



VLAN-и в RouterOS

MikroTik Net Camp 2018

Троян

Петър Димитров

За мен:

- ❖ Име: Петър Димитров
- ❖ MikroTik Trainer: от 2013 г.
- ❖ Ubiquiti Trainer: от 2018 г.

❖ Предлагани обучения:

Въведение в компютърните мрежи

MTCNA, MTCWE, MTCRE, MTCTSE, MTCUME, MTCIPv6E, MTCINE

UBWS, UBWA, UBRSS, UBRSA, UEWA



VLAN-и в RouterOS, Петър Димитров

© PG.NET PRO

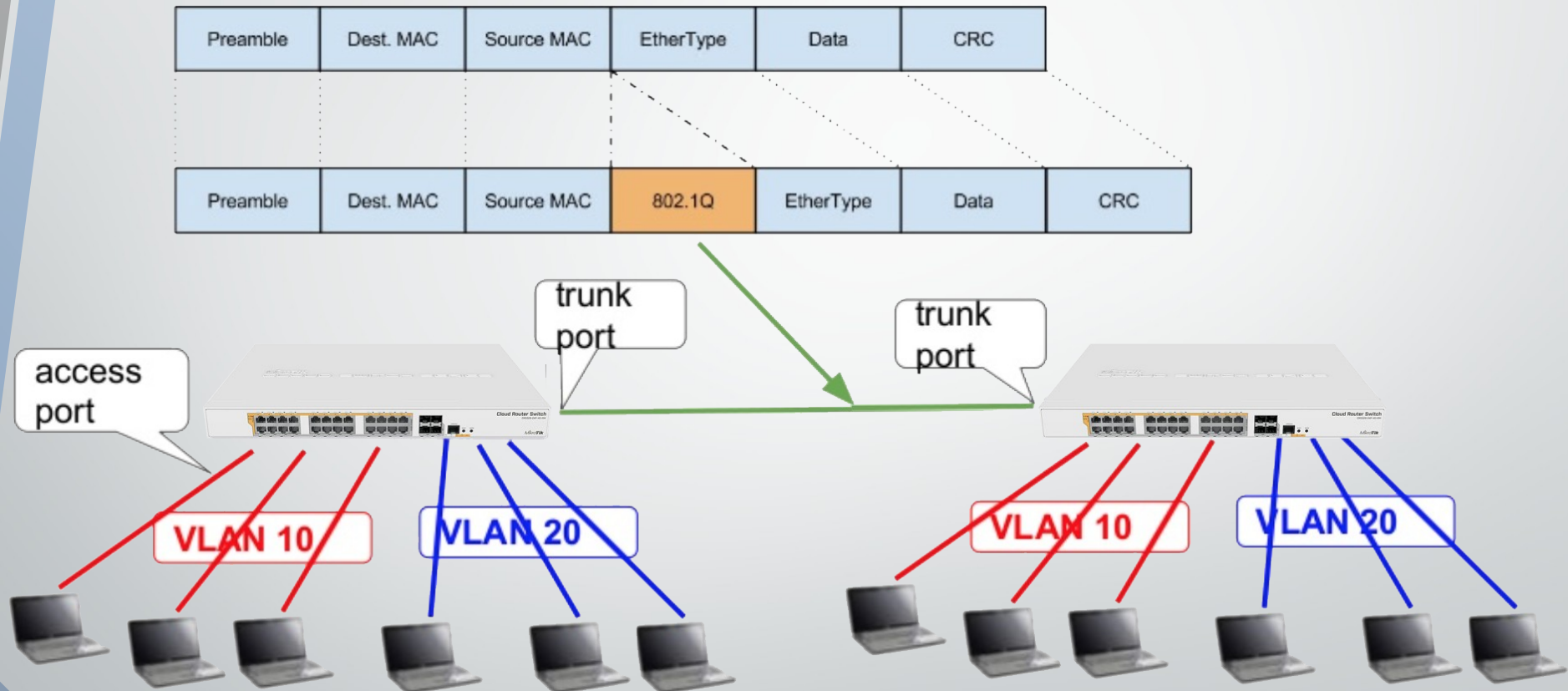
Virtual LAN (802.1Q/ 802.1ad)

- ❖ Virtual LAN (VLAN) позволява мрежовите устройства да бъдат разделени в независими подгрупи, дори да се намират в един Layer 2 сегмент.
- ❖ В Ethernet хедъра се вмъква VLAN tag, който разполага с 12 бита идентификатор (VLAN-и от 0 до 4095) и 3 бита за пренос на CoS.

VLAN и управляеми комутатори

- ❖ Портовете на switch-овете, които поддържат vlan-и, могат да участват към една или повече vlan групи. Всеки порт може да бъде в режим:
 - ❖ tagged – добавя собствения tag при предаване на фрейм, приема само фреймове с неговия tag
 - ❖ untagged (access) – премахва vlan tag-а при предаване на фрейм, приема само фреймове без tag
 - ❖ trunk – tagged порт в няколко vlan групи

VLAN и управляеми комутатори



VLAN на интерфейс на рутер

Interface List

IP Tunnel GRE Tunnel **VLAN** VRRP Bonding LTE

New Interface

General Loop Protect Status Traffic OK

Name: vlan1 Cancel

Type: VLAN Apply

MTU: 1500 Disable

Actual MTU: Comment

L2 MTU: Copy

MAC Address: Remove

ARP: enabled ARP Timeout: Torch

VLAN ID: 123

Interface: ether1-LAN

Use Service Tag

enabled running slave

802.1Q (vlan ethertype=0x8100)
Позволява употребата на 1 tag

Interface List

IP Tunnel GRE Tunnel **VLAN** VRRP Bonding LTE

New Interface

General Loop Protect Status Traffic OK

Name: vlan1 Cancel

Type: VLAN Apply

MTU: 1500 Disable

Actual MTU: Comment

L2 MTU: Copy

MAC Address: Remove

ARP: enabled ARP Timeout: Torch

VLAN ID: 123

Interface: ether1-LAN

Use Service Tag

enabled running slave

802.1ad (QinQ, vlan ethertype=0x88a8)
Позволява употребата на вложени tag-ове

VLAN-и в RouterOS, Петър Димитров

VLAN на интерфейс на рутер

- ❖ VLAN може да бъде добавян в Bridge.
- ❖ VLAN не е тунелиращ протокол, не съдържа отделни полета за хардуерен адрес на източник и местоназначение
- ❖ Ако интерфейси на рутера са добавени като портове на Bridge, то тогава създаваме VLAN интерфейсите на Bridge-а.

VLAN на интерфейс на рутер



- ❖ Ще бъдат създадени няколко VLAN интерфейса с различни id-та на рутер
- ❖ На тях ще бъдат вдигнати различни IP адреси
- ❖ Ще бъдат осигурени DHCP сървъри за всеки VLAN
- ❖ Така подготвената конфигурация ще бъде използвана и в следващите демонстрации

VLAN и switch chip на routerboard

- ❖ Портовете на routerboard към някои switch chip-ове поддържат работа с vlan-и
- ❖ Режим на работа с кадри (frames) с vlan tag-ове се указва за всеки порт в менюто `/interface ethernet switch port`
- ❖ Таблицата с vlan-и се управлява от менюто `/interface ethernet switch vlan`

VLAN и switch chip на routerboard

- ❖ Vlan mode – при приемане на кадри (frames)
 - ❖ **disabled** - не проверява vlan таблицата, третира всички кадри (frames) все едно нямат vlan tag.
 - ❖ **fallback** (режима по подразбиране) - третира кадрите (frames) с неописани във vlan таблицата tag-ове все едно нямат vlan tag. Тези, описани в таблицата, но идващи на порт, който не е описан, не се унищожават.
 - ❖ **check** - кадрите (frames) с неописани във vlan таблицата tag-ове, се унищожават. Тези, описани в таблицата, но идващи на порт, който не е описан, не се унищожават.
 - ❖ **secure** - кадрите (frames) с неописани във vlan таблицата tag-ове или идващи на порт, който не е описан, се унищожават.

VLAN и switch chip на routerboard

- ❖ Vlan-header – при предаване на кадри (frames)
- ❖ **leave-as-is** - при предаване vlan tag-а не се променя.
- ❖ **always-strip** - ако има vlan tag, при предаване tag-а се премахва
- ❖ **add-if-missing** - ако няма vlan tag, при предаване се добавя собствения vlan tag на port-а
- ❖ Default VLAN ID – собствен VLAN на порта

VLAN и switch chip на routerboard



- ❖ На рутер със switch chip ще бъдат "поети" подготвените в предната демонстрация VLAN-и:
 - ❖ Порта за връзка към другия рутер ще бъде конфигуриран като trunk порт
 - ❖ Ще бъдат конфигурирани различни портове като access портове за различните VLAN-и

VLAN и switch chip на routerboard



```
/interface bridge
add name=bridge1 protocol-mode=none
/interface bridge port
add bridge=bridge1 interface=ether2
add bridge=bridge1 interface=ether3
add bridge=bridge1 interface=ether4
add bridge=bridge1 interface=ether5
```

```
/interface ethernet switch vlan
add ports=ether2,ether5 switch=switch1 vlan-id=10
add ports=ether3,ether5 switch=switch1 vlan-id=20
add ports=ether4,ether5 switch=switch1 vlan-id=30
```

```
/interface ethernet switch port
set ether2 vlan-mode=secure vlan-header=always-strip default-vlan-id=10
set ether3 vlan-mode=secure vlan-header=always-strip default-vlan-id=20
set ether4 vlan-mode=secure vlan-header=always-strip default-vlan-id=30
set ether5 vlan-mode=secure vlan-header=add-if-missing
```

VLAN и новия Bridge

- ❖ Във всички версии 6.41+ на RouterOS има нова имплементация на Bridge.
- ❖ Разполагаме с аналогична на поддържаната от switch chip-овете на малките рутери функционалност по отношение на VLAN-ите със сходна логика на конфигуриране.
- ❖ Въпреки, че switch chip-а може да работи с VLAN-и, ако използвате тази функционалност в Bridge, с изключение на CRS3xx всички останали рутери спират да ползват Hardware Offload!

VLAN и новия Bridge

Interface <bridge1>

General STP **VLAN** Status Traffic

VLAN Filtering

PVID: 1

Bridge Port <ether3>

General STP **VLAN** Status

PVID: 10

Frame Types: admit all

Ingress Filtering

enabled

enabled inactive Hw. Offload

Bridge VLAN <10>

Bridge: bridge1

VLAN IDs: 10

Tagged: ether5

Untagged: ether3

Current Tagged:

Current Untagged:

enabled

VLAN и новия Bridge



- ❖ Да направим отново port-based VLAN като в предната демонстрация, но този път използвайки само Bridge:

```
/interface bridge
add fast-forward=no name=bridge1 protocol-mode=none vlan-filtering=yes
/interface bridge port
add bridge=bridge1 ingress-filtering=yes interface=ether2 pvid=10
add bridge=bridge1 ingress-filtering=yes interface=ether3 pvid=20
add bridge=bridge1 ingress-filtering=yes interface=ether4 pvid=30
add bridge=bridge1 ingress-filtering=yes interface=ether5
/interface bridge vlan
add bridge=bridge1 tagged=ether5 untagged=ether2 vlan-ids=10
add bridge=bridge1 tagged=ether5 untagged=ether3 vlan-ids=20
add bridge=bridge1 tagged=ether5 untagged=ether4 vlan-ids=30
```


VLAN при CRS3xx

- ❖ При 3-та серия CRS работата с VLAN-и се извършва чрез Bridge функционалността.
- ❖ Активирането на VLAN filtering при CRS3xx не изключва Hardware Offload
- ❖ Менюто switch е подобно на това при рутерите, но липсва работа с VLAN-и, там има само специфичните за свича настройки като ограничения на скоростта, управление на broadcast и multicast и др.

VLAN при CRS3xx



- ❖ Можем да направим port-based VLAN чрез Bridge по същия начин. Резултатата:

The screenshot displays the RouterOS configuration interface for a Bridge. The main window shows the 'Bridge' configuration for 'bridge1' with the 'Ports' tab selected. A table lists the configured ports, with the 'Hardware Offload' column highlighted in green. Below this, the 'Interface <bridge1>' configuration window is open, showing 'VLAN Filtering' checked and 'PVID' set to 1. To the right, a smaller window shows the 'Bridge' configuration for 'bridge1' with the 'VLANs' tab selected, displaying a table of configured VLANs.

#	Interface	Bridge	Hardware Offload	Priority ...	Path ...	PVID	Role	Root Pat...
0	H ether2	bridge1	yes	80	10	10	designated port	
1	H ether3	bridge1	yes	80	10	20	designated port	
2	H ether4	bridge1	yes	80	10	30	designated port	
3	H ether5	bridge1	yes	80	10	1	designated port	

Bridge	VLAN IDs	Current Tagged	Current Untagged
bridge1	1		bridge1, ether5
bridge1	10	ether5	ether2
bridge1	20	ether5	ether3
bridge1	30	ether5	ether4

VLAN при CRS1xx и CRS2xx

- ❖ При CRS1xx и CRS2xx работата с VLAN-и се извършва чрез Switch менюто.
- ❖ До версия 6.40.x включително за използване на switch функционалността интерфейсите се конфигурираха с master-port, от версия 6.41 интерфейсите се добавят като портове на Bridge
- ❖ От switch менюто конфигурацията е малко по-сложна, но към момента дава и повече опции

VLAN при CRS1xx и CRS2xx

Меню /interface ethernet switch vlan

- ❖ Дефинира VLAN-и
- ❖ Определя кои портове участват в дефинираните VLAN-и
- ❖ Задава настройки за съответните VLAN-и

VLAN при CRS1xx и CRS2xx

- ❖ Меню /interface ethernet switch vlan
 - ❖ Дефинира VLAN-и
 - ❖ Определя кои портове участват в дефинираните VLAN-и
- ❖ Меню /interface ethernet switch egress-vlan-tag
 - ❖ Определя кои от портовете ще работят с tag-нати VLAN-и

VLAN при CRS1xx и CRS2xx

- ❖ Меню /interface ethernet switch ingress-vlan-translation
 - ❖ Правила за манипулиране на входящия трафик
- ❖ Меню /interface ethernet switch egress-vlan-translation
 - ❖ Правила за манипулиране на изходящия трафик

VLAN при CRS1xx и CRS2xx



- ❖ За да направим port based VLAN като в предните демонстрации, ще е необходимо:
 - ❖ Да създадем Bridge и да добавим като портове в него всички участващи интерфейси
 - ❖ Да дефинираме VLAN-и 10, 20 и 30
 - ❖ Да укажем ether5 като tagged порт за всеки VLAN
 - ❖ Да създадем за ether2, ether3 и ether4 правила за входящ/изходящ трафик, които добавят/премахват съответните VLAN-и

VLAN при CRS1xx и CRS2xx



```
/interface bridge
add name=bridge1 protocol-mode=none
/interface bridge port
add bridge=bridge1 interface=ether2
add bridge=bridge1 interface=ether3
add bridge=bridge1 interface=ether4
add bridge=bridge1 interface=ether5
```

```
/interface ethernet switch vlan
add ports=ether2,ether5 vlan-id=10
add ports=ether3,ether5 vlan-id=20
add ports=ether4,ether5 vlan-id=30
```

```
/interface ethernet switch egress-vlan-tag
add tagged-ports=ether5 vlan-id=10
add tagged-ports=ether5 vlan-id=20
add tagged-ports=ether5 vlan-id=30
```


VLAN при CRS1xx и CRS2xx



```
/interface ethernet switch ingress-vlan-translation
add ports=ether2 new-customer-vid=10
add ports=ether3 new-customer-vid=20
add ports=ether4 new-customer-vid=30
```

```
/interface ethernet switch egress-vlan-translation
add ports=ether2 customer-vid=10 new-customer-vid=0
add ports=ether3 customer-vid=20 new-customer-vid=0
add ports=ether4 customer-vid=30 new-customer-vid=0
```

MAC based VLAN

- ❖ CRS1xx и CRS2xx поддържат определяне на VLAN на база MAC адрес
- ❖ Това позволява да направим конфигурация, при която определено устройство "живее" в определен VLAN, независимо на кой порт е свързано

MAC based VLAN

- ❖ Освен гореописаните процеси за vlan* и egress-vlan-tag, за работа с MAC based VLAN:
 - ❖ В конфигурацията на портовете разрешете MAC Based VLAN Translate
 - ❖ В MAC Based VLAN опишете желаните MAC адреси в съответните VLAN-и

* Уверете се, че VLAN-ите, които ще ползвате, са разрешени на всички портове, на които може да се появят съответните MAC адреси

MAC based VLAN



- ❖ Да направим конфигурация, при която XX:XX:XX:XX:XX:XX живее във VLAN 10:

```
/interface ethernet switch vlan set  
ports=ether2,ether3,ether4,ether5 [find vlan-id=10]
```

```
/interface ethernet switch port  
set ether2 allow-fdb-based-vlan-translate=yes  
set ether3 allow-fdb-based-vlan-translate=yes  
set ether4 allow-fdb-based-vlan-translate=yes
```

```
/interface ethernet switch mac-based-vlan  
add new-customer-vid=10 src-mac-address=XX:XX:XX:XX:XX:XX
```

Благодаря за
вниманието!