

Contents

- 1 Description
- 2 Illustration of port and vLAN mappings
 - ◆ 2.1 Illustration of mappings (pre-N, B/G devices)
- 3 vLan assignments and port numbering described
 - ◆ 3.1 Other network device interfaces

Description

The DD-WRT user interface is an awesome piece of work, as it front-ends what amounts to a really quite sophisticated combination of switching and routing. This document is a concise and hopefully mostly accurate description of what the UI is actually configuring for you from the perspective of internal network devices and the data flows between them.

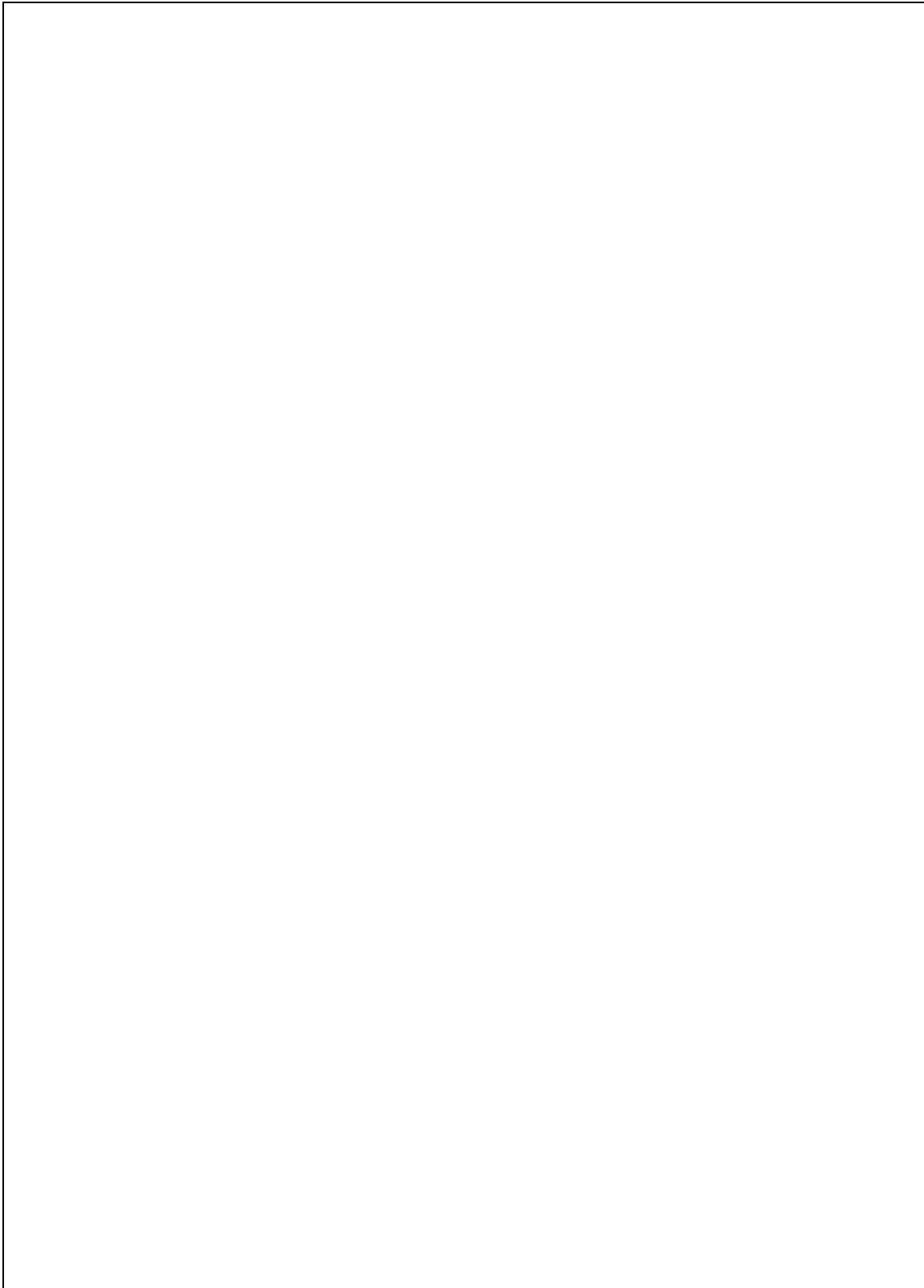
Illustration of port and vLAN mappings

Default_internal_device_network



The default internal device mappings in an 802.11n or newer (AC+) network device (K3x). In network devices previous to 802.11n, the internal numbering of ports and vlans are different. **Some current/newer devices continue to use port 5.** See the command below to find out which one your model uses.

Illustration of mappings (pre-N, B/G devices)



Default_internal_device_network

The internal device mappings in a B/G (pre-N) device - specifically the default configuration of a DD-WRT V23-SP2 firmware on a Linksys WRT54G v2.

By way of a narrative, the default configuration works like this (*parenthesis are pre-N hardware*):

The internal network switch device has 6 ports. One is the WAN RJ45 socket (0), four are the RJ45 sockets (numbered 1-4), and **one is an electrically hardwired vlan trunk interface** (8 '(5)') that connects the switch to the internal router. **The mapping of port numbers to physically exposed external sockets varies according to model.** Do the command: `nvramp show | grep vlan.*ports` to find out which one your model uses. See [this page](#) for details.

vLan assignments and port numbering described

Current routers, typically N or AC, K3.x. Older devices (B/G) have a different vLAN and port numbering scheme. To show the difference between the newer and older, the older is enclosed in italic parenthesis.

The internal port that connects to the router is usually port number 8 (*port 5*). **Some N-and-newer models continue to use port 5 instead of 8. Do a `nvramp show | grep vlan.*ports` to find out which one your model uses. The internal port has a '*'.**

Within the switch entity there are defined two VLANs - vlan1 and vlan2 (*vlan0 and vlan1*). Vlan1 (*vLan0*) is the one on which all of the numbered (1-4) RJ45 sockets on the back belong to. Vlan2 (*vLan1*) is the one on which the WAN socket resides.

In order for it to be possible to move traffic outside the domain of any vLan, it is necessary for traffic to be manipulated by routing and filtering logic. In a DD-WRT device, you make this possible by assuring that any vLan that you've defined has port 8 (*port 5*) as one of its member ports, thus tying the vLan to the router via trunking (note that by extension you can completely isolate a network by *not* including port 8 (*port 5*)). Further, you can declare that **one** of the vLans be the "default": when a packet flows through the router that is otherwise missing vLan tag information, it will behave as if it were a packet on the default vLan. In DD-WRT's default case, this is vLan1 (*vLan0*).

So at this point in the narrative, we have five physical ports mapped to two vLans, both of which connect to the routing system on port 8 (*port 5*). Vlan1 (*vLan0*) is the default one, which includes the LAN ethernet ports.

If we go no further, we have a programmable switch that can describe as many as 5 vlans, between which we can do sophisticated routing and filtering. But that ignores the wireless part of it, which adds some extra complexity.

From the router's perspective, that special port, port 8 (*port 5*) connection to the switch's vLans is the eth0. When port 8 (*port 5*) is associated with a vLan, an additional device named vLan# becomes available to the router by virtue of vLan tagging through the eth0 device. All movement (or non-movement) of packets between the virtual lans is determined by routing logic behind those devices (which is normal linux routing, filtering, etc). **The wireless device is on a separate interface called eth1.** This interface, which is not part of the switch, is available to routing logic just as eth0 and the vLans are. However, DD-WRT by default does not use routing logic per se to move traffic between vLan1 (*vLan0*) and eth1; rather, **it employs a bridge device - who's interface is called br0** - that logically combines vlan1 (*vLan0*) and eth1 into a single interface.

Default_internal_device_network

The effect of this, given that vLan1 (*vLan0*) is the default one and packets from eth1 do not carry vlan tagging, is that vLan1 (*vLan0*) and the wireless network (however that might be arranged) logically behave as a single vlan. No routing is required to get traffic from the wireless domain to vLan1 (*vLan0*); the router will act as though the wireless packets are on the wired segment.

In the default scenario, however, the "enemy" is considered to live on the WAN port (and thus vlan2 (*vLan1*)). So routing and filtering **does** come into play when moving traffic between br0: (vlan1 and eth1) and vlan2

*(So routing and filtering *does* come into play when moving traffic between br0: (vlan0 and eth1) and vlan1).*

Other network device interfaces

There are a few other network device interfaces in addition to ethX, brX, and vlanX:

- teql0 - load sharing device I believe this is used for the "Link Aggregation on Ports 3 & 4" option on the Setup/Vlans page.
- imq0,1 - QOS device or IMQ Device

DD-WRT's distribution includes command line utilities to manipulate all of these devices and mechanisms.

In this dd-wrt forum thread, the default configuration is explained simpler:

- Simple Default Config Explanation