

# VLAN

Document revision 1.3 (October 11, 2007, 17:38 GMT)

This document applies to V3.0

## Table of Contents

[Table of Contents](#)

[General Information](#)

[Summary](#)

[Specifications](#)

[Description](#)

[Additional Documents](#)

[VLAN Setup](#)

[Property Description](#)

[Notes](#)

[Example](#)

[Application Example](#)

[VLAN example on MikroTik Routers](#)

## General Information

### Summary

VLAN is an implementation of the 802.1Q VLAN protocol for MikroTik RouterOS. It allows you to have multiple Virtual LANs on a single ethernet or wireless interface, giving the ability to segregate LANs efficiently. It supports up to 4095 VLAN interfaces, each with a unique VLAN ID, per ethernet device. VLAN priorities may also be used and manipulated. Many routers, including Cisco and Linux based, and many Layer 2 switches use VLAN to enable multiple independent, isolated networks to exist on the same physical network.

A VLAN is a logical grouping that allows end users to communicate as if they were physically connected to a single isolated LAN, independent of the physical configuration of the network. VLAN support adds a new dimension of security and cost savings permitting the sharing of a physical network infrastructure and interfaces/ports while logically maintaining separation among unrelated users.

### Specifications

Packages required: *system*

License required: *level1 (limited to 1 vlan), level3*

Home menu level: */interface vlan*

Standards and Technologies: [VLAN \(IEEE 802.1Q\)](#)

Hardware usage: *Not significant*

### Description

VLANs are simply a way of grouping a set of switch ports together so that they form a logical network, separate from any other such group. It may also be understood as breaking one physical switch into several

independent parts. Within a single switch this is straightforward local configuration. When the VLAN extends over more than one switch, the inter-switch links have to become trunks, on which packets are tagged to indicate which VLAN they belong to.

You can use MikroTik RouterOS (as well as Cisco IOS, Linux and other router systems) to mark these packets as well as to accept and route marked ones.

As VLAN works on OSI Layer 2, it can be used just as any other network interface without any restrictions. VLAN successfully passes through regular Ethernet bridges.

You can also transport VLANs over wireless links and put multiple VLAN interfaces on a single wireless interface. Note that as VLAN is not a full tunnel protocol (i.e., it does not have additional fields to transport MAC addresses of sender and recipient), the same limitation applies to bridging over VLAN as to bridging plain wireless interfaces. In other words, while wireless clients may participate in VLANs put on wireless interfaces, it is not possible to have VLAN put on a wireless interface in **station** mode bridged with any other interface.

## Currently supported Ethernet interfaces

This is a list of network interfaces on which VLAN was tested and worked. Note that there might be many other interfaces that support VLAN, but they just were not checked. Most modern Ethernet interfaces support VLAN.

- Realtek 8139
- Intel PRO/100
- Intel PRO1000 server adapter
- National Semiconductor DP83816 based cards (RouterBOARD200 onboard Ethernet, RouterBOARD 24 card)
- National Semiconductor DP83815 (Soekris onboard Ethernet)
- VIA VT6105M based cards (RouterBOARD 44 card)
- VIA VT6105
- VIA VT6102 (VIA EPIA onboard Ethernet)

This is a list of network interfaces on which VLAN was tested and worked, but **WITHOUT LARGE PACKET (>1496 bytes) SUPPORT**:

- 3Com 3c59x PCI
- DEC 21140 (tulip)

## Additional Documents

- <http://www.csd.uwo.ca/courses/CS457a/reports/handin/jpbojtos/A2/trunking.htm>
- <http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t3/dtbridge.htm#xtocid114533>
- <http://www.cisco.com/warp/public/473/27.html#tagging>
- <http://www.cisco.com/warp/public/538/7.html>

- <http://www.nwfusion.com/news/tech/2001/0305tech.html>
- [http://www.intel.com/network/connectivity/resources/doc\\_library/tech\\_brief/virtual\\_lans.htm](http://www.intel.com/network/connectivity/resources/doc_library/tech_brief/virtual_lans.htm)

## VLAN Setup

Home menu level: */interface vlan*

### Property Description

**arp** (*disabled* | *enabled* | *proxy-arp* | *reply-only*; default: **enabled**) - Address Resolution Protocol mode

- **disabled** - the interface will not use ARP protocol
- **enabled** - the interface will fully use ARP protocol
- **proxy-arp** - the interface will be an ARP proxy
- **reply-only** - the interface will only reply to the requests for to its own IP addresses, but neighbor MAC addresses will be gathered from /ip arp statically set table only

**interface** (*name*) - physical interface to the network where the VLAN is put

**mtu** (*integer*; default: **1500**) - Maximum Transmission Unit

**name** (*name*) - interface name for reference

**vlan-id** (*integer*; default: **1**) - Virtual LAN identifier or tag that is used to distinguish VLANs. Must be equal for all computers that belong to the same VLAN.

### Notes

MTU should be set to 1500 bytes as on Ethernet interfaces. But this may not work with some Ethernet cards that do not support receiving/transmitting of full size Ethernet packets with VLAN header added (1500 bytes data + 4 bytes VLAN header + 14 bytes Ethernet header). In this situation MTU 1496 can be used, but note that this will cause packet fragmentation if larger packets have to be sent over interface. At the same time remember that MTU 1496 may cause problems if path MTU discovery is not working properly between source and destination.

### Example

To add and enable a VLAN interface named **test** with **vlan-id=1** on interface **ether1**:

```
[admin@MikroTik] interface vlan> add name=test vlan-id=1 interface=ether1
[admin@MikroTik] interface vlan> print
Flags: X - disabled, R - running
#   NAME           MTU  ARP      VLAN-ID  INTERFACE
0 X test           1500 enabled  1        ether1
[admin@MikroTik] interface vlan> enable 0
[admin@MikroTik] interface vlan> print
Flags: X - disabled, R - running
#   NAME           MTU  ARP      VLAN-ID  INTERFACE
0 R test           1500 enabled  1        ether1
[admin@MikroTik] interface vlan>
```

### Application Example

## VLAN example on MikroTik Routers

Let us assume that we have two or more MikroTik RouterOS routers connected with a hub. Interfaces to the physical network, where the VLAN is to be created is **ether1** for all of them (it is needed only for example simplification, it is NOT a must).

To connect computers through VLAN they must be connected physically and unique IP addresses should be assigned them so that they could ping each other. Then on each of them the VLAN interface should be created:

```
[admin@MikroTik] interface vlan> add name=test vlan-id=32 interface=ether1
[admin@MikroTik] interface vlan> print
Flags: X - disabled, R - running
#   NAME      MTU  ARP      VLAN-ID  INTERFACE
0  R test      1500 enabled  32       ether1
[admin@MikroTik] interface vlan>
```

If the interfaces were successfully created, both of them will be **running**. If computers are connected incorrectly (through network device that does not retransmit or forward VLAN packets), either both or one of the interfaces will not be **running**.

When the interface is running, IP addresses can be assigned to the VLAN interfaces.

On the Router 1:

```
[admin@MikroTik] ip address> add address=10.10.10.1/24 interface=test
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
#   ADDRESS      NETWORK      BROADCAST      INTERFACE
0   10.0.0.204/24  10.0.0.0     10.0.0.255     ether1
1   10.20.0.1/24   10.20.0.0    10.20.0.255    pcl
2   10.10.10.1/24  10.10.10.0   10.10.10.255   test
[admin@MikroTik] ip address>
```

On the Router 2:

```
[admin@MikroTik] ip address> add address=10.10.10.2/24 interface=test
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
#   ADDRESS      NETWORK      BROADCAST      INTERFACE
0   10.0.0.201/24  10.0.0.0     10.0.0.255     ether1
1   10.10.10.2/24  10.10.10.0   10.10.10.255   test
[admin@MikroTik] ip address>
```

If it set up correctly, then it is possible to ping Router 2 from Router 1 and vice versa:

```
[admin@MikroTik] ip address> /ping 10.10.10.1
10.10.10.1 64 byte pong: ttl=255 time=3 ms
10.10.10.1 64 byte pong: ttl=255 time=4 ms
10.10.10.1 64 byte pong: ttl=255 time=10 ms
10.10.10.1 64 byte pong: ttl=255 time=5 ms
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 3/10.5/10 ms
[admin@MikroTik] ip address> /ping 10.10.10.2
10.10.10.2 64 byte pong: ttl=255 time=10 ms
10.10.10.2 64 byte pong: ttl=255 time=11 ms
10.10.10.2 64 byte pong: ttl=255 time=10 ms
10.10.10.2 64 byte pong: ttl=255 time=13 ms
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 10/11/13 ms
[admin@MikroTik] ip address>
```