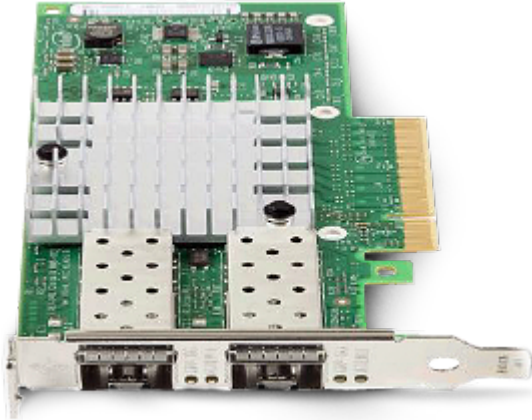


[coderazzi.net](http://coderazzi.net)

[http://coderazzi.net/linux/dual\\_nic\\_linux.htm](http://coderazzi.net/linux/dual_nic_linux.htm)

## Using multiple NICs on Linux



A Linux machine -or virtual machine-, can require having multiple ethernet cards, each on a separate LAN. One reason for this can be to separate network traffic; for example, all management traffic can be routed to one NIC, while the another NIC handles all other traffic.

With one single NIC, the default Linux installation will automatically set it up as a static or dynamic IP address, and a default gateway. When there are multiple NICs, gateway routing becomes an issue. Basically, there is initially only one default gateway, associated to one of the NICs, and all outgoing traffic will be sent on that default gateway / NIC. If there is incoming traffic on the other NIC, the answer is not sent back via the same route, being then effectively lost.

This [page](#) explains correctly the problem happening when using multiple NICs.

The solution is to create routing tables on each interface; to enable multiple routing tables, it is needed to edit the file

`/etc/iproute2/rt_tables`

Add to the end of this file one entry for each NIC. For this example, let's assume just two NICs, `eth0` and `eth1`. We add two lines to the

`/etc/iproute2/rt_tables`

file

```
1      eth0
2      eth1
```

The names on the second column is the names given to the tables, and do not need to match the ethernet interface. It could be, instead:

```
1      eth0table
2      eth1table
```

Now, it is needed to setup the routes. This differ in

**debian**

and

**centos**

## Debian : setting up multiple routes

In Debian, all information is setup in the file

`/etc/network/interfaces`

For example, assuming static IP addresses, this file could look like:

`source /etc/network/interfaces.d/*`

`auto lo`

`iface lo inet loopback`

`allow-hotplug eth0`

`iface eth0 inet static`

`hwaddress ether 94:57:a5:5e:5a:e4`

`address 172.30.250.231`

`netmask 255.255.255.0`

`network 172.30.250.0`

`broadcast 172.30.250.255`

`up ip route add 172.30.250.0/24 dev eth0 table eth0`

`up ip route add default via 172.30.250.254 dev eth0 table eth0`

`up ip rule add from 172.30.250.231/32 table eth0`

`up ip rule add to 172.30.250.231/32 table eth0`

`allow-hotplug eth1`

`iface eth1 inet static`

`hwaddress ether 94:57:a5:5e:5a:e5`

`address 172.16.251.211`

`netmask 255.255.255.0`

`network 172.16.251.0`

`broadcast 172.16.251.255`

`gateway 172.16.251.254`

`dns-nameservers 8.8.8.8`

`up ip route add 172.16.251.0/24 dev eth1 table eth1`

`up ip route add default via 172.16.251.254 dev eth1 table eth1`

`up ip rule add from 172.16.251.211/32 table eth1`

`up ip rule add to 172.16.251.211/32 table eth1`

Note that, for each interface, we are adding two lines to setup the routes, and two lines to setup the rules. Also, note that the table names provided in the file `/etc/iproute2/route_tables` are here fully required. For the interface **eth0**, we have:

1. **up ip route add 172.30.250.0/24 dev eth0 table eth0** That is, when the interface gets up, add the route to the subnet of this interface (172.30.250.0/24) via this device (eth0). Furthermore, this route is define in the table eth0
2. **up ip route add default via 172.30.250.254 dev eth0 table eth0** Here, the default gateway (172.30.250.254) is specified.
3. **up ip rule add from 172.30.250.231/32 table eth0** Basic rule. Traffic from this interface's address (172.30.250.231) is handled by the table route eth0
4. **up ip rule add to 172.30.250.231/32 table eth0** Basic rule. Traffic to this interface's address (172.30.250.231) is handled by the table route eth0

Finally, it is important to note that in this file, only eth1 defines a **gateway**. eth0 does not have one defined.

## CentOS : setting up multiple routes

The information is now split into multiple files, all located in `/etc/sysconfig/network-scripts/` (plus the file `/etc/iproute2/route_tables` already described)

First, let's assume that NetworkManager is not used. If used, the first step is to disable it:

```
service NetworkManager stop
chkconfig NetworkManager off
service network start
chkconfig network on
```

Each NIC has its own files in `/etc/sysconfig/network-scripts/`. For eth0, the files of interest are:

- ifcfg-eth0
- route-eth0
- rule-eth0

Assuming now that this interface is setup dynamically using DHCP, the file **ifcfg-eth0** could look like:

```
NM_CONTROLLED="no"
BOOTPROTO=dhcp
DEFROUTE="yes"
DEVICE="eth0"
HWADDR=00:16:3e:91:30:8e
NAME="eth0"
ONBOOT=yes
PREFIX="24"
TYPE="Ethernet"
UUID=0e2935a0-f57d-4f44-96f3-3bffa8f8606ea
```

That is: not controlled by network manager, using DHCP, and defining a default route.

The file **route-eth0**, using the ip syntax, would look like:

```
192.168.201.0 dev eth0 src 192.168.201.132 table eth0
default via 192.168.201.1 dev eth0 table eth0
```

That is, we define the route to subnet 192.168.201.0 using this interface, and we provide a default gateway (192.168.201.1) for this traffic

The file **rule-eth0** just defines the conditions to use this route table:

```
from 192.168.201.132 table eth0
to 192.168.201.132 table eth0
```

Very much like in the Debian case. However, this is the dynamic IP address, makes no sense dumping here the received IP, which can eventually change. There is no need in this case to have the files **route-eth0** or **rule-eth0**. But the files for eth1 are definitely required. These files can look very similar. In the case of **ifcfg-eth1** two details are important:

```
DEFROUTE=no
#GATEWAY=not defined
```

Copyright (c) Luis M. Pena - [lu@coderazzi.net](mailto:lu@coderazzi.net)