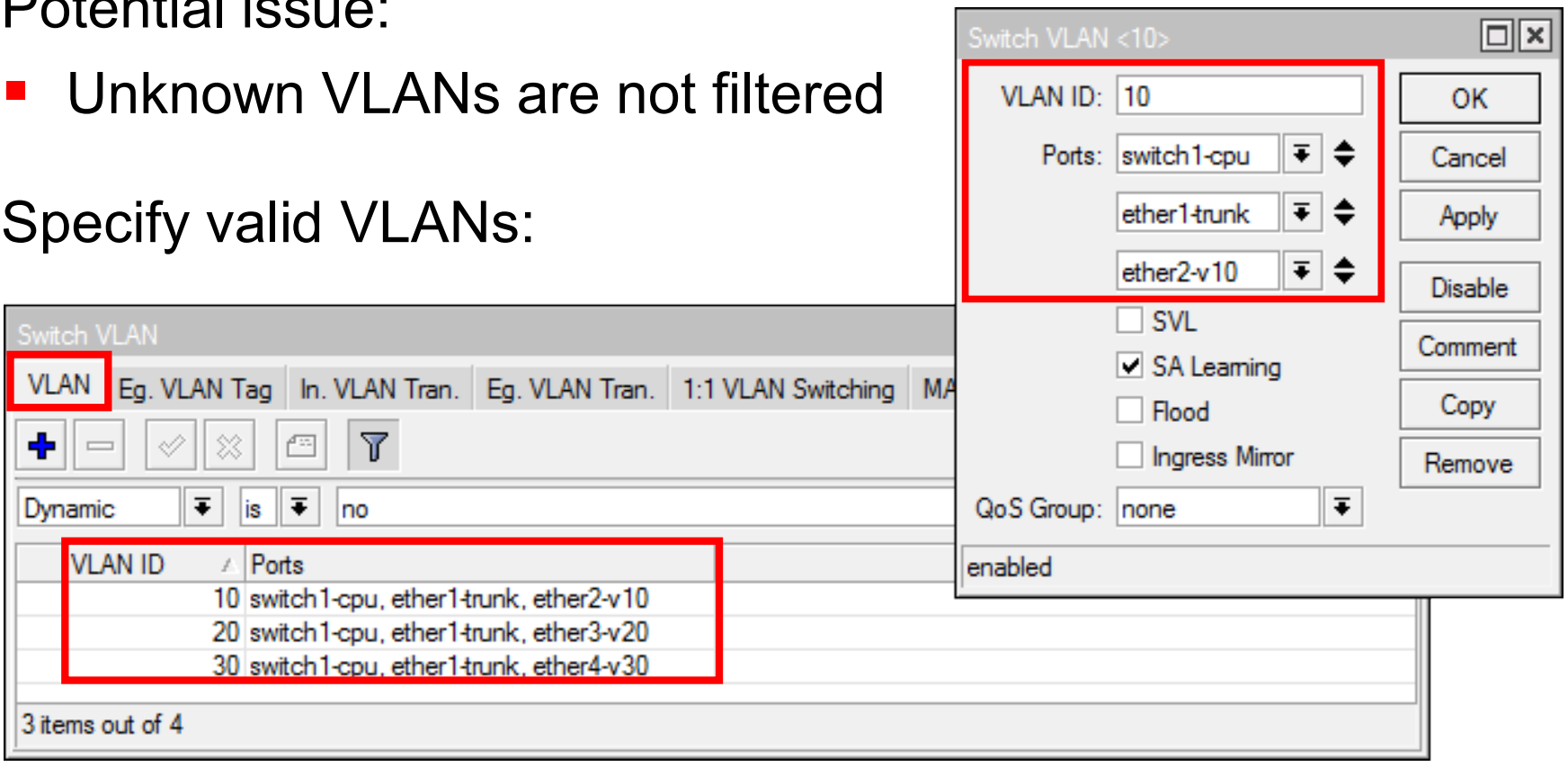
```
# Split your trunk, part 2

/interface vlan
add interface=ether1-trunk name=vlan10.ether1 vlan-id=10
add interface=ether1-trunk name=vlan20.ether1 vlan-id=20
add interface=ether1-trunk name=vlan30.ether1 vlan-id=30

/ip address
add address=10.20.20.20/24 interface=vlan20.ether1
add address=10.10.10.10/24 interface=vlan10.ether1
add address=10.30.30.30/24 interface=vlan30.ether1

/interface ethernet switch egress-vlan-tag
add tagged-ports=ether1-trunk,switch1-cpu vlan-id=10
add tagged-ports=ether1-trunk,switch1-cpu vlan-id=20
add tagged-ports=ether1-trunk,switch1-cpu vlan-id=30
```

Unknown VLANs

Potential issue:

- Unknown VLANs are not filtered

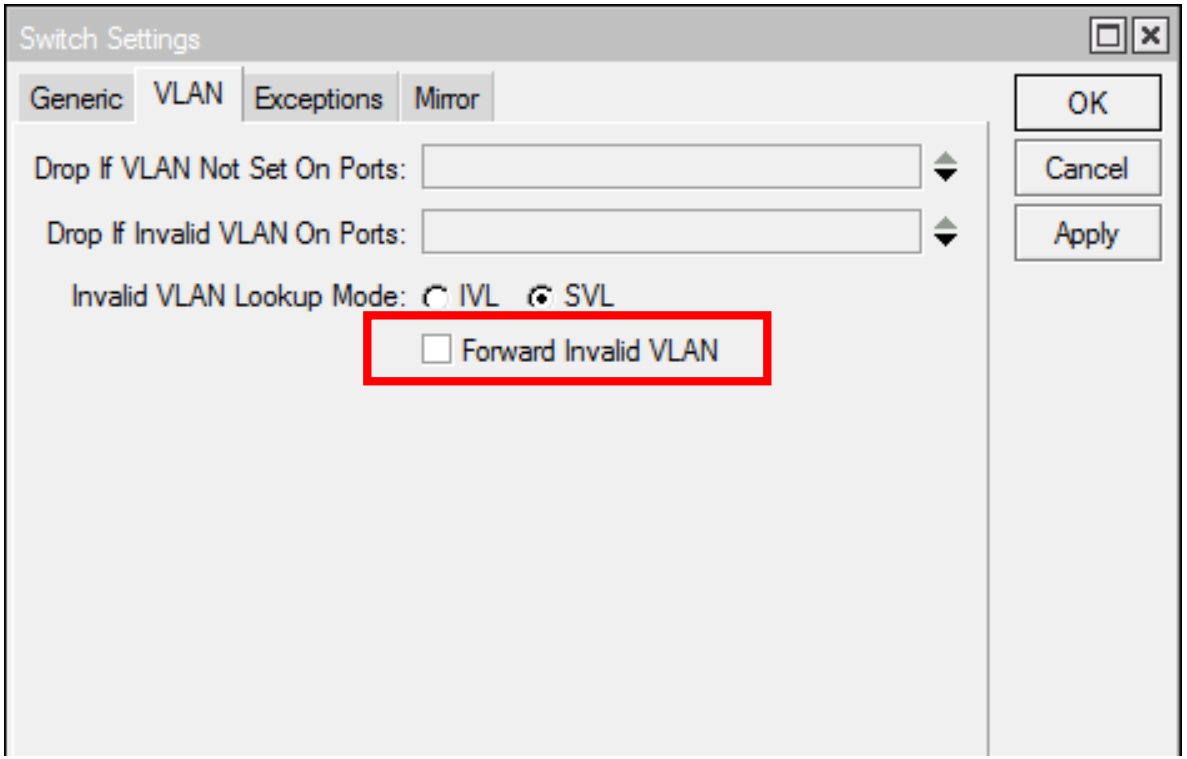
Specify valid VLANs:



Switch → VLAN

Unknown VLANs

Disable forwarding for unspecified VLANs



Switch → Settings → VLAN

Reference

```
# Define (all) valid VLANs
/interface ethernet switch vlan
add ports=switch1-cpu,ether2-v10,ether1-trunk vlan-id=10
add ports=switch1-cpu,ether3-v20,ether1-trunk vlan-id=20
add ports=switch1-cpu,ether4-v30,ether1-trunk vlan-id=30

# Disable forwarding of unknown VLANs
/interface ethernet switch set forward-unknown-vlan=no
```

Note

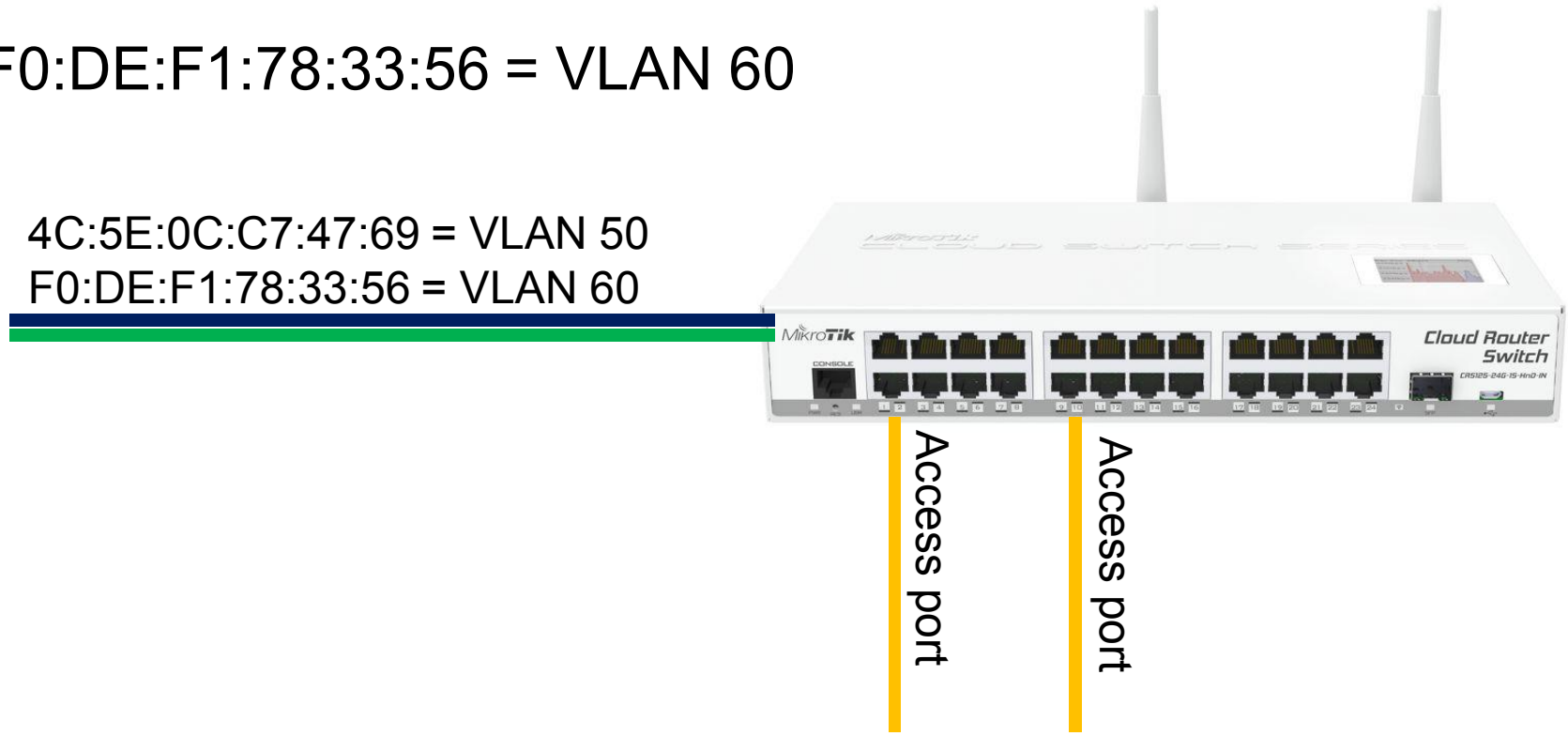
```
# # Be careful: forward-unknown-vlan=no -> define all used VLANs on that device
# #
# # Switch 1: No VLAN
# # Switch 2: Only VLAN 10
# #
# # Use:
#
# /interface ethernet switch vlan
# add ports=ether5-sw1,ether6-sw1 vlan-id=0
# add ports=ether7-sw2,ether8-sw2 vlan-id=10
# /interface ethernet switch set forward-unknown-vlan=no
```

Dynamic VLAN definitions

MAC based dynamic VLAN

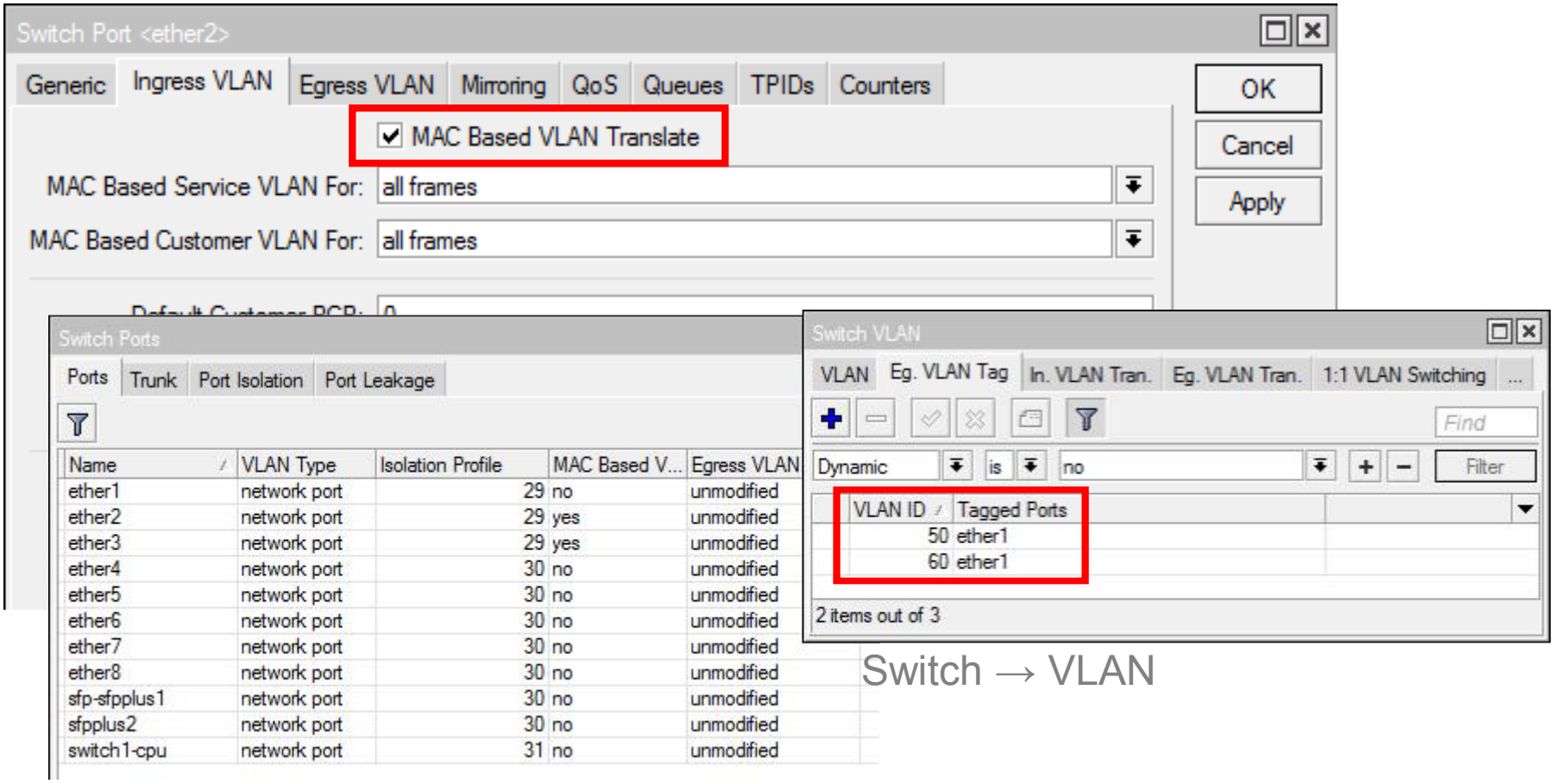
VLAN definition, based on MAC address

- 4C:5E:0C:C7:47:69 = VLAN 50
- F0:DE:F1:78:33:56 = VLAN 60



MAC based dynamic VLAN

Switch → Ports → Ports → ether*X*

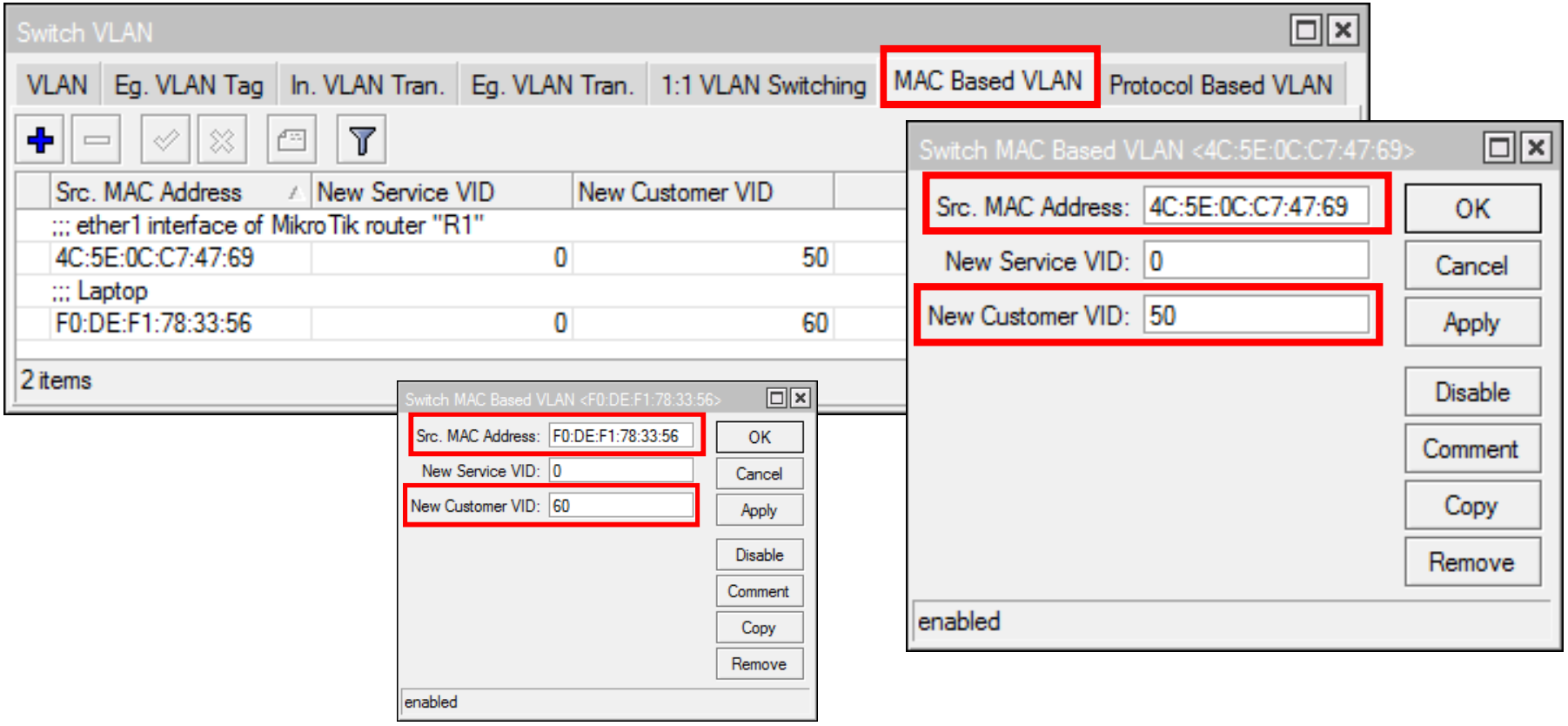


Switch → VLAN

MAC based dynamic VLAN

Specific MAC address required (no mask)

Switch → VLAN → MAC Based VLAN



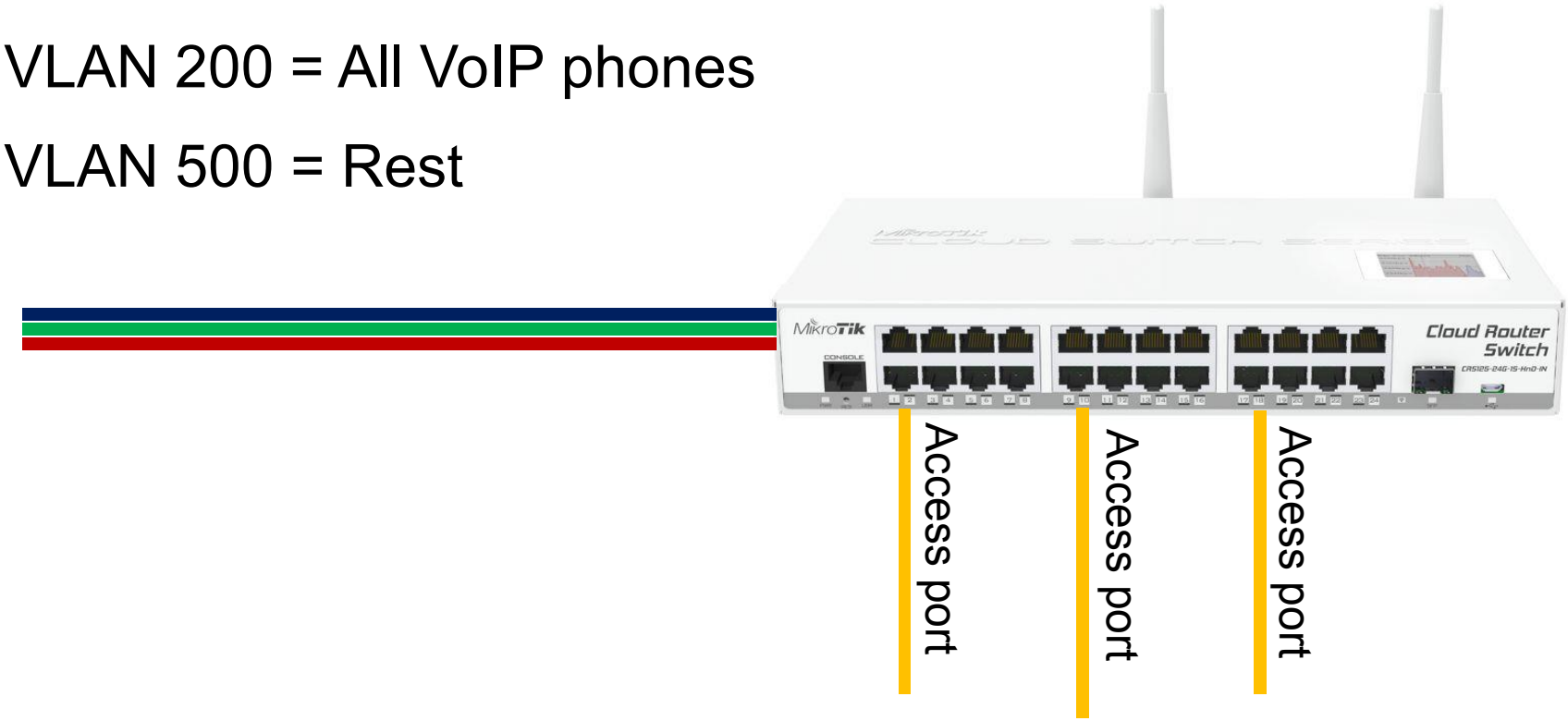
Reference

```
# Create switch
/interface ethernet
set ether2 master-port=ether1
set ether3 master-port=ether1

# Define trunk port
/interface ethernet switch egress-vlan-tag
add tagged-ports=ether1 vlan-id=50
add tagged-ports=ether1 vlan-id=60

# enable MAC based VLAN translation
/interface ethernet switch port
set ether2 allow-fdb-based-vlan-translate=yes
set ether3 allow-fdb-based-vlan-translate=yes

# Assign VLANs to MAC addresses
/interface ethernet switch mac-based-vlan
add src-mac=4c:5E:0c:C7:47:69 new-customer-vid=50
add src-mac=F0:DE:F1:78:33:56 new-customer-vid=60
```

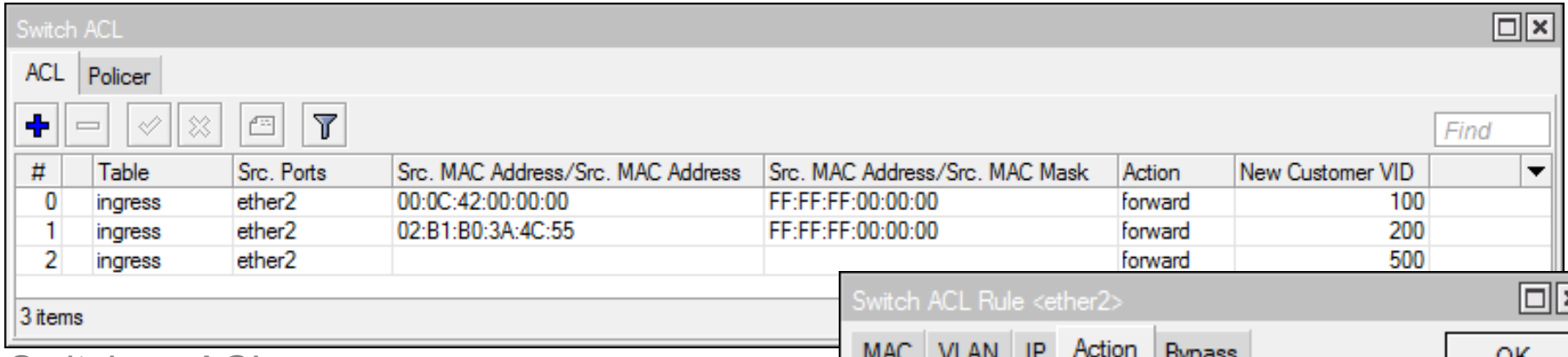
MAC based dynamic VLAN (ACL)

VLAN definition, based on MAC address

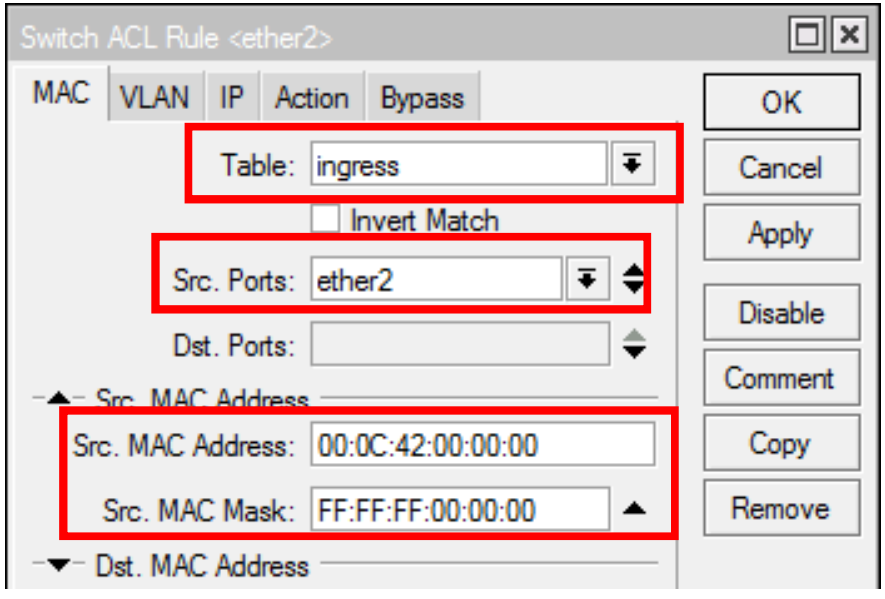
- VLAN 100 = MikroTik devices
- VLAN 200 = All VoIP phones
- VLAN 500 = Rest

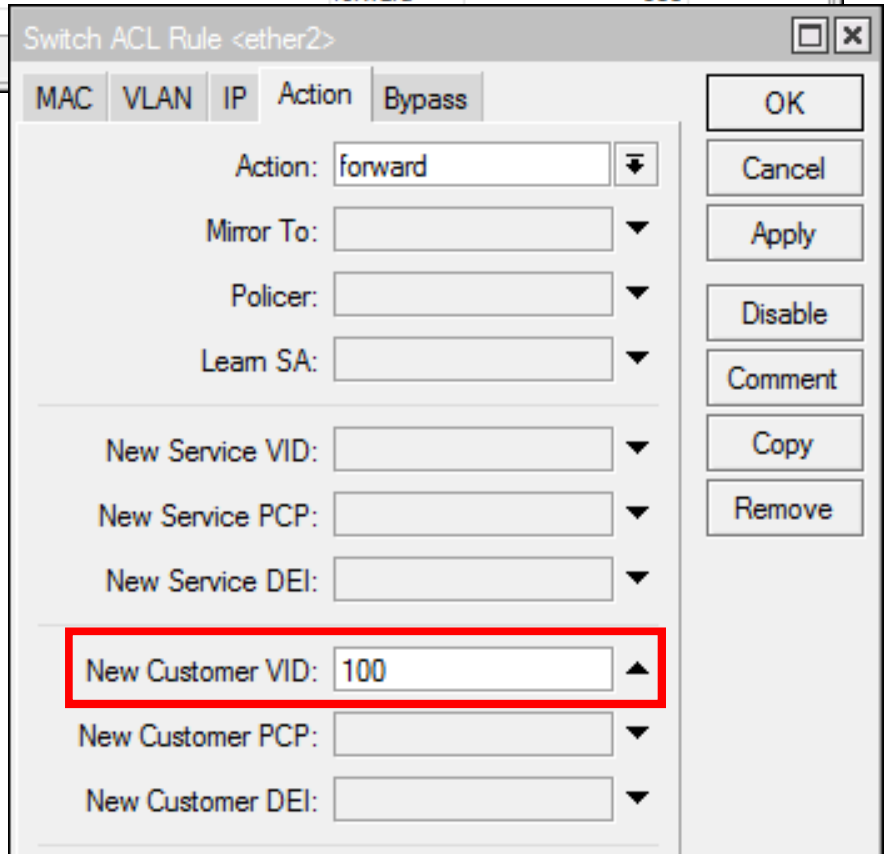


MAC based dynamic VLAN (ACL)



Switch → ACL





Reference

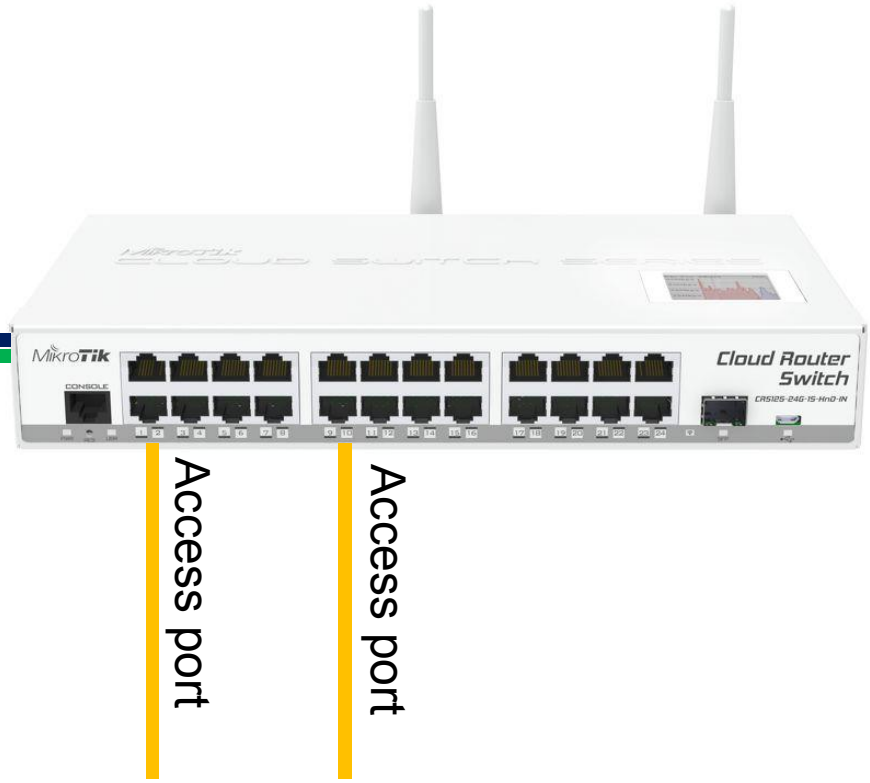
```
/interface ethernet
set [ find default-name=ether2 ] master-port=ether1

/interface ethernet switch acl
# MikroTik devices with MAC 00:0C:42:*:*:* -> VLAN 100
add table=ingress action=forward \
        mac-src-address=00:0C:42:00:00:00/FF:FF:FF:00:00:00 \
        new-customer-vid=100 src-ports=ether2
# VoIP phones with MAC 02:B1:B0:*:*:* -> VLAN 200
add table=ingress action=forward \
        mac-src-address=02:B1:B0:3A:4C:55/FF:FF:FF:00:00:00 \
        new-customer-vid=200 src-ports=ether2
# Rest -> VLAN 500
add table=ingress action=forward new-customer-vid=500 src-ports=ether2
```

Protocol based dynamic VLAN

VLAN definition, based on protocol

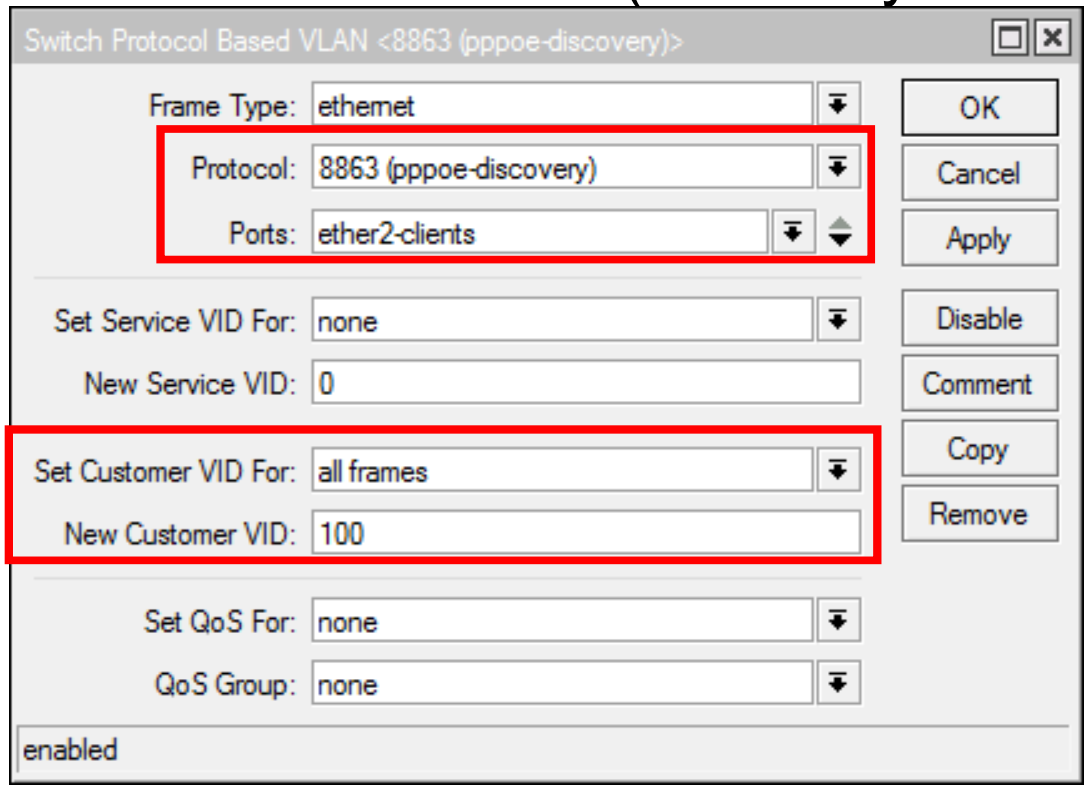
- PPPoE = VLAN 100
- IP = VLAN 200

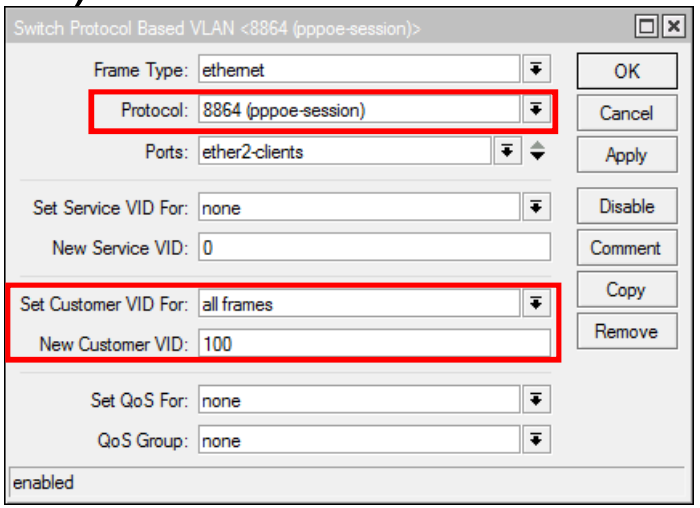
VLAN 100 = PPPoE
VLAN 200 = IP-traffic



Protocol based dynamic VLAN

Set VLAN for PPPoE (discovery & session)

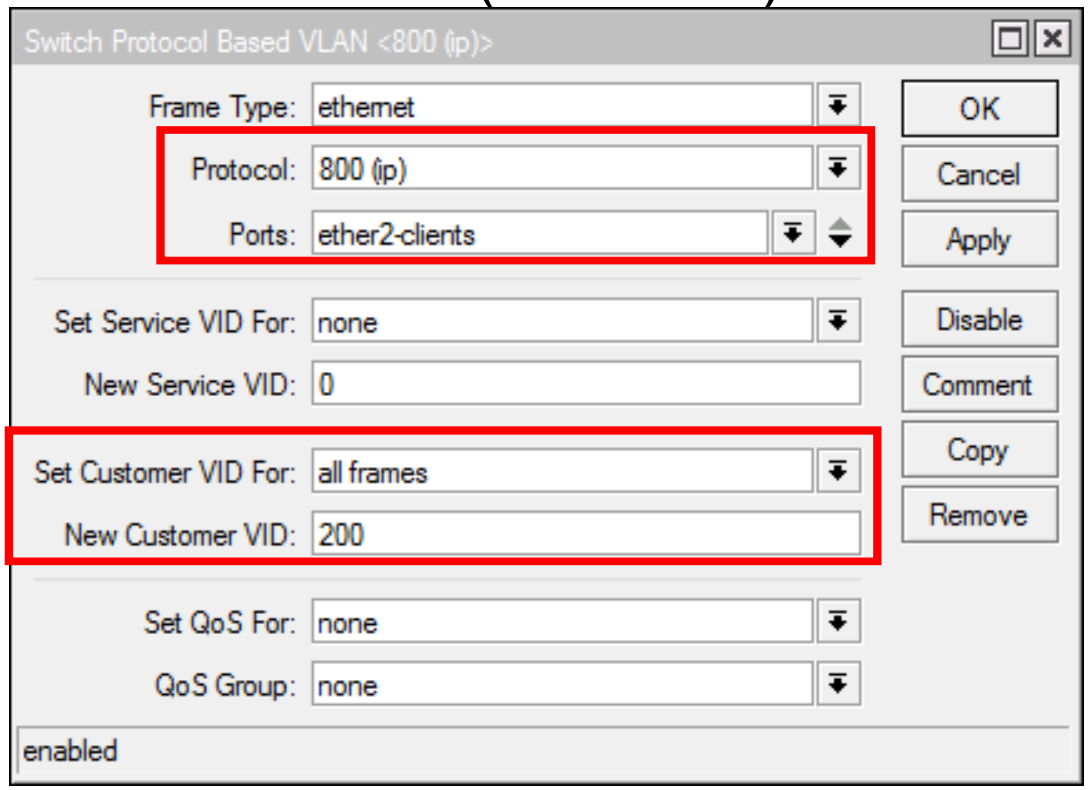


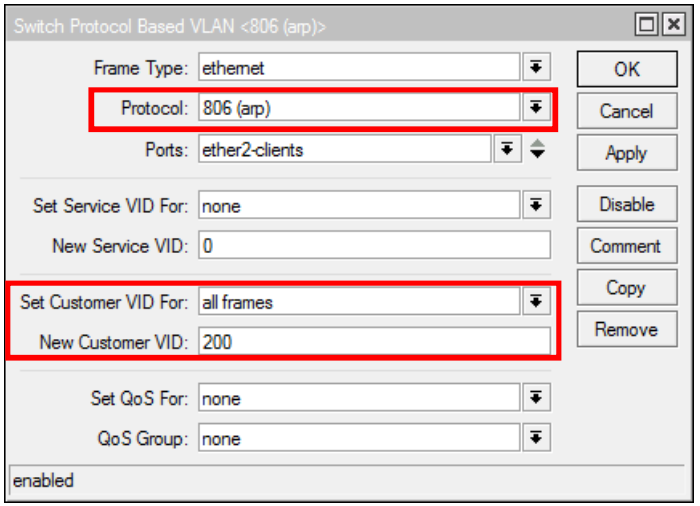


Switch → VLAN → Protocol Based VLAN

Protocol based dynamic VLAN

Set VLAN for IP (IP & ARP)

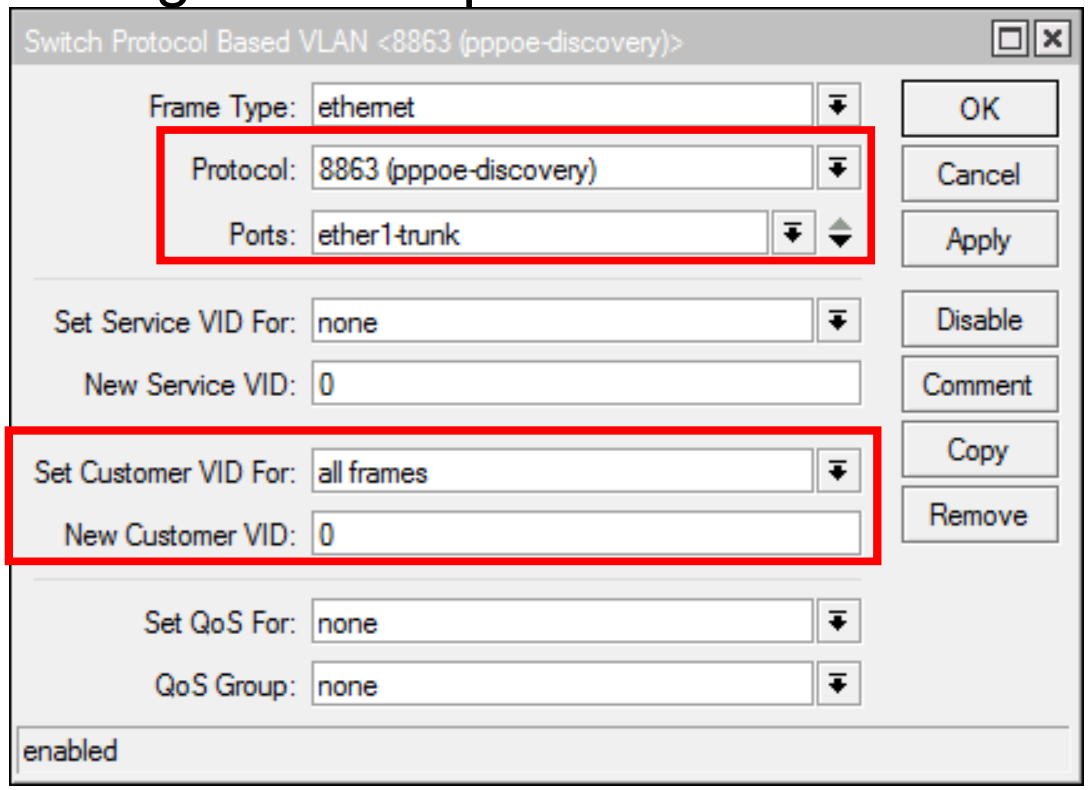




Switch → VLAN → Protocol Based VLAN

Protocol based dynamic VLAN

Configure trunk port



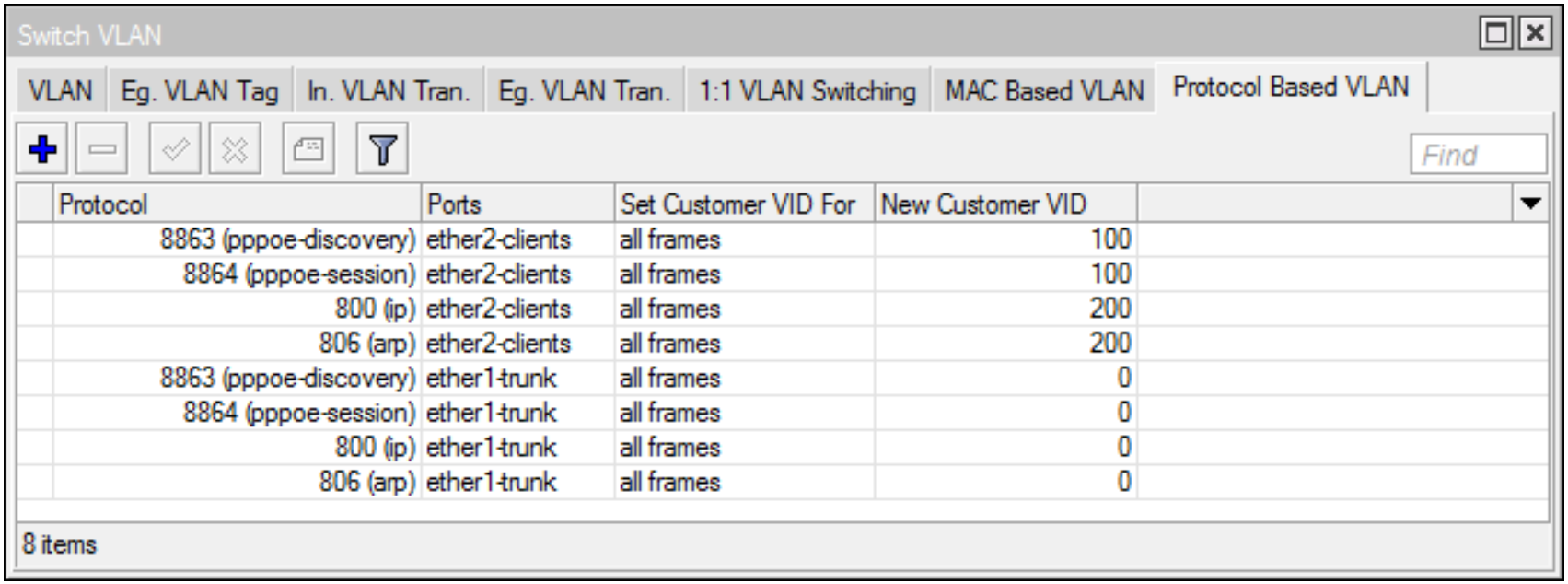
Switch → VLAN → Protocol Based VLAN

Protocol

- pppoe-discovery
- pppoe-session
- ip
- arp

Protocol based dynamic VLAN

Switch → VLAN



Reference

```
/interface ethernet
set [ find default-name=ether1 ] name=ether1-trunk
set [ find default-name=ether2 ] master-port=ether1-trunk name=ether2-clients

/interface ethernet switch protocol-based-vlan

add ports=ether2-clients protocol=pppoe-discovery set-customer-vid-for=all \
        new-customer-vid=100 set-service-vid-for=none
add ports=ether2-clients protocol=pppoe set-customer-vid-for=all \
        new-customer-vid=100 set-service-vid-for=none
add ports=ether2-clients protocol=ip set-customer-vid-for=all \
        new-customer-vid=200 set-service-vid-for=none
add ports=ether2-clients protocol=arp set-customer-vid-for=all \
        new-customer-vid=200 set-service-vid-for=none

add ports=ether1-trunk protocol=pppoe-discovery set-customer-vid-for=all \
        new-customer-vid=0 set-service-vid-for=none
add ports=ether1-trunk protocol=pppoe set-customer-vid-for=all \
        new-customer-vid=0 set-service-vid-for=none
add ports=ether1-trunk protocol=ip set-customer-vid-for=all \
        new-customer-vid=0 set-service-vid-for=none
add ports=ether1-trunk protocol=arp set-customer-vid-for=all \
        new-customer-vid=0 set-service-vid-for=none
```

Dynamic VLAN

Remember the question about bridge or switch?

- How do you bridge *some packets* from one interface with a VLAN interface?

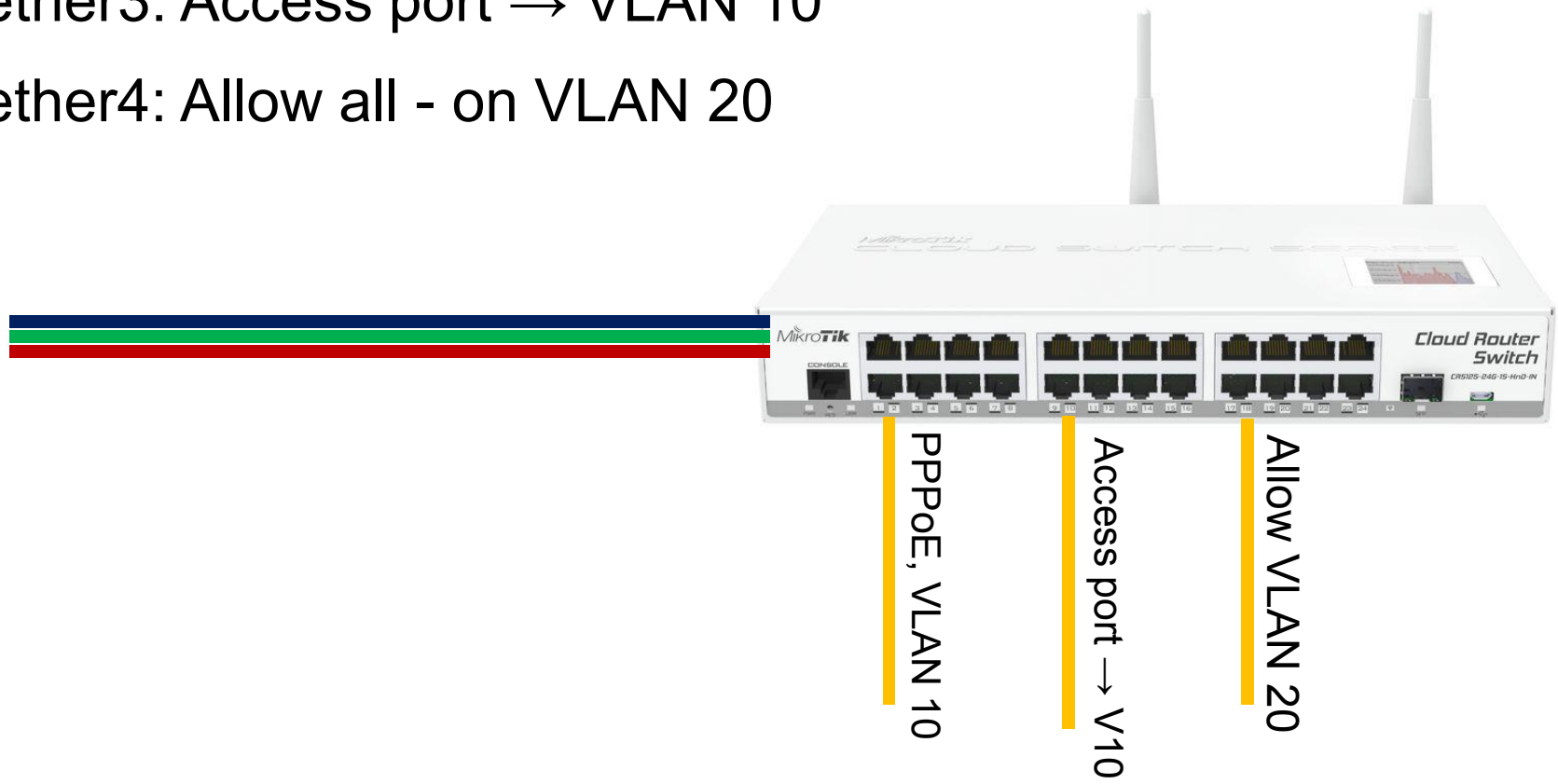
Note: Protocol based VLAN and MAC based VLAN

- CRS switch chip: Yes
- RB switch chip: No

Advanced traffic control (ACL)

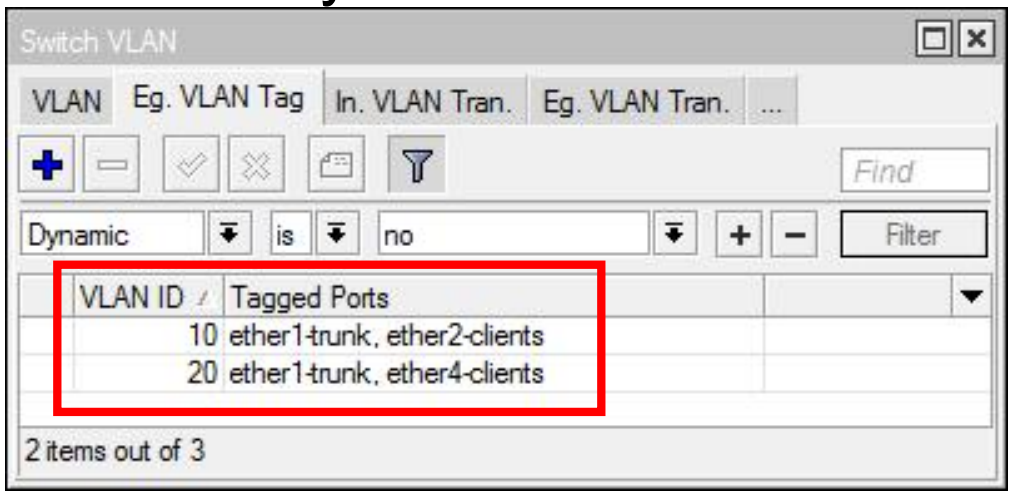
Advanced traffic control (ACL)

- ether2: Only PPPoE with VLAN 10
- ether3: Access port → VLAN 10
- ether4: Allow all - on VLAN 20



Advanced traffic control (ACL)

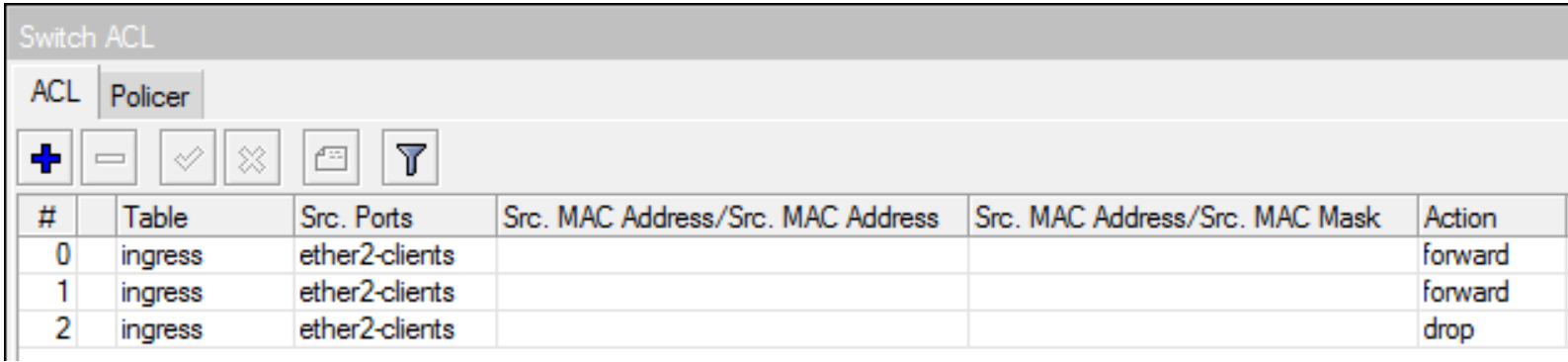
ether2: Only PPPoE with VLAN 10



Switch → VLAN

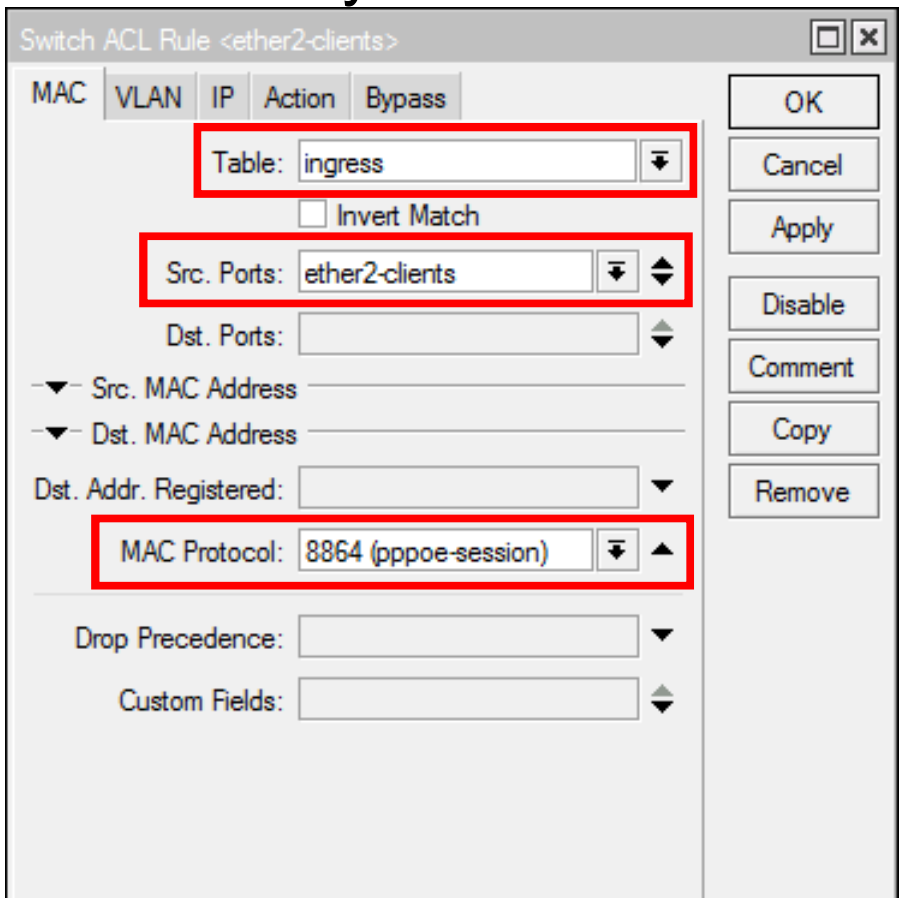
- Define egress: VLAN 10, VLAN 20 (for ether4)
- 3 ACL rules
  - 2x fwd pppoe
  - drop other

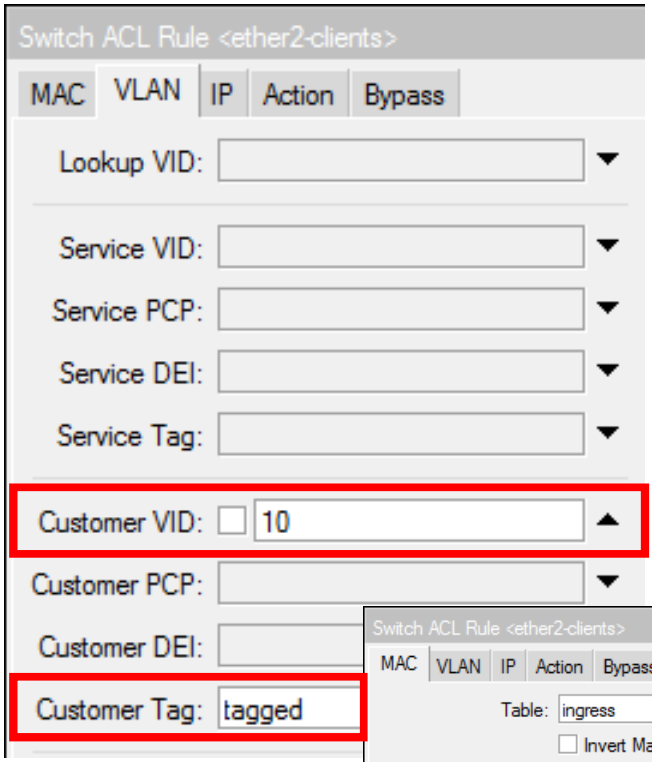
Switch → ACL



Advanced traffic control (ACL)

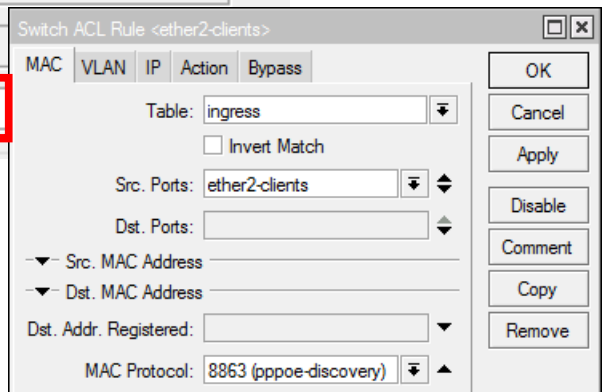
ether2: Only PPPoE with VLAN 10





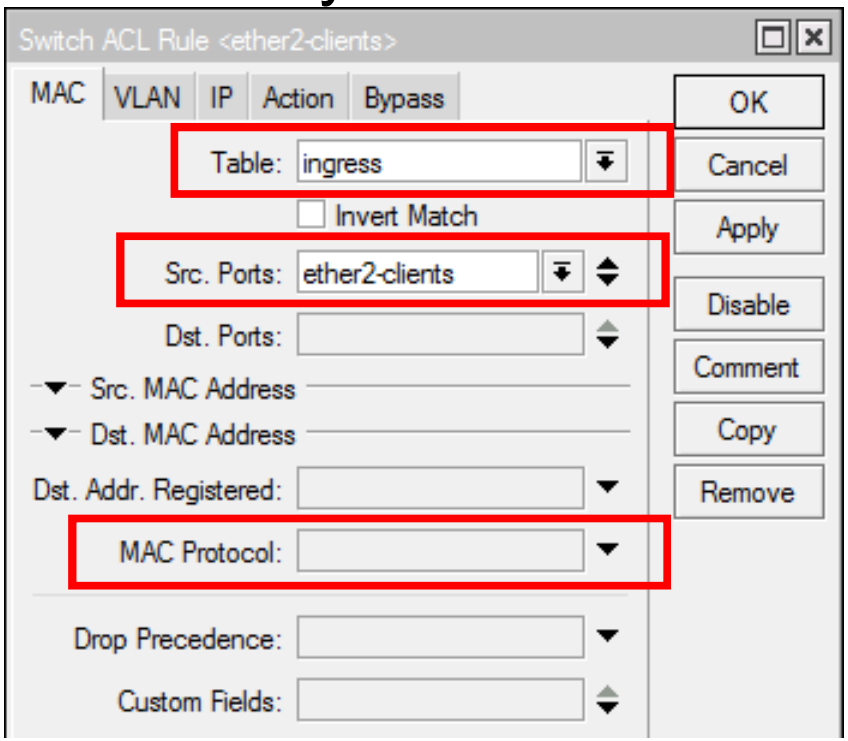
Switch → ACL

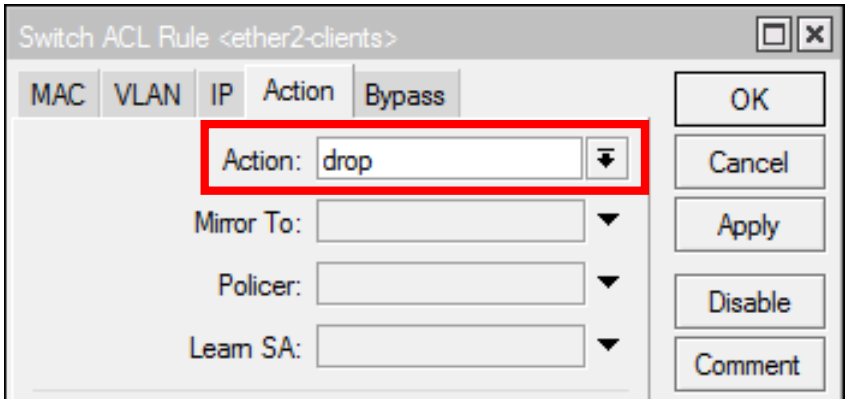
Action: forward



Advanced traffic control (ACL)

ether2: Only PPPoE with VLAN 10

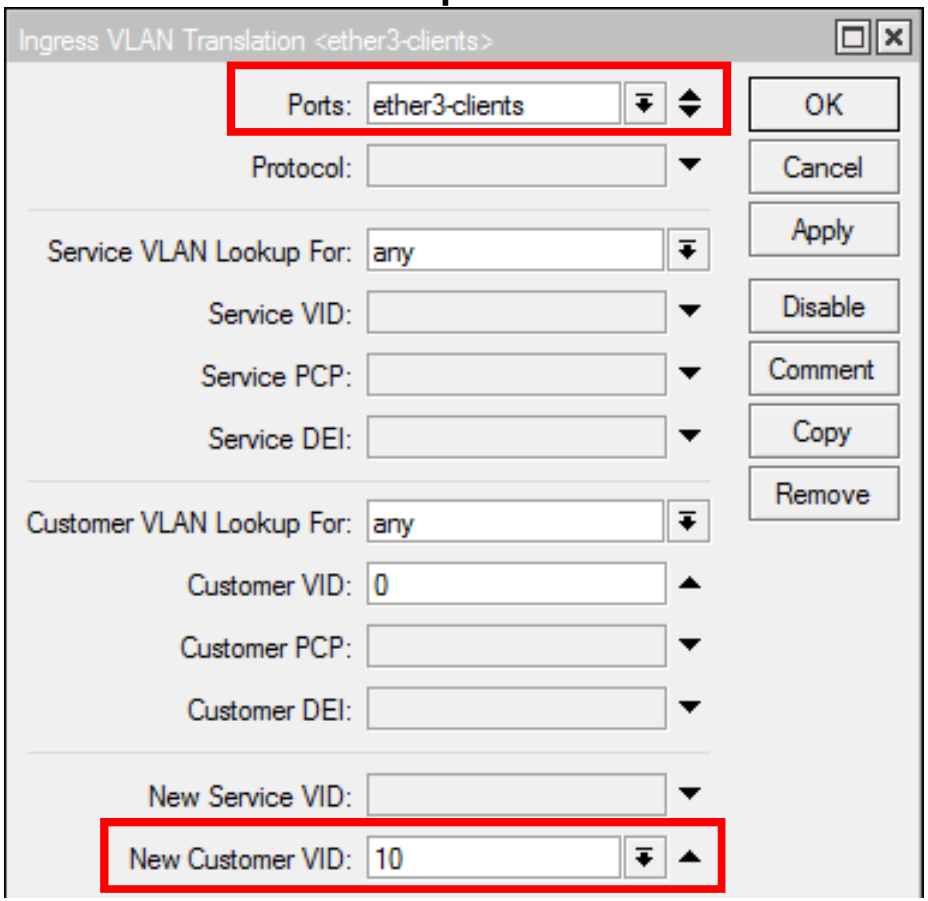


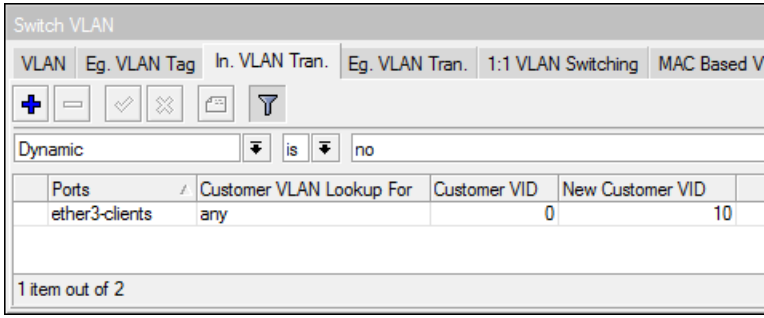


Switch → ACL

Advanced traffic control (ACL)

ether3: Access port → VLAN 10

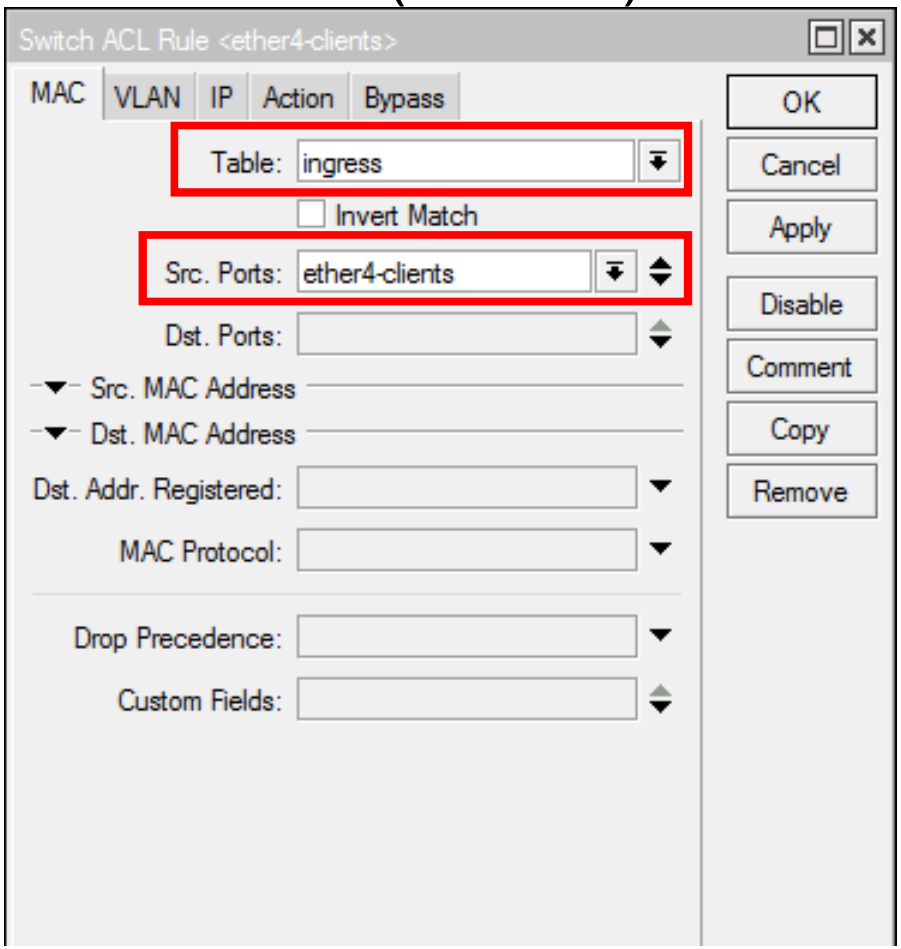


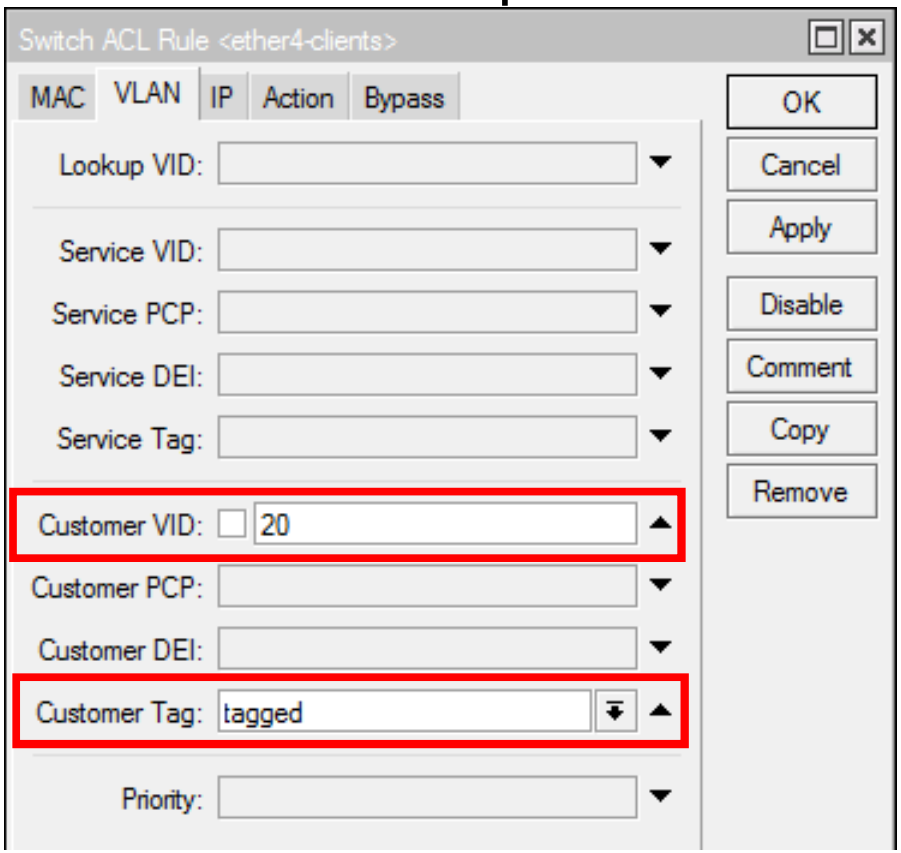


Switch → VLAN → Ingress VLAN Tran.

Advanced traffic control (ACL)

ether4: Allow (forward) all on VLAN 20. Then: Drop rest.





Switch → ACL

Reference

```
# Create switch
/interface ethernet
set [ find default-name=ether1 ] name=ether1-trunk
set [ find default-name=ether2 ] master-port=ether1-trunk name=ether2-clients
set [ find default-name=ether3 ] master-port=ether1-trunk name=ether3-clients
set [ find default-name=ether4 ] master-port=ether1-trunk name=ether4-clients

# ether1 is uplink / trunk port: VLAN 10, 20
/interface ethernet switch egress-vlan-tag
add tagged-ports=ether1-trunk,ether2-clients vlan-id=10
add tagged-ports=ether1-trunk,ether4-clients vlan-id=20

# ether2: Block everything apart from PPPoE on VLAN 10
/interface ethernet switch acl
add table=ingress action=forward customer-tag=tagged customer-vid=10 \
        mac-protocol=pppoe-discovery src-ports=ether2-clients
add table=ingress action=forward customer-tag=tagged customer-vid=10 \
        mac-protocol=pppoe src-ports=ether2-clients
add table=ingress action=drop src-ports=ether2-clients
```

Reference

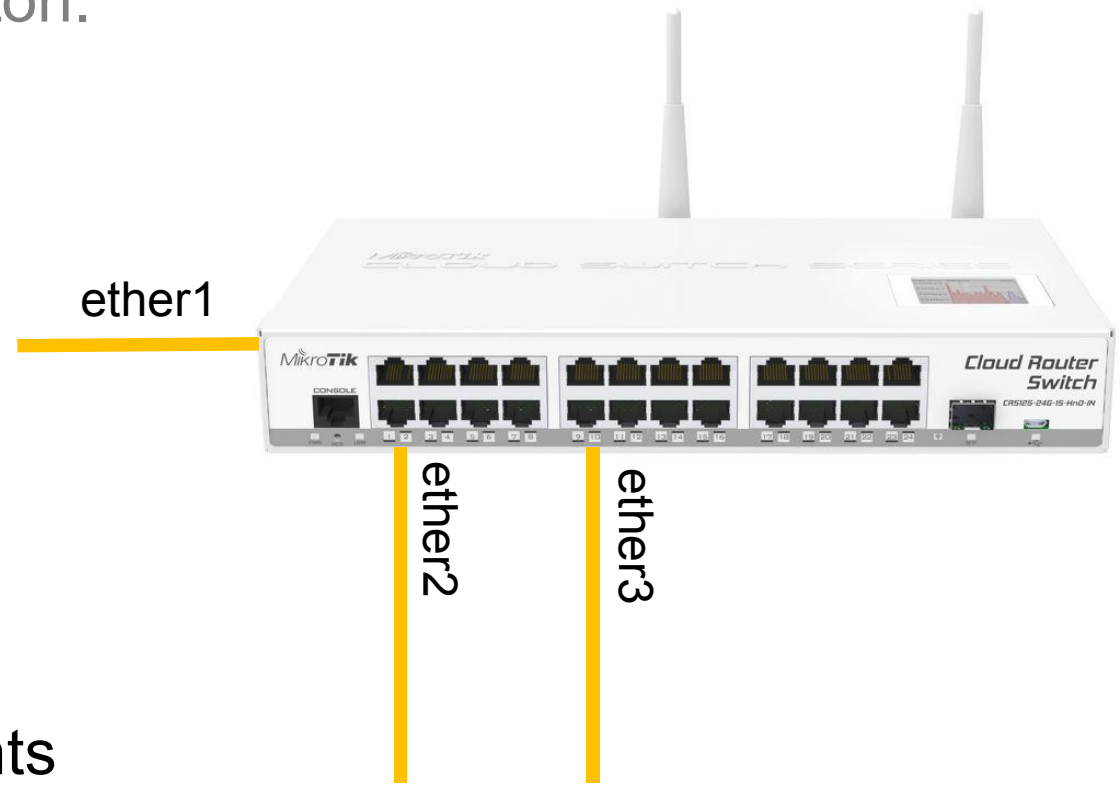
```
# ether3: Automatically VLAN 10 (connect to pppoe server)
/interface ethernet switch ingress-vlan-translation
add customer-vid=0 new-customer-vid=10 ports=ether3-clients

# ether4: Allow everything on VLAN 20
/interface ethernet switch acl
add table=ingress action=forward customer-tag=tagged customer-vid=20 \
        src-ports=ether4-clients
add table=ingress action=drop src-ports=ether4-clients
```

Client isolation

Client isolation

Client communication blocked.

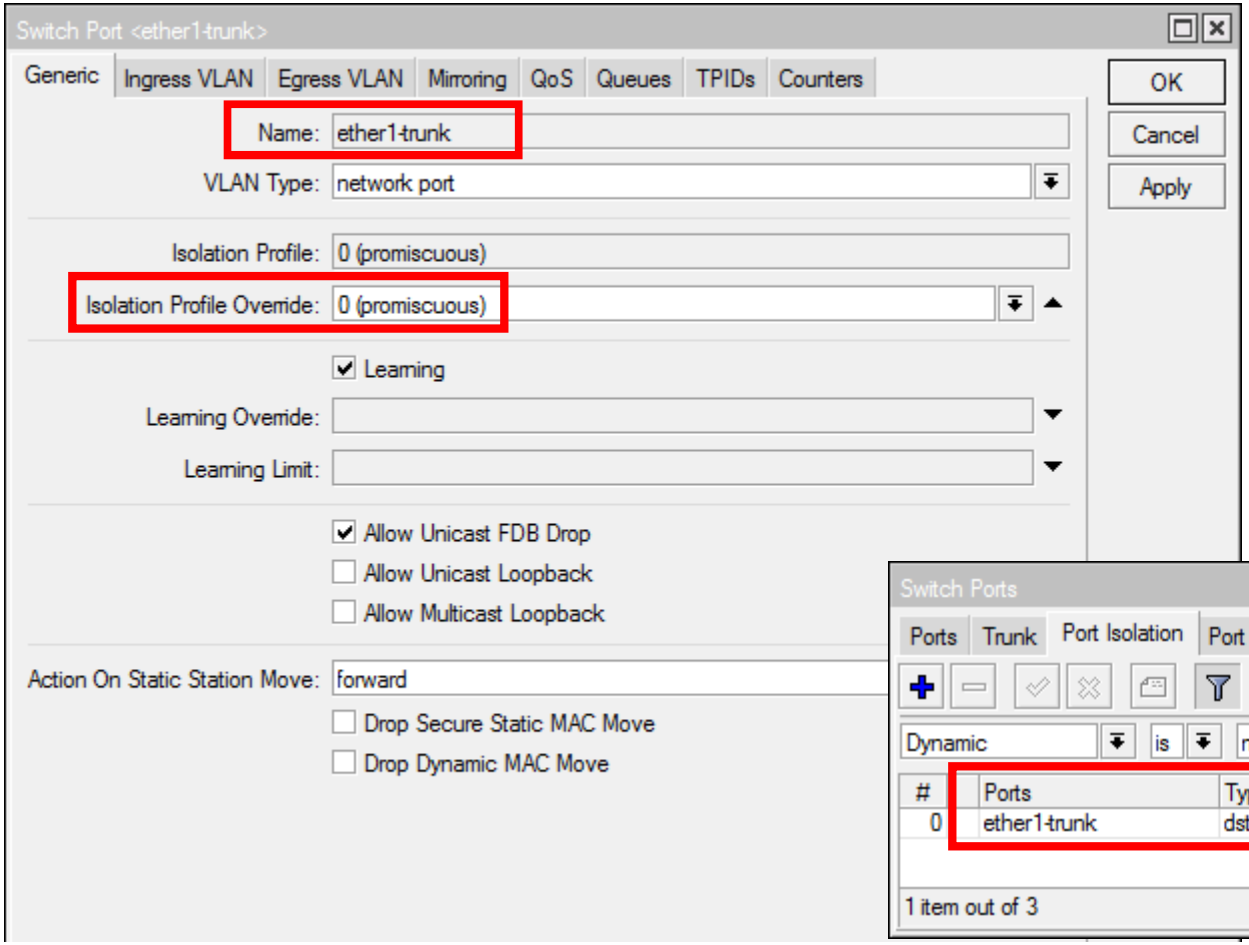
Bridge would use horizon.



- ether1: Uplink
- ether2, ether3: Clients

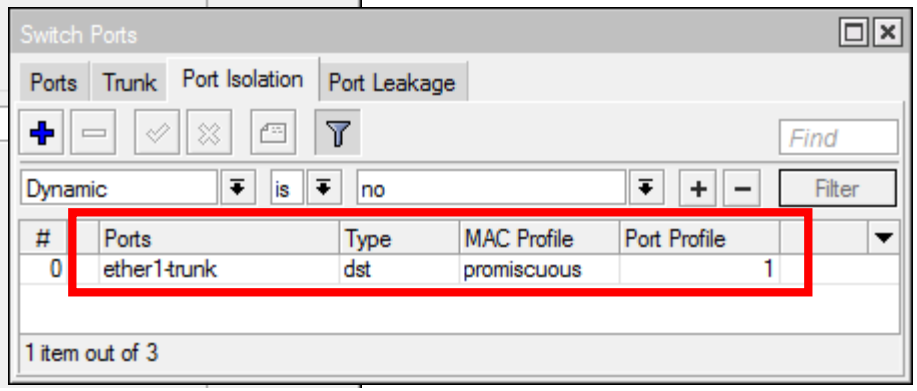
Client isolation

Switch → Ports → ether1



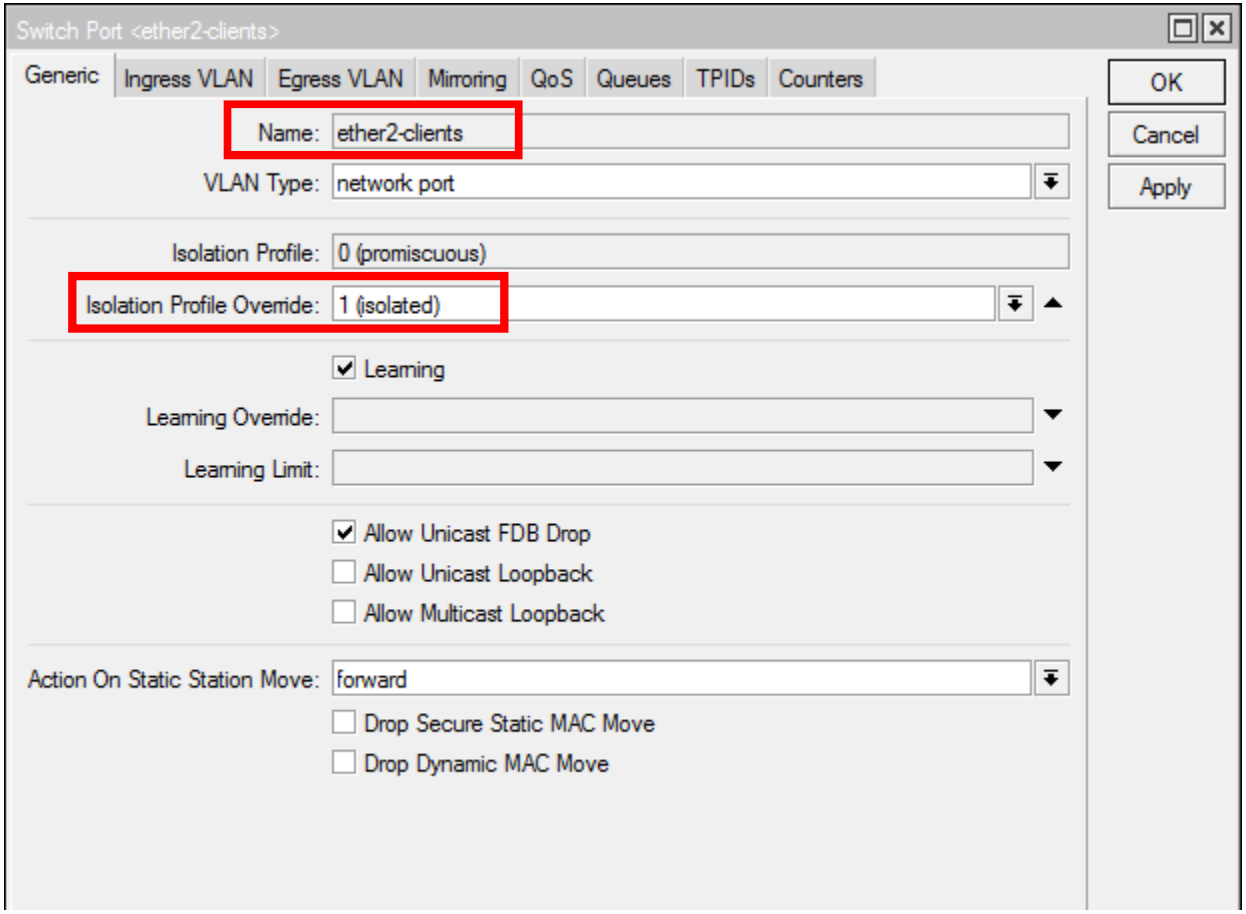
Isolation profile 0

Switch → Ports



Client isolation

Switch → Ports → ether2 and ether3



Isolation profile 1

Isolation Profile?

Isolation Profile	Function	Description
0	Uplink port	Communicate with all ports
1	Isolated port	Communication only with uplink port
2-31	Community port	Communication with uplink port and ports of same community

Winbox: Isolation Profile Override

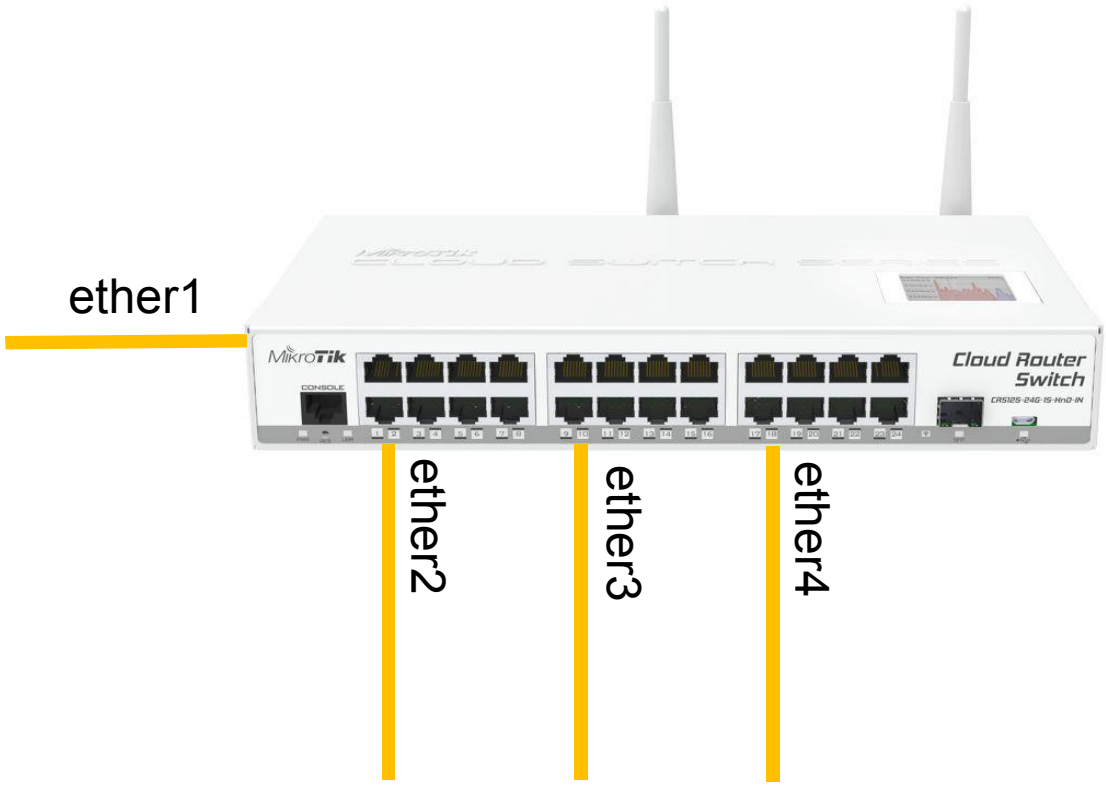
CLI: isolation-leakage-profile-override

Reference

```
# Create switch
/interface ethernet
set [ find default-name=ether1 ] name=ether1-trunk
set [ find default-name=ether2 ] master-port=ether1-trunk name=ether2-clients
set [ find default-name=ether3 ] master-port=ether1-trunk name=ether3-clients

/interface ethernet switch port
set ether1-trunk isolation-leakage-profile-override=0
set ether2-clients isolation-leakage-profile-override=1
set ether3-clients isolation-leakage-profile-override=1

# type dst -> egress packets
/interface ethernet switch port-isolation
add port-profile=1 ports=ether1-trunk type=dst mac-profile=promiscuous
```

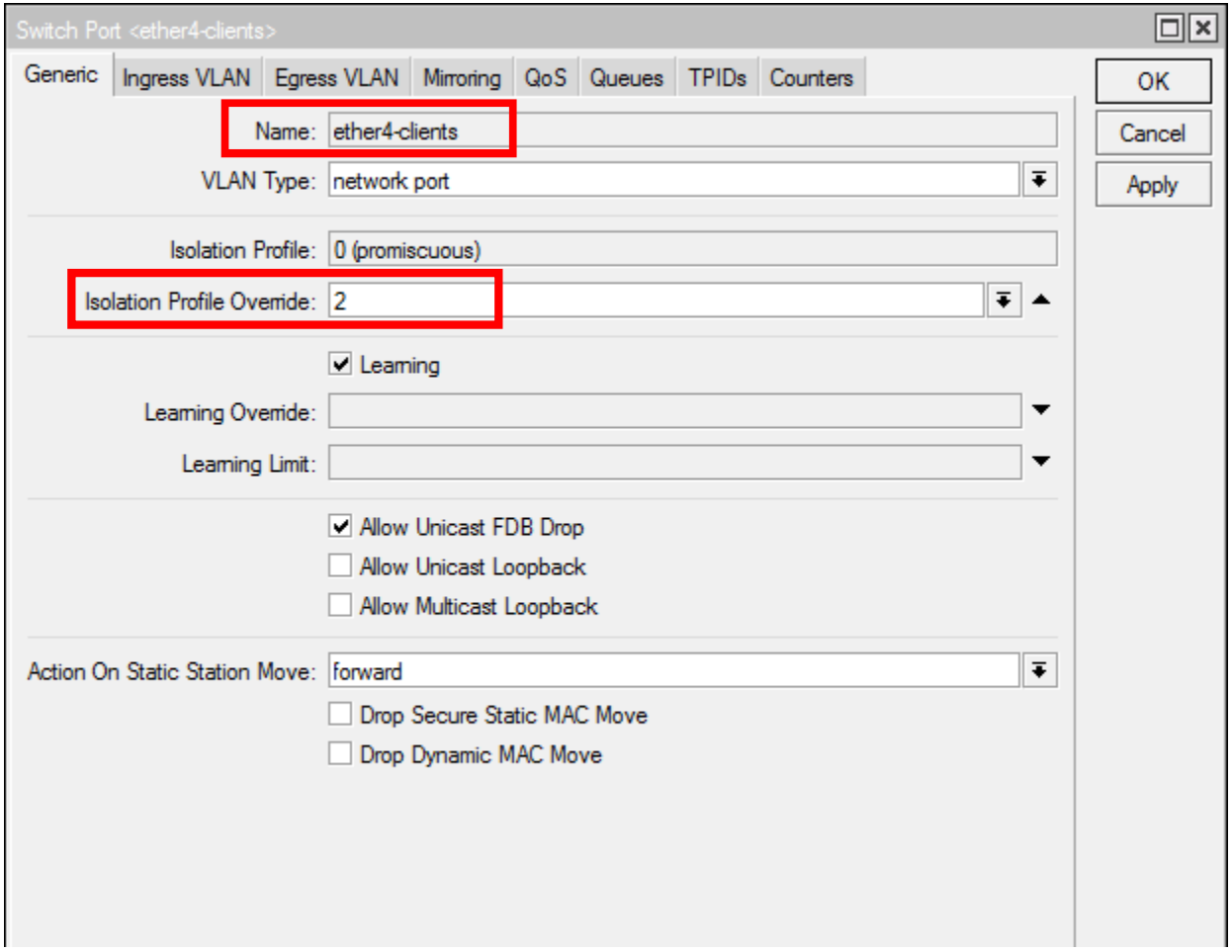
Evil DHCP server(s)

Block DHCP servers at customer site(s)



Evil DHCP server(s)

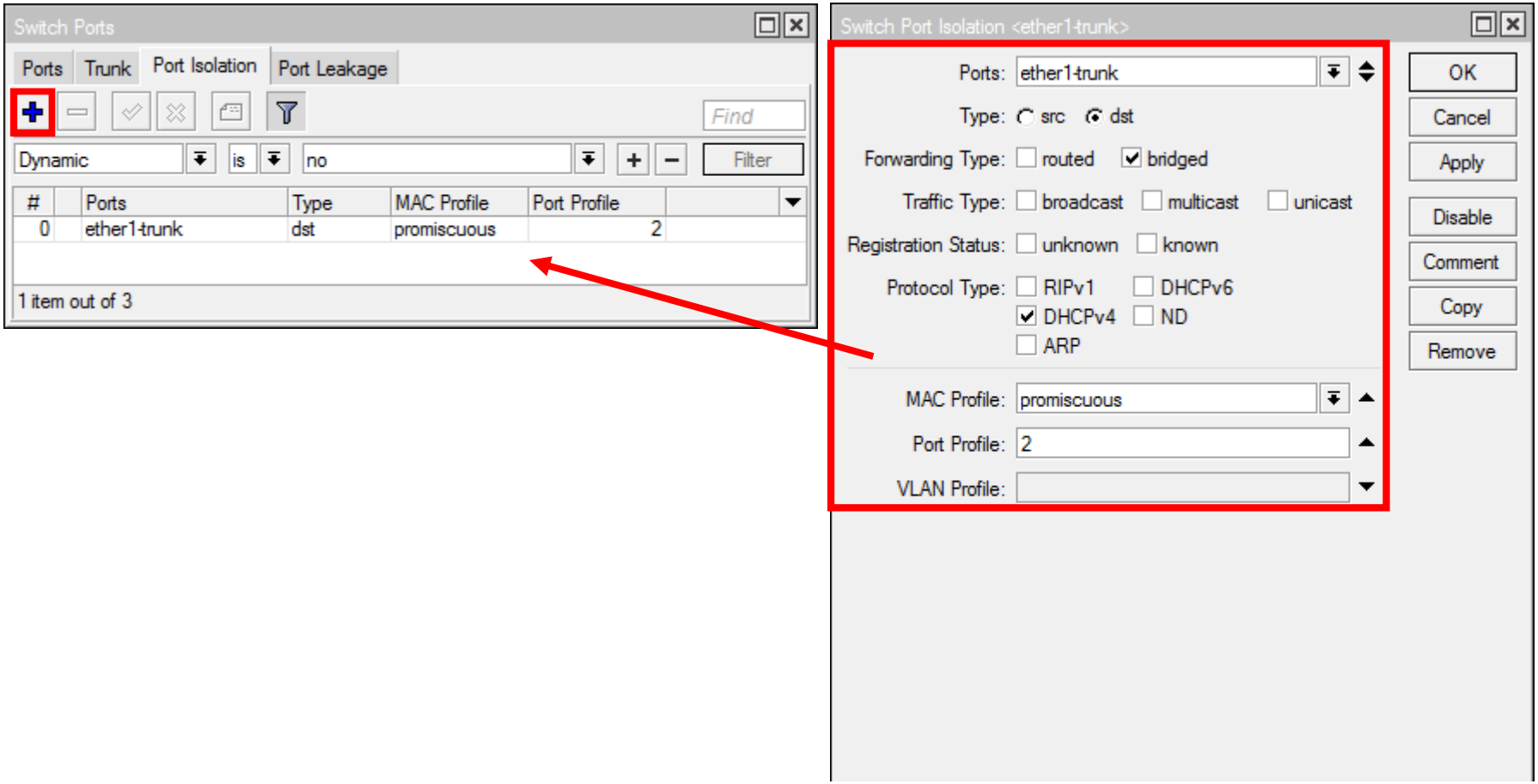
Switch → Ports → ether2, ether3 and ether4



Isolation profile 2

Evil DHCP server(s)

Switch → Ports → Port Isolation



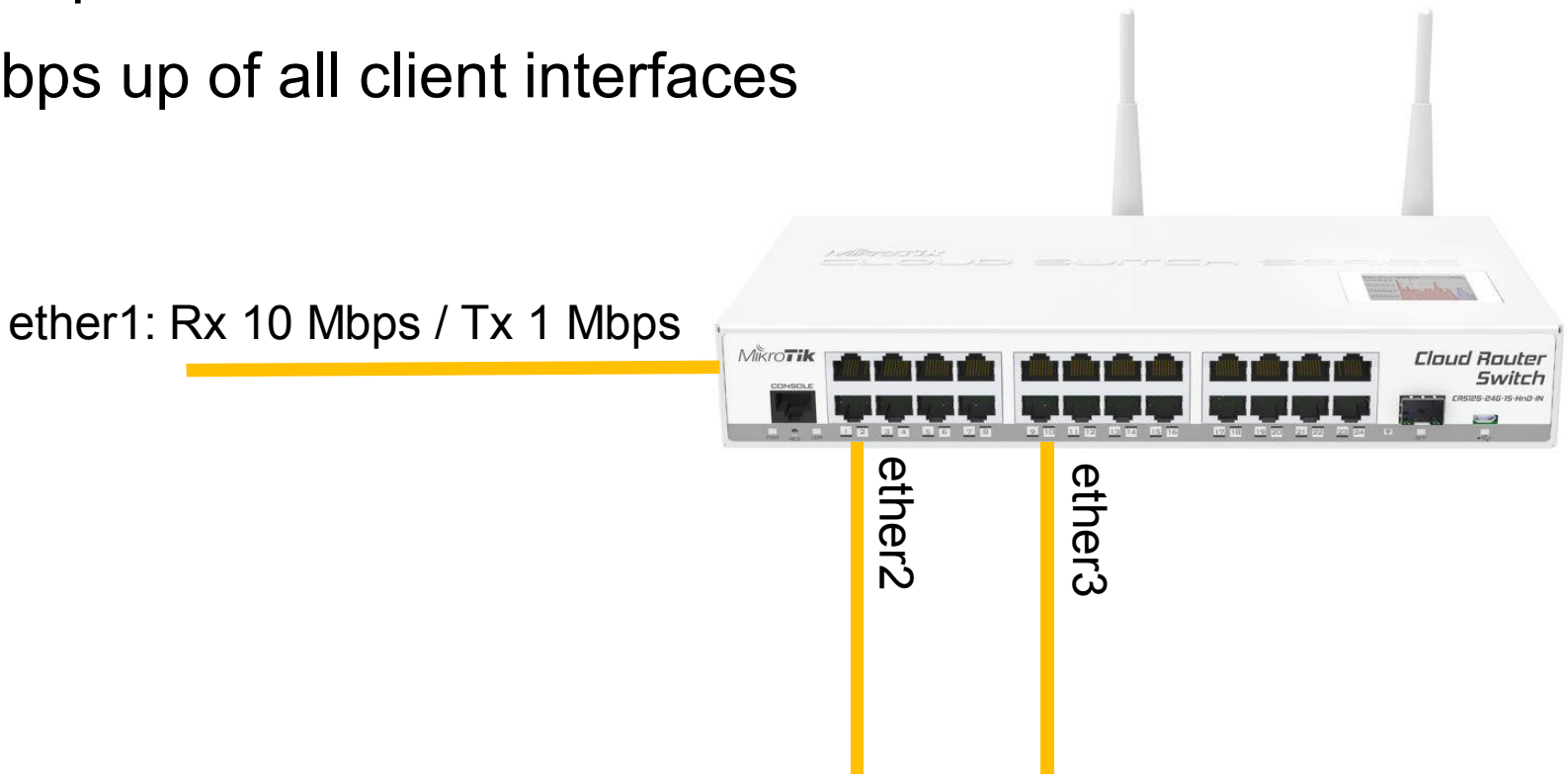
Reference

```
# Create switch
/interface ethernet
set [ find default-name=ether1 ] name=ether1-trunk
set [ find default-name=ether2 ] master-port=ether1-trunk name=ether2-clients
set [ find default-name=ether3 ] master-port=ether1-trunk name=ether3-clients
set [ find default-name=ether4 ] master-port=ether1-trunk name=ether4-clients

/interface ethernet switch port
set ether2-clients isolation-leakage-profile-override=2
set ether3-clients isolation-leakage-profile-override=2
set ether4-clients isolation-leakage-profile-override=2

# Allow DHCPv4 out to ether1-trunk
/interface ethernet switch port-isolation
add port-profile=2 protocol-type=dhcpv4 type=dst forwarding-type=bridged \
        ports=ether1-trunk registration-status="" traffic-type=""
```

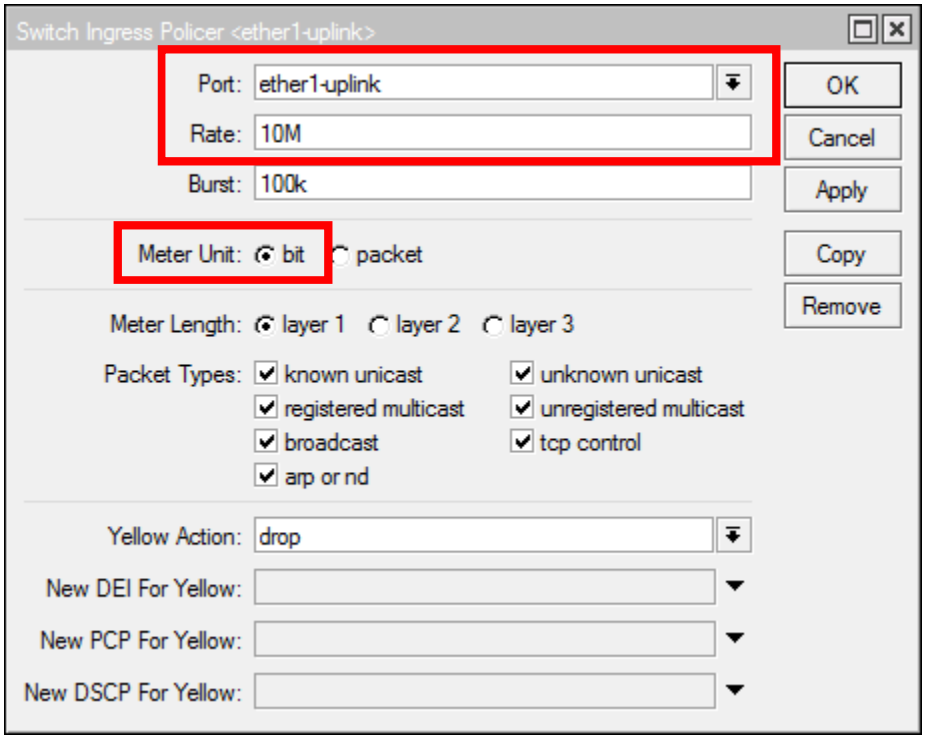
Max throughput #1

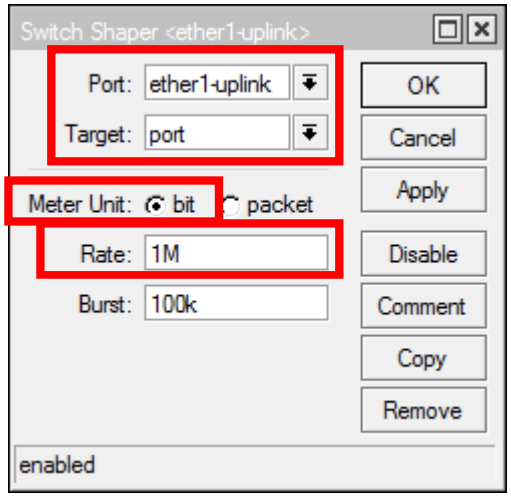
Define max. throughput without queues:

- 10 Mbps down of all client interfaces
- 1 Mbps up of all client interfaces



Max throughput #1





Switch → QoS → Shaper

Switch → QoS → Ingress Port Policer

ether1 (uplink)	Rate	Result
ingress-port-policer	10M	Download of all interfaces
shaper	1M	Upload of all interfaces

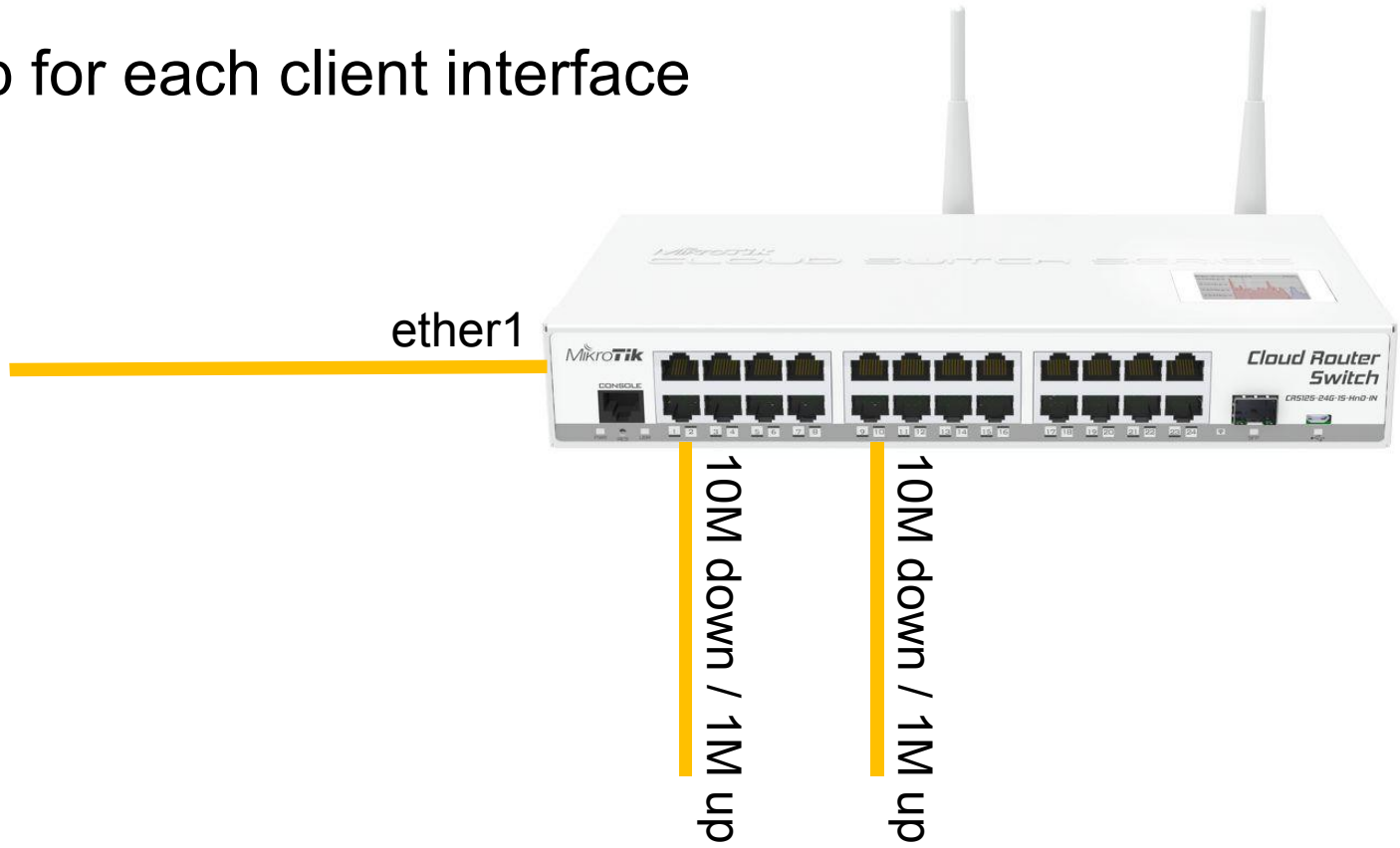
Reference

```
/interface ethernet
set [ find default-name=ether1 ] name=ether1-uplink
set [ find default-name=ether2 ] master-port=ether1-uplink name=ether2-clients
set [ find default-name=ether3 ] master-port=ether1-uplink name=ether3-clients

/interface ethernet switch ingress-port-policer
add port=ether1-uplink meter-unit=bit rate=10M

/interface ethernet switch shaper
add port=ether1-uplink meter-unit=bit rate=1M
```

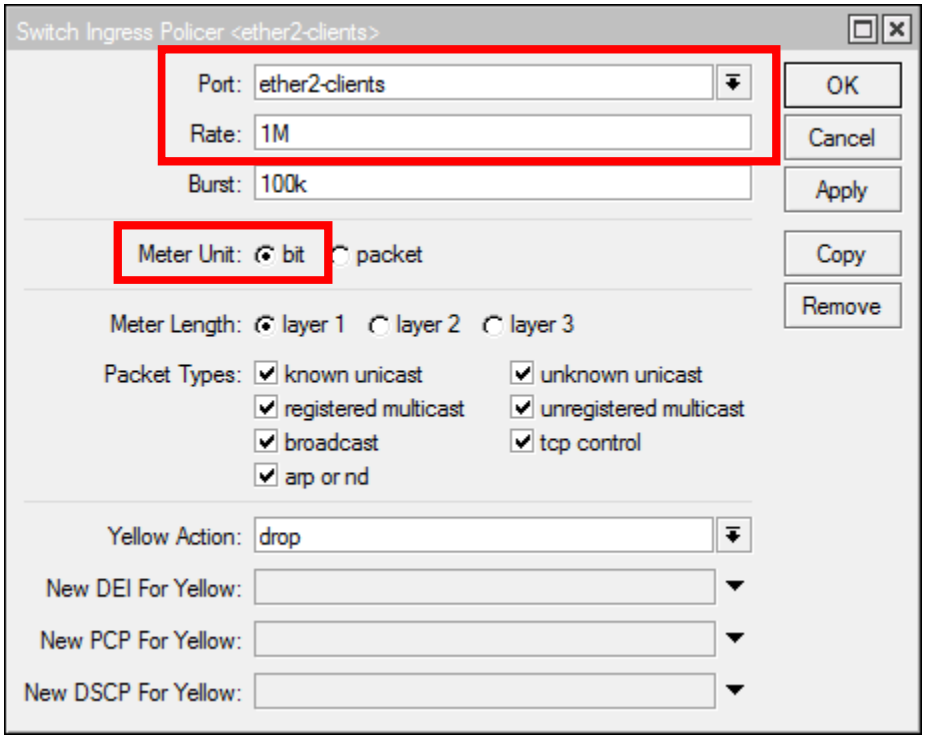
Max throughput #2

Define max. throughput without queues:

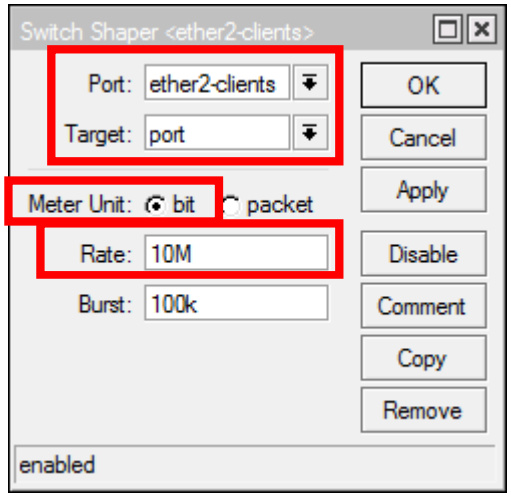
- 10 Mbps down for each client interface
- 1 Mbps up for each client interface



Max throughput #2



Switch → QoS → Ingress Port Policer



Switch → QoS → Shaper

ether2 (client)	Rate	Result
ingress-port-policer	1M	Download of client(s) on ether2
shaper	10M	Upload of client(s) on ether2

Reference

```
/interface ethernet
set [ find default-name=ether1 ] name=ether1-uplink
set [ find default-name=ether2 ] master-port=ether1-uplink name=ether2-clients
set [ find default-name=ether3 ] master-port=ether1-uplink name=ether3-clients

/interface ethernet switch ingress-port-policer
add port=ether2-clients rate=1M
add port=ether3-clients rate=1M

/interface ethernet switch shaper
add port=ether2-clients rate=10M
add port=ether3-clients rate=10M
```

Thank you!

FMS Internetservice GmbH

Phone: +49 761 2926500
Web: www.fmsweb.de
Shop: www.mikrotik-shop.de
Email: sales@fmsweb.de
Twitter: https://twitter.com/fmsweb_de