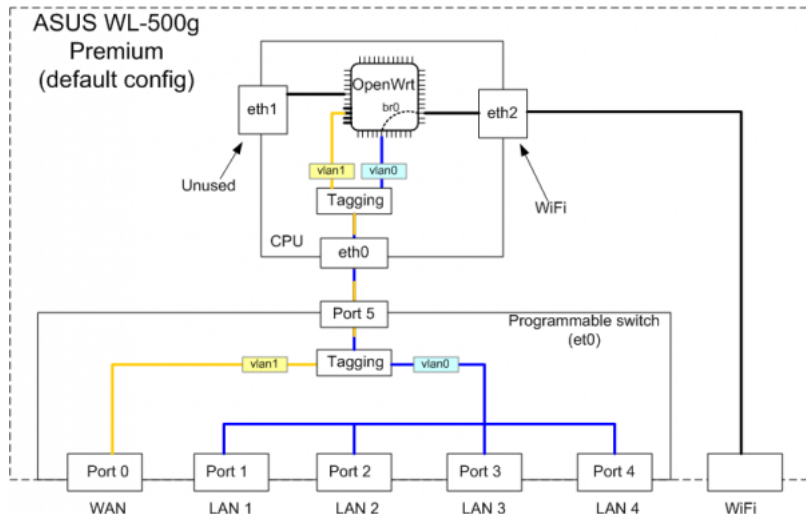


Switch Documentation

Make sure you can safemode or TTL before changing network/switch settings

Fix Me! This page assumes you know what this is and why you want it. (Someone please add info or another page and a link).

If your device has more than 1 LAN port, it may contain a special connection between the different ports called **switch**. Most likely the internals may look like in the following picture



If you want to change how these ports are connected to each other you need to configure the **switch** of your device (see also [network.interfaces](#))

Different routers have different switch layouts, so look at the Wiki for your specific device. The TP-Link Archer C7 has eth0 = WAN, and eth1 = LAN (the 4 switch ports). Port 0 of the switch = eth1 (labelled CPU in Luci), Port 6 = eth0. Port 1 is labelled WAN in Luci. Look at the wiki for your router. Every router is different. The popular TP-Link WDR4300 only has eth0.

UCI config, swconfig style

Known problems

- if a switch interface (for the cpu point of view) is controlling several 'physical interfaces', every time than one physical interface is connected all the switch interface result connected (that means all the ports ethN.Y are UP) and therefore every rule (routing for example) is applied. This could cause serious problem, for example if one relies on the automatic mechanism of routing metric when one route is not available anymore.
 - One way to detect this is: `swconfig dev <switch_interface_name> show | grep link` or see [swconfig](#)

Assumptions

Fix Me! Some of the assumptions, does not see to add up with the provided diagram. Someone familiar with the matter, should either fix them or add a better explanation.

- device is running kernel 2.6 or 3
- device uses an swconfig type switch configuration
- The switch is on eth1. **Fix Me!** (Many are on eth0) [Howto find out: → /proc/switch directory appears to contain the right eth number for the switch. please confirm], and also on chips like rt18366s
- Five-port switch with 0-3 connected externally, 4 not connected, and 5 connected to the CPU's eth1 interface (which adds up to six ports except that 4 is not counted)
- vlan0 is to be all external ports but the last one
- vlan1 is only the last external port [Howto find out which Port corresponds:]
- vlan0 is the default vlan, meaning if a packet is untagged, it will be treated a vlan0 packet

The configuration

The Switch

```
in /etc/config/network

config 'switch' 'eth1'
  option 'enable'      '1'
  option 'enable_vlan' '1'
  option 'reset'       '1'
```

VLAN: switch config

```
in /etc/config/network
```

Notes

The number of the VLAN is specified on the option `vlan` line. The VID (VLAN ID) associated with a VLAN is by default the same as the number of the VLAN. This is overridden by using an option `vid` line so, for example, that VLAN 1 could use VID 100.

In the option `ports` line, a number indicates that the specified vlan includes the port with that number. If the number is followed by a "t" then packets **transmitted** out that port on this VLAN are tagged, and that packets **received** on that port may be received with this VLAN tag. 5 is generally the CPU or 'internal' port and is most often used as tagged. Other suffixes are ignored on devices using `swconfig` but Broadcom `kmod-switch` style interfaces (`/proc/switch/`) use "*" and "u" to indicate PVID and untagged ports respectively (as they have the CPU port implicitly tagged one needs to use "u" to untag it).

So, '0 1 2 3 5t' would mean that packets on this VLAN are transmitted untagged when leaving ports 0, 1, 2 and 3, but tagged when leaving port 5 (generally the CPU internal port as described above).

Tagged packets received on a port will be directed to the VLAN indicated by the VID contained in the packet. Untagged packets received on a port will be directed to the default port VLAN (usually called the PVID). A separate config `switch_port` section is required to set the default port VLAN.

The relevant standards document is 801.2q which says that VID values 0 and 4095 may not be used for tagging packets as they denote reserved values - VID 0 is the default 'native' vlan - leaving 4094 valid values in between, although VID 1 is often reserved for network management (see Dell 2708 for example). This means `vlan0` can be used as a VLAN within or between devices, but you cannot tag packets with it.

The config sections

```
config 'switch_vlan'
    option 'vlan'      '0'
    option 'device'     'eth1'
    option 'ports'      '0 1 2 5t'
```

```
config 'switch_vlan'
    option 'vlan'      '1'
    option 'device'     'eth1'
    option 'ports'      '3 5t'
```

```
config 'switch_port'
    option 'port'       '3'
    option 'pvid'        '1'
```

VLAN: interface/network config

VLAN interface sections look just like regular interface sections, except that instead of `eth1` (or `eth0`, or whatever), you have `eth1.0`, `eth1.1`, etc. where a digit after a `.` is a VLAN number. (that is, for kernel 2.6; 2.4 kernels do something different).

The following example is for a two-interface router, with `eth0` being the WAN and `eth1` being the five-port switch configured as above. It goes in `/etc/config/network`

e.g.

```
config 'interface' 'lan'
    option 'ifname' 'eth1.0'
    option 'proto'  'static'
    option 'ipaddr' '192.168.1.1'
    option 'netmask' '255.255.255.0'
    option 'defaultroute' '0'
    option 'peerdns' '0'
    option 'nat'      '1'
```

```
config 'interface' 'extranet'
    option 'ifname' 'eth1.1'
    option 'proto'  'dhcp'
```

```
config 'interface' 'wan'
    option 'ifname' 'eth0.2'
    option 'proto'  'pppoe'
    option 'username' 'szabozsolt-em'
    option 'password' 'M3IuWBt4'
```

Of course, if you only had a five port switch on `eth0` (and no other interfaces), you might make the `wan` interface `eth0.1` and the `lan` `eth0.0` with appropriately matching `switch`, `switch_vlan` and `switch_port` sections.

See also [backplane](https://en.wikipedia.org/wiki/backplane) [https://en.wikipedia.org/wiki/backplane].

Examples**Example on the asus wl500gp v2 , openwrt 10.03, every physical port**

```
config 'switch' 'eth0'
    option 'enable' '1'
```

```
config 'switch_vlan' 'eth0_0'
    option 'device' 'eth0'
    option 'vlan' '0'
    option 'ports' '4 5' #wan
```

```
config 'switch_vlan' 'eth0_1'
    option 'device' 'eth0'
    option 'vlan' '1'
    option 'ports' '3 5' #lan 1
```

```
config 'switch_vlan' 'eth0_2'
    option 'device' 'eth0'
    option 'vlan' '2'
    option 'ports' '2 5' #lan2
```

```


config 'switch_vlan' 'eth0_3'
    option 'device' 'eth0'
    option 'vlan' '3'
    option 'ports' '1 5' #lan3

config 'switch_vlan' 'eth0_4'
    option 'device' 'eth0'
    option 'vlan' '4'
    option 'ports' '0 5' #lan4

```

#note that to use a particular port in an interface the ifname
#should be 'devicename.vlan' . So for example ifname 'eth0.3'

Example vmware linux guest, openwrt x86 generic 12.09 combined, 2virtualized intel e1000

 More research on vlan on x86 devices has to be done to collect more information on the wiki.

The majority of x86 devices do not have any programmable switch, but it does not seem to be a problem. The syntax used on devices with programmable switches seems completely not necessary.

For example we want to create two 'virtual interfaces' associated to the same physical interface, eth1. To do this, we do the following in /etc/config/network

..other lines...

```

config interface lan1
    option ifname eth1.100
    ...further config...

```

```

config interface lan2
    option ifname eth1.101
    ... further config ...

```

According to what the contributors of this section have read online, so far seems that the packet will be tagged by default, because they are associated to one physical ports that at most will have one PVID (port vlan id) but more than one virtual interfaces. Therefore, having multiple virtual interfaces, the packets must be tagged else it won't make sense, they won't be able to reach the interfaces or to go out.

The tests seems to confirm that because (using a vmware switch and portgroups) to let two openwrt x86 vmware guests reach each other the portgroups had to be configured with the trunk vlan id (that is: vlan id 4095, According to white papers: *VMware Virtual Networking Concepts* and *VMware ESX Server 3 802.1Q VLAN Solutions*).

Side note: if different virtual interfaces related to different vlan are in the same logical network, there will be conflict in terms of metrics, in that case bridging the interfaces could be a solution (has to be tested).

doc/uci/network/switch.txt · Last modified: 2015/12/14 21:34 by tmomas