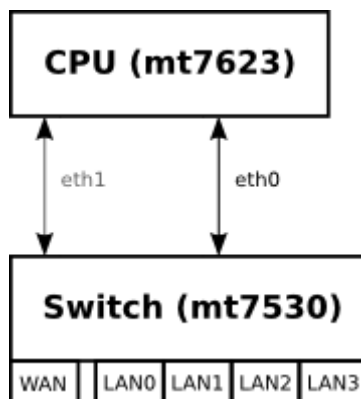


Network-Configuration

ip-command needs package iproute2

Configuration on this page is based on debian stretch, should work the same way with Ubuntu needs Kernel 4.14 or above (DSA-driver for Port-separation)

in Kernel 4.14 eth0 is the connection between CPU and the Switch-Circuit (mt7530), on which the Ports wan and lan0-4 are connected. this connection have to be set to “up” first.



bringing up then cpu-port(s)

```
ip link set eth0 up
ip link set eth1 up
```

or via /etc/network/interfaces

```
auto eth0
iface eth0 inet manual
pre-up ip link set $IFACE up
post-down ip link set $IFACE down

auto eth1
iface eth1 inet manual
pre-up ip link set $IFACE up
post-down ip link set $IFACE down
```

the mapping of ports to gmac is defined in dts-file and can be shown with “ip a”

With 4.14 >.52 on my repo gmac #2 (eth1) is added and wan is connected to this.

by default each lan-port is separated and needs an own ip-configuration in different subnets

most users like to use all lan-ports in 1 network-segment, so these can be [bridged together](#) to make only 1 ip-configuration for “LAN”

MAC-Address

The MAC-address can only be set for the GMAC (connection between Switch and CPU). In Kernel 4.14 only 1 GMAC is detected (eth0). There are 2 GMACs in Hardware.

UDEV

[here](#)

```
$ cat /etc/udev/rules.d/00-static-mac-address.rules
ACTION=="add", SUBSYSTEM=="net", KERNELS=="1b100000.ethernet",
RUN+="/sbin/ip link set dev %k address ae:fc:de:ad:be:ef"
```

interfaces-file

/etc/network/interfaces

```
iface lan0 inet static
    address 192.168.0.10
    netmask 255.255.255.0
    gateway 192.168.0.5
# pre-up ip link set $IFACE up
pre-up ip link set $IFACE address 02:01:02:03:04:08 up
```

using systemd

/etc/systemd/network/10-wan.link

```
[Match]
OriginalName=wan

[Link]
MACAddress=XX:XX:XX:XX:XX:XX
```

<http://forum.banana-pi.org/t/set-mac-address-on-boot/7224/7>

device-tree

```
local-mac-address = [00 0a 35 00 00 01];
mac-address = [00 0a 35 00 00 01];
```

<http://forum.banana-pi.org/t/set-mac-address-on-boot/7224/4>

this can also be used in devicetree-overlays

set via uboot

if devicetree (with mac-address property) is loaded separately (fdt), an alias for ethernet-node is defined and ethaddr-variable is set in uboot this is used in linux

<http://forum.banana-pi.org/t/set-mac-address-on-boot/7224/6>

interface-name

Ubuntu 18.4+ (and debiaan 10+) using new interface names. Wifi devices are no more named like wlanX more like wlpXsY

to avoid this, "net.ifnames=0" can be added to Kernel-Cmdline (uEnv.txt for uboot) and/or rename via udev

/etc/udev/rules.d/70-persistent-net.rules

```
SUBSYSTEM=="net", ACTION=="add", ATTR{address}=="f8:62:aa:50:15:c8",  
NAME="wlan1"
```

find attributes like this

```
udevadm info --attribute-walk /sys/class/net/<interface-name>
```

to apply the rule(s) (but does not rename back):

```
udevadm control --reload-rules && udevadm trigger
```

after reboot all works

if driver is compiled as module it can be reloaded (after activating the udev-rules)

```
modprobe -r mt76x2e  
modprobe mt76x2e
```

IP

permanent

/etc/network/interfaces:

```
#first set the upstream-Port (NIC between CPU and MT7530-Switch) up  
auto eth0  
iface eth0 inet manual  
pre-up ip link set $IFACE up  
post-down ip link set $IFACE down
```

```
auto eth1
iface eth1 inet manual
    pre-up ip link set $IFACE up
    post-down ip link set $IFACE down

#then configure the lan-ports
auto lan0
iface lan0 inet static
    hwaddress ether 08:00:00:00:00:00 # if you want to set MAC manually
    address 192.168.0.10
    netmask 255.255.255.0
    gateway 192.168.0.5
    pre-up ip link set $IFACE up
    post-down ip link set $IFACE down
```

temporary

```
ifconfig lan0 192.168.0.10 netmask 255.255.255.0 broadcast 192.168.0.255
```

```
ip addr add 192.168.0.10/24 broadcast 192.168.0.255 dev lan0
```

make sure only 1 port is in the specific subnet.

```
ip a
#or
ip addr show lan0
```

DHCP

Client

/etc/network/interfaces:

```
auto lan3
allow-hotplug lan3
iface lan3 inet dhcp
```

Renew ip via

```
sudo dhclient -v -r lan3
```

Server

/etc/dnsmasq.conf (activate line by removing # on begin of line)

```
conf-dir=/etc/dnsmasq.d
```

```
/etc/dnsmasq.d/interfaces.conf
```

```
interface=wlan1
interface=ap0

# DHCP-Server not active for Interface
no-dhcp-interface=eth0
no-dhcp-interface=eth1

#dhcp-authoritative (interface+range+leasetime, default-gateway-ip as option
3)
dhcp-range=ap0,192.168.10.100,192.168.10.150,255.255.255.0,48h
dhcp-option=ap0,3,192.168.10.1
dhcp-range=wlan1,192.168.11.100,192.168.11.150,255.255.255.0,48h
dhcp-option=wlan1,3,192.168.11.1
```

```
/etc/dnsmasq.d/interfaces.conf
```

```
service dnsmasq start
```

more info here: [dnsmasq](#)

IPv6

disabling

https://www.thomas-krenn.com/de/wiki/IPv6_deaktivieren

temporary

```
echo 1 > /proc/sys/net/ipv6/conf/all/disable_ipv6
```

to make it permanent create file /etc/sysctl.d/01-disable-ipv6.conf with content:

```
net.ipv6.conf.all.disable_ipv6 = 1
```

testing:

```
ip addr show | grep inet6
```

NAT/Routing

NAT

to enable Network Adress Translation (net with private IPs behind one public IP)

```
ipt=/sbin/iptables
if_wan=wan
${ipt} -t nat -A POSTROUTING -o ${if_wan} -j MASQUERADE
```

HW-Nat

HW-Nat is currently only available in LEDE (Kernel 4.9)

i have merged the Lede-Patches to my 4.9-main and ported to 4.14 (4.14-hnat), see [HW-NAT](#)

Routing

enable routing for IPv4

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

alternative:

```
nano /etc/sysctl.conf
#activate net.ipv4.ip_forward=1 and net.ipv6.conf.all.forwarding=1 by
removing # at beginning of line
sysctl -p /etc/sysctl.conf
```

manipulating default route:

```
ip route del default
ip route add default via 192.168.50.2
```

show routing table

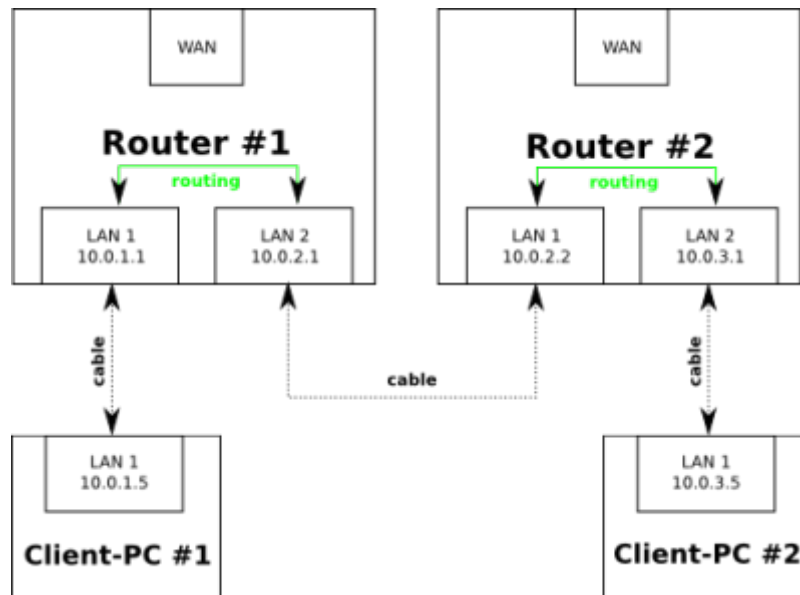
```
ip route show
```

remember you need DNS-resolving (/etc/resolv.conf) for translating domains to ip-addresses

adding static routes to other networks

Pakets are sent to the default-gateway, if the net is not known (directly connected or route available). In normal home-networks there is only 1 router and in this the default-gateway is the Internet-interface and on client-PCs the default-gateway is this router.

static routes are needed, if a net is not directly connected to a router and not accessible via its default-gateway



- in router #1 a static route must be added for net 10.0.3.0/24 with next-hop 10.0.2.2 (send packets over lan#2)
 - `ip route add 10.0.3.0/24 via 10.0.2.2`
- in router #2 a static route must be added for net 10.0.1.0/24 with next-hop 10.0.2.1 (send packets over lan#1)
 - `ip route add 10.0.1.0/24 via 10.0.2.1`

example for net 192.168.50.x behind router with ip 192.168.0.10

```
ip route add 192.168.50.0/24 via 192.168.0.10
```

DNS

/etc/resolv.conf

contains ip-adress to nameserver, e.g.

```
nameserver 192.168.0.10
```

Netbridge

if 2 or more lan-ports should use same network-segment (configure only 1 IP-address for "LAN"), you can bridge ports together.

```
apt-get install bridge-utils
```

/etc/network/interfaces:

```
auto lan1
iface lan1 inet manual
auto lan2
iface lan2 inet manual

auto br0
iface br0 inet static
    address 192.168.40.1
    netmask 255.255.255.0
    bridge_ports lan1 lan2
    bridge_fd 5
    bridge_stp no
```

temporary bridge

```
brctl addbr br0
brctl addif br0 lan1
brctl addif br0 lan2
ip addr add 192.168.40.1/24 dev br0
ip link set br0 up

brctl show br0
```

VLAN

[vlan on dsa-ports need](#)

additional Patch

/etc/network/interfaces:

```
auto lan3.60
iface lan3.60 inet static
    address 192.168.60.10
    netmask 255.255.255.0
```

temporary

```
ip addr add 192.168.40.11/24 dev lan1
ip link set lan1 up
ip link add link lan1 name vlan500 type vlan id 500
ip addr add 192.168.50.1/24 dev vlan500
ip link set vlan500 up
```


vlan aware bridge

With 4.16 vlan aware bridge support was added.

⚠️ `vlan_filtering` needs to be enabled before dsa-ports are added to the bridge, else all traffic (untagged too) is blocked after this setting.

```
#!/bin/bash
BRDEV=br-lan
LANDEV=lan2
BRIP=192.168.40.11/24
VLAN=500
VLANIP=192.168.50.11/24

#first create bridge with vlan-suport and add dsa-port(s)
ip link set eth0 up #ifconfig eth0 up
brctl addbr $BRDEV
ip add add $BRIP dev $BRDEV
ip link set $BRDEV type bridge vlan_filtering 1
brctl addif $BRDEV $LANDEV
ip link set $BRDEV up
ip link set $LANDEV up

#now adding vlan
bridge vlan add vid $VLAN dev $LANDEV master
bridge vlan add vid $VLAN dev $BRDEV self
ip link add link $BRDEV name $BRDEV.$VLAN type vlan id $VLAN
ip add add $VLANIP dev $BRDEV.$VLAN
ip link set $BRDEV.$VLAN up
bridge vlan show
```

testing

```
sudo tcpdump -ei lan1 arp or icmp
```

-e shows link-layer information like vlan

```
sudo tcpdump -XXi lan1 arp or icmp
```

shows arp and icmp-packets as hex-dump on the interface

offset 0x0c should show 8100 followed by hex-value of vlan-number (here vlan 500 = 0x01f4)

```
12:16:26.491644 IP 192.168.50.11 > frank-G5: ICMP echo reply, id 4294, seq
5, length 64
0x0000: 3c18 a003 c3a4 c63a 3897 5920 8100 01f4 <.....:8.Y.....
```

Firewall (iptables)

[iptables](#)

Monitoring

```
sudo tcpdump -i eth0 port not 22 > tcpdump.log  
sudo tcpdump -XXi lan1 arp or icmp
```

From:

<http://www.fw-web.de/dokuwiki/> - **FW-WEB -Wiki**

Permanent link:

<http://www.fw-web.de/dokuwiki/doku.php?id=en:bpi-r2:network:start>

Last update: **2020/09/21 11:18**

