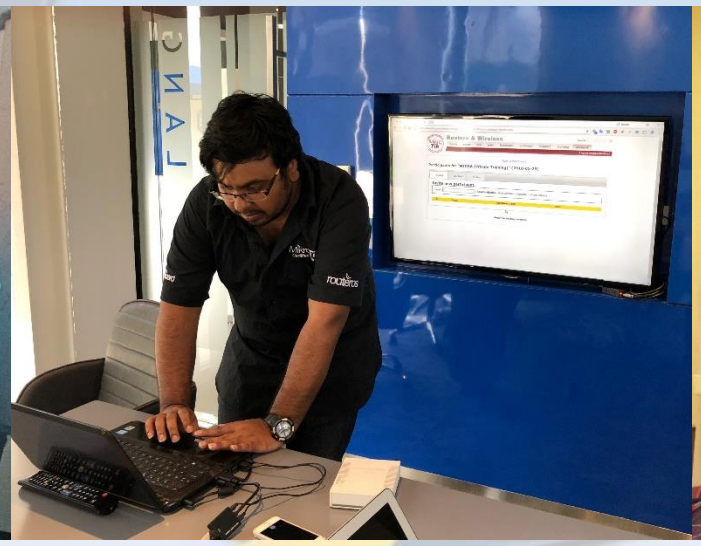
Switching, VLAN, QinQ in Ros 6.41 Onwards and their application to CRS 3.xx models.

SOUMIL GUPTA BHAYA

Mikortik Certified Trainer

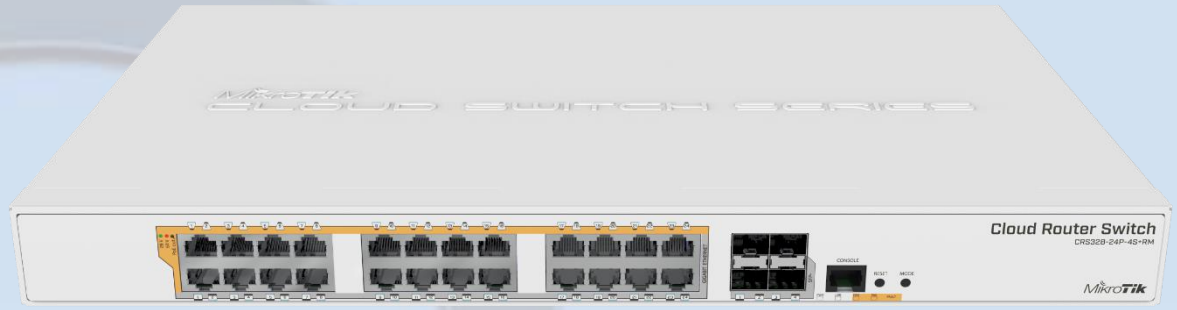
About


- MTCNA, MTCWE, MTCTCE, MTCRE, MTCINE, MTCIPV6E
- Ten years of Mikrotik Experience
- Mikrotik Certified Trainer Since 2012

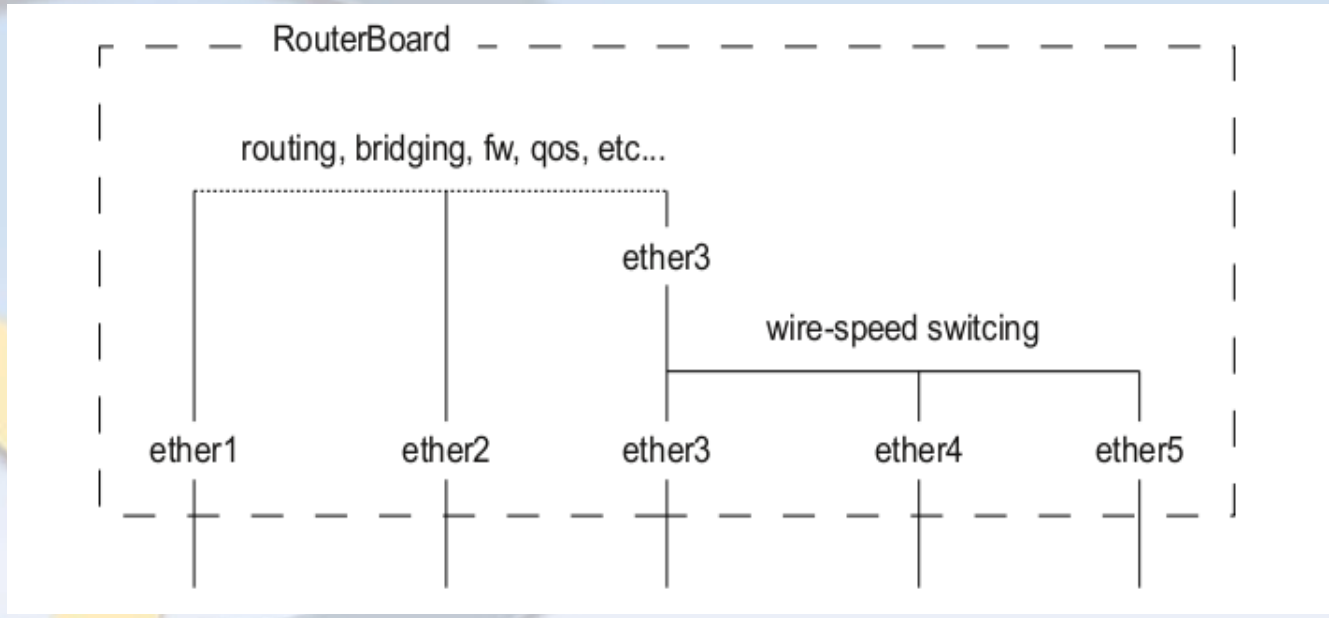
Blinknet Solutions Pvt. Ltd.

Switching



- Switching feature allows wire speed traffic passing among a group of ports.
- We configure this feature by setting a "master-port" property to one or more ports in /interface ethernet menu.
- Since RouterOS v6.41 RouterBoard master-port configuration is converted into a bridge with hardware offloading.
- Bridges will handle all Layer2 forwarding and the use of switch chip **(hw-offload)** will automatically turn on if appropriate conditions are met.

Example with Master Port (<6.41)




```
[admin@MikroTik] > interface ethernet export
/interface ethernet
set [ find default-name=ether4 ] master-port=ether3
set [ find default-name=ether5 ] master-port=ether2
[admin@MikroTik] > interface ethernet print

Flags: X - disabled, R - running, S - slave
 #    NAME        MTU MAC-ADDRESS       ARP        MASTER-PORT      SWITCH
 0 R  ether1      1500 D4:CA:6D:E2:64:64 enabled      none             switch1
 1 R  ether2      1500 D4:CA:6D:E2:64:65 enabled      none             switch1
 2 R ether3       1500 D4:CA:6D:E2:64:66 enabled      none             switch1
 3 RS ether4      1500 D4:CA:6D:E2:64:67 enabled      ether3           switch1
 4 RS ether5      1500 D4:CA:6D:E2:64:68 enabled      ether3           switch1
```

“HW-OFFLOAD”

- By default all newly created bridge ports have hw=yes option and it allows enabling of hw-offload when possible.

RouterBoard/[Switch Chip] Model	Features in Switch menu	Bridge STP/RSTP	Bridge MSTP	Bridge IGMP Snooping	Bridge VLAN Filtering	Bonding
CRS3xx series	+	+	+	+	+	+
CRS1xx/CRS2xx series	+	+	-	+	-	-
[QCA8337]	+	+	-	-	-	-
[AR8327]	+	+	-	-	-	-
[AR8227]	+	+	-	-	-	-
[AR8316]	+	+	-	-	-	-
[AR7240]	+	+	-	-	-	-
[MT7621]	+	-	-	-	-	-
RB1100AHx4 [RTL8367]	+	-	-	-	-	-
[ICPlus175D]	+	-	-	-	-	-

“**+**” :- Enabling this feature **maintains** hw-offload. || “**-**” :- Enabling this feature **turns off** hw-offload.

Example with Bridge HW offloading:

```
admin@MikroTik] > interface bridge export
/interface bridge
add name=bridge1 igmp-snooping=no  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
 [admin@MikroTik] > interface bridge port print

Flags: X - disabled, I - inactive, D - dynamic, H - hw-offload
 #    INTERFACE          BRIDGE          HW  PVID PRIORITY  PATH-COST INTERNAL-PATH-COST
 0  H ether2             bridge1         yes    1    0x80       10           10      none
 1  H ether3             bridge1         yes    1    0x80       10           10      none
 2  H ether4             bridge1         yes    1    0x80       10           10      none
 3  H ether5             bridge1         yes    1    0x80       10           10      none
```

Port isolation

- Since RouterOS v6.43rc11 it is possible to create an uplink port and isolated ports.
- Allows each device connected to a switch port to be isolated from other ports.
- Devices are only capable of communicating with other devices through the uplink port.
- Filter unwanted packets and limit access between devices that are behind switch ports.

Port Isolation

/interface bridge port

add interface=sfp1 bridge=bridge1 hw=yes

add interface=ether1 bridge=bridge1 hw=yes

add interface=ether2 bridge=bridge1 hw=yes

add interface=ether3 bridge=bridge1 hw=yes

Override the egress port for each switch port that needs to be isolated (excluding the uplink port):

/interface ethernet switch port-isolation

set ether1 forwarding-override=sfp1

set ether2 forwarding-override=sfp1

set ether3 forwarding-override=sfp1

Note: It is possible to set multiple uplink ports for a single switch chip, this can be done by specifying multiple interfaces and separating them with a comma.

VLAN

- Virtual Local Area Network (VLAN).
- Layer 2 method that allows multiple Virtual LANs on a single physical interface (ethernet, wireless, etc.).
- Ability to segregate LANs efficiently.
- Each VLAN is treated as a separate subnet.
- A trunk carries the traffic of multiple VLANs.

VLAN (Trunk and Access Ports)

/interface bridge

add name=bridge1 igmp-snooping=no protocol-mode=none

/interface bridge port

add bridge=bridge1 interface=ether2 hw=yes

add bridge=bridge1 interface=ether3 hw=yes

add bridge=bridge1 interface=ether4 hw=yes

add bridge=bridge1 interface=ether5 hw=yes

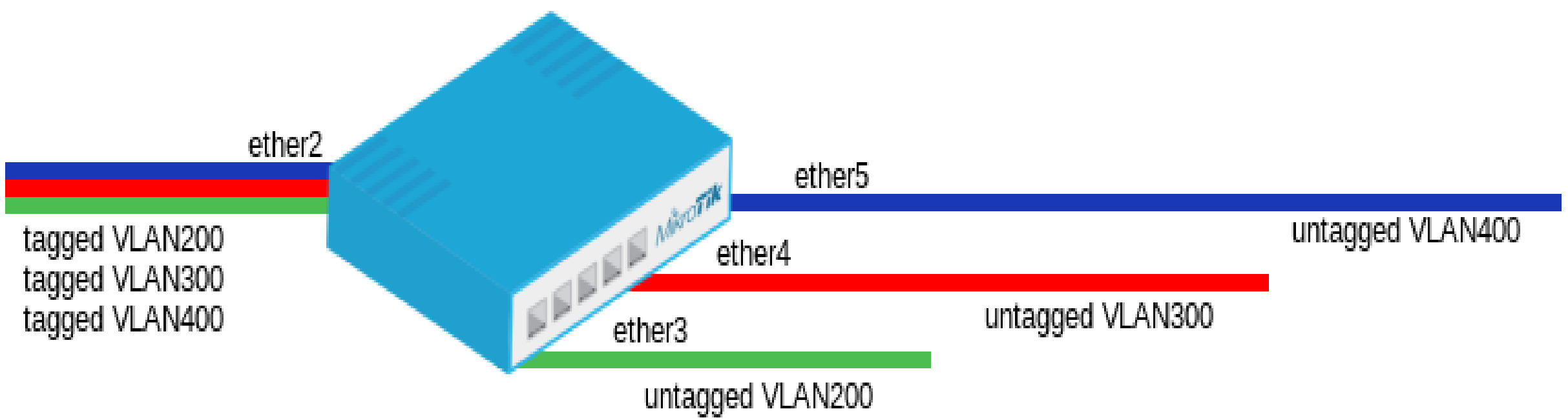
Add VLAN table entries to allow frames with specific VLAN IDs between ports.

/interface ethernet switch vlan

add ports=ether2,ether3 switch=switch1 vlan-id=200

add ports=ether2,ether4 switch=switch1 vlan-id=300

add ports=ether2,ether5 switch=switch1 vlan-id=400



VLAN - Settings

- We assign "vlan-mode" and "vlan-header" mode for each port and also "default-vlan-id" on ingress for each access port.
- Setting "vlan-mode=secure" ensures strict use of VLAN table.
- Setting "vlan-header=always-strip" for access ports removes VLAN header from frame when it leaves the switch chip.
- Setting "vlan-header=add-if-missing" for trunk port adds VLAN header to untagged frames.
- "Default-vlan-id" specifies what VLAN ID is added for untagged ingress traffic of the access port.

VLAN (Example Contd.)

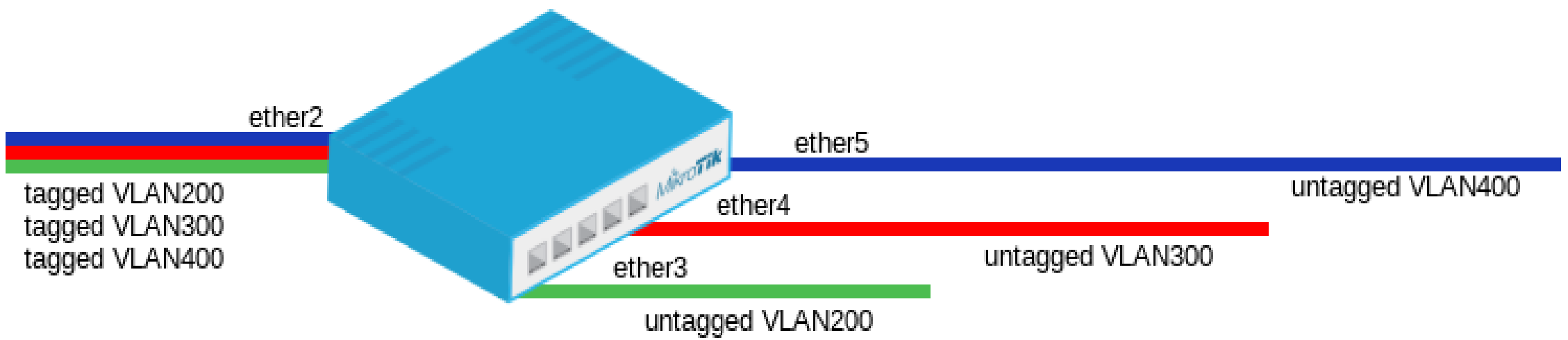
/interface ethernet switch port

set ether2 vlan-mode=secure vlan-header=add-if-missing

set ether3 vlan-mode=secure vlan-header=always-strip default-vlan-id=200

set ether4 vlan-mode=secure vlan-header=always-strip default-vlan-id=300

set ether5 vlan-mode=secure vlan-header=always-strip default-vlan-id=400



VLAN (Trunk and Hybrid Ports)

- VLAN Hybrid ports which can forward both tagged and untagged traffic are supported only by some Gigabit switch chips (QCA8337, AR8327)

/interface bridge

add name=bridge1 igmp-snooping=no protocol-mode=none

/interface bridge port

add bridge=bridge1 interface=ether2 hw=yes

add bridge=bridge1 interface=ether3 hw=yes

add bridge=bridge1 interface=ether4 hw=yes

add bridge=bridge1 interface=ether5 hw=yes

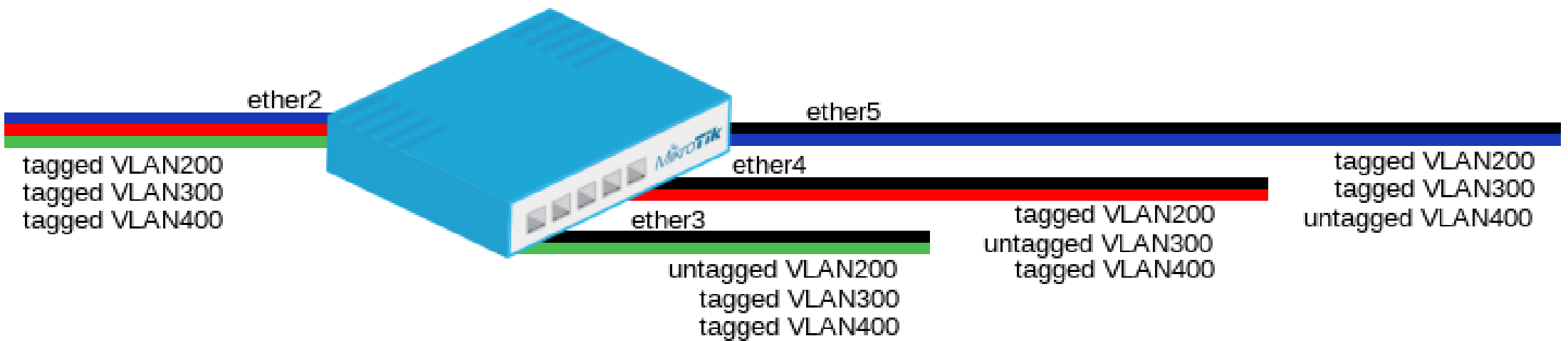
Add VLAN table entries to allow frames with specific VLAN IDs between ports.

/interface ethernet switch vlan

add ports=ether2,ether3,ether4,ether5 switch=switch1 vlan-id=200

add ports=ether2,ether3,ether4,ether5 switch=switch1 vlan-id=300

add ports=ether2,ether3,ether4,ether5 switch=switch1 vlan-id=400



VLAN (Example Contd.)

```
/interface ethernet switch port
set ether2 vlan-mode=secure vlan-header=leave-as-is
set ether3 vlan-mode=secure vlan-header=leave-as-is default-vlan-id=200
set ether4 vlan-mode=secure vlan-header=leave-as-is default-vlan-id=300
set ether5 vlan-mode=secure vlan-header=leave-as-is default-vlan-id=400
```

- In Gigabit switch chips when "vlan-mode=secure", it ignores switch port "vlan-header" options.
- VLAN table entries handle all the egress tagging/untagging and works as "vlan-header=leave-as-is" on all ports.
- It means what comes in tagged, goes out tagged as well, only "default-vlan-id" frames are untagged at the egress of port.

Management Port Configuration

- Management port needed to access router when using VLAN

/interface bridge
add name=bridge1 protocol-mode=none
/interface bridge port
add interface=ether1 bridge=bridge1 hw=yes
add interface=ether2 bridge=bridge1 hw=yes
In these examples it will be assumed that ether1 is the trunk port and ether2 is the access port, for configuration as the following:
/interface ethernet switch port
set ether1 vlan-header=add-if-missing
set ether2 default-vlan-id=100 vlan-header=always-strip
/interface ethernet switch vlan
add ports=ether1,ether2,switch1-cpu switch=switch1 vlan-id=100

Management port configuration (Tagged)

- In order to make the device accessible only from a certain VLAN, you need to create a new VLAN interface on the bridge/master-port interface and assign an IP address to it:

```
/interface vlan
add name=MGMT vlan-id=99 interface=bridge1
/ip address
add address=192.168.99.1/24 interface=MGMT
```

Specify from which interfaces it is allowed to access the device:

```
/interface ethernet switch vlan
add ports=ether1,switch1-cpu switch=switch1 vlan-id=99
```

- When VLAN table is configured, you can enable vlan-mode=secure to limit access to the CPU:

```
/interface ethernet switch port
set ether1 vlan-header=add-if-missing vlan-mode=secure
set ether2 default-vlan-id=100 vlan-header=always-strip vlan-mode=secure
set switch1-cpu vlan-header=leave-as-is vlan-mode=secure
```

Management Port Configuration (Untagged)

- In order to make the device accessible from the access port, create a VLAN interface with the same VLAN ID as set in default-vlan-id, for example VLAN 100, and add an IP address to it:

/interface vlan

add name=VLAN100 vlan-id=100 interface=bridge1

/ip address

add address=192.168.100.1/24 interface=VLAN100

Specify which access (untagged) ports are allowed to access the CPU:

/interface ethernet switch vlan

add ports=ether1,ether2,switch1-cpu switch=switch1 vlan-id=100

Management Port Configuration (Untagged)

- It is possible to allow access to the device from the trunk (tagged) port with untagged traffic.
- To do so, assign an IP address on the bridge/master-port interface.

/ip address

add address=10.0.0.1/24 interface=bridge1

Specify the trunk port to be able to access the CPU for the default-vlan-id for the trunk port, by default it is set to 1:

/interface ethernet switch vlan

add ports=ether1,switch1-cpu switch=switch1 vlan-id=1

When VLAN table is configured, you can enable vlan-mode=secure to limit access to the CPU:

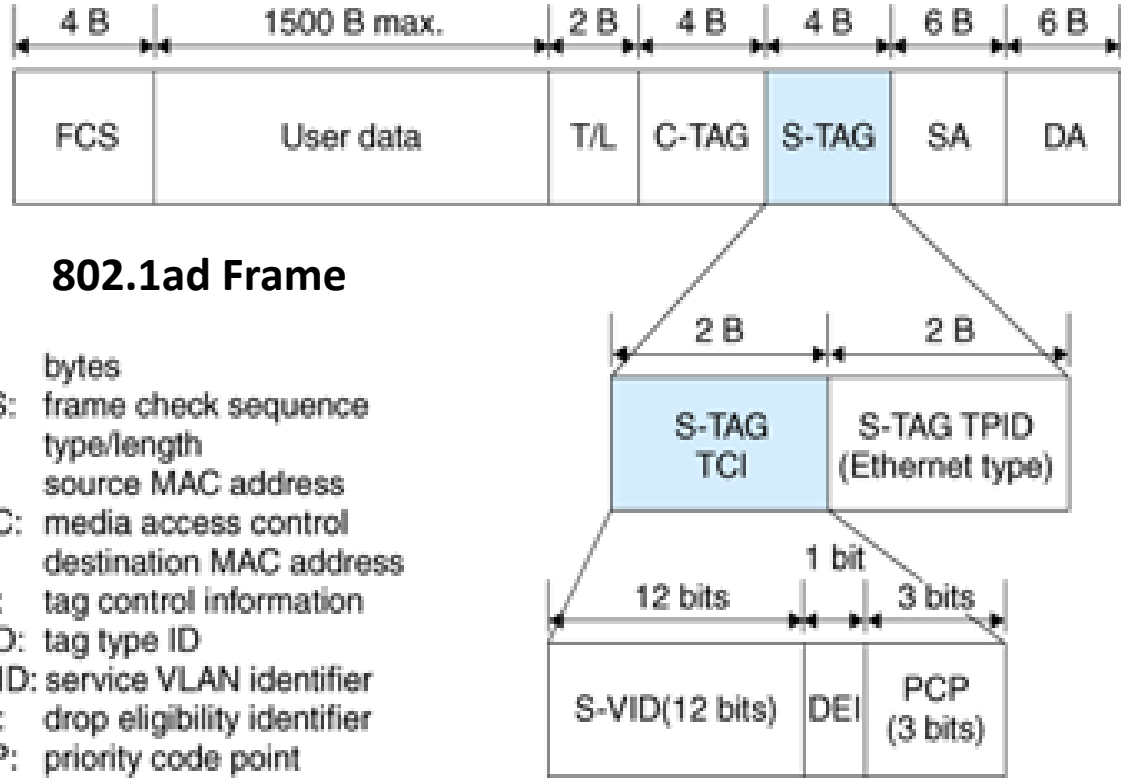
/interface ethernet switch port

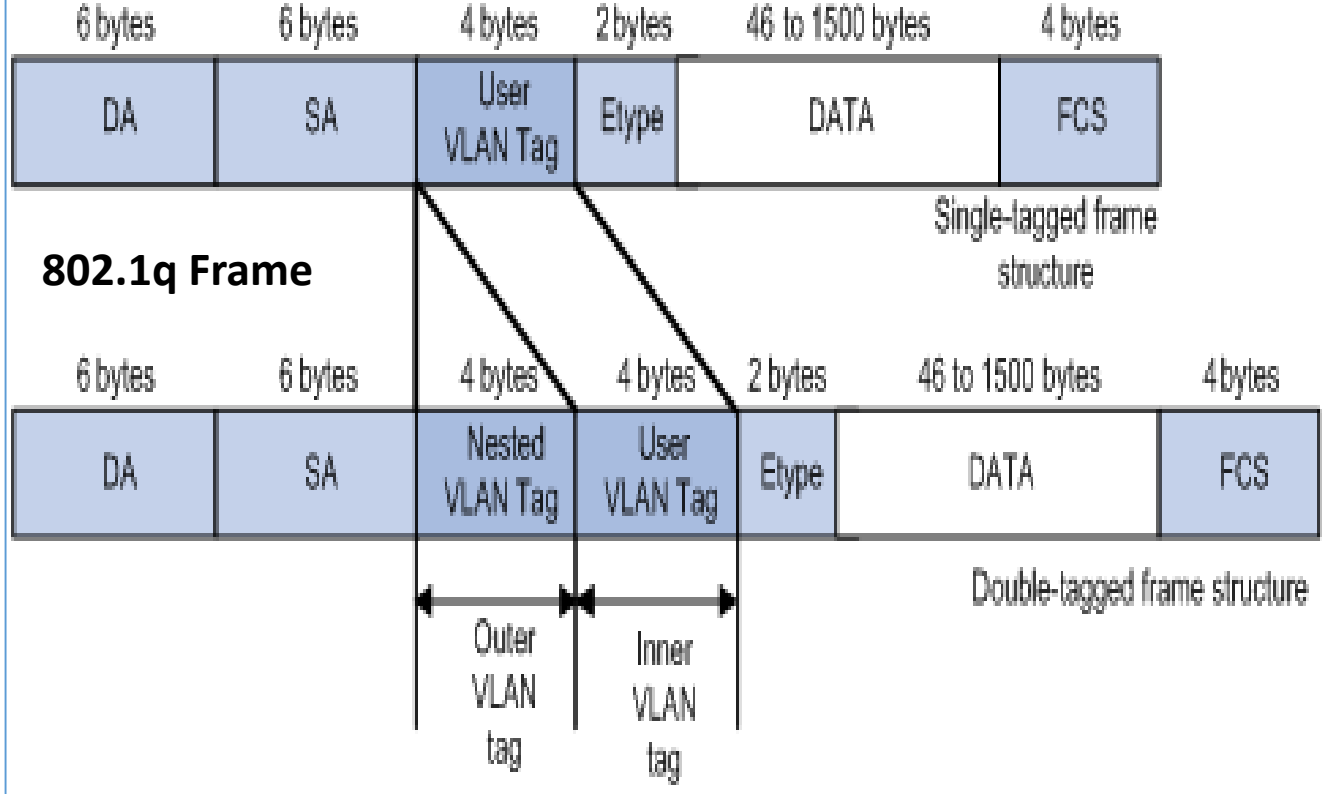
set ether1 default-vlan-id=1 vlan-header=add-if-missing vlan-mode=secure

set switch1-cpu vlan-header=leave-as-is vlan-mode=secure

VLAN Tunneling (Q-in-Q)

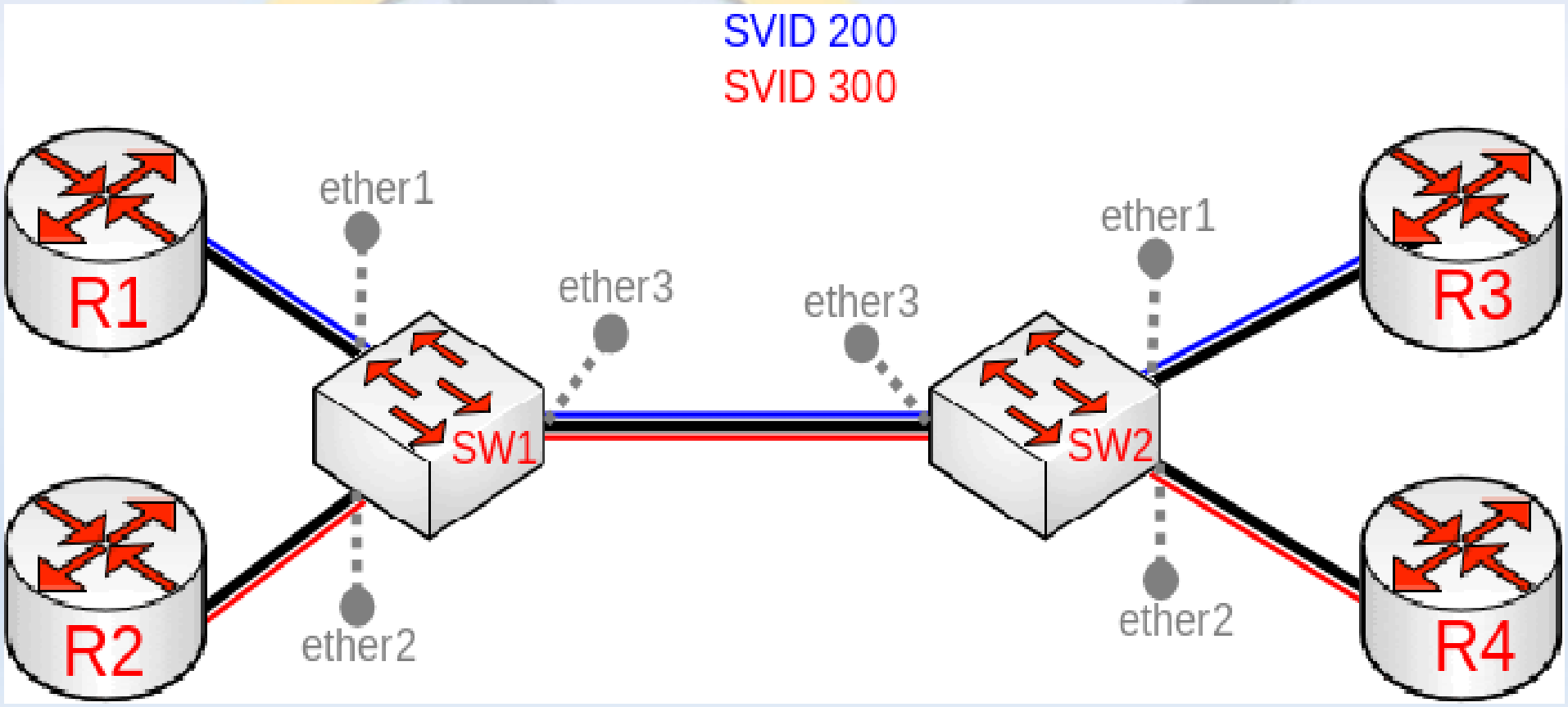
- Since RouterOS v6.43rc14 the RouterOS bridge is IEEE 802.1ad compliant.
- It is possible to filter VLAN IDs based on Service VLAN ID (0x88A8) rather than Customer VLAN ID (0x8100).
- The same principals can be applied as with IEEE 802.1Q VLAN filtering





VLAN Tunneling (Q-in-Q) 802.11ad

- In this example R1, R2, R3 and R4 might be sending any VLAN tagged traffic by 802.1Q (CVID), but SW1 and SW2 needs isolate traffic between routers in a way that R1 is able to communicate only with R3 and R2 is only able to communicate with R4



VLAN Tunneling (Q-in-Q) 802.11ad

- Tag all ingress traffic with a SVID and only allow these VLANs on certain ports.
- Start by enabling 802.1ad VLAN protocol on the bridge, use these commands on SW1 and SW2:

```
/interface bridge
add name=bridge1 vlan-filtering=no ether-type=0x88a8
```

In this setup ether1 and ether2 are going to be access ports (untagged), use the pvid parameter to tag all ingress traffic on each port:

```
/interface bridge port
add interface=ether1 bridge=bridge1 pvid=200
add interface=ether2 bridge=bridge1 pvid=300
add interface=ether3 bridge=bridge1
```

Specify tagged and untagged ports in the bridge VLAN table:

```
/interface bridge vlan
add bridge=bridge1 tagged=ether3 untagged=ether1 vlan-ids=200
add bridge=bridge1 tagged=ether3 untagged=ether2 vlan-ids=300
```

When bridge VLAN table is configured, you can enable bridge VLAN filtering:

```
/interface bridge set bridge1 vlan-filtering=yes
```

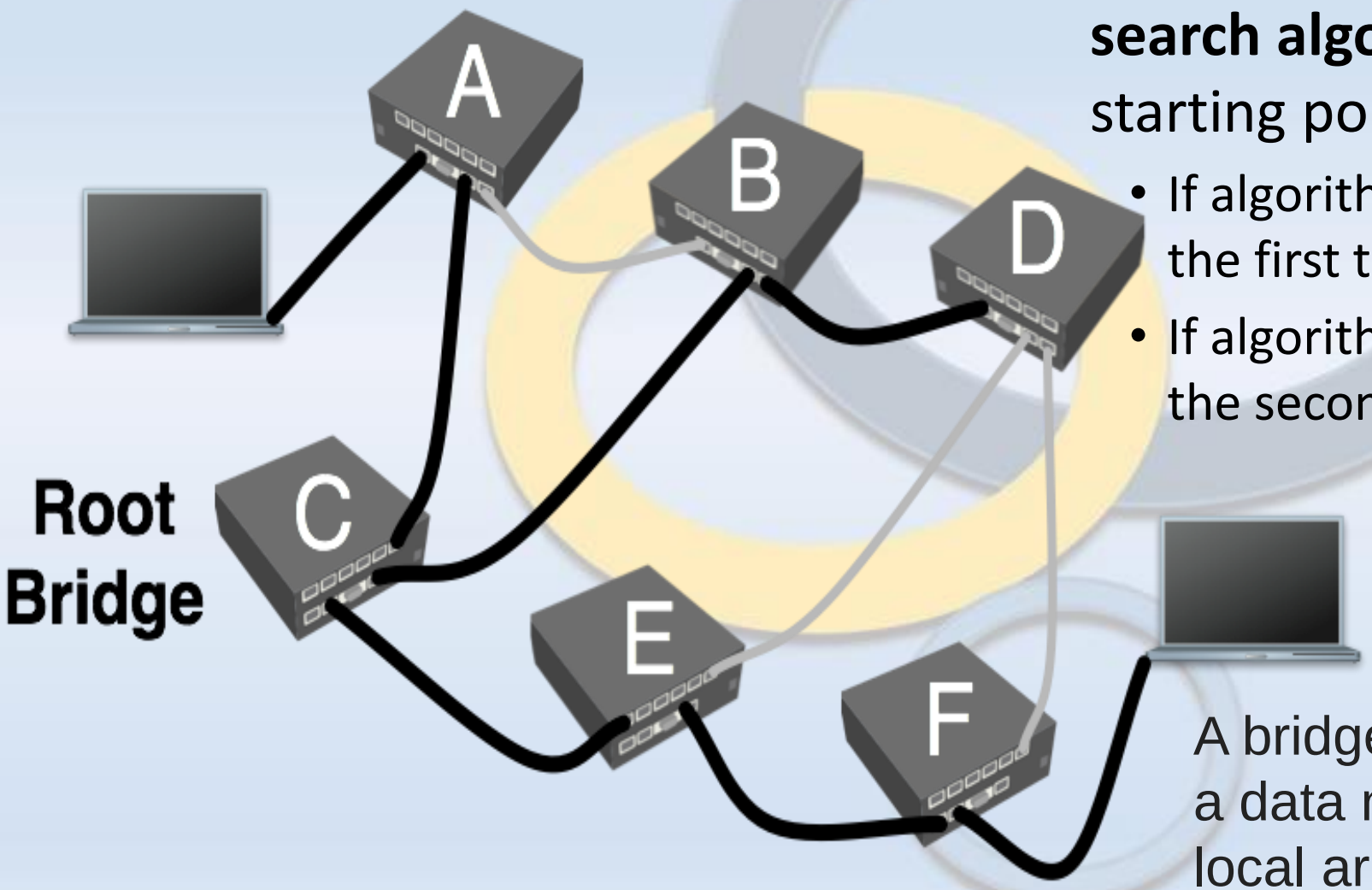
802.11ad – Ether types and Management Port

- Before enabling VLAN filtering you should make sure that you set up a Management port
- The difference between using different EtherTypes is that you must use a Service VLAN interface.
- Service VLAN interfaces can be created as regular VLAN interface, but the use-service-tag parameter toggles if the interface will use Service VLAN tag.
- If the bridge receives a packet with an outer tag that has a different EtherType, it will mark the packet as untagged.

MSTP

- Since RouterOS v6.41 it is possible to enable Multiple Spanning Tree Protocol (MSTP) on a bridge interface.
- Ensure loop-free topology across multiple VLANs.
- MSTP can also provide Layer2 redundancy and can be used as a load balancing technique for VLANs.
- MSTP operates very similarly to (R)STP.

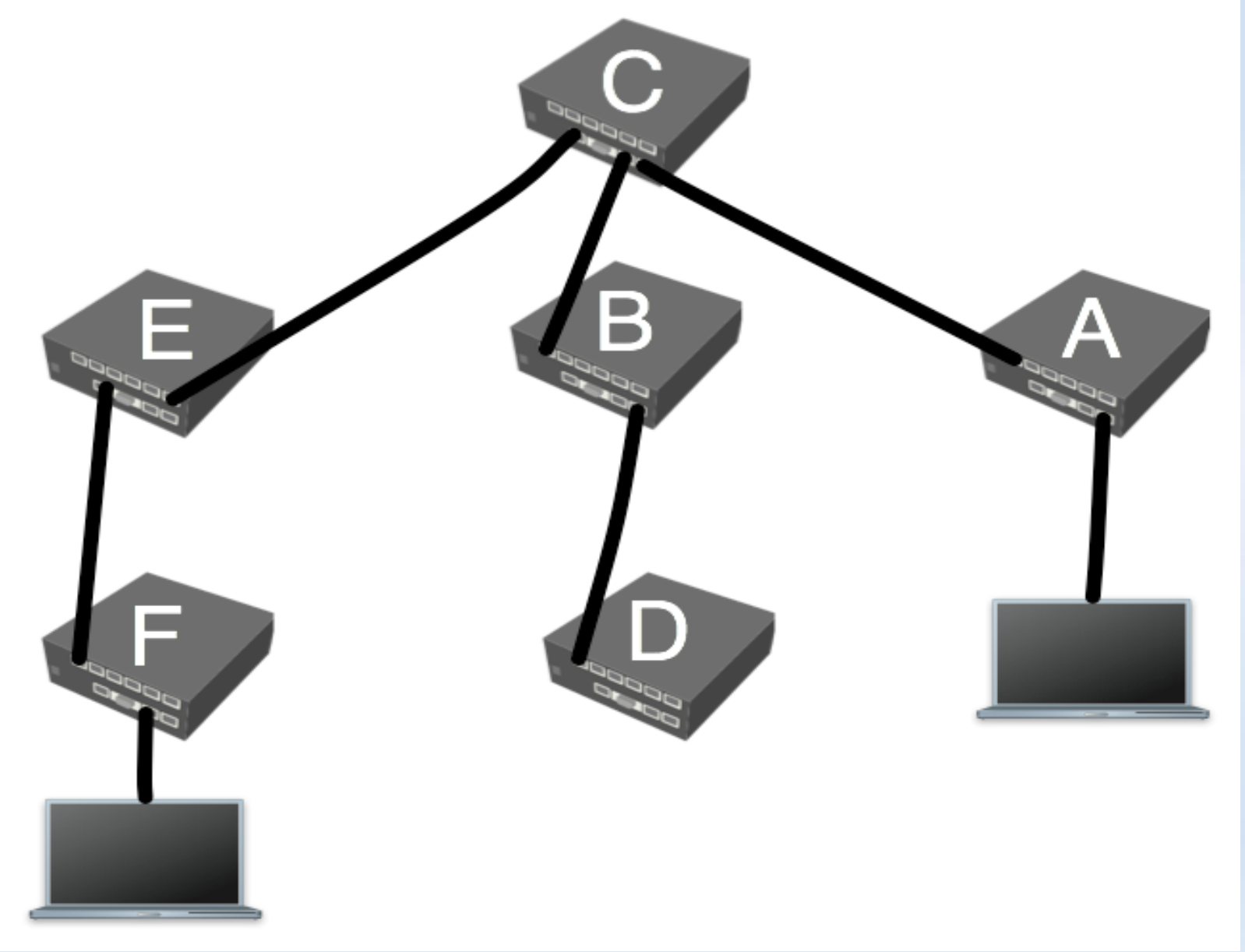
(R)STP Process



- First (R)STP will elect a root bridge based on smallest bridge ID
- Then (R)STP will use **breadth-first search algorithm** taking **root bridge** as starting point
  - If algorithm reaches the MAC address for the first time – it leaves the link active
  - If algorithm reaches the MAC address for the second time – it disables the link

A bridge protocol data unit (**BPDU**) is a data message transmitted across a local area network to detect loops in network topologies.

(R)STP Topology

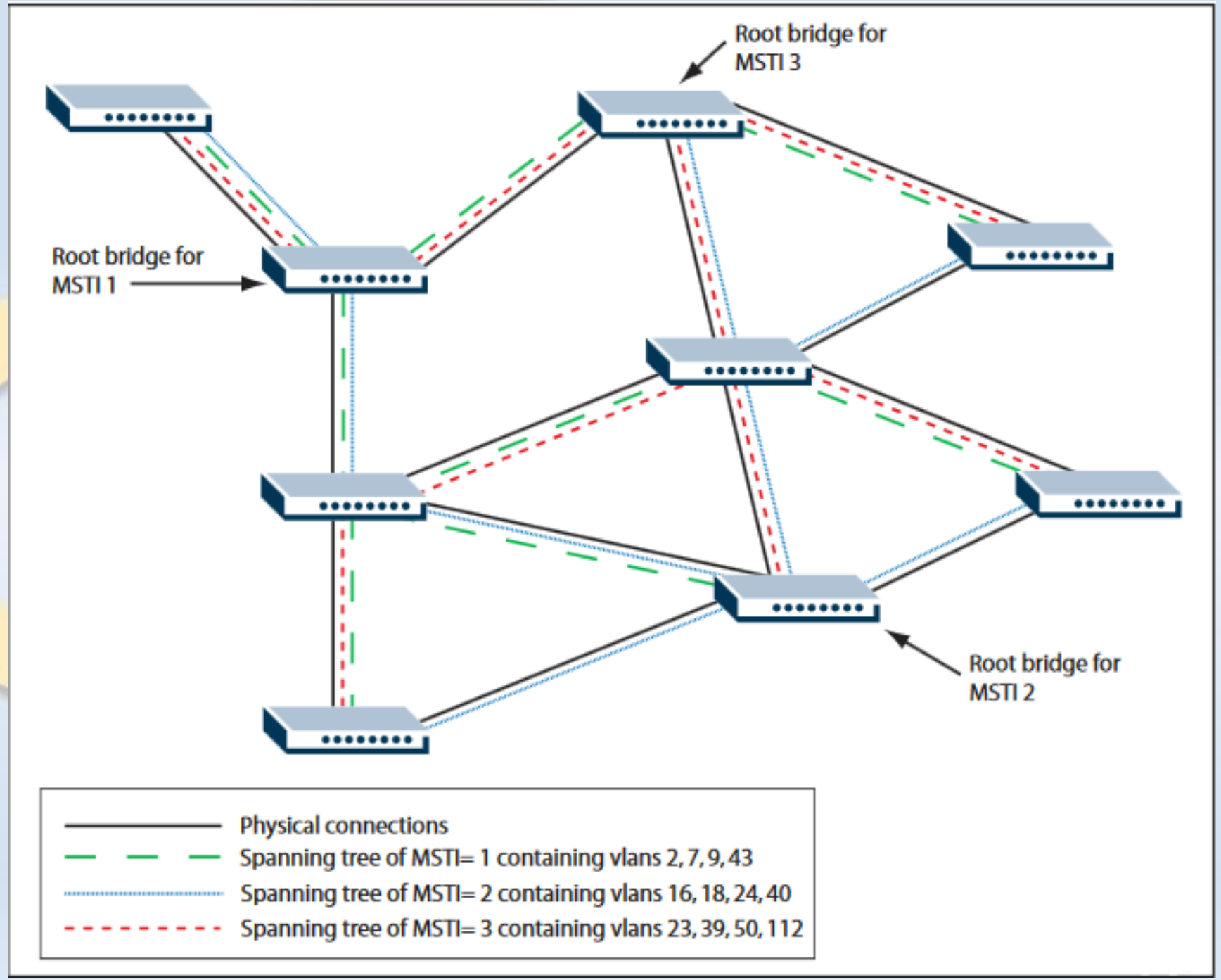


MSTP vs (R)STP

- In case (R)STP is used, the BPDUs are sent across all physical interfaces in a bridge to determine loops.
- In case there is a loop inside a certain VLAN, (R)STP might not be able to detect it.
- MSTP tends to solve both problems by using MST instances that can define a group of VLANs (VLAN mapping) that can be used for load balancing and redundancy.
- Each VLAN group can have a different root bridge and a different path.

MSTP Diagram

In this Example multiple VLANs are there between these switches and MSTP creates loop free environment by creating separate spanning trees.




That's all Folks!

Afterthoughts

- Most material and examples from Mikrotik Wiki. Please check for more details and examples:

https://wiki.mikrotik.com/wiki/Manual:Interface/Bridge#Bridge_VLAN_Filtering

https://wiki.mikrotik.com/wiki/Manual:Switch_Chip_Features

- Those of you who want to do configuration in SwOs can check this excellent presentation out:

MikroTik SwitchOS Basic VLAN Tagging and Trunk by Firdhyan Adhie Lesmana (PowerNet Liberia, Indonesia)

Thank You for Your Attention

Questions???