

Interface Bonding

Document revision 1.1 (oct-26-2004)

This document applies to V2.9

Table of Contents

[Table of Contents](#)

[Summary](#)

[Quick Setup Guide](#)

[Specifications](#)

[Related Documents](#)

[Description](#)

[Property Description](#)

[Notes](#)

[Bonding two Eoip tunnels](#)

General Information

Summary

Bonding is a technology that allows to aggregate multiple ethernet-like interfaces into a single virtual link, thus getting higher data rates and providing failover.

Quick Setup Guide

Let us assume that we have 2 NICs in each router (**Router1** and **Router2**) and want to get maximum data rate between 2 routers. To make this possible, follow these steps:

1. Make sure that you do not have IP addresses on interfaces which will be enslaved for bonding interface!
2. Add **bonding** interface on **Router1**:

```
[admin@Router1] interface bonding> add slaves=ether1,ether2
```

And on **Router2**:

```
[admin@Router2] interface bonding> add slaves=ether1,ether2
```

3. Add addresses to bonding interfaces:

```
[admin@Router1] ip address> add address=172.16.0.1/24 interface=bonding1
```

```
[admin@Router2] ip address> add address=172.16.0.2/24 interface=bonding1
```

4. Test the link from **Router1**:

```
[admin@Router1] interface bonding> /pi 172.16.0.2
172.16.0.2 ping timeout
172.16.0.2 ping timeout
172.16.0.2 ping timeout
172.16.0.2 64 byte ping: ttl=64 time=2 ms
172.16.0.2 64 byte ping: ttl=64 time=2 ms
```

Note that bonding interface needs a couple of seconds to get connectivity with its peer.

Specifications

Packages required: *system*

License required: *level1*

Home menu level: */interface bonding*

Standards and Technologies: *None*

Hardware usage: *Not significant*

Related Documents

- [Linux Ethernet Bonding Driver mini-howto](#)

Description

To provide a proper failover, you should specify **link-monitoring** parameter. It can be:

- MII (Media Independent Interface) type1 or type2 - Media Independent Interface is an abstract layer between the operating system and the NIC which detects whether the link is running (it performs also other functions, but in our case this is the most important).
- ARP - Address Resolution Protocol periodically (for **arp-interval** time) checks the link status.

link-monitoring is used to check whether the link is up or not.

Property Description

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

- **disabled** - the interface will not use ARP
- **enabled** - the interface will use ARP
- **proxy-arp** - the interface will use the ARP proxy feature
- **reply-only** - the interface will only reply to the requests originated to its own IP addresses. Neighbour MAC addresses will be resolved using /ip arp statically set table only

arp-interval (*time*; default: **00:00:00.100**) - time in milliseconds which defines how often to monitor ARP requests

arp-ip-targets (*IP address*; default: **""**) - IP target address which will be monitored if link-monitoring is set to arp. You can specify multiple IP addresses, separated by comma

down-delay (*time*; default: **00:00:00**) - if a link failure has been detected, bonding interface is disabled for down-delay time. Value should be a multiple of mii-interval

lACP-rate (*Isec | 30secs*; default: **30secs**) - Link Aggregation Control Protocol rate specifies how often to exchange with LACPDU's between bonding peer. Used to determine whether link is up or other changes have occurred in the network. LACP tries to adapt to these changes providing failover.

link-monitoring (*arp | mii-type1 | mii-type2 | none*; default: **none**) - method to use for monitoring the link (whether it is up or down)

- **arp** - uses Address Resolution Protocol to determine whether the remote interface is reachable

- **mii-type1** - uses Media Independent Interface type1 to determine link status. Link status determination relies on the device driver. If bonding shows that the link status is up, when it should not be, then it means that this card don't support this possibility.
- **mii-type2** - uses MII type2 to determine link status (used if mii-type1 is not supported by the NIC)
- **none** - no method for link monitoring is used. If a link fails, it is not considered as down (but no traffic passes through it, thus).

mac-address (*read-only: MAC address*) - MAC address of the bonding interface

mii-interval (*time*; default: **00:00:00.100**) - how often to monitor the link for failures (parameter used only if link-monitoring is mii-type1 or mii-type2)

mtu (*integer*: 68..1500; default: **1500**) - Maximum Transmit Unit in bytes

mode (*802.3ad | active-backup | balance-alb | balance-rr | balance-tlb | balance-xor | broadcast*; default: **balance-rr**) - interface bonding mode. Can be one of:

- **802.3ad** - IEEE 802.3ad dynamic link aggregation. In this mode, the interfaces are aggregated in a group where each slave shares the same speed. If you use a switch between 2 bonding routers, be sure that this switch supports IEEE 802.3ad standard. Provides fault tolerance and load balancing.
- **active-backup** - provides link backup. Only one slave can be active at a time. Another slave becomes active only, if first one fails.
- **balance-alb** - adaptive load balancing. It includes balance-tlb and received traffic is also balanced. Device driver should support for setting the mac address, then it is active. Otherwise balance-alb doesn't work. No special switch is required.
- **balance-rr** - round-robin load balancing. Slaves in bonding interface will transmit and receive data in sequential order. Provides load balancing and fault tolerance.
- **balance-tlb** - Outgoing traffic is distributed according to the current load on each slave. Incoming traffic is received by the current slave. If receiving slave fails, then another slave takes the MAC address of the failed slave. Doesn't require any special switch support.
- **balance-xor** - Use XOR policy for transmit. Provides only failover (in very good quality), but not load balancing, yet.
- **broadcast** - Broadcasts the same data on all interfaces at once. This provides fault tolerance but slows down traffic throughput on some slow machines.

name (*name*) - descriptive name of bonding interface

primary (*name*; default: **none**) - Interface is used as primary output media. If primary interface fails, only then others slaves will be used. This value works only with mode=active-backup

slaves (*name*) - at least two ethernet-like interfaces separated by a comma, which will be used for bonding

up-delay (*time*; default: **00:00:00**) - if a link has been brought up, bonding interface is disabled for up-delay time and after this time it is enabled. Value should be a multiple of mii-interval

Notes

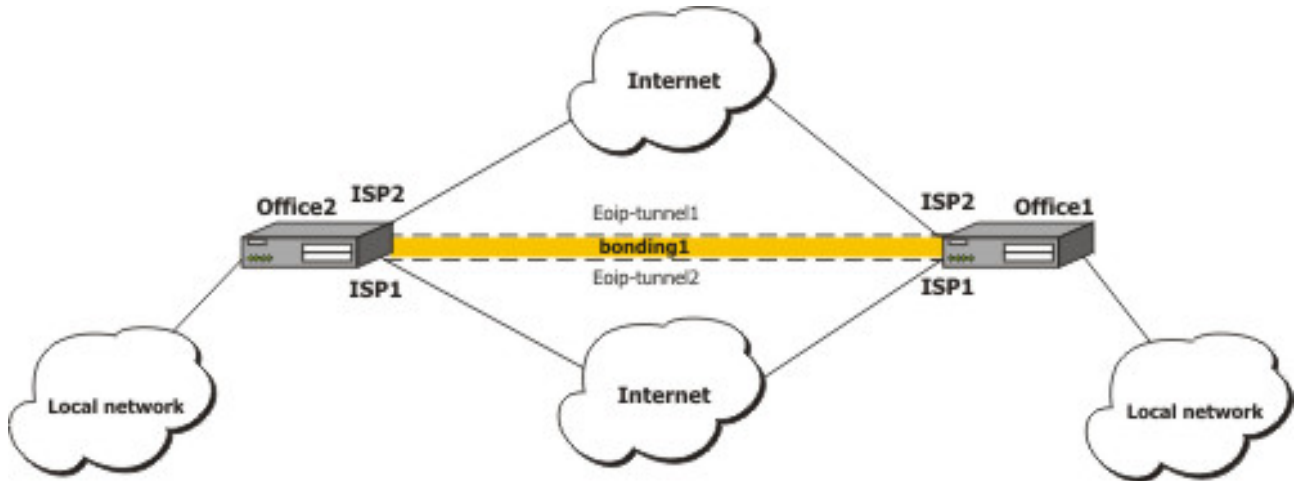
Link failure detection and failover is working significantly better with expensive network cards, for example, made by Intel, then with more cheap ones. For example, on Intel cards failover is taking place in less than a second after link loss, while on some other cards, it may require up to 20 seconds. Also, the Active load

balancing (mode=**balance-alb**) does not work on some cheap cards.

Application Examples

Bonding two Eoip tunnels

Assume you need to configure the MikroTik router for the following network setup, where you have two offices with 2 ISP for each. You want combine links for getting double speed and provide failover:



We are assuming that connections to Internet through two ISP are configured for both routers.

- Configuration on routers
 - on **Office1**

```
[admin@office1] > /interface print
Flags: X - disabled, D - dynamic, R - running
#  NAME          TYPE          RX-RATE  TX-RATE  MTU
0  R isp1         ether         0         0 1500
1  R isp2         ether         0         0 1500

[admin@office1] > /ip address print
Flags: X - disabled, I - invalid, D - dynamic
#  ADDRESS        NETWORK        BROADCAST  INTERFACE
0  1.1.1.1/24     1.1.1.0        1.1.1.255  isp2
1  10.1.0.111/24 10.1.0.0       10.1.0.255 isp1
```

- on **Office2**

```
[admin@office2] interface> print
Flags: X - disabled, D - dynamic, R - running
#  NAME          TYPE          RX-RATE  TX-RATE  MTU
0  R isp2         ether         0         0 1500
1  R isp1         ether         0         0 1500
[admin@office2] interface> /ip add print
Flags: X - disabled, I - invalid, D - dynamic
#  ADDRESS        NETWORK        BROADCAST  INTERFACE
0  2.2.2.1/24     2.2.2.0        2.2.2.255  isp2
1  10.1.0.112/24 10.1.0.0       10.1.0.255 isp1
```

- Eoip tunnel configuration

- for **Office1** through ISP1

```
[admin@office1] > interface eoip add remote-address=10.1.0.112 tunnel-id=2
\... mac-address=FE:FD:00:00:00:04
[admin@office1] > interface eoip print
Flags: X - disabled, R - running
0 R name="eoip-tunnel2" mtu=1500 mac-address=FE:FD:00:00:00:04 arp=enabled
\... remote-address=10.1.0.112 tunnel-id=2
```

- for **Office2** through ISP1

```
[admin@office2] > interface eoip add remote-address=10.1.0.111 tunnel-id=2
\... mac-address=FE:FD:00:00:00:02
[admin@office2] > interface eoip print
Flags: X - disabled, R - running
0 R name="eoip-tunnel2" mtu=1500 mac-address=FE:FD:00:00:00:02 arp=enabled
\... remote-address=10.1.0.111 tunnel-id=2
```

- for **Office1** through ISP2

```
[admin@office1] > interface eoip add remote-address=2.2.2.1 tunnel-id=1
\... mac-address=FE:FD:00:00:00:03
[admin@office1] interface eoip> print
Flags: X - disabled, R - running
0 R name="eoip-tunnel1" mtu=1500 mac-address=FE:FD:00:00:00:03 arp=enabled
  remote-address=2.2.2.1 tunnel-id=1

1 R name="eoip-tunnel2" mtu=1500 mac-address=FE:FD:00:00:00:04 arp=enabled
  remote-address=10.1.0.112 tunnel-id=2
```

- for **Office2** through ISP2

```
[admin@office2] > interface eoip add remote-address=1.1.1.1 tunnel-id=1
\... mac-address=FE:FD:00:00:00:01
[admin@office2] interface eoip> print
Flags: X - disabled, R - running
0 R name="eoip-tunnel1" mtu=1500 mac-address=FE:FD:00:00:00:01 arp=enabled
  remote-address=1.1.1.1 tunnel-id=1

1 R name="eoip-tunnel2" mtu=1500 mac-address=FE:FD:00:00:00:02 arp=enabled
  remote-address=10.1.0.111 tunnel-id=2
```

- Bonding configuration

- for **Office1**

```
[admin@office1] interface bonding> add slaves=eoip-tunnel1,eoip-tunnel2
[admin@office1] interface bonding> print
Flags: X - disabled, R - running
0 R name="bonding1" mtu=1500 mac-address=00:0C:42:03:20:E7 arp=enabled
  slaves=eoip-tunnel1,eoip-tunnel2 mode=balance-rr primary=none
  link-monitoring=none arp-interval=00:00:00.100 arp-ip-targets=""
  mii-interval=00:00:00.100 down-delay=00:00:00 up-delay=00:00:00
  lacp-rate=30secs
[admin@office1] ip address> add address=3.3.3.1/24 interface=bonding1
[admin@office1] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
# ADDRESS NETWORK BROADCAST INTERFACE
0 1.1.1.1/24 1.1.1.0 1.1.1.255 isp2
1 10.1.0.111/24 10.1.0.0 10.1.0.255 isp1
2 3.3.3.1/24 3.3.3.0 3.3.3.255 bonding1
```

- for **Office2**

```
[admin@office2] interface bonding> add slaves=eoip-tunnell1,eoip-tunnel2
[admin@office2] interface bonding> print
Flags: X - disabled, R - running
 0 R name="bonding1" mtu=1500 mac-address=00:0C:42:03:20:E7 arp=enabled
    slaves=eoip-tunnell1,eoip-tunnel2 mode=balance-rr primary=none
    link-monitoring=none arp-interval=00:00:00.100 arp-ip-targets=""
    mii-interval=00:00:00.100 down-delay=00:00:00 up-delay=00:00:00
    lacp-rate=30secs
[admin@office2] ip address> add address=3.3.3.2/24 interface=bonding1
[admin@office2] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
#  ADDRESS          NETWORK          BROADCAST        INTERFACE
0  2.2.2.1/24        2.2.2.0          2.2.2.255        isp2
1  10.1.0.112/24     10.1.0.0         10.1.0.255       isp1
2  3.3.3.2/24        3.3.3.0          3.3.3.255        bonding1
[admin@office2] ip address> /ping 3.3.3.1
3.3.3.1 64 byte ping: ttl=64 time=2 ms
3.3.3.1 64 byte ping: ttl=64 time=2 ms
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 2/2.0/2 ms
```