Manual:CRS Router

From MikroTik Wiki

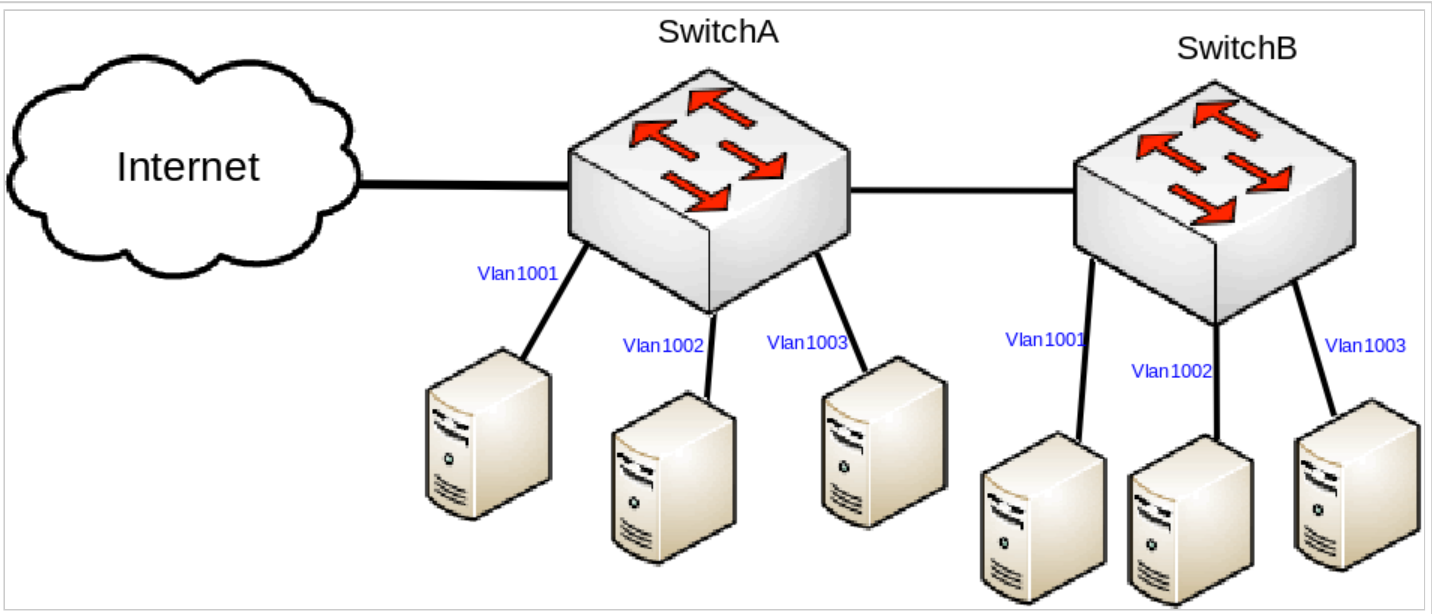
Contents

Applies to RouterOS: v6.41 +

Summary

MikroTik's CRS series devices are powerful switches that also have routing capabilities. In some cases it is sufficient to use the CRS's built-in CPU to handle some functions that are meant to be done by a router, not a switch. The CRS series devices can be used as a router and as a switch at the same time, this is useful for networks that focus on internal network throughput and don't require a large throughput to the Internet.

Warning: CRS series devices are NOT designed to handle large amounts of traffic through the CPU, for this reason be very careful when designing your network since large amounts of traffic that are passing through the CPU will overload it. Functions that depend on the CPU (for example, NAT and DHCP) will not work properly when the CPU is overloaded.



CRS setup where CRS is used as a Router

In this setup SwitchA is going to be our Switch-Router that will use port based VLANs while SwitchB is going to extend the amount of ports. Switch's ports are going to be divided into 3 groups, each group will tag the ingress traffic (access ports) with the appropriate VLAN ID, while the SFP+ port will be used as a trunk port to forward traffic between switches. In this setup a large throughput between each port is expected (except for the WAN port). This guide is written for CRS326-24G-2S+, but it can be used for any other CRS series device that is capable of running RouterOS.

Port switching

All switches in this setup require that ports are added to a bridge. Use these commands on **SwitchA** and **SwitchB**:

```
/interface bridge
add name=bridge vlan-filtering=no
```

- In case you are using a CRS3xx series device:

There will be different ports assigned to each switch to a bridge since one switch will have a WAN port. Use these commands on **SwitchA**:

```
/interface bridge port
add bridge=bridge interface=ether2 pvid=1001
add bridge=bridge interface=ether3 pvid=1001
add bridge=bridge interface=ether4 pvid=1001
add bridge=bridge interface=ether5 pvid=1001
add bridge=bridge interface=ether6 pvid=1001
add bridge=bridge interface=ether7 pvid=1001
add bridge=bridge interface=ether8 pvid=1001
add bridge=bridge interface=ether9 pvid=1002
add bridge=bridge interface=ether10 pvid=1002
add bridge=bridge interface=ether11 pvid=1002
add bridge=bridge interface=ether12 pvid=1002
add bridge=bridge interface=ether13 pvid=1002
add bridge=bridge interface=ether14 pvid=1002
add bridge=bridge interface=ether15 pvid=1002
add bridge=bridge interface=ether16 pvid=1002
add bridge=bridge interface=ether17 pvid=1003
add bridge=bridge interface=ether18 pvid=1003
add bridge=bridge interface=ether19 pvid=1003
add bridge=bridge interface=ether20 pvid=1003
add bridge=bridge interface=ether21 pvid=1003
add bridge=bridge interface=ether22 pvid=1003
add bridge=bridge interface=ether23 pvid=1003
add bridge=bridge interface=ether24 pvid=1003
add bridge=bridge interface=sfp-sfpplus1
```

Since the other switch will not have a WAN port, use these commands on **SwitchB**:

```
/interface bridge port
add bridge=bridge interface=ether1 pvid=1001
add bridge=bridge interface=ether2 pvid=1001
add bridge=bridge interface=ether3 pvid=1001
add bridge=bridge interface=ether4 pvid=1001
add bridge=bridge interface=ether5 pvid=1001
add bridge=bridge interface=ether6 pvid=1001
add bridge=bridge interface=ether7 pvid=1001
add bridge=bridge interface=ether8 pvid=1001
add bridge=bridge interface=ether9 pvid=1002
add bridge=bridge interface=ether10 pvid=1002
add bridge=bridge interface=ether11 pvid=1002
add bridge=bridge interface=ether12 pvid=1002
add bridge=bridge interface=ether13 pvid=1002
add bridge=bridge interface=ether14 pvid=1002
add bridge=bridge interface=ether15 pvid=1002
add bridge=bridge interface=ether16 pvid=1002
add bridge=bridge interface=ether17 pvid=1003
add bridge=bridge interface=ether18 pvid=1003
add bridge=bridge interface=ether19 pvid=1003
add bridge=bridge interface=ether20 pvid=1003
add bridge=bridge interface=ether21 pvid=1003
add bridge=bridge interface=ether22 pvid=1003
add bridge=bridge interface=ether23 pvid=1003
add bridge=bridge interface=ether24 pvid=1003
add bridge=bridge interface=sfp-sfpplus1
```

- In case you are using a CRS1xx/CRS2xx series device:

There will be different ports assigned to each switch to a bridge since one switch will have a WAN port. Use these commands on **SwitchA**:

```
/interface bridge port
```

```
add bridge=bridge interface=ether2
add bridge=bridge interface=ether3
add bridge=bridge interface=ether4
add bridge=bridge interface=ether5
add bridge=bridge interface=ether6
add bridge=bridge interface=ether7
add bridge=bridge interface=ether8
add bridge=bridge interface=ether9
add bridge=bridge interface=ether10
add bridge=bridge interface=ether11
add bridge=bridge interface=ether12
add bridge=bridge interface=ether13
add bridge=bridge interface=ether14
add bridge=bridge interface=ether15
add bridge=bridge interface=ether16
add bridge=bridge interface=ether17
add bridge=bridge interface=ether18
add bridge=bridge interface=ether19
add bridge=bridge interface=ether20
add bridge=bridge interface=ether21
add bridge=bridge interface=ether22
add bridge=bridge interface=ether23
add bridge=bridge interface=ether24
add bridge=bridge interface=sfp-sfpplus1
```

Since the other switch will not have a WAN port, use these commands on **SwitchB**:

```
/interface bridge port
add bridge=bridge interface=ether1
add bridge=bridge interface=ether2
add bridge=bridge interface=ether3
add bridge=bridge interface=ether4
add bridge=bridge interface=ether5
add bridge=bridge interface=ether6
add bridge=bridge interface=ether7
add bridge=bridge interface=ether8
add bridge=bridge interface=ether9
add bridge=bridge interface=ether10
add bridge=bridge interface=ether11
add bridge=bridge interface=ether12
add bridge=bridge interface=ether13
add bridge=bridge interface=ether14
add bridge=bridge interface=ether15
add bridge=bridge interface=ether16
add bridge=bridge interface=ether17
add bridge=bridge interface=ether18
add bridge=bridge interface=ether19
add bridge=bridge interface=ether20
add bridge=bridge interface=ether21
add bridge=bridge interface=ether22
add bridge=bridge interface=ether23
add bridge=bridge interface=ether24
add bridge=bridge interface=sfp-sfpplus1
```

Disable the SFP2+ interface for security reasons (in case it is not being used):

```
/interface ethernet set [find where name="sfp-sfpplus2"] disabled=yes
```

Note: Create a bridge with VLAN filtering disabled at first. If you have misconfigured the VLAN table, you will not be able to access to switch. Enable VLAN filtering only on CRS3xx series devices and only when you have finished configuring VLANs. Currently VLAN filtering on bridge interfaces is not supported on CRS1xx/CRS2xx series devices, you must use the `/interface ethernet switch` section istead, otherwise hardware offloading will not be possible.

Management port

There are multiple ways how to add a management port, in this example we will use a VLAN interface that accepts already tagged traffic with VLAN ID 99. We will allow management traffic only from ether3 and ether4 on **both** switches.

Warning: Since a switch was never designed to be a router, then it will be required to have a firewall that blocks unwanted traffic that is destined to the switch. This must be kept in mind since it will be required to allow special packets such as DHCP to the switch that will have a DHCP Server since these packets will be sent to the CPU and they must not be blocked in the switch chip. If a firewall is not implemented, then management port is unneeded since access to the CPU will be granted either way. You can find an example firewall that will block unwanted traffic to the CPU. Keep in mind that each firewall rule will add extra load to the CPU.

For this guide we are going to use these addresses for each device:

Address	Device
192.168.99.1	SwitchA
192.168.99.2	SwitchB

Use these commands on **SwitchA**:

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

And use these commands on **SwitchB**:

```
/interface vlan
add interface=bridge name=MGMT vlan-id=99
/ip address
add address=192.168.99.2/24 interface=MGMT
/ip route
add gateway=192.168.99.1
```

- In case you are using CRS3xx series device:

Use these commands on **SwitchA** and **SwitchB**:

```
/interface bridge vlan
add bridge=bridge tagged=bridge,ether3,ether4,sfp-sfpplus1 vlan-ids=99
```

- In case you are using CRS1xx/CRS2xx series device:

```
/interface ethernet switch vlan
add ports=switch1-cpu,ether3,ether4,sfp-sfpplus1 vlan-id=99
/interface ethernet switch egress-vlan-tag
add tagged-ports=ether3,ether4,switch1-cpu vlan-id=99
```

Note: SWitchB is a pure switch, it does not require a firewall to block unwanted traffic, this can be done in the switch chip instead and it is the preferred way on a switch.

DHCP Server

To get the DHCP Server working for each VLAN ID, you must first create a VLAN interface that will have access the CPU for each VLAN ID, use these commands on **SwitchA**:

```
/interface vlan
add interface=bridge name=VLAN1001 vlan-id=1001
add interface=bridge name=VLAN1002 vlan-id=1002
add interface=bridge name=VLAN1003 vlan-id=1003
```

Crate a address pool for each VLAN ID, use these commands on **SwitchA**:

```
/ip pool
add name=VLAN1001_pool ranges=192.168.1.100-192.168.1.200
add name=VLAN1002_pool ranges=192.168.2.100-192.168.2.200
add name=VLAN1003_pool ranges=192.168.3.100-192.168.3.200
```

Assign an IP address for each VLAN ID, use these commands on **SwitchA**:

```
/ip address
add address=192.168.1.1/24 interface=VLAN1001
add address=192.168.2.1/24 interface=VLAN1002
add address=192.168.3.1/24 interface=VLAN1003
```

Create a DHCP Server for each VLAN interface, use these commands on **SwitchA**:

```
/ip dhcp-server
add address-pool=VLAN1001_pool disabled=no interface=VLAN1001 name=VLAN1001_DHCP
add address-pool=VLAN1002_pool disabled=no interface=VLAN1002 name=VLAN1002_DHCP
add address-pool=VLAN1003_pool disabled=no interface=VLAN1003 name=VLAN1003_DHCP
/ip dhcp-server network
add address=192.168.1.0/24 dns-server=8.8.8.8 gateway=192.168.1.1
add address=192.168.2.0/24 dns-server=8.8.8.8 gateway=192.168.2.1
add address=192.168.3.0/24 dns-server=8.8.8.8 gateway=192.168.3.1
```

Port based VLANs

- In case you are using a CRS3xx series device:

Ingress traffic is going to be tagged to the VLAN ID specified when each port was added to the bridge. It is required to add each VLAN ID to appropriate ports to the VLAN table that servers as a access list and a

egress VLAN table. Tagged ports are our trunk ports and untagged ports are our access ports.

Note: Since one of our switch is going to be a router that requires access to the CPU from all ports that will want to access the Internet, we must add the bridge port itself as tagged port. This must be done only on the switch that will work as a router, otherwise devices will not be able to receive DHCP leases and access the Internet.

Use these commands on **SwitchA**:

```
/interface bridge vlan
add bridge=bridge tagged=sfp-sfpplus1,bridge untagged="ether2,ether3,ether4,ether5,ether6,ether7,\
ether8" vlan-ids=1001
add bridge=bridge tagged=sfp-sfpplus1,bridge untagged="ether9,ether10,ether11,ether12,ether13,\
ether14,ether15,ether16" vlan-ids=1002
add bridge=bridge tagged=sfp-sfpplus1,bridge untagged="ether17,ether18,ether19,ether20,ether21,\
ether22,ether23,ether24" vlan-ids=1003
```

Similarly add entries to the VLAN table for the other switch, note that bridge port is not listed as tagged port since we don't need anything accessing the CPU to that switch. Use these commands on **SwitchB**:

```
/interface bridge vlan
add bridge=bridge tagged=sfp-sfpplus1 untagged="ether1,ether2,ether3,ether4,ether5,ether6,ether7,\
ether8" vlan-ids=1001
add bridge=bridge tagged=sfp-sfpplus1 untagged="ether9,ether10,ether11,ether12,ether13,ether14,\
ether15,ether16" vlan-ids=1002
add bridge=bridge tagged=sfp-sfpplus1 untagged="ether17,ether18,ether19,ether20,ether21,ether22,\
ether23,ether24" vlan-ids=1003
```

- In case you are using a CRS1xx/CRS2xx series device:

Ingress traffic is going to be tagged using the ingress-vlan-translation section where all untagged traffic is going to be assigned with a specific VLAN ID. Use these commands on **SwitchA** and **SwitchB**:

```
/interface ethernet switch ingress-vlan-translation
add customer-vid=0 new-customer-vid=1001 ports="ether1,ether2,ether3,ether4,ether5,\
ether6,ether7,ether8"
add customer-vid=0 new-customer-vid=1002 ports="ether9,ether10,ether11,ether12,ether13,\
ether14,ether15,ether16"
add customer-vid=0 new-customer-vid=1003 ports="ether17,ether18,ether19,ether20,ether21,\
ether22,ether23,ether24"
```

To specify which ports will be trunk ports, we need to add entries to the egress-vlan-tag section, this section will determine which ports will need to send out only tagged traffic. Use these commands on **SwitchA**:

```
/interface ethernet switch egress-vlan-tag
add tagged-ports=switch1-cpu,sfp-sfpplus1 vlan-id=1001
add tagged-ports=switch1-cpu,sfp-sfpplus1 vlan-id=1002
add tagged-ports=switch1-cpu,sfp-sfpplus1 vlan-id=1003
```

Note: It is required to add `switch1-cpu` as tagged port in order to be able to receive tagged traffic on the VLAN interfaces, otherwise devices from the access ports will not receive a lease from the DHCP Server or route the traffic.

Similarly specify trunk ports for the other switch, use these commands on **SwitchB**:

```
/interface ethernet switch egress-vlan-tag
add tagged-ports=sfp-sfpplus1 vlan-id=1001
add tagged-ports=sfp-sfpplus1 vlan-id=1002
add tagged-ports=sfp-sfpplus1 vlan-id=1003
```

It is required to specify which ports are allowed to forward each VLAN ID, use these commands on **SwitchA**:

```
/interface ethernet switch vlan
add ports="switch1-cpu,ether1,ether2,ether3,ether4,ether5,ether6,ether7,\
ether8,sfp-sfpplus1" vlan-id=1001
add ports="switch1-cpu,ether9,ether10,ether11,ether12,ether13,ether14,\
ether15,ether16,sfp-sfpplus1" vlan-id=1002
add ports="switch1-cpu,ether17,ether18,ether19,ether20,ether21,ether22,\
ether23,ether24,sfp-sfpplus1" vlan-id=1003
```

Similarly specify VLAN membership for the other switch, use these commands on **SwitchB**:

```
/interface ethernet switch vlan
add ports=ether1,ether2,ether3,ether4,ether5,ether6,ether7,ether8,sfp-sfpplus1 vlan-id=1001
add ports=ether9,ether10,ether11,ether12,ether13,ether14,ether15,ether16,sfp-sfpplus1 vlan-id=1002
add ports=ether17,ether18,ether19,ether20,ether21,ether22,ether23,ether24,sfp-sfpplus1 vlan-id=1003
```

Firewall

It is always important to have a proper firewall that secures your device. For this setup it is sufficient to use the default firewall that comes with most RouterBOARDs.

Warning: Each firewall rule is processed by the CPU, be careful when designing your firewall.

Use these commands on **SwitchA**:

```
/interface list
add name=WAN
add name=LAN
/interface list member
add interface=bridge list=LAN
add interface=ether1 list=WAN
/ip firewall filter
add action=accept chain=input connection-state=established,related,untracked
add action=drop chain=input connection-state=invalid
add action=accept chain=input protocol=icmp
add action=drop chain=input in-interface-list=!LAN
add action=accept chain=forward ipsec-policy=in,ipsec
add action=accept chain=forward ipsec-policy=out,ipsec
add action=fasttrack-connection chain=forward connection-state=established,related
add action=accept chain=forward connection-state=established,related,untracked
add action=drop chain=forward connection-state=invalid
add action=drop chain=forward connection-nat-state=!dstnat connection-state=new in-interface-list=W
/ip firewall nat
add action=masquerade chain=srcnat ipsec-policy=out,none out-interface-list=WAN
```

For extra security you can disable discovery protocols on all ports except for the management port, use these commands on **SwitchA** and **SwitchB**:

```
/interface list
add name=neighbors
/interface list member
add interface=MGMT list=neighbors
/ip neighbor discovery-settings
set discover-interface-list=neighbors
```

InterVLAN Routing

Since we created a VLAN interface and added IP address to it, that enabled InterVLAN routing. If your setup requires InterVLAN routing, then you can skip this step. If access from different subnets is not desirable, then it is possible to block communications between different subnets. This can be done using firewall rules, but this will use the CPU's resources. Instead it is recommended to do this using ACL since CRS series devices have highly configurable switch chips. Since InterVLAN routing is enabled only on one switch, then ACL rules are only required for the switch that has the VLAN interfaces.

- In case you are using a CRS3xx series device:

Use these commands on **SwitchA**:

```
/interface ethernet switch rule
add dst-address=192.168.2.0/24 new-dst-ports="" ports="ether2,ether3,ether4,ether5,ether6,ether7\
,ether8,sfp-sfpplus1" src-address=192.168.1.0/24 switch=switch1
add dst-address=192.168.3.0/24 new-dst-ports="" ports="ether2,ether3,ether4,ether5,ether6,ether7,\
ether8,sfp-sfpplus1" src-address=192.168.1.0/24 switch=switch1
add dst-address=192.168.1.0/24 new-dst-ports="" ports="ether9,ether10,ether11,ether12,ether13,\
ether14,ether15,ether16,sfp-sfpplus1" src-address=192.168.2.0/24 switch=switch1
add dst-address=192.168.3.0/24 new-dst-ports="" ports="ether9,ether10,ether11,ether12,ether13,\
ether14,ether15,ether16,sfp-sfpplus1" src-address=192.168.2.0/24 switch=switch1
add dst-address=192.168.1.0/24 new-dst-ports="" ports="ether17,ether18,ether19,ether20,ether21,\
ether22,ether23,ether24,sfp-sfpplus1" src-address=192.168.3.0/24 switch=switch1
add dst-address=192.168.2.0/24 new-dst-ports="" ports="ether17,ether18,ether19,ether20,ether21,\
ether22,ether23,ether24,sfp-sfpplus1" src-address=192.168.3.0/24 switch=switch1
```

Note: There are multiple ways to achieve the same result. For this example we used DST and SRC IP matcher to distinguish packets that are coming from the trunk port (sfp-sfpplus1). Ports must be specified since that is a requirement for the switch chip type that the CRS3xx series devices use.

- In case you are using a CRS2xx series device:

```
/interface ethernet switch acl
add action=drop ip-dst=192.168.2.0/24 ip-src=192.168.1.0/24 mac-protocol=ip
add action=drop ip-dst=192.168.3.0/24 ip-src=192.168.1.0/24 mac-protocol=ip
add action=drop ip-dst=192.168.1.0/24 ip-src=192.168.2.0/24 mac-protocol=ip
add action=drop ip-dst=192.168.3.0/24 ip-src=192.168.2.0/24 mac-protocol=ip
add action=drop ip-dst=192.168.1.0/24 ip-src=192.168.3.0/24 mac-protocol=ip
add action=drop ip-dst=192.168.2.0/24 ip-src=192.168.3.0/24 mac-protocol=ip
```

- In case you are using a CRS1xx series device:

CRS1xx series devices don't have a built-in capability to filter packet at the switch chip's level. For this reason it will not be possible to use the switch chip to offload some packet filter, instead regular firewall filter rules will have to be used. Use these commands on **SwitchA**:

```
add action=drop chain=forward dst-address=192.168.2.0/24 src-address=192.168.1.0/24
add action=drop chain=forward dst-address=192.168.3.0/24 src-address=192.168.1.0/24
add action=drop chain=forward dst-address=192.168.1.0/24 src-address=192.168.2.0/24
add action=drop chain=forward dst-address=192.168.3.0/24 src-address=192.168.2.0/24
add action=drop chain=forward dst-address=192.168.1.0/24 src-address=192.168.3.0/24
add action=drop chain=forward dst-address=192.168.2.0/24 src-address=192.168.3.0/24
```

Note: Make sure you place these firewall filter rules before accepting other packets, in this example you should place these rules before allow traffic that is not DST-NATed.

Invalid/Unknown VLAN filtering

When all VLANs are configured, you should enable VLAN filtering.

- In case you are using a CRS3xx series device:

Use this command on **SwitchA** and **SwitchB**:

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

- In case you are using a CRS1xx/CRS2xx series device:

Use this these commands on **SwitchA**:

```
/interface ethernet switch
set drop-if-invalid-or-src-port-not-member-of-vlan-on-ports="ether2,ether3,ether4,ether5,\
ether6,ether7,ether8,ether9,ether10,ether11,ether12,ether13,ether14,ether15,ether16,ether17,\
ether18,ether19,ether20,ether21,ether22,ether23,ether24,sfp-sfpplus1"
```

Use this these commands on **SwitchB**:

```
/interface ethernet switch
set drop-if-invalid-or-src-port-not-member-of-vlan-on-ports="ether1,ether2,ether3,ether4,ether5,\
ether6,ether7,ether8,ether9,ether10,ether11,ether12,ether13,ether14,ether15,ether16,ether17,\
ether18,ether19,ether20,ether21,ether22,ether23,ether24,sfp-sfpplus1"
```

See also

- CRS examples
- CRS features
- NAT examples
- Firewall filter examples
- VLAN

[Top | Back to Content **]**

- This page was last edited on 23 November 2017, at 15:47.