

# Manual:Interface/Bridge

From MikroTik Wiki  
< Manual:Interface

## Summary

**Sub-menu:** /interface bridge

**Standards:** IEEE802.1D (<http://standards.ieee.org/getieee802/download/802.1D-2004.pdf>)

Applies  
to



RouterOS: v3, v4+

Ethernet-like networks (Ethernet, Ethernet over IP, IEEE802.11 in ap-bridge or bridge mode, WDS, VLAN) can be connected together using MAC bridges. The bridge feature allows the interconnection of hosts connected to separate LANs (using EoIP, geographically distributed networks can be bridged as well if any kind of IP network interconnection exists between them) as if they were attached to a single LAN. As bridges are transparent, they do not appear in traceroute list, and no utility can make a distinction between a host working in one LAN and a host working in another LAN if these LANs are bridged (depending on the way the LANs are interconnected, latency and data rate between hosts may vary).

Network loops may emerge (intentionally or not) in complex topologies. Without any special treatment, loops would prevent network from functioning normally, as they would lead to avalanche-like packet multiplication. Each bridge runs an algorithm which calculates how the loop can be prevented. STP and RSTP allows bridges to communicate with each other, so they can negotiate a loop free topology. All other alternative connections that would otherwise form loops, are put to standby, so that should the main connection fail, another connection could take its place. This algorithm exchanges configuration messages (BPDU - Bridge Protocol Data Unit) periodically, so that all bridges are updated with the newest information about changes in network topology. (R)STP selects a root bridge which is responsible for network reconfiguration, such as blocking and opening ports on other bridges. The root bridge is the bridge with the lowest bridge ID.

## Bridge Interface Setup

**Sub-menu:** /interface bridge

To combine a number of networks into one bridge, a bridge interface should be created (later, all the desired interfaces should be set up as its ports). One MAC address will be assigned to all the bridged interfaces (the smallest MAC address will be chosen automatically).

## Properties

Property	Description
----------	-------------

<b>admin-mac</b> ( <i>MAC address</i> ; Default: )	Static MAC address of the bridge (takes effect if <code>auto-mac=no</code> )
<b>ageing-time</b> ( <i>time</i> ; Default: <b>00:05:00</b> )	How long a host's information will be kept in the bridge database
<b>arp</b> ( <i>disabled   enabled   proxy-arp   reply-only</i> ; Default: <b>enabled</b> )	Address Resolution Protocol setting <ul style="list-style-type: none"> <li>■ <b>disabled</b> - the interface will not use ARP</li> <li>■ <b>enabled</b> - the interface will use ARP</li> <li>■ <b>proxy-arp</b> - the interface will use the ARP proxy feature</li> <li>■ <b>reply-only</b> - the interface will only reply to requests originated from matching IP address/MAC address combinations which are entered as static entries in the "/ip arp" table. No dynamic entries will be automatically stored in the "/ip arp" table. Therefore for communications to be successful, a valid static entry must already exist.</li> </ul>
<b>auto-mac</b> ( <i>yes   no</i> ; Default: <b>yes</b> )	Automatically select the smallest MAC address of bridge ports as a bridge MAC address
<b>forward-delay</b> ( <i>time</i> ; Default: <b>00:00:15</b> )	Time which is spent during the initialization phase of the bridge interface (i.e., after router startup or enabling the interface) in listening/learning state before the bridge will start functioning normally
<b>l2mtu</b> ( <i>integer</i> ; read-only)	Layer2 Maximum transmission unit. read more»
<b>max-message-age</b> ( <i>time</i> ; Default: <b>00:00:20</b> )	How long to remember Hello messages received from other bridges
<b>mtu</b> ( <i>integer</i> ; Default: <b>1500</b> )	Maximum Transmission Unit
<b>name</b> ( <i>text</i> ; Default: <b>bridgeN</b> )	Name of the bridge interface
<b>priority</b> ( <i>integer: 0..65535 decimal format or 0x0000-0xffff hex format</i> ; Default: <b>32768 / 0x8000</b> )	Spanning tree protocol priority for bridge interface. Bridge with the smallest (lowest) bridge ID becomes a Root-Bridge. Bridge ID consists of two numbers - priority and MAC address of the bridge. To compare two bridge IDs, the priority is compared first. If two bridges have equal priority, then the MAC addresses are compared.
<b>protocol-mode</b> ( <i>none   rstp   stp</i> ; Default: <b>rstp</b> )	Select Spanning tree protocol (STP) or Rapid spanning tree protocol (RSTP) to ensure a loop-free topology for any bridged LAN. RSTP provides for faster spanning tree convergence after a topology change.
<b>transmit-hold-count</b> ( <i>integer: 1..10</i> ; Default: <b>6</b> )	The Transmit Hold Count used by the Port Transmit state machine to limit transmission rate

## (Rapid) Spanning Tree Protocol

[http://en.wikipedia.org/wiki/Spanning\\_Tree\\_Protocol](http://en.wikipedia.org/wiki/Spanning_Tree_Protocol)

### Example

To add and enable a bridge interface that will forward all the protocols:

```
[admin@MikroTik] /interface bridge> add
[admin@MikroTik] /interface bridge> print
Flags: X - disabled, R - running
0 R name="bridge1" mtu=1500 l2mtu=65535 arp=enabled
   mac-address=00:00:00:00:00:00 protocol-mode=none priority=0x8000
   auto-mac=yes admin-mac=00:00:00:00:00:00 max-message-age=20s
   forward-delay=15s transmit-hold-count=6 ageing-time=5m
[admin@MikroTik] /interface bridge>
```

## Bridge Settings

**Sub-menu:** /interface bridge settings

Property	Description
<b>allow-fast-path</b> ( <i>yes</i>   <i>no</i> ; Default: <b>yes</b> )	Allows fast path
<b>use-ip-firewall</b> ( <i>yes</i>   <i>no</i> ; Default: <b>no</b> )	Send bridged traffic to also be processed by 'IP firewall'. This does not apply to routed traffic.
<b>use-ip-firewall-for-pppoe</b> ( <i>yes</i>   <i>no</i> ; Default: <b>no</b> )	Send bridged un-encrypted PPPoE traffic to also be processed by 'IP firewall' (requires <b>use-ip-firewall=yes</b> to work)
<b>use-ip-firewall-for-vlan</b> ( <i>yes</i>   <i>no</i> ; Default: <b>no</b> )	Send bridged VLAN traffic to also be processed by 'IP firewall' (requires <b>use-ip-firewall=yes</b> to work)

## Port Settings

**Sub-menu:** /interface bridge port

Port submenu is used to enslave interfaces in a particular bridge interface.

Property	Description
<b>auto-isolate</b> ( <i>yes</i>   <i>no</i> ; Default: <b>no</b> )	Prevents STP blocking port from erroneously moving into a forwarding state if no BPDU's are received on the bridge.
<b>bridge</b> ( <i>name</i> ; Default: <b>none</b> )	The bridge interface the respective interface is grouped in

**edge** (*auto* | *no* | *no-discover* | *yes* | *yes-discover*; Default: **auto**)

Set port as edge port or non-edge port, or enable automatic detection. Edge ports are connected to a LAN that has no other bridges attached. If the port is configured to discover edge port then as soon as the bridge detects a BPDU coming to an edge port, the port becomes a non-edge port.

**external-fdb** (*auto* | *no* | *yes*; Default: **auto**)

Whether to use wireless registration table to speed up bridge host learning

**horizon** (*none* | *integer 0..429496729*; Default: **none**)

Use split horizon bridging to prevent bridging loops. [read more»](#)

**interface** (*name*; Default: **none**)

Name of the interface

**path-cost** (*integer: 0..65535*; Default: **10**)

Path cost to the interface, used by STP to determine the "best" path

**point-to-point** (*auto* | *yes* | *no*; Default: **auto**)

**priority** (*integer: 0..255*; Default: **128**)

The priority of the interface in comparison with other going to the same subnet

## Example

To group **ether1** and **ether2** in the already created **bridge1** bridge

```
[admin@MikroTik] /interface bridge port> add bridge=bridge1 interface=ether1
[admin@MikroTik] /interface bridge port> add bridge=bridge1 interface=ether2
[admin@MikroTik] /interface bridge port> print
Flags: X - disabled, I - inactive, D - dynamic
#   INTERFACE      BRIDGE      PRIORITY  PATH-COST  HORIZON
0   ether1          bridge1      0x80      10         none
1   ether2          bridge1      0x80      10         none
[admin@MikroTik] /interface bridge port>
```

## Bridge Monitoring

**Sub-menu:** /interface bridge monitor

Used to monitor the current status of a bridge.

Property	Description
<b>current-mac-address</b> ( <i>MAC address</i> )	Current MAC address of the bridge
<b>designated-port-count</b> ( <i>integer</i> )	Number of designated bridge ports
<b>port-count</b> ( <i>integer</i> )	Number of the bridge ports
<b>root-bridge</b> ( <i>yes</i>   <i>no</i> )	Shows whether bridge is the root bridge of the spanning tree

<b>root-bridge-id</b> ( <i>text</i> )	The root bridge ID, which is in form of bridge-priority.bridge-MAC-address
<b>root-path-cost</b> ( <i>integer</i> )	The total cost of the path to the root-bridge
<b>root-port</b> ( <i>name</i> )	Port to which the root bridge is connected to
<b>state</b> ( <i>enabled</i>   <i>disabled</i> )	State of the bridge

## Example

To monitor a bridge:

```
[admin@MikroTik] /interface bridge> monitor bridge1
      state: enabled
current-mac-address: 00:0C:42:52:2E:CE
      root-bridge: yes
root-bridge-id: 0x8000.00:00:00:00:00:00
root-path-cost: 0
      root-port: none
      port-count: 2
designated-port-count: 0

[admin@MikroTik] /interface bridge>
```

## Bridge Port Monitoring

**Sub-menu:** /interface bridge port monitor

Statistics of an interface that belongs to a bridge.

Property	Description
<b>edge-port</b> ( <i>yes</i>   <i>no</i> )	Whether port is an edge port or not
<b>edge-port-discovery</b> ( <i>yes</i>   <i>no</i> )	Whether port is set to automatically detect edge ports
<b>external-fdb</b> ( <i>yes</i>   <i>no</i> )	Shows whether registration table is used instead of forwarding data base
<b>forwarding</b> ( <i>yes</i>   <i>no</i> )	Port state
<b>learning</b> ( <i>yes</i>   <i>no</i> )	Port state
<b>port-number</b> ( <i>integer 1..4095</i> )	Port identifier
<b>point-to-point-port</b> ( <i>yes</i>   <i>no</i> )	
<b>role</b> ( <i>designated</i>   <i>root port</i>   <i>alternate</i>   <i>backup</i>   <i>disabled</i> )	(R)STP algorithm assigned role of the port: <ul style="list-style-type: none"><li>▪ <b>Disabled port</b> - not strictly part of STP, a network administrator can manually disable a port</li><li>▪ <b>Root port</b> – a forwarding port that is the best port from</li></ul>



Nonroot-bridge to Rootbridge

- **Alternative port** – an alternate path to the root bridge. This path is different than using the root port
- **Designated port** – a forwarding port for every LAN segment
- **Backup port** – a backup/redundant path to a segment where another bridge port already connects.

**sending-rstp** (*yes* | *no*)

Whether the port is sending BPDU messages

**status** (*in-bridge* | *inactive*)

Port status

## Example

To monitor a bridge port:

```
[admin@MikroTik] /interface bridge port> monitor 0
      status: in-bridge
      port-number: 1
        role: designated-port
      edge-port: no
edge-port-discovery: yes
point-to-point-port: no
  external-fdb: no
    sending-rstp: no
      learning: yes
    forwarding: yes

[admin@MikroTik] /interface bridge port>
```

## Bridge Host Monitoring

**Sub-menu:** /interface bridge host

Property	Description
<b>age</b> ( <i>read-only: time</i> )	The time since the last packet was received from the host
<b>bridge</b> ( <i>read-only: name</i> )	The bridge the entry belongs to
<b>external-fdb</b> ( <i>read-only: flag</i> )	Whether the host was learned using wireless registration table
<b>local</b> ( <i>read-only: flag</i> )	Whether the host entry is of the bridge itself (that way all local interfaces are shown)
<b>mac-address</b> ( <i>read-only: MAC address</i> )	Host's MAC address
<b>on-interface</b> ( <i>read-only: name</i> )	Which of the bridged interfaces the host is connected to

## Example

To get the active host table:

```
[admin@MikroTik] /interface bridge host> print
Flags: L - local, E - external-fdb
BRIDGE      MAC-ADDRESS      ON-INTERFACE      AGE
bridge1     00:00:00:00:00:01 ether2      3s
bridge1     00:01:29:FF:1D:CC ether2      0s
L bridge1   00:0C:42:52:2E:CF ether2      0s
bridge1     00:0C:42:52:2E:D0 ether2      3s
bridge1     00:0C:42:5C:A5:AE ether2      0s
[admin@MikroTik] /interface bridge host>
```

## Bridge Firewall

**Sub-menu:** /interface bridge filter, /interface bridge nat

The bridge firewall implements packet filtering and thereby provides security functions that are used to manage data flow to, from and through bridge.

Packet flow diagram shows how packets are processed through router. It is possible to force bridge traffic to go through `/ip firewall filter` rules (see: Bridge Settings)

There are two bridge firewall tables:

- **filter** - bridge firewall with three predefined chains:
  - **input** - filters packets, where the destination is the bridge (including those packets that will be routed, as they are destined to the bridge MAC address anyway)
  - **output** - filters packets, which come from the bridge (including those packets that has been routed normally)
  - **forward** - filters packets, which are to be bridged (note: this chain is not applied to the packets that should be routed through the router, just to those that are traversing between the ports of the same bridge)
- **nat** - bridge network address translation provides ways for changing source/destination MAC addresses of the packets traversing a bridge. Has two built-in chains:
  - **srcnat** - used for "hiding" a host or a network behind a different MAC address. This chain is applied to the packets leaving the router through a bridged interface
  - **dstnat** - used for redirecting some packets to other destinations

You can put packet marks in bridge firewall (filter and NAT), which are the same as the packet marks in IP firewall put by `'/ip firewall mangle'`. In this way, packet marks put by bridge firewall can be used in 'IP firewall', and vice versa.

General bridge firewall properties are described in this section. Some parameters that differ between nat and filter rules are described in further sections.

### Properties

Property	Description
<b>802.3-sap</b> ( <i>integer</i> )	DSAP (Destination Service Access Point) and SSAP (Source Service Access Point) are 2 one byte fields, which identify the network protocol entities which use the link layer service. These

bytes are always equal. Two hexadecimal digits may be specified here to match a SAP byte

**802.3-type** (*integer*)

Ethernet protocol type, placed after the IEEE 802.2 frame header. Works only if 802.3-sap is 0xAA (SNAP - Sub-Network Attachment Point header). For example, AppleTalk can be indicated by SAP code of 0xAA followed by a SNAP type code of 0x809B

**arp-dst-address** (*IP address; default:* )

ARP destination address

**arp-dst-mac-address** (*MAC address; default:* )

ARP destination MAC address

**arp-gratuitous** (*yes | no; default:* )

Matches ARP gratuitous packets

**arp-hardware-type** (*integer; default: 1*)

ARP hardware type. This is normally Ethernet (Type 1)

**arp-opcode** (*arp-nak | drarp-error | drarp-reply | drarp-request | inarp-reply | inarp-request | reply | reply-reverse | request | request-reverse*)

ARP opcode (packet type)

- **arp-nak** - negative ARP reply (rarely used, mostly in ATM networks)
- **drarp-error** - Dynamic RARP error code, saying that an IP address for the given MAC address can not be allocated
- **drarp-reply** - Dynamic RARP reply, with a temporary IP address assignment for a host
- **drarp-request** - Dynamic RARP request to assign a temporary IP address for the given MAC address
- **inarp-reply** - InverseARP Reply
- **inarp-request** - InverseARP Request
- **reply** - standard ARP reply with a MAC address
- **reply-reverse** - reverse ARP (RARP) reply with an IP address assigned
- **request** - standard ARP request to a known IP address to find out unknown MAC address
- **request-reverse** - reverse ARP (RARP) request to a known MAC address to find out unknown IP address (intended to be used by hosts to find out their own IP address, similarly to DHCP service)

**arp-packet-type** (*integer: 0..65535 decimal format or 0x0000-0xffff hex format*)

ARP Packet Type

**arp-src-address** (*IP address; default:* )

ARP source address

**arp-src-mac-address** (*MAC address; default:* )

ARP source MAC address

**chain** (*text*)

Bridge firewall chain, which the filter is functioning in (either a built-in one, or a user defined)

**dst-address** (*IP address; default:* )

Destination IP address (only if MAC protocol is set to IPv4)



**dst-mac-address** (*MAC address; default:* )

Destination MAC address

**dst-port** (*integer 0..65535*)

Destination port number or range (only for TCP or UDP protocols)

**in-bridge** (*name*)

Bridge interface through which the packet is coming in

**in-interface** (*name*)

Physical interface (i.e., bridge port) through which the packet is coming in

**ingress-priority** (*integer 0..63*)

Matches ingress priority of the packet. Priority may be derived from VLAN, WMM or MPLS EXP bit. [read more»](#)

**ip-protocol** (*ddp | egp | encap | etherip | ggp | gre | hmp | icmp | icmpv6 | idpr-cmtip | igmp | ipencap | ipip | ipsec-ah | ipsec-esp | ipv6 | ipv6-frag | ipv6-nonxt | ipv6-opts | ipv6-route | iso-tp4 | l2tp | ospf | pim | pup | rdp | rspf | rsvp | st | tcp | udp | vmtp | vrrp | xns-idp | xtp*)

IP protocol (only if MAC protocol is set to IPv4)

- **ddp** - datagram delivery protocol
- **egp** - exterior gateway protocol
- **encap** - ip encapsulation
- **etherip** -
- **ggp** - gateway-gateway protocol
- **gre** - general routing encapsulation
- **hmp** - host monitoring protocol
- **icmp** - IPv4 internet control message protocol
- **icmpv6** - IPv6 internet control message protocol
- **idpr-cmtip** - idpr control message transport
- **igmp** - internet group management protocol
- **ipencap** - ip encapsulated in ip
- **ipip** - ip encapsulation
- **ipsec-ah** - IPsec AH protocol
- **ipsec-esp** - IPsec ESP protocol
- **ipv6** -
- **ipv6-frag** -
- **ipv6-nonxt** -
- **ipv6-opts** -
- **ipv6-route** -
- **iso-tp4** - iso transport protocol class 4
- **l2tp** -
- **ospf** - open shortest path first
- **pim** - protocol independent multicast
- **pup** - parc universal packet protocol
- **rspf** - radio shortest path first
- **rsvp** -
- **rdp** - reliable datagram protocol
- **st** - st datagram mode
- **tcp** - transmission control protocol
- **udp** - user datagram protocol
- **vmtp** - versatile message transport
- **vrrp** - Virtual Router Redundancy Protocol
- **xns-idp** - xerox ns idp
- **xtp** - xpress transfer protocol

**jump-target** (*name*)

If `action=jump` specified, then specifies the user-defined firewall chain to process the packet

**limit** (*integer/time, integer*)

Restricts packet match rate to a given limit.

- **count** - maximum average packet rate, measured in packets per second (pps), unless followed by Time option
- **time** - specifies the time interval over which the packet rate is measured
- **burst** - number of packets to match in a burst

**log-prefix** (*text*)

Defines the prefix to be printed before the logging information

**mac-protocol** (*802.2 | arp | ip | ipv6 | ipx | length | mpls-multicast | mpls-unicast | pppoe | pppoe-discovery | rarp | vlan or integer: 0..65535 decimal format or 0x0000-0xffff hex format*)

Ethernet payload type (MAC-level protocol)

- **802.2**
- **arp** - Type 0x0806 - ARP
- **ip** - Type 0x0800 - IPv4
- **ipv6** - Type 0x86dd - IPv6
- **ipx** - Type 0x8137 - "Internetwork Packet Exchange"
- **length**
- **mpls-multicast** - Type 0x8848 - MPLS Multicast
- **mpls-unicast** - Type 0x8847 - MPLS Unicast
- **ppoe** - Type 0x8864 - PPPoE Session
- **ppoe-discovery** - Type 0x8863 - PPPoE Discovery
- **rarp** - Type 0x8035 - Reverse ARP
- **vlan** - Type 0x8100 - 802.1Q tagged VLAN

**out-bridge** (*name*)

Outgoing bridge interface

**out-interface** (*name*)

Interface that the packet is leaving the bridge through

**packet-mark** (*name*)

Match packets with certain packet mark

**packet-type** (*broadcast | host | multicast | other-host*)

MAC frame type:

- **broadcast** - broadcast MAC packet
- **host** - packet is destined to the bridge itself
- **multicast** - multicast MAC packet
- **other-host** - packet is destined to some other unicast address, not to the bridge itself

**src-address** (*IP address; default:* )

Source IP address (only if MAC protocol is set to IPv4)

**src-mac-address** (*MAC address; default:* )

Source MAC address

**src-port** (*integer 0..65535*)

Source port number or range (only for TCP or UDP protocols)

**stp-flags** (*topology-change | topology-change-ack*)

The BPDU (Bridge Protocol Data Unit) flags. Bridge exchange configuration messages named BPDU periodically for preventing loops

- **topology-change** - topology change flag is set when a bridge detects port state change, to force all other bridges to drop their host tables and recalculate network topology
- **topology-change-ack** - topology change acknowledgement flag is sen in replies to the notification packets

**stp-forward-delay** (*time 0..65535*)

Forward delay timer

**stp-hello-time** (*time 0..65535*)

STP hello packets time

**stp-max-age** (*time 0..65535*)

Maximal STP message age

**stp-msg-age** (*time 0..65535*)

STP message age

**stp-port** (*integer 0..65535*)

STP port identifier

**stp-root-address** (*MAC address*)

Root bridge MAC address

**stp-root-cost** (*integer 0..65535*)

Root bridge cost

**stp-root-priority** (*integer 0..65535*)

Root bridge priority

**stp-sender-address** (*MAC address*)

STP message sender MAC address

**stp-sender-priority** (*integer 0..65535*)

STP sender priority

**stp-type** (*config | tcn*)

The BPDU type:

- **config** - configuration BPDU
- **tcn** - topology change notification

**vlan-encap** (*802.2 | arp | ip | ipv6 | ipx | length | mpls-multicast | mpls-unicast | pppoe | pppoe-discovery | rarp | vlan or integer: 0..65535 decimal format or 0x0000-0xffff hex format*)

the MAC protocol type encapsulated in the VLAN frame

**vlan-id** (*integer 0..4095*)

VLAN identifier field

**vlan-priority** (*integer 0..7*)

The user priority field

## Notes

- STP matchers are only valid if destination MAC address is 01:80:C2:00:00:00/FF:FF:FF:FF:FF:FF (Bridge Group address), also `stp` should be enabled.
- ARP matchers are only valid if `mac-protocol` is `arp` or `rarp`

- VLAN matchers are only valid for `vlan` ethernet protocol
- IP-related matchers are only valid if `mac-protocol` is set as `ipv4`
- 802.3 matchers are only consulted if the actual frame is compliant with IEEE 802.2 and IEEE 802.3 standards (**note**: it is not the industry-standard Ethernet frame format used in most networks worldwide!). These matchers are ignored for other packets.

## Bridge Packet Filter

**Sub-menu:** `/interface bridge filter`

This section describes bridge packet filter specific filtering options, that are specific to `'/interface bridge filter'`.

### Properties

Property	Description
<code>action</code> ( <i><code>accept</code>   <code>drop</code>   <code>jump</code>   <code>log</code>   <code>mark-packet</code>   <code>passthrough</code>   <code>return</code>   <code>set-priority</code></i> )	<ul style="list-style-type: none"><li>▪ <b>accept</b> - accept the packet. No action, i.e., the packet is passed through without undertaking any action, and no more rules are processed in the relevant list/chain</li><li>▪ <b>drop</b> - silently drop the packet (without sending the ICMP reject message)</li><li>▪ <b>jump</b> - jump to the chain specified by the value of the <code>jump-target</code> argument</li><li>▪ <b>log</b> - log the packet</li><li>▪ <b>mark</b> - mark the packet to use the mark later</li><li>▪ <b>passthrough</b> - ignore this rule and go on to the next one. Acts the same way as a disabled rule, except for ability to count packets</li><li>▪ <b>return</b> - return to the previous chain, from where the jump took place</li><li>▪ <b>set-priority</b> - set priority specified by the <code>new-priority</code> parameter on the packets sent out through a link that is capable of transporting priority (VLAN or WMM-enabled wireless interface). <a href="#">Read more</a>&gt;</li></ul>

## Bridge NAT

**Sub-menu:** `/interface bridge nat`

This section describes bridge NAT options, that are specific to `'/interface bridge nat'`.



## Properties

Property	Description
<b>action</b> ( <i>accept</i>   <i>drop</i>   <i>jump</i>   <i>mark-packet</i>   <i>redirect</i>   <i>set-priority</i>   <i>arp-reply</i>   <i>dst-nat</i>   <i>log</i>   <i>passthrough</i>   <i>return</i>   <i>src-nat</i> )	<ul style="list-style-type: none"> <li>▪ <b>accept</b> - accept the packet. No action, i.e., the packet is passed through without undertaking any action, and no more rules are processed in the relevant list/chain</li> <li>▪ <b>arp-reply</b> - send a reply to an ARP request (any other packets will be ignored by this rule) with the specified MAC address (only valid in dstnat chain)</li> <li>▪ <b>drop</b> - silently drop the packet (without sending the ICMP reject message)</li> <li>▪ <b>dst-nat</b> - change destination MAC address of a packet (only valid in dstnat chain)</li> <li>▪ <b>jump</b> - jump to the chain specified by the value of the jump-target argument</li> <li>▪ <b>log</b> - log the packet</li> <li>▪ <b>mark</b> - mark the packet to use the mark later</li> <li>▪ <b>passthrough</b> - ignore this rule and go on to the next one. Acts the same way as a disabled rule, except for ability to count packets</li> <li>▪ <b>redirect</b> - redirect the packet to the bridge itself (only valid in dstnat chain)</li> <li>▪ <b>return</b> - return to the previous chain, from where the jump took place</li> <li>▪ <b>set-priority</b> - set priority specified by the new-priority parameter on the packets sent out through a link that is capable of transporting priority (VLAN or WMM-enabled wireless interface). <a href="#">Read more</a>&gt;</li> <li>▪ <b>src-nat</b> - change source MAC address of a packet (only valid in srcnat chain)</li> </ul>
<b>to-arp-reply-mac-address</b> ( <i>MAC address</i> )	Source MAC address to put in Ethernet frame and ARP payload, when <code>action=arp-reply</code> is selected
<b>to-dst-mac-address</b> ( <i>MAC address</i> )	Destination MAC address to put in Ethernet frames, when <code>action=dst-nat</code> is selected
<b>to-src-mac-address</b> ( <i>MAC address</i> )	Source MAC address to put in Ethernet frames, when <code>action=src-nat</code> is selected

[ Top | Back to Content ]

Retrieved from "http://wiki.mikrotik.com/index.php?title=Manual:Interface/Bridge&oldid=26902"

Categories: Manual | Interface

- This page was last modified on 29 December 2014, at 10:27.
- This page has been accessed 332,135 times.



