



MikroTik CloudRouterSwitch

Features and configurations



Overview

Big picture

- Company Profile
- Introduction
- Hardware STP
- Switch Configurations



FMS Internetservice GmbH

Company Profile



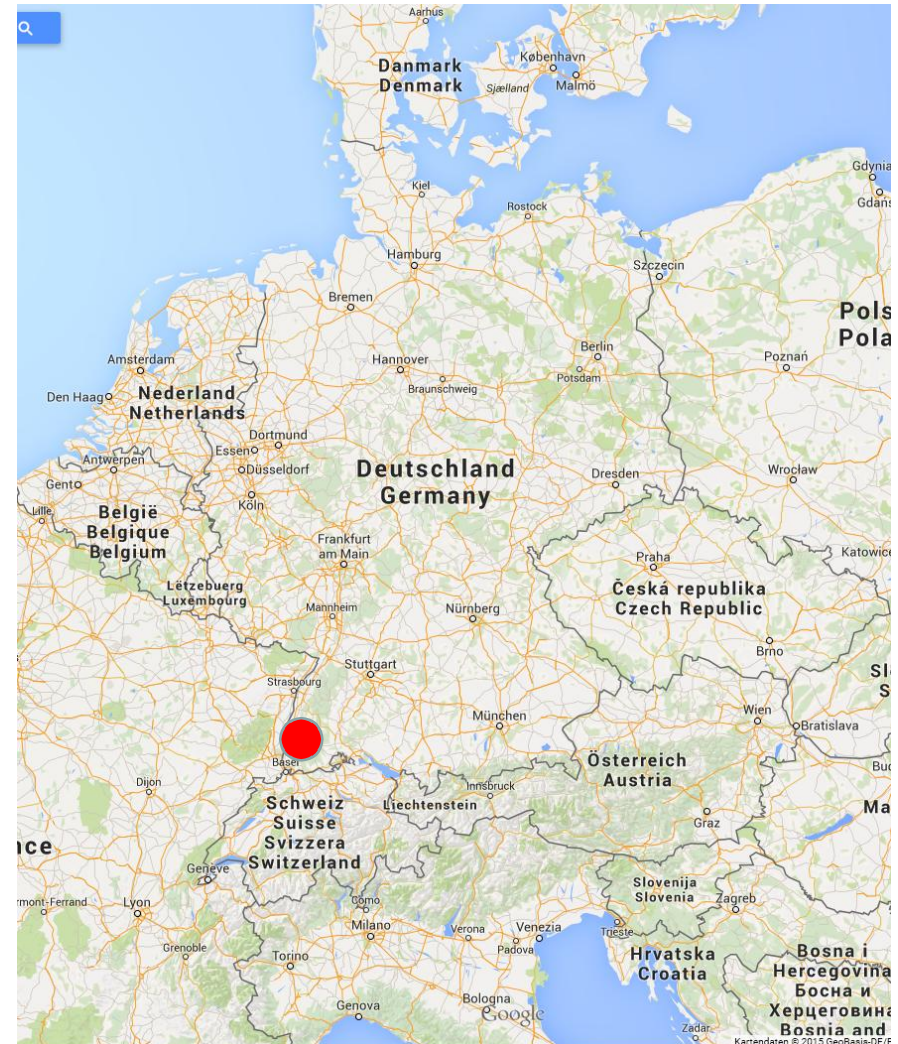
About me

- 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

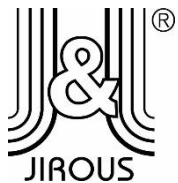
Sebastian Inacker: TR11

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

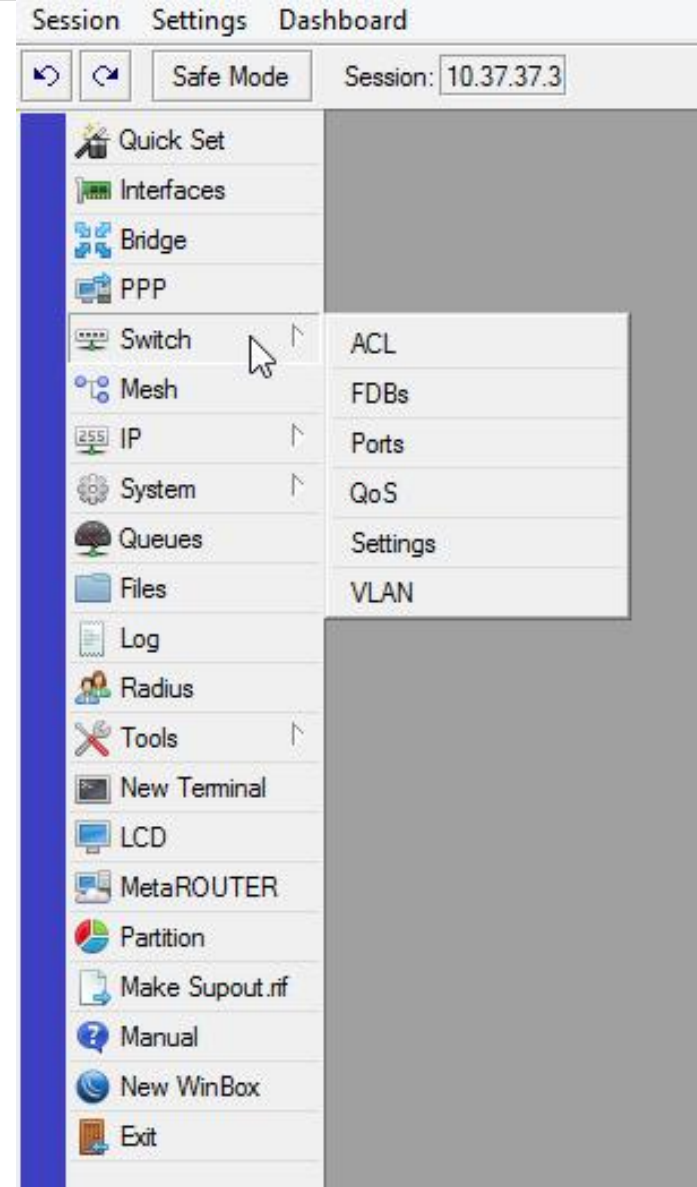
Topic:

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

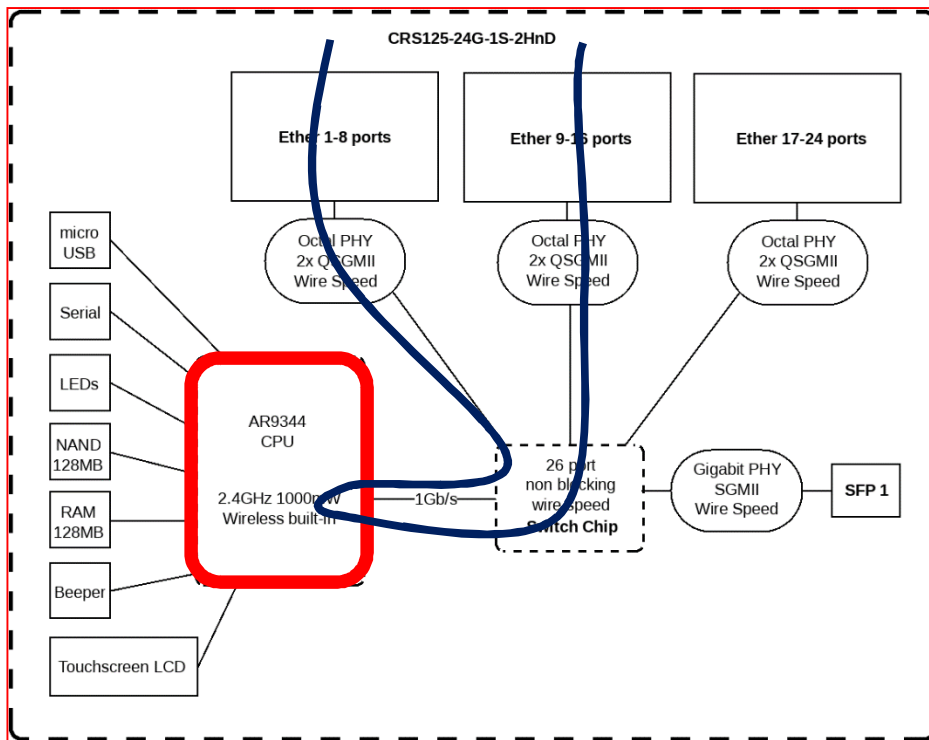
You can use them differently. Success depends on your needs

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

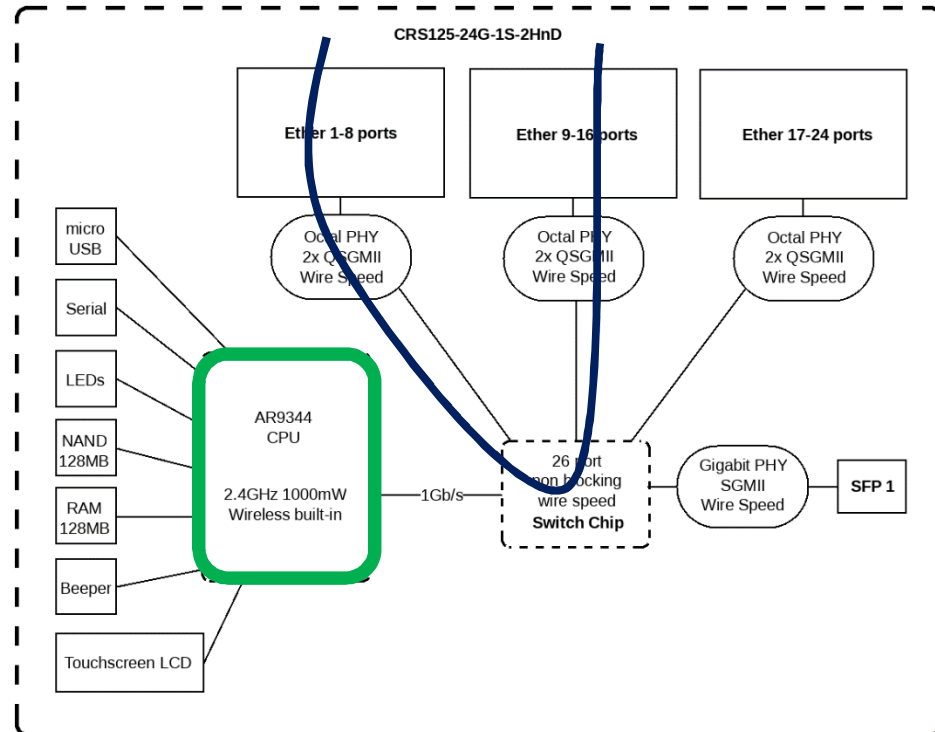


Bridge or switch chip (on CRS)?

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 comparison 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

Interface <ether2-slave>

General | Ethernet | Loop Protect | Overall Stats | Rx Stats | ...

Name: ether2-slave

Type: Ethernet

MTU: 1500

Actual MTU: 1500

L2 MTU: 1588

Max L2 MTU: 4064

MAC Address: E4:8D:8C:A1:E5:95

ARP: enabled

ARP Timeout:

Master Port: ether1-master

Bandwidth (Rx/Tx): unlimited / unlimited

Switch: switch1

OK
Cancel
Apply
Disable
Comment
Torch
Cable Test
Blink
Reset MAC Address
Reset Counters

enabled | running | slave | no link



Hardware STP

Create RSTP bridge

Interface <bridge1>

General STP Status Traffic

Protocol Mode: none stp rstp

Priority: 8000 hex

Max Message Age: 00:00:20

Forward Delay: 00:00:15

Transmit Hold Count: 6

Ageing Time: 00:05:00

OK Cancel Apply Disable Comment Copy Remove Torch

Add master port

New Bridge Port

General Status

Interface: ether1-master

Bridge: bridge1

Priority: 80 hex

Path Cost: 10

Horizon: [dropdown]

Edge: auto

Point To Point: auto

External FDB: auto

OK Cancel Apply Disable Comment Copy Remove

Result: ether2 dynamic

Bridge

Bridge Ports Filters NAT Hosts

Interface	Bridge	Forwarding
ether1-master	bridge1	yes
ether2-slave	bridge1	yes

2 items



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



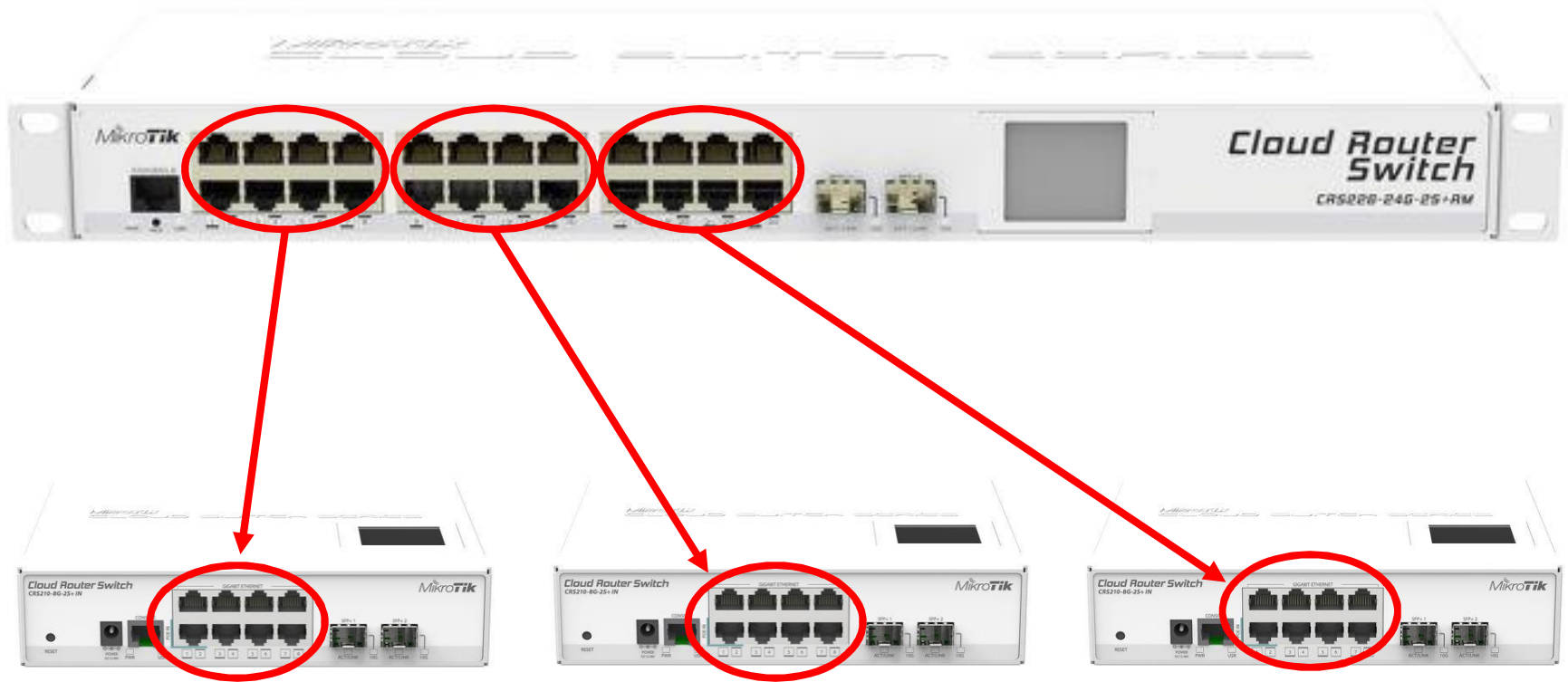
Common configurations

- 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

	Name	Type	Master Port	Switch
RS	ether01	Ethernet	none	switch 1
S	ether02	Ethernet	ether01	switch 1
S	ether03	Ethernet	ether01	switch 1
S	ether04	Ethernet	ether01	switch 1
S	ether05	Ethernet	ether01	switch 1
S	ether06	Ethernet	ether01	switch 1
S	ether07	Ethernet	ether01	switch 1
S	ether08	Ethernet	ether01	switch 1
RS	ether09	Ethernet	none	switch 1
S	ether10	Ethernet	ether09	switch 1
S	ether11	Ethernet	ether09	switch 1
S	ether12	Ethernet	ether09	switch 1
S	ether13	Ethernet	ether09	switch 1
S	ether14	Ethernet	ether09	switch 1
RS	ether15	Ethernet	ether09	switch 1
S	ether16	Ethernet	ether09	switch 1
RS	ether17	Ethernet	none	switch 1
S	ether18	Ethernet	ether17	switch 1
S	ether19	Ethernet	ether17	switch 1
S	ether20	Ethernet	ether17	switch 1
S	ether21	Ethernet	ether17	switch 1
RS	ether22	Ethernet	ether17	switch 1
S	ether23	Ethernet	ether17	switch 1
S	ether24	Ethernet	ether17	switch 1
	sfp 1	Ethernet	none	switch 1

25 items out of 29



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

Interface <sfpplus2>

Overall Stats Rx Stats Tx Stats Status Traffic ...

Tx/Rx Rate: 9.8 Gbps / 9.8 Gbps

Tx/Rx Packet Rate: 820 178 p/s / 820 210 p/s

FP Tx/Rx Rate: 0 bps / 0 bps

FP Tx/Rx Packet Rate: 0 p/s / 0 p/s

Tx/Rx Bytes: 4701.8 GiB / 5036.0 GiB

Tx/Rx Packets: 3580 841 990 / 3599 141 128

OK
Cancel
Apply
Disable
Comment
Torch
Cable Test
Blink
Reset MAC Address
Reset Counters

Switch ACL

ACL Policer

+ - ✓ ✗ [] [] Find

#	Table	Src. Ports	Src. MAC Address/Src. ...	Src. MAC Address/Src....	Action	
121	ingress	sfpplus2	4C:5E:0C:00:00:01	FF:FF:FF:00:00:00	forward	
122	ingress	sfpplus2	E4:8D:8C:00:00:01	FF:FF:FF:00:00:00	forward	
123	ingress	sfpplus2	D4:CA:6D:00:00:01	FF:FF:FF:00:00:00	forward	
124	ingress	sfpplus2	6C:3B:6B:00:00:01	FF:FF:FF:00:00:00	forward	
125	ingress	sfpplus2	00:0C:42:00:00:01	FF:FF:FF:00:00:00	forward	
126	ingress	sfpplus2	64:D1:54:00:00:01	FF:FF:FF:00:00:00	forward	
127	ingress	sfpplus2			drop	

128 items

enabled running slave link ok



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

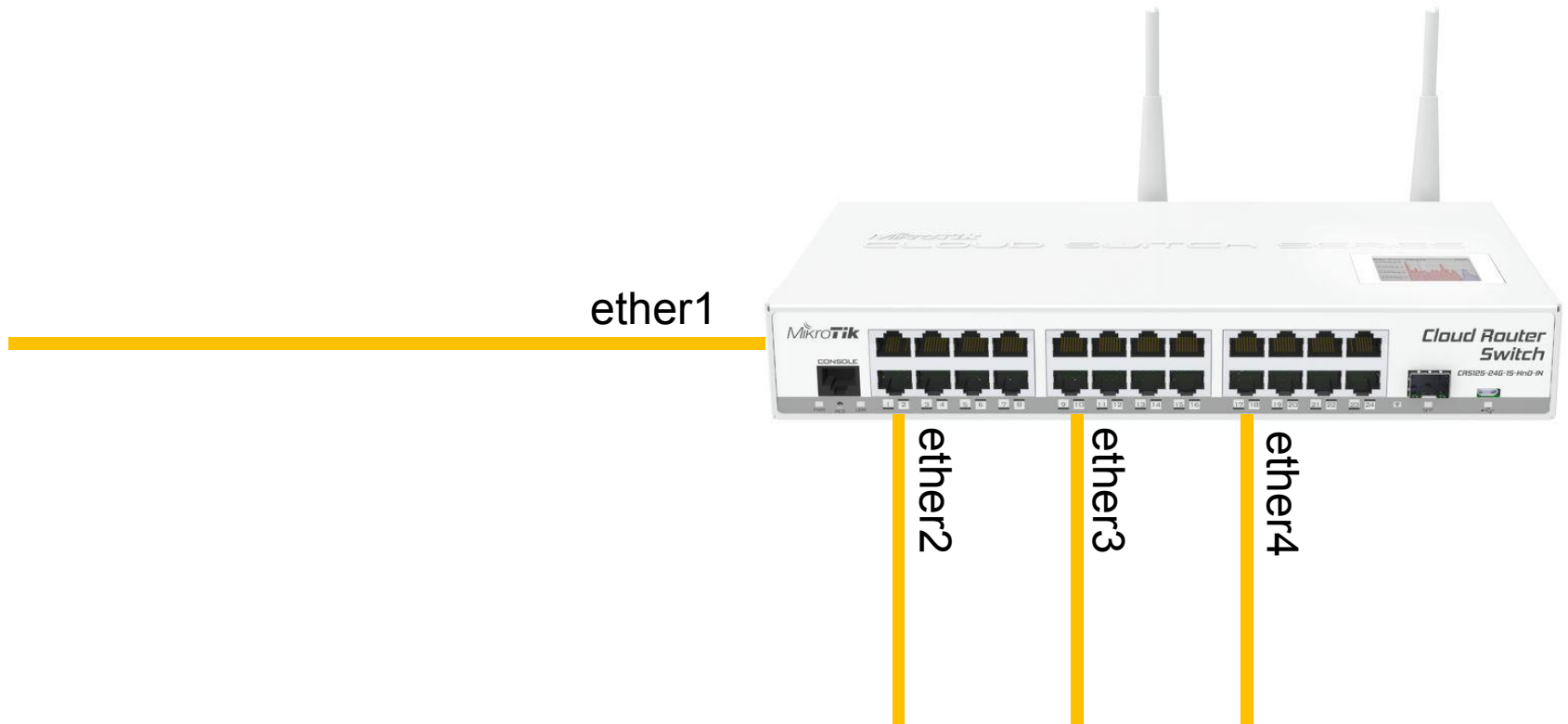


ACL setups



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:

Switch ACL Rule <ether2>

MAC | VLAN | IP | Action | Bypass

Table: ingress

Invert Match

Src. Ports: ether2

Dst. Ports:

Src. MAC Address

Src. MAC Address: 02:DE:AB:CD:EF:11

Src. MAC Mask: FF:FF:FF:FF:FF:FF

Dst. MAC Address

Dst. Addr. Registered:

MAC Protocol:

Drop Precedence:

Custom Fields:

OK
Cancel
Apply
Disable
Comment
Copy
Remove

Switch ACL Rule <ether2>

MAC | VLAN | IP | Action | Bypass

Action: forward

Mirror To:

Policer:

Learn SA:

New Service VID:

New Service PCP:

New Service DEI:

OK
Cancel
Apply
Disable
Comment
Copy
Remove



ACL – stay on your port!

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

Switch ACL Rule <>

MAC | VLAN | IP | Action | Bypass

Table: ingress

Invert Match

Src. Ports:

Dst. Ports:

Src. MAC Address

Src. MAC Address: 02:DE:AB:CD:EF:11

Src. MAC Mask: FF:FF:FF:FF:FF:FF

Dst. MAC Address

Dst. Addr. Registered:

MAC Protocol:

Drop Precedence:

Custom Fields:

OK

Cancel

Apply

Disable

Comment

Copy

Remove

Switch ACL Rule <>

MAC | VLAN | IP | Action | Bypass

Action: drop

Mirror To:

Policer:

Learn SA:

New Service VID:

New Service PCP:

New Service DEI:

OK

Cancel

Apply

Disable

Comment

Copy

Remove



ACL – stay on your port!

Drop anything (other) on ether2:

Switch ACL Rule <ether2>

MAC | VLAN | IP | Action | Bypass

Table: ingress

Invert Match

Src. Ports: ether2

Dst. Ports:

OK
Cancel
Apply
Disable
Comment

Switch ACL Rule <ether2>

MAC | VLAN | IP | Action | Bypass

Action: drop

Mirror To:

Policer:

Learn SA:

OK
Cancel
Apply
Disable
Comment

Switch ACL

ACL | Policer

+ - ✓ ✗ [Filter Icon] Find

#	Table	Src. Ports	Src. MAC Address/Src. MAC Address	Src. MAC Address/Src. MAC Mask	Action
::: Allow MAC on ether2					
0	ingress	ether2	02:DE:AB:CD:EF:11	FF:FF:FF:FF:FF:FF	forward
::: Deny MAC on any (other) port					
1	ingress		02:DE:AB:CD:EF:11	FF:FF:FF:FF:FF:FF	drop
::: Deny anything (other) on ether2					
2	ingress	ether2			drop

3 items



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"

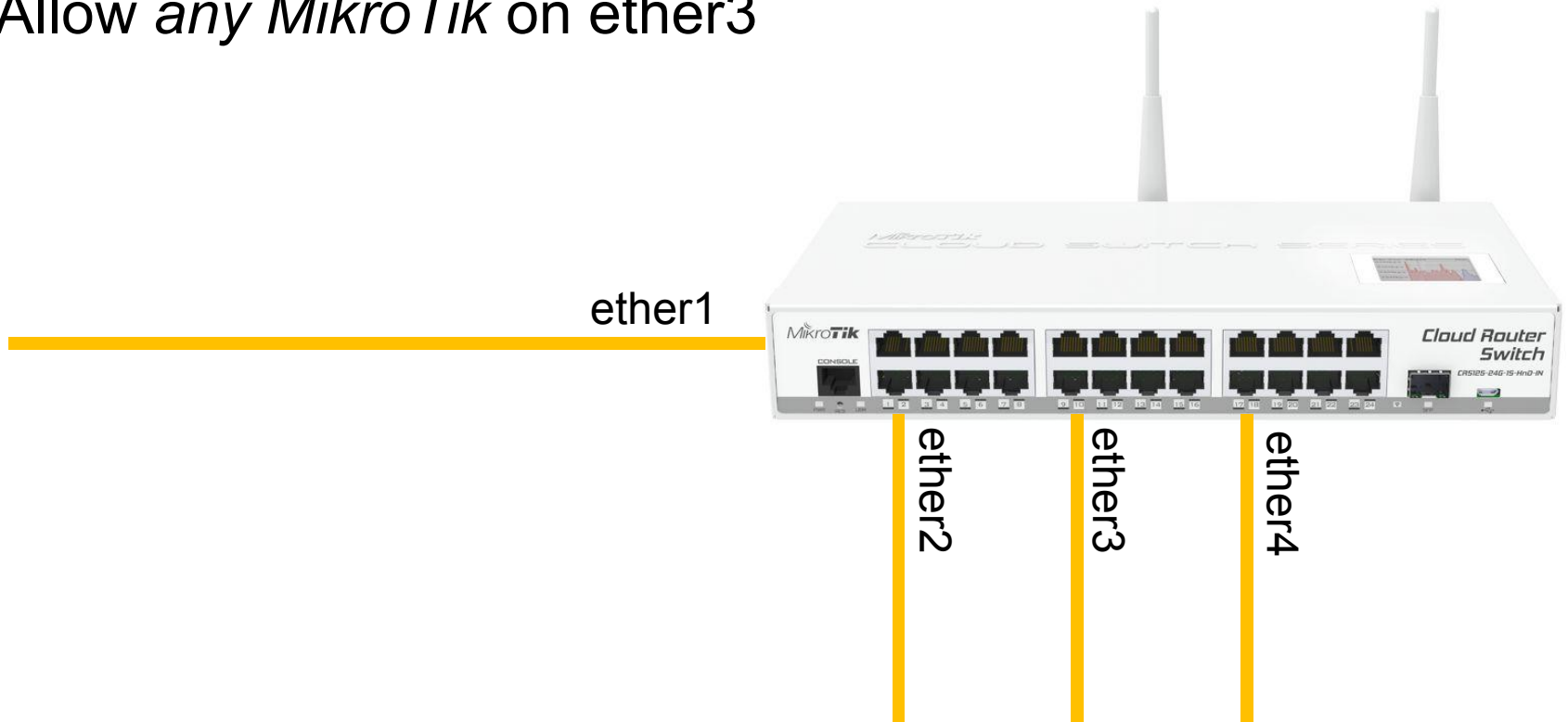
add table=ingress action=drop mac-src-address=02:DE:AB:CD:EF:11 \
    comment="Deny MAC 02:DE:AB:CD:EF:11 on any (other) port"

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

Switch ACL Rule <ether3>

Table: ingress

Src. Ports: ether3

Src. MAC Address: 4C:5E:0C:00:00:01

Src. MAC Mask: FF:FF:FF:00:00:00

Switch ACL

ACL Policer

Src. Ports: ether3 contains

#	Table	Src. Ports	Src. MAC Address/Src. MAC Address	Src. MAC Address/Src. MAC Mask	Action
3	ingress	ether3	4C:5E:0C:00:00:01	FF:FF:FF:00:00:00	forward
4	ingress	ether3	E4:8D:8C:00:00:01	FF:FF:FF:00:00:00	forward
5	ingress	ether3	D4:CA:6D:00:00:01	FF:FF:FF:00:00:00	forward
6	ingress	ether3	6C:3B:6B:00:00:01	FF:FF:FF:00:00:00	forward
7	ingress	ether3	00:0C:42:00:00:01	FF:FF:FF:00:00:00	forward
8	ingress	ether3	64:D1:54:00:00:01	FF:FF:FF:00:00:00	forward
9	ingress	ether3			drop

7 items out of 10 (1 selected)



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

#	Action	Chain	Interfaces...	Src. MAC Address/Src. MAC Address	Src. MAC Address/Src. MAC Mask	
0	✓ acc...	forward	ether3	4C:5E:0C:00:00:01	FF:FF:FF:00:00:00	
1	✓ acc...	forward	ether3	E4:8D:8C:00:00:01	FF:FF:FF:00:00:00	
2	✓ acc...	forward	ether3	D4:CA:6D:00:00:01	FF:FF:FF:00:00:00	
3	✓ acc...	forward	ether3	6C:3B:6B:00:00:01	FF:FF:FF:00:00:00	
4	✓ acc...	forward	ether3	00:0C:42:00:00:01	FF:FF:FF:00:00:00	
5	✓ acc...	forward	ether3	64:D1:54:00:00:01	FF:FF:FF:00:00:00	
6	✗ drop	forward	ether3			

7 items

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

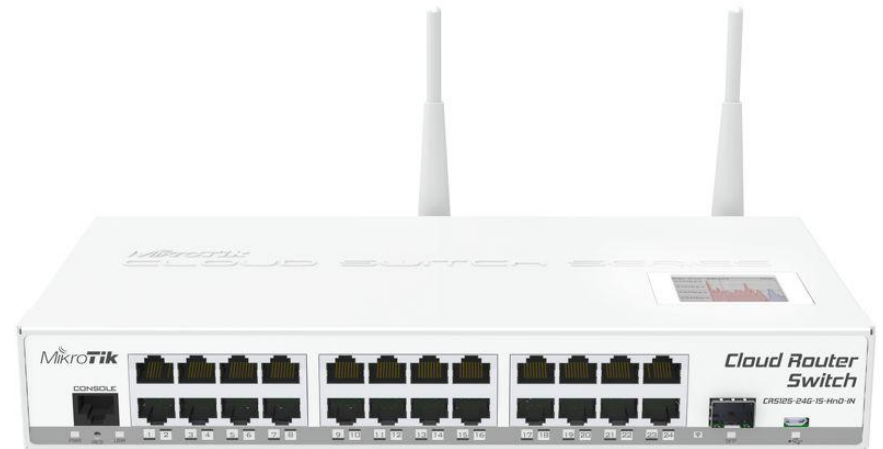
ACL = wirespeed



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





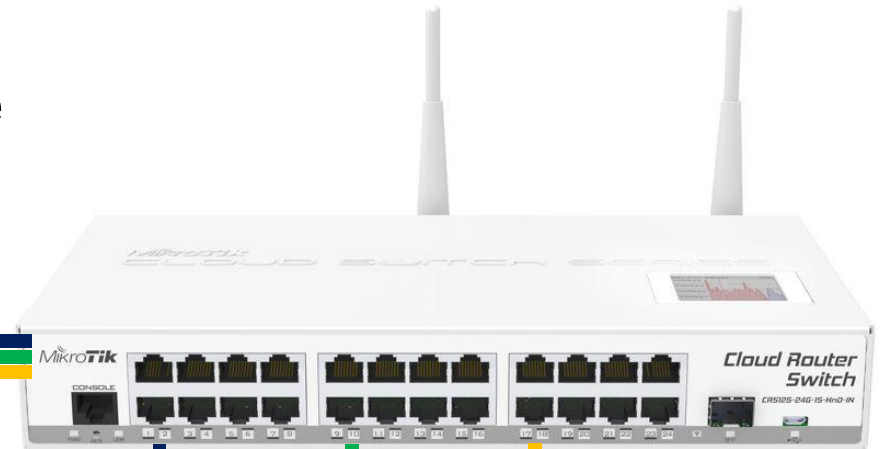
Multiple possible setups



Split your trunk

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

ether1: VLAN 10, 20, 30



ether2: VLAN 10

ether3: VLAN 20

ether4: VLAN 30

ether1: Trunk port

ether2 – ether4: Access ports



Egress, outgoing to trunk port

Switch → VLAN

Switch VLAN

VLAN	Eg. VLAN Tag	In. VLAN Tran.	Eg. VLAN Tran.	1:1 VLAN Switching	...
	10				
	20				
	30				
D	4095				

4 items

Switch Egress Tag VLAN <10>

VLAN ID: 10

Tagged Ports: ether1-trunk

OK Cancel Apply Disable Comment Copy Remove

enabled

Switch Egress Tag VLAN <20>

VLAN ID: 20

Tagged Ports: ether1-trunk

OK Cancel Apply Disable Comment Copy Remove

enabled

Switch Egress Tag VLAN <30>

VLAN ID: 30

Tagged Ports: ether1-trunk

OK Cancel Apply Disable Comment Copy Remove

enabled



Ingress, incomming from access port

Switch → VLAN

VLAN	Eg. VLAN Tag	In. VLAN Tran.	Eg. VLAN Tran.	1:1	
+	-	✓	✗	📄	🔍
Dynamic	is	no			
Ports	Customer VLAN Lookup For	Customer V			
ether2-v10	any				
ether3-v20	any				
ether4-v30	any				

3 items out of 4

Ingress VLAN Translation <ether2-v10>

Ports: ether2-v10

Protocol:

Service VLAN Lookup For: any

Service VID:

Service PCP:

Service DEI:

Customer VLAN Lookup For: any

Customer VID: 0

Customer PCP:

Customer DEI:

New Service VID:

New Customer VID: 10

PCP Propagation

SA Learning

enabled

OK
Cancel
Apply
Disable
Comment
Copy
Remove



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

	Name	Type	VLAN ID	Interface
R	↔ Service VLAN ID	VLAN	50	ether5
R	↔ Customer VLAN ID	VLAN	200	Service VLAN ID

2 items out of 15



Split your trunk

Done! Wait... IP management?

Address	Network	Interface
10.10.10.10/24	10.10.10.0	ether1-trunk
10.20.20.20/24	10.20.20.0	ether1-trunk
10.30.30.30/24	10.30.30.0	ether1-trunk

3 items out of 4

Address <10.10.10.10/24>

Address: 10.10.10.10/24

Network: 10.10.10.0

Interface: ether1-trunk

OK Cancel Apply

Address <10.20.20.20/24>

Address: 10.20.20.20/24

Network: 10.20.20.0

Interface: ether1-trunk

OK Cancel Apply

Address <10.30.30.30/24>

Address: 10.30.30.30/24

Network: 10.30.30.0

Interface: ether1-trunk

OK Cancel Apply

IP reachable from access port side.

Not from trunk port side!



Split your trunk

Done! Wait... IP management?

Address	Network	Interface
10.10.10.10/24	10.10.10.0	vlan10.ether1
10.20.20.20/24	10.20.20.0	vlan20.ether1
10.30.30.30/24	10.30.30.0	vlan30.ether1

3 items out of 4

Address <10.10.10.10/24>

Address: 10.10.10.10/24

Network: 10.10.10.0

Interface: vlan10.ether1

OK Cancel Apply

Address <10.20.20.20/24>

Address: 10.20.20.20/24

Network: 10.20.20.0

Interface: vlan20.ether1

OK Cancel Apply

Address <10.30.30.30/24>

Address: 10.30.30.30/24

Network: 10.30.30.0

Interface: vlan30.ether1

OK Cancel Apply

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

The screenshot shows the Mikrotik WinBox interface for configuring VLANs. The main window is titled "Switch VLAN" and has several tabs: "VLAN", "Eg. VLAN Tag", "In. VLAN Tran.", "Eg. VLAN Tran.", and "1:1 VLAN Switching". The "Eg. VLAN Tag" tab is active, showing a table of VLAN configurations. A red box highlights the table content:

VLAN ID	Tagged Ports
10	switch1-cpu, ether1-trunk
20	switch1-cpu, ether1-trunk
30	switch1-cpu, ether1-trunk

Below the table, it says "3 items out of 4".

An overlay dialog box titled "Switch Egress Tag VLAN <10>" is open. It has a "VLAN ID" field set to "10". The "Tagged Ports" section is highlighted with a red box and contains two entries: "switch1-cpu" and "ether1-trunk", each with a dropdown arrow and a double-headed arrow icon. The dialog also has buttons for "OK", "Cancel", "Apply", "Disable", "Comment", "Copy", and "Remove". At the bottom, it says "enabled".



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:

The screenshot shows a network management interface with a 'Switch VLAN' window. The window has a table of VLANs and a configuration dialog for a selected VLAN.

VLAN ID	Ports
10	switch1-cpu, ether1-trunk, ether2-v10
20	switch1-cpu, ether1-trunk, ether3-v20
30	switch1-cpu, ether1-trunk, ether4-v30

The configuration dialog for 'Switch VLAN <10>' shows the following settings:

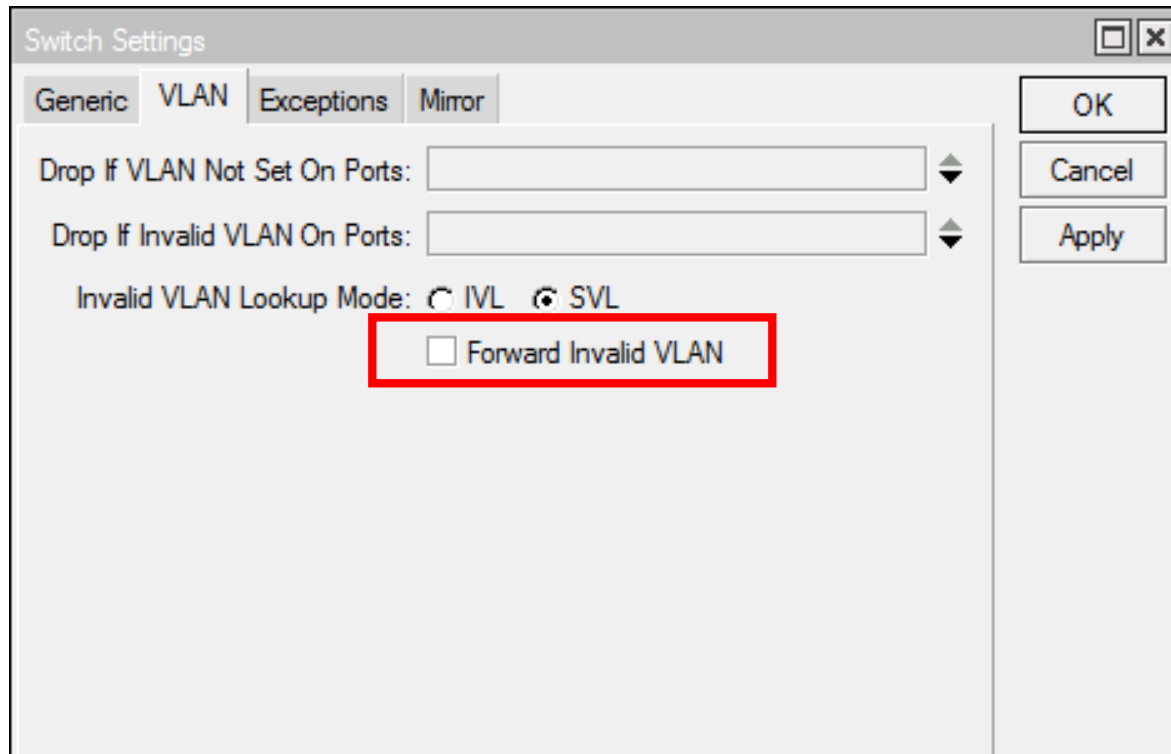
- VLAN ID: 10
- Ports: switch1-cpu, ether1-trunk, ether2-v10
- SA Learning:
- QoS Group: none

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



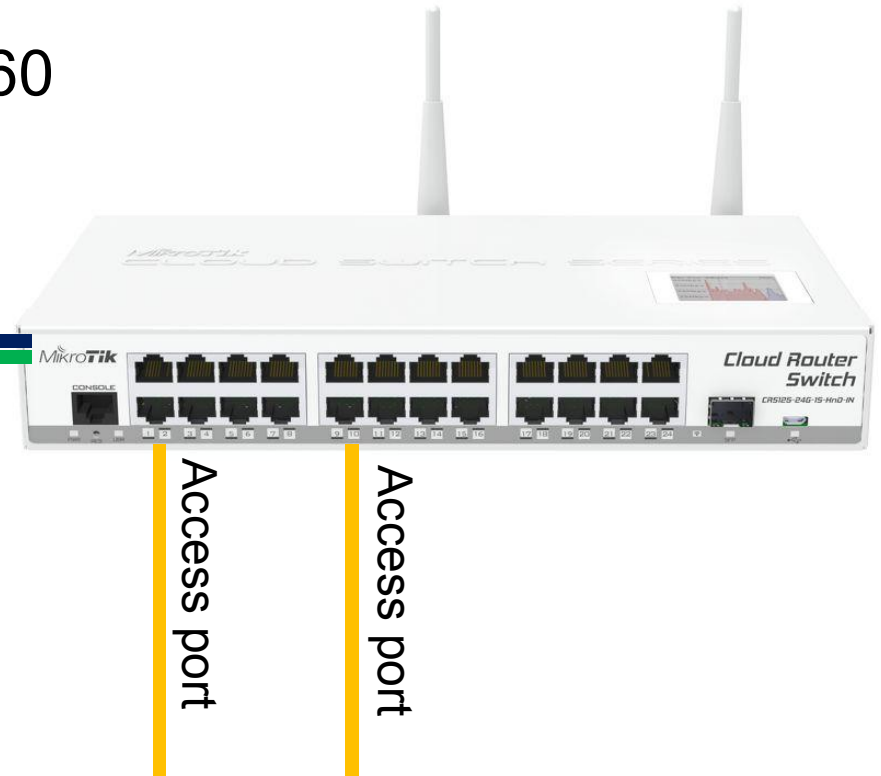
MAC based dynamic VLAN

VLAN definition, based on MAC address

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

4C:5E:0C:C7:47:69 = VLAN 50

F0:DE:F1:78:33:56 = VLAN 60





MAC based dynamic VLAN

Switch → Ports → Ports → etherX

Switch Port <ether2>

Generic | Ingress VLAN | Egress VLAN | Mirroring | QoS | Queues | TPIDs | Counters

MAC Based VLAN Translate

MAC Based Service VLAN For: all frames

MAC Based Customer VLAN For: all frames

Default Customer PCB: 0

OK
Cancel
Apply

Switch Ports

Ports | Trunk | Port Isolation | Port Leakage

Name	VLAN Type	Isolation Profile	MAC Based V...	Egress VLAN
ether1	network port		29 no	unmodified
ether2	network port		29 yes	unmodified
ether3	network port		29 yes	unmodified
ether4	network port		30 no	unmodified
ether5	network port		30 no	unmodified
ether6	network port		30 no	unmodified
ether7	network port		30 no	unmodified
ether8	network port		30 no	unmodified
sfp-sfpplus1	network port		30 no	unmodified
sfpplus2	network port		30 no	unmodified
switch1-cpu	network port		31 no	unmodified

Switch VLAN

VLAN | Eg. VLAN Tag | In. VLAN Tran. | Eg. VLAN Tran. | 1:1 VLAN Switching | ...

Dynamic | is | no

VLAN ID	Tagged Ports
50	ether1
60	ether1

2 items out of 3

Switch → VLAN



MAC based dynamic VLAN

Specific MAC address required (no mask)

Switch → VLAN → MAC Based VLAN

The screenshot illustrates the configuration of MAC-based dynamic VLANs in Mikrotik WinBox. It shows three windows: the main 'Switch VLAN' window, and two 'Switch MAC Based VLAN' configuration windows.

Switch VLAN Window: The 'MAC Based VLAN' tab is selected. A table lists the configuration for two items:

Src. MAC Address	New Service VID	New Customer VID
::: ether1 interface of MikroTik router "R1" 4C:5E:0C:C7:47:69	0	50
::: Laptop F0:DE:F1:78:33:56	0	60

Switch MAC Based VLAN <4C:5E:0C:C7:47:69> Window: Shows configuration for the router's MAC address. The 'Src. MAC Address' is 4C:5E:0C:C7:47:69, 'New Service VID' is 0, and 'New Customer VID' is 50.

Switch MAC Based VLAN <F0:DE:F1:78:33:56> Window: Shows configuration for the laptop's MAC address. The 'Src. MAC Address' is F0:DE:F1:78:33:56, 'New Service VID' is 0, and 'New Customer VID' is 60.



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

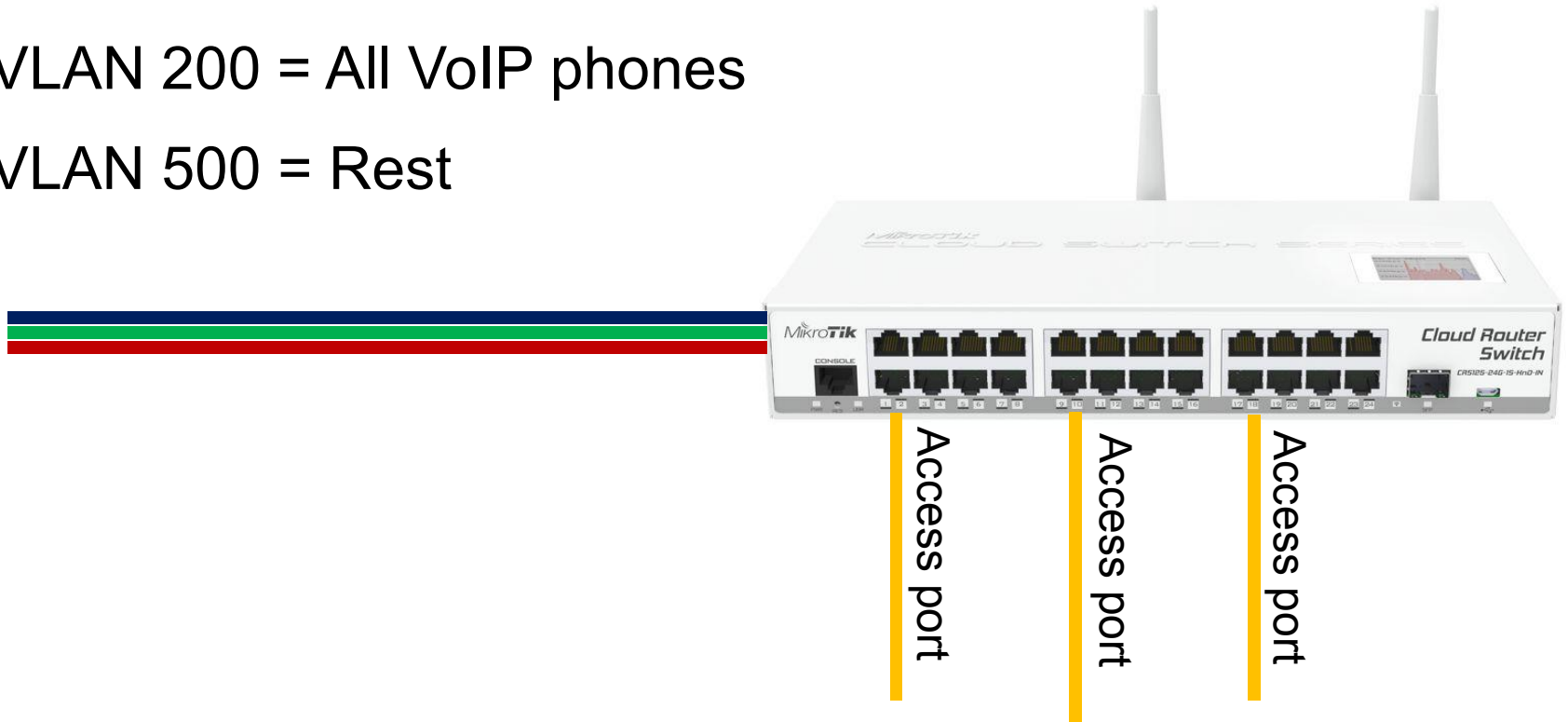
# Assign VLANs to MAC addresses
/interface ethernet switch mac-based-vlan
add src-mac=4c:5E:0c:C7:47:69 new-customer-vid=50
add src-mac=F0:DE:F1:78:33:56 new-customer-vid=60
```



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

ACL Policer

#	Table	Src. Ports	Src. MAC Address/Src. MAC Address	Src. MAC Address/Src. MAC Mask	Action	New Customer VID
0	ingress	ether2	00:0C:42:00:00:00	FF:FF:FF:00:00:00	forward	100
1	ingress	ether2	02:B1:B0:3A:4C:55	FF:FF:FF:00:00:00	forward	200
2	ingress	ether2			forward	500

3 items

Switch → ACL

Switch ACL Rule <ether2>

MAC VLAN IP Action Bypass

Action: forward

Mirror To:

Policer:

Learn SA:

New Service VID:

New Service PCP:

New Service DEI:

New Customer VID: 100

New Customer PCP:

New Customer DEI:

OK Cancel Apply Disable Comment Copy Remove

Switch ACL Rule <ether2>

MAC VLAN IP Action Bypass

Table: ingress

Invert Match

Src. Ports: ether2

Dst. Ports:

Src. MAC Address: 00:0C:42:00:00:00

Src. MAC Mask: FF:FF:FF:00:00:00

Dst. MAC Address:

OK Cancel Apply Disable Comment Copy Remove



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
```

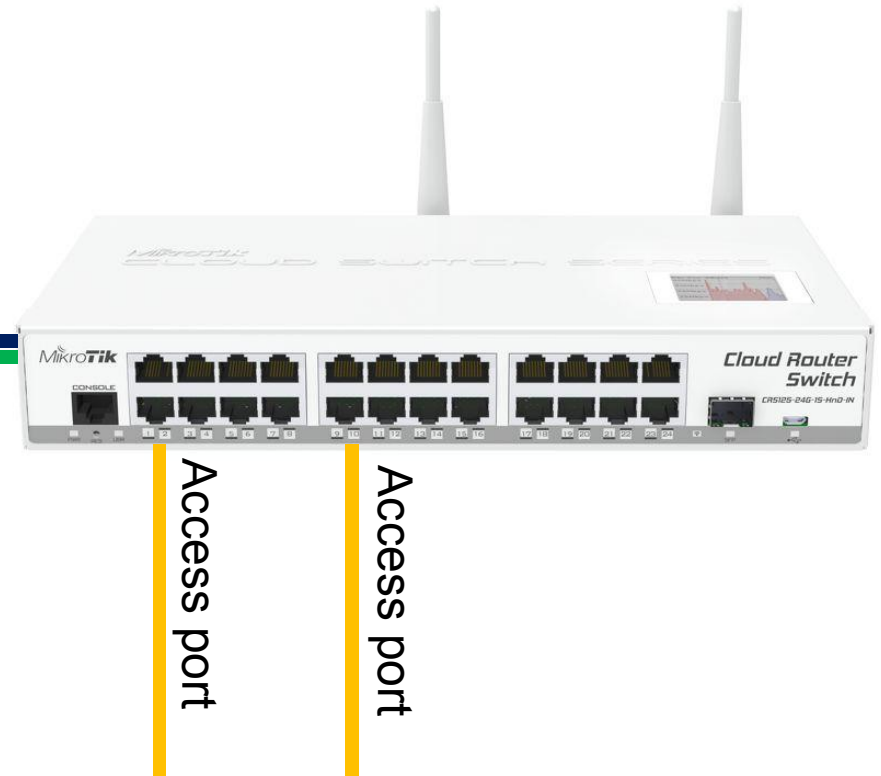


Protocol based dynamic VLAN

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 Protocol Based VLAN <8863 (pppoe-discovery)>

Frame Type: ethernet

Protocol: 8863 (pppoe-discovery)

Ports: ether2-clients

Set Service VID For: none

New Service VID: 0

Set Customer VID For: all frames

New Customer VID: 100

Set QoS For: none

QoS Group: none

enabled

OK
Cancel
Apply
Disable
Comment
Copy
Remove

Switch Protocol Based VLAN <8864 (pppoe-session)>

Frame Type: ethernet

Protocol: 8864 (pppoe-session)

Ports: ether2-clients

Set Service VID For: none

New Service VID: 0

Set Customer VID For: all frames

New Customer VID: 100

Set QoS For: none

QoS Group: none

enabled

OK
Cancel
Apply
Disable
Comment
Copy
Remove

Switch → VLAN → Protocol Based VLAN



Protocol based dynamic VLAN

Set VLAN for IP (IP & ARP)

Switch Protocol Based VLAN <800 (ip)>

Frame Type: ethernet

Protocol: 800 (ip)

Ports: ether2-clients

Set Service VID For: none

New Service VID: 0

Set Customer VID For: all frames

New Customer VID: 200

Set QoS For: none

QoS Group: none

enabled

OK
Cancel
Apply
Disable
Comment
Copy
Remove

Switch Protocol Based VLAN <806 (arp)>

Frame Type: ethernet

Protocol: 806 (arp)

Ports: ether2-clients

Set Service VID For: none

New Service VID: 0

Set Customer VID For: all frames

New Customer VID: 200

Set QoS For: none

QoS Group: none

enabled

OK
Cancel
Apply
Disable
Comment
Copy
Remove

Switch → VLAN → Protocol Based VLAN



Protocol based dynamic VLAN

Configure trunk port

Switch Protocol Based VLAN <8863 (pppoe-discovery)>

Frame Type: ethernet

Protocol: 8863 (pppoe-discovery)

Ports: ether1-trunk

Set Service VID For: none

New Service VID: 0

Set Customer VID For: all frames

New Customer VID: 0

Set QoS For: none

QoS Group: none

enabled

OK
Cancel
Apply
Disable
Comment
Copy
Remove

Protocol

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

Switch → VLAN → Protocol Based VLAN



Protocol based dynamic VLAN

Switch → VLAN

Switch VLAN						
VLAN	Eg. VLAN Tag	In. VLAN Tran.	Eg. VLAN Tran.	1:1 VLAN Switching	MAC Based VLAN	Protocol Based VLAN
						<input type="text" value="Find"/>
Protocol	Ports	Set Customer VID For	New Customer VID			
8863 (pppoe-discovery)	ether2-clients	all frames	100			
8864 (pppoe-session)	ether2-clients	all frames	100			
800 (ip)	ether2-clients	all frames	200			
806 (arp)	ether2-clients	all frames	200			
8863 (pppoe-discovery)	ether1-trunk	all frames	0			
8864 (pppoe-session)	ether1-trunk	all frames	0			
800 (ip)	ether1-trunk	all frames	0			
806 (arp)	ether1-trunk	all frames	0			

8 items



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

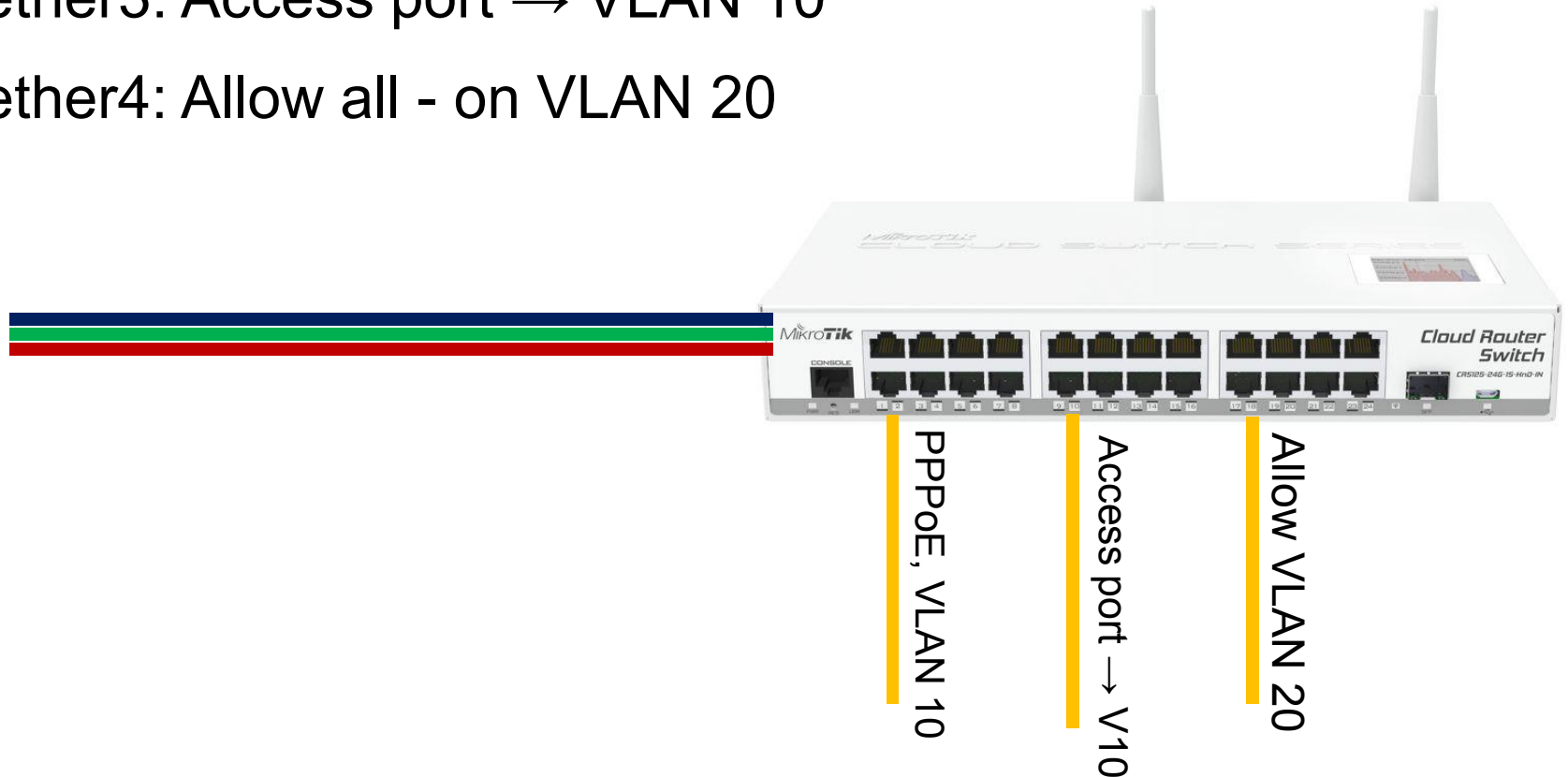


Advanced traffic control (ACL)



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

VLAN ID /	Tagged Ports
10	ether1-trunk, ether2-clients
20	ether1-trunk, ether4-clients

2 items out of 3

Switch → VLAN

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

Switch → ACL

#	Table	Src. Ports	Src. MAC Address/Src. MAC Address	Src. MAC Address/Src. MAC Mask	Action
0	ingress	ether2-clients			forward
1	ingress	ether2-clients			forward
2	ingress	ether2-clients			drop



Advanced traffic control (ACL)

ether2: Only PPPoE with VLAN 10

Switch ACL Rule <ether2-clients>

MAC	VLAN	IP	Action	Bypass
Table: ingress				
<input type="checkbox"/> Invert Match				
Src. Ports: ether2-clients				
Dst. Ports:				
Src. MAC Address				
Dst. MAC Address				
Dst. Addr. Registered:				
MAC Protocol: 8864 (pppoe-session)				
Drop Precedence:				
Custom Fields:				

Buttons: OK, Cancel, Apply, Disable, Comment, Copy, Remove

Switch ACL Rule <ether2-clients>

MAC	VLAN	IP	Action	Bypass
Lookup VID:				
Service VID:				
Service PCP:				
Service DEI:				
Service Tag:				
Customer VID: <input type="checkbox"/> 10				
Customer PCP:				
Customer DEI:				
Customer Tag: tagged				

Action:
forward

Switch → ACL

Switch ACL Rule <ether2-clients>

MAC	VLAN	IP	Action	Bypass
Table: ingress				
<input type="checkbox"/> Invert Match				
Src. Ports: ether2-clients				
Dst. Ports:				
Src. MAC Address				
Dst. MAC Address				
Dst. Addr. Registered:				
MAC Protocol: 8863 (pppoe-discovery)				

Buttons: OK, Cancel, Apply, Disable, Comment, Copy, Remove



Advanced traffic control (ACL)

ether2: Only PPPoE with VLAN 10

Switch ACL Rule <ether2-clients>

MAC	VLAN	IP	Action	Bypass
Table: ingress				
<input type="checkbox"/> Invert Match				
Src. Ports: ether2-clients				
Dst. Ports:				
Src. MAC Address				
Dst. MAC Address				
Dst. Addr. Registered:				
MAC Protocol:				
Drop Precedence:				
Custom Fields:				

OK
Cancel
Apply
Disable
Comment
Copy
Remove

Switch ACL Rule <ether2-clients>

MAC	VLAN	IP	Action	Bypass
Action: drop				
Mirror To:				
Policer:				
Learn SA:				

OK
Cancel
Apply
Disable
Comment

Switch → ACL



Advanced traffic control (ACL)

ether3: Access port → VLAN 10

Ingress VLAN Translation <ether3-clients>

Ports: ether3-clients

Protocol:

Service VLAN Lookup For: any

Service VID:

Service PCP:

Service DEI:

Customer VLAN Lookup For: any

Customer VID: 0

Customer PCP:

Customer DEI:

New Service VID:

New Customer VID: 10

OK
Cancel
Apply
Disable
Comment
Copy
Remove

Switch VLAN

VLAN	Eg. VLAN Tag	In. VLAN Tran.	Eg. VLAN Tran.	1:1 VLAN Switching	MAC Based V
+	-	✓	✗	📄	🔍
Dynamic		is	no		
Ports	Customer VLAN Lookup For	Customer VID	New Customer VID		
ether3-clients	any	0	10		

1 item out of 2

Switch → VLAN → Ingress VLAN Tran.



Advanced traffic control (ACL)

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

Switch ACL Rule <ether4-clients>

MAC	VLAN	IP	Action	Bypass
Table: ingress				
<input type="checkbox"/> Invert Match				
Src. Ports: ether4-clients				
Dst. Ports:				
Src. MAC Address				
Dst. MAC Address				
Dst. Addr. Registered:				
MAC Protocol:				
Drop Precedence:				
Custom Fields:				

OK
Cancel
Apply
Disable
Comment
Copy
Remove

Switch ACL Rule <ether4-clients>

MAC	VLAN	IP	Action	Bypass
Lookup VID:				
Service VID:				
Service PCP:				
Service DEI:				
Service Tag:				
Customer VID: <input type="checkbox"/> 20				
Customer PCP:				
Customer DEI:				
Customer Tag: tagged				
Priority:				

OK
Cancel
Apply
Disable
Comment
Copy
Remove

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
```



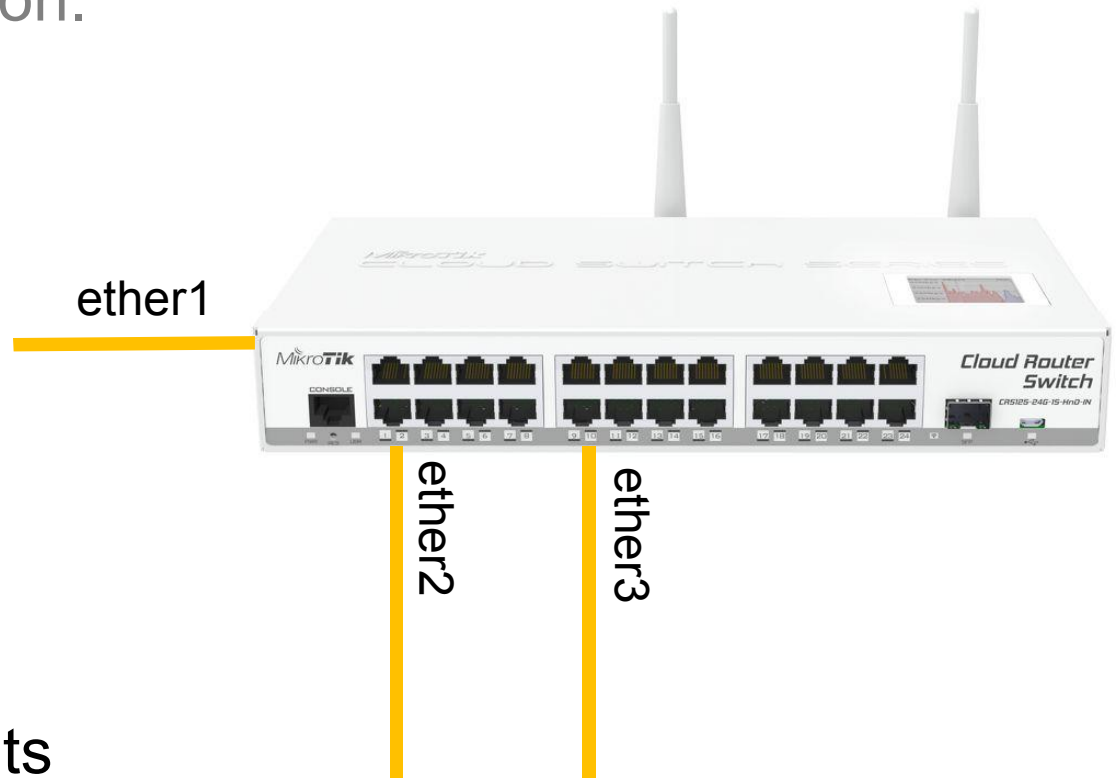
Client isolation



Client isolation

Client communication blocked.

Bridge would use horizon.



- ether1: Uplink
- ether2, ether3: Clients



Client isolation

Switch → Ports → ether1

Switch Port <ether1-trunk>

Generic | Ingress VLAN | Egress VLAN | Mirroring | QoS | Queues | TPIDs | Counters

Name: ether1-trunk

VLAN Type: network port

Isolation Profile: 0 (promiscuous)

Isolation Profile Override: 0 (promiscuous)

Learning

Learning Override: []

Learning Limit: []

Allow Unicast FDB Drop

Allow Unicast Loopback

Allow Multicast Loopback

Action On Static Station Move: forward

Drop Secure Static MAC Move

Drop Dynamic MAC Move

OK

Cancel

Apply

Isolation profile 0

Switch → Ports

Switch Ports

Ports | Trunk | Port Isolation | Port Leakage

Dynamic is no

#	Ports	Type	MAC Profile	Port Profile
0	ether1-trunk	dst	promiscuous	1

1 item out of 3



Client isolation

Switch → Ports → ether2 and ether3

Switch Port <ether2-clients>

Generic | Ingress VLAN | Egress VLAN | Mirroring | QoS | Queues | TPIDs | Counters

Name: ether2-clients

VLAN Type: network port

Isolation Profile: 0 (promiscuous)

Isolation Profile Override: 1 (isolated)

Learning

Learning Override:

Learning Limit:

Allow Unicast FDB Drop

Allow Unicast Loopback

Allow Multicast Loopback

Action On Static Station Move: forward

Drop Secure Static MAC Move

Drop Dynamic MAC Move

OK

Cancel

Apply

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

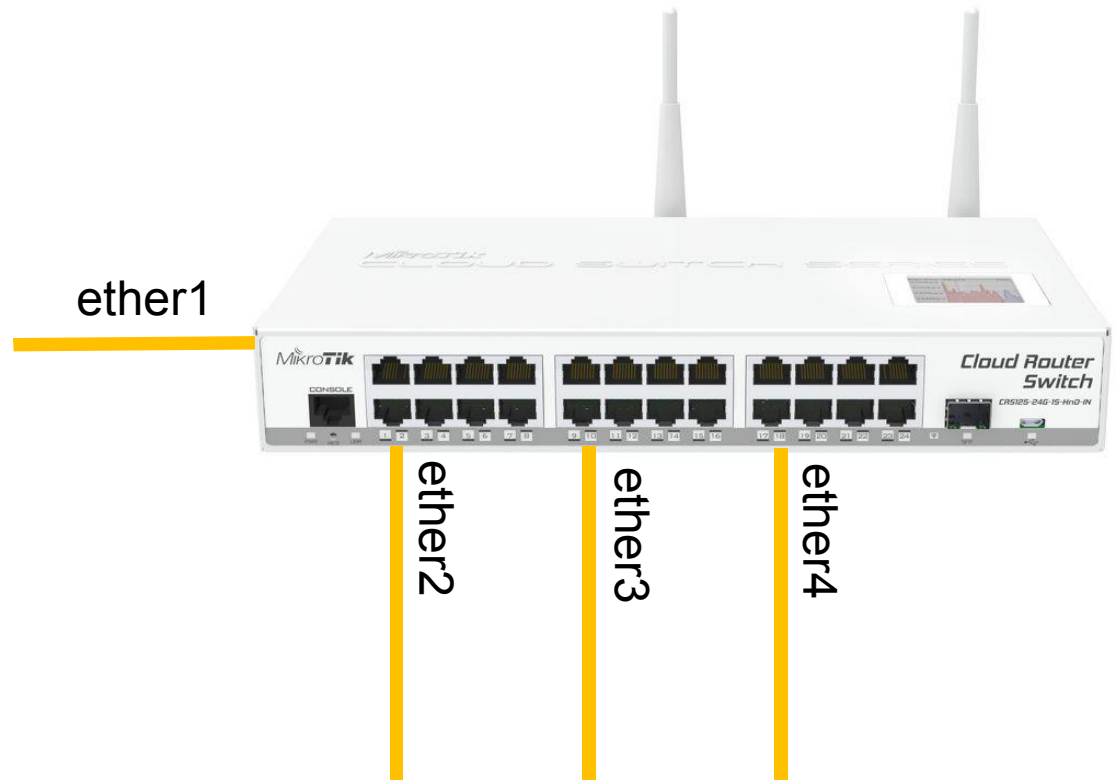
/interface ethernet switch port
set ether1-trunk isolation-leakage-profile-override=0
set ether2-clients isolation-leakage-profile-override=1
set ether3-clients isolation-leakage-profile-override=1

# 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

Switch Port <ether4-clients>

Generic | Ingress VLAN | Egress VLAN | Mirroring | QoS | Queues | TPIDs | Counters

Name: ether4-clients

VLAN Type: network port

Isolation Profile: 0 (promiscuous)

Isolation Profile Override: 2

Learning

Learning Override:

Learning Limit:

Allow Unicast FDB Drop

Allow Unicast Loopback

Allow Multicast Loopback

Action On Static Station Move: forward

Drop Secure Static MAC Move

Drop Dynamic MAC Move

OK

Cancel

Apply

Isolation profile 2



Evil DHCP server(s)

Switch → Ports → Port Isolation

Switch Ports

Ports Trunk Port Isolation Port Leakage

+ - ✓ ✕ [icon] [icon] Find

Dynamic [dropdown] is [dropdown] no [dropdown] + - Filter

#	Ports	Type	MAC Profile	Port Profile
0	ether1-trunk	dst	promiscuous	2

1 item out of 3

Switch Port Isolation <ether1-trunk>

Ports: ether1-trunk [dropdown]

Type: src dst

Forwarding Type: routed bridged

Traffic Type: broadcast multicast unicast

Registration Status: unknown known

Protocol Type: RIPv1 DHCPv6
 DHCPv4 ND
 ARP

MAC Profile: promiscuous [dropdown]

Port Profile: 2 [dropdown]

VLAN Profile: [dropdown]

OK
Cancel
Apply
Disable
Comment
Copy
Remove



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=""
```

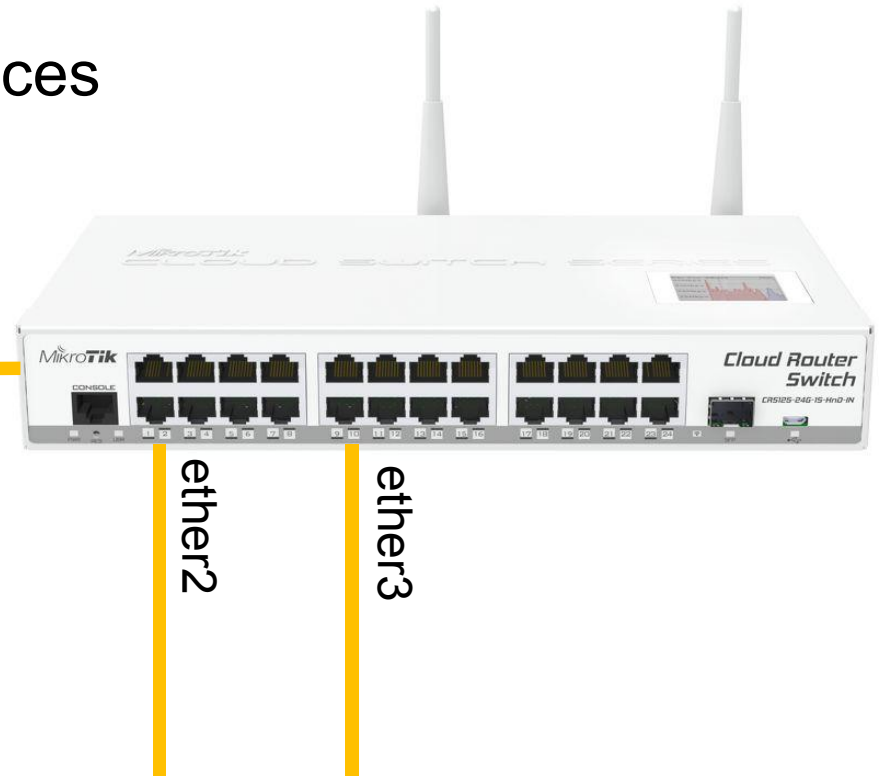


Max throughput #1

Define max. throughput without queues:

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

ether1: Rx 10 Mbps / Tx 1 Mbps





Max throughput #1

Switch Ingress Policer <ether1-uplink>

Port: ether1-uplink

Rate: 10M

Burst: 100k

Meter Unit: bit packet

Meter Length: layer 1 layer 2 layer 3

Packet Types:

- known unicast
- unknown unicast
- registered multicast
- unregistered multicast
- broadcast
- tcp control
- arp or nd

Yellow Action: drop

New DEI For Yellow:

New PCP For Yellow:

New DSCP For Yellow:

Buttons: OK, Cancel, Apply, Copy, Remove

Switch Shaper <ether1-uplink>

Port: ether1-uplink

Target: port

Meter Unit: bit packet

Rate: 1M

Burst: 100k

Buttons: OK, Cancel, Apply, Disable, Comment, Copy, Remove

enabled

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

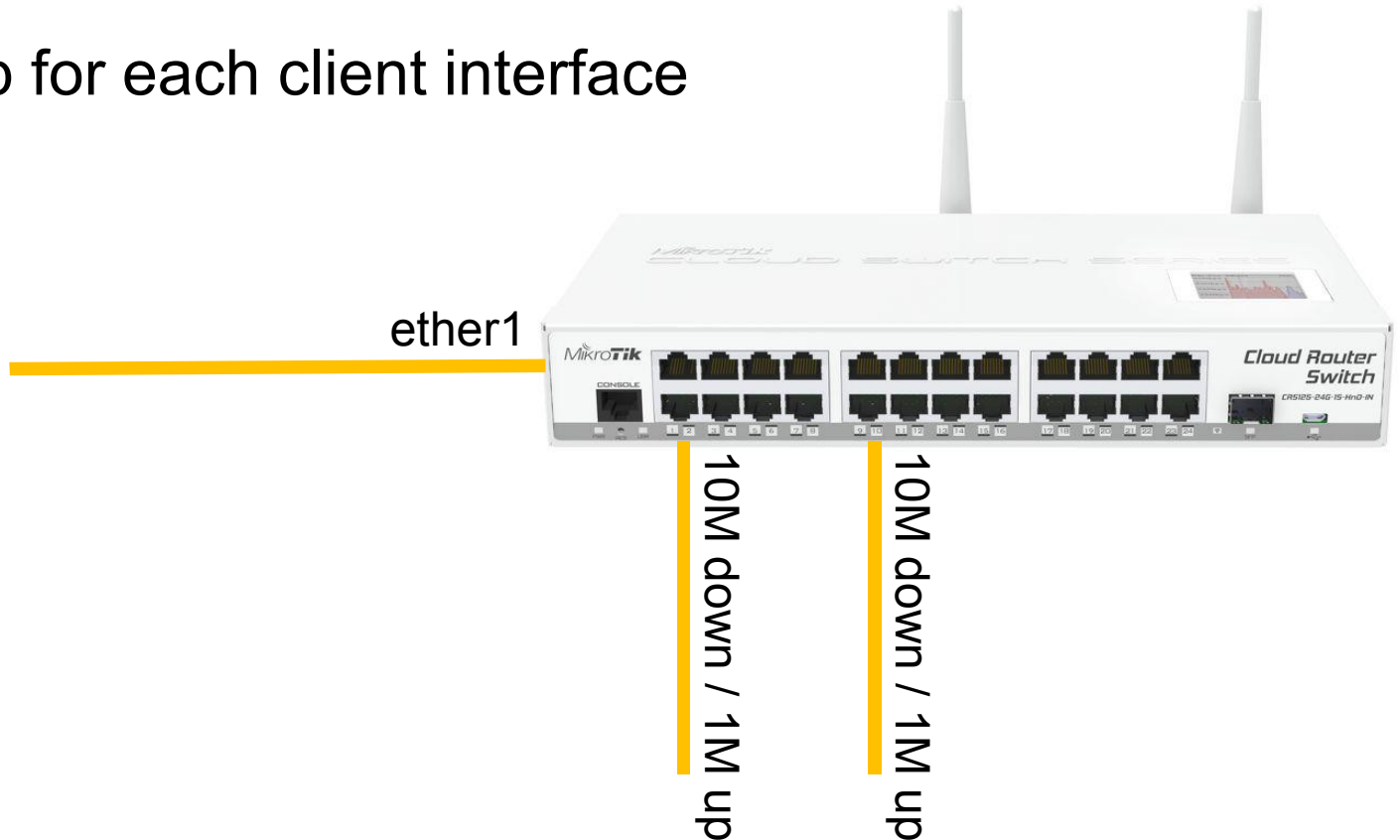
/interface ethernet switch shaper
add port=ether1-uplink meter-unit=bit rate=1M
```



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 Ingress Policer <ether2-clients>

Port: ether2-clients

Rate: 1M

Burst: 100k

Meter Unit: bit packet

Meter Length: layer 1 layer 2 layer 3

Packet Types:

- known unicast
- unknown unicast
- registered multicast
- unregistered multicast
- broadcast
- tcp control
- arp or nd

Yellow Action: drop

New DEI For Yellow:

New PCP For Yellow:

New DSCP For Yellow:

Buttons: OK, Cancel, Apply, Copy, Remove

Switch Shaper <ether2-clients>

Port: ether2-clients

Target: port

Meter Unit: bit packet

Rate: 10M

Burst: 100k

Buttons: OK, Cancel, Apply, Disable, Comment, Copy, Remove

enabled

Switch → QoS → Shaper

Switch → QoS → Ingress Port Policer

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