
	Linksys WRT54GS UC3M router configuration manual	
Telematics Engineering Department - UC3M		2006-2007
Version: 2007-03-26		

1. INTRODUCTION

The Linksys WRT54GS is a SOHO (Small Office Home Office) router having 5 Ethernet interfaces and one wireless IEEE 802.11b/g interface:

- Ethernet interfaces. As can be seen in Figure 2, there is a first connector (from left to right) labeled “Internet” in the router, which internally is called `eth0.0`. To its right there are 4 more connectors labeled from 1 to 4 in the router, called internally `eth0.1`, ..., `eth0.4`. Initially (after turn-on) these interfaces are pre-configured with the following IP addresses:
 - `eth0.0`: 192.168.0.1/24
 - `eth0.1`: 192.168.1.1/24
 - `eth0.2`: 192.168.2.1/24
 - `eth0.3`: 192.168.3.1/24
 - `eth0.4`: 192.168.4.1/24
- Wireless interface (WLAN). It is labeled internally `wlan0`. At start-up this interface has the following IP address:
 - `wlan0`: 192.168.5.1/24

Be aware that for specific laboratory practices it may be convenient that you erase the addresses configured by default after adding the ones you actually need.



Figure 1: WRT54GS front view



Figure 2: WRT54GS rear view

The router executes Linux as its Operating System, allowing the installation a series of different applications and utilities. For the Telematics Engineering Department laboratory practices, the installed software¹ allows the configuration of several of the router characteristics and to experiment with router configuration aspects in a way very similar to larger commercial routers. There are 2 complementary ways to access the router:

¹ Consult appendix 1 to see additional details about the installed software.

- Via *TELNET* (to any of the IP addresses configured in the router), accessing a router console². Through that console it is possible to configure all the parameters related with the interfaces (turn-on, turn-off), IP (addresses, static routes) and dynamic routing protocols (RIP, OSPF and BGP).
- Via *SSH*, accessing a Linux console. Such console provides the basic commands found in a Linux system. This way of accessing the router is used to configure the wireless interface. The required access parameters are:
 - User: alumno
 - Password: alumno13

Depending on the type of laboratory practice, the student will have to use one of the access methods (and sometimes both) to configure the router. The laboratory teacher will inform him of the method to be used on each practice. This manual focuses on the first access method: via *TELNET*.

Note: The router's default configuration is restored every time the router is turned on. Therefore, all changes previously done are lost if the router is reset or turned off.

2. BASIC COMMANDS

This section includes the basic router configuration commands that can be accessed through the configuration console via *telnet*. For this, you must configure in your PC an IP address that allows you to connect to the WRT54GS router (the address will depend on the router network's interface to be used to access it from the PC). Once the PC interface is adequately configured, the correct cable connected, and the router turned on (the router needs a few seconds to start-up, during which it is not possible to have access to it), we can do a *telnet* from the PC:

```
telnet 192.168.1.1
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.
```

```
Hello, this is Quagga (version 0.98.4).
Copyright 1996-2005 Kunihiro Ishiguro, et al.
```

```
router#
```

When the *prompt* “router#” appears, it means that we have accessed the router's console (terminal mode). From this start-up console we can perform different actions on the router. To find out the available commands (in this and in any other router mode) we can type “?”:

```
router# ?
clear          Reset functions
configure      Configuration from vty interface
copy           Copy from one file to another
debug          Debugging functions (see also 'undebug')
disable        Turn off privileged mode command
end            End current mode and change to enable mode
exit           Exit current mode and down to previous mode
list           Print command list
no             Negate a command or set its defaults
ping           Send echo messages
quit           Exit current mode and down to previous mode
show           Show running system information
ssh            Open an ssh connection
telnet         Open a telnet connection
terminal       Set terminal line parameters
```

² This console is usually called CLI (*Command Line Interface*)

```

tracert      Trace route to destination
undebg      Disable debugging functions (see also 'debug')
write       Write running configuration to memory, network, or
           terminal
router#

```

As can be seen, there exists a large number of commands accessible from this mode. Some of them can accept additional parameters that can modify their behavior. As before, we can use “?” after a given command to obtain more information about the parameters and their use. For example, typing “ping ?” from this mode, we obtain information about the parameters that the command accepts (in this case, the destination address or name we want to ping, and additionally, if we want to do it using IPv4 or the IPv6 protocol):

```

router# ping ?
WORD Ping destination address or hostname
ip      IP echo
ipv6    IPv6 echo
router# ping

```

Of all the previously listed commands, it may be interesting to describe the following ones³:

- `configure terminal`
 - Reverts to configuration mode. This command is the first step to configure any router parameter:

```

router# configure terminal
router(config)#

```

Note that the router’s *prompt* changes to `router(config)#` when it reverts to the configuration mode.

- `list`
 - Lists all possible commands that can be executed from this mode. The list shows the complete commands, including possible modifiers.
- `ping DESTINATION`
 - Sends a *ping* to the IP address or name provided as parameter. Admits a modifier in front of the destination to specify if it is IPv4 (`ip`) or IPv6 (`ipv6`).
- `show`
 - Shows information about the router’s present configuration. This parameter admits multiple additional modifiers that allow specifying the particular information to be obtained. As an example, the following command shows information about a network interface (use “?” to find out about all the possibilities available with this command):

```

router# show interface eth0.1
Interface eth0.1 is up, line protocol detection is disabled
index 4 metric 1 mtu 1500 <UP,BROADCAST,RUNNING,PROMISC,MULTICAST>
HWaddr: 00:14:bf:d1:eb:07
inet 192.168.1.1/24 broadcast 192.168.1.255
inet 10.10.10.1/24 broadcast 10.10.10.255
input packets 0, bytes 0, dropped 0, multicast packets 0
input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 0, missed 0
output packets 0, bytes 0, dropped 0
output errors 0, aborted 0, carrier 0, fifo 0, heartbeat 0, window 0
collisions 0

```

- `tracert`
 - Sends a *tracert* to the IP address or name given as a parameter. Admits a modifier in front of the destination to specify if it is IPv4 (`ip`) or IPv6 (`ipv6`).

³ There exist other useful commands, specific to other protocols. The usage of such commands is described (in case it is necessary) in the sections referring to each one of the particular protocols.

Within the router's console (CLI), it is possible to use the following movement commands (note that the character <C> means keep the Ctrl, Control key pressed):

- C-f o <RIGHT CURSOR>
 - Moves ahead one character
- C-b o <LEFT CURSOR>
 - Moves back one character
- M-f
 - Displaces the cursor ahead one word
- M-b
 - Displaces the cursor up one word
- C-a
 - Displaces the cursor to the beginning of the line
- C-e
 - Displaces the cursor to the end of the line
- C-c
 - Interrupts the present input and moves to the next line
- C-z
 - Finalizes the present configuration and moves to the initial mode
- C-n o <DOWN CURSOR>
 - Moves to the next line in the history *buffer*
- C-p o <UP CURSOR>
 - Moves to the line before in the history *buffer*
- <TAB>
 - Completes the line pressing on the tab key

The most important configuration commands are described below. As a general rule, to undo/eliminate a particular command's configuration, you should place "no" in front of the command (the next sections show examples of use).

2.1 Interface Commands

Starting from the router's configuration mode (remember that to access the mode you should type `configure terminal` from the start-up console), it is possible to access the network interface configuration sub-menu, typing `interface NAME_OF_THE_NETWORK_INTERFACE`. For example, to access the `vlan1` interface configuration, it is required to type the following command (from the configuration mode):

```
router(config)# interface vlan1
```

The following commands are available inside the network interface configuration sub-menu:

- `shutdown`
 - Disables the interface (in the current interface)
- `no shutdown`
 - Enables the interface (in the current interface)
- `ip address DIRECCIÓN/PREFIJO`
 - Configures (adds) the IP address in the current interface.
- `ip6 address DIRECCIÓN/PREFIJO`
 - Configures (adds) the IPv6 address in the current interface.
- `no ip address DIRECCIÓN/PREFIJO`
 - Eliminates the IP address in the current interface.
- `no ip6 address DIRECCIÓN/PREFIJO`
 - Eliminates the IPv6 address in the current interface.
- `ip address DIRECCIÓN/PREFIJO secondary`
 - Configures (adds) the IP address as secondary. This makes OSPF to not consider the address as one from a different subnet.
- `no ip address DIRECCIÓN/PREFIJO secondary`
 - Eliminates the secondary IP address.

- `description DESCRIPCIÓN ...`
 - Configures the description of the current interface.
- `no description`
 - Eliminates the description of the current interface.
- `multicast`
 - Enables the multicast *flag* of the current interface.
- `no multicast`
 - Disables the multicast *flag* of the current interface.
- `bandwidth <1-100000000>`
 - Configures the current interface's bandwidth. This parameter is taken into consideration to calculate the cost in OSPF. This command does not modify the physical interface configuration.
- `no bandwidth <1-100000000>`
- `exit / quit`
 - Exits the interface configuration sub-menu and returns to the previous menu.

The terminal mode (the router's console start-up mode, identified by the *prompt* "router#"), allows the obtaining of information about the current configuration of a particular interface. To do it, use the following command:

- `show interface NAME_OF_THE_NETWORK_INTERFACE`
 - Shows information about the current configuration of the named interface (NAME_OF_THE_NETWORK_INTERFACE). If the name of the interface (NAME_OF_THE_NETWORK_INTERFACE) is omitted, the command returns the information of all the equipment's interfaces. Example:

```
router# show interface eth0.1
Interface eth0.1 is up, line protocol detection is disabled
index 5 metric 1 mtu 1500 <UP,BROADCAST,RUNNING,PROMISC,MULTICAST>
HWaddr: 00:14:bf:d1:eb:0a
inet 192.168.2.1/24 broadcast 192.168.2.255
input packets 0, bytes 0, dropped 0, multicast packets 0
input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 0, missed 0
output packets 0, bytes 0, dropped 0
output errors 0, aborted 0, carrier 0, fifo 0, heartbeat 0, window 0
collisions 0
```

2.2 Static IP routes configuration commands

In the router's configuration mode (remember that to access it, you have to type configure terminal from the start-up console), it is possible to configure static IP routes. Next, we show a few examples of configuring static IP routes:

- `ip route NETWORK NEXT_HOP`
 - Configures (adds) a route to the destination network NETWORK (with format A.B.C.D/M) through the next hop NEXT_HOP. NEXT_HOP can be the IP address of the next hop (with format A.B.C.D) or the name of the output interface⁴ (in case the destination network is directly connected). Examples:

```
router(config)# ip route 10.0.0.0/8 192.168.0.4
router(config)# ip route 10.0.0.0/8 eth0.2
router(config)# ip route 10.0.0.0/8 null0
```
- `ip route DESTINATION MASK NEXT_HOP`
 - Command identical to the previous one, but using the DESTINATION (with format A.B.C.D) MASK (with format A.B.C.D), instead of NETWORK (indicating the prefix length). Examples:

⁴ If the null0 interface is used, the route is installed as unreachable.

```

router(config)# ip route 10.0.0.0 255.0.0.0 192.168.0.4
router(config)# ip route 255.0.0.0 eth0.2
router(config)# ip route 10.0.0.0 255.0.0.0 null0

```

- `ip route NETWORK NEXT_HOP DISTANCE /`
`ip route DESTINATION MASK NEXT_HOP DISTANCE`
 - Configures (adds) a route with the distance specified in DISTANCE.
- `ip route NETWORK {blackhole/reject} /`
`ip route DESTINATION MASK {blackhole/reject}`
 - In addition, it is possible to configure the router to discard packets going to a destination, specifying the router's behaviour when discarding those packets:
 - ◆ `blackhole`: the routers discards packets silently.
 - ◆ `reject`: the router discards packets and sends a Destination Host Unreachable ICMP message.

To erase a configured route, it is sufficient to put in front of the (previously described) command used to create the route the word “no”.

To add static IPv6 routes, the syntax is the same as before for IPv4, with the difference that the commands have the form “`ipv6 route ...`” instead of “`ip route ...`”

The terminal mode (the router console's start-up mode, identified by the *prompt* “router#”), allows the obtaining of information about the configured routes in the router (independent of whether they were configured statically or using a routing protocol). For that, the following command is used:

- `show ip route`
 - Shows the configured routes currently available in the router, presenting information about the origin of the route. For example:

```

router# sh ip ro
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
I - ISIS, B - BGP, > - selected route, * - FIB route
R>* 10.0.0.0/24 [120/8] via 10.0.2.173, eth0.4, 2d03h03m
R>* 10.0.2.0/26 [120/8] via 10.0.2.166, eth0.3, 2d03h03m
R>* 10.0.2.64/26 [120/8] via 10.0.2.170, eth0.2, 2d03h03m
C>* 10.0.2.128/27 is directly connected, eth0.1
R>* 10.0.2.160/30 [120/8] via 10.0.2.166, eth0.3, 2d03h03m
C>* 10.0.2.164/30 is directly connected, eth0.3
C>* 10.0.2.168/30 is directly connected, eth0.2
C>* 10.0.2.172/30 is directly connected, eth0.4
C>* 127.0.0.0/8 is directly connected, lo
C>* 192.168.0.0/24 is directly connected, eth0.0
R>* 192.168.4.0/24 [120/8] via 10.0.2.166, eth0.3, 2d03h03m
C>* 192.168.5.0/24 is directly connected, wlan0

```

Note that the first output rows returned by the command include a short explanatory legend about some codes in front of each router table input. These codes provide information about the origin of routing table entries: “C” for directly connected routes, “S” for static routes, “R” for routes learned by RIP, “O” for routes obtained by OSPF, “K” for routes configured by the operating system kernel, etc... The symbol “*” implies that the route is in the *Forwarding Information Base* (FIB), that is, the routing table used by the equipment for actual routing. This is useful in order be able to determine which one(s) of the possible routes available for the same destination is being used to route the traffic. Another useful information that this command shows is the administrative distance and the metric of the routes, using the following format: [ADMINISTRATIVE_DISTANCE, METRIC] (if the administrative distance is 0 – which means we are dealing with a directly connected route – this information is not shown).

- The command accepts additional parameters that allow the filtering of the output or to obtain specific information about certain types of routes:

```

router# show ip route ?
<cr>
bgp                Border Gateway Protocol (BGP)

```

connected	Connected
isis	ISO IS-IS (ISIS)
kernel	Kernel
ospf	Open Shortest Path First (OSPF)
rip	Routing Information Protocol (RIP)
static	Static routes
A.B.C.D	Network in the IP routing table to display
A.B.C.D/M	IP prefix <network>/<length>, e.g., 35.0.0.0/8
supernets-only	Show supernet entries only

For example, more information can be obtained regarding a destination network:

```
router# show ip ro 192.168.2.3
Routing entry for 192.168.2.0/24
  Known via "connected", distance 0, metric 0, best
  * directly connected, eth0.2
```

- To show more information about IPv6 routes, use the same syntax but employing “show ipv6 route” instead of “show ip route”.

3. ROUTING PROTOCOLS CONFIGURATION

This section includes information about the RIP, OSPF and BGP routing protocols configuration.

3.1 RIP Configuration

From the router’s configuration mode (remember that to access it, type `configure terminal` from the start-up console), the RIP routing protocol can be enabled by typing `router rip`. To disable it, type `no router rip`. It is necessary to enable the RIP protocol to access its configuration commands. The following command enables the RIP protocol in the router and provides access to the RIP protocol configuration sub-menu:

```
router(config)# router rip
router(config-router)#
```

Within the RIP protocol configuration sub-menu, the following commands are available:

- `network NETWORK`
 - This command enables the RIP protocol in those interfaces that have IP addresses belonging to NETWORK. For example, if the network 10.0.0.0/24 is enabled by RIP, this means that the addresses from 10.0.0.0 till 10.0.0.255 are enabled by RIP. The command `no network` will disable RIP in the specified network.
- `no network NETWORK`
 - Disables the RIP protocol in the network interfaces that have IP addresses belonging to the specified network.
- `network NAME_OF_NETWORK_INTERFACE`
 - This command enables the RIP protocol in the network interface NAME_OF_NETWORK_INTERFACE. Sending and receiving RIP packets are enabled in the specified network interface.
- `no network NAME_OF_NETWORK_INTERFACE`
 - Disables the RIP protocol in the specified network interface.
- `network neighbor A.B.C.D`
 - Specifies the node with IP address A.B.C.D as a RIP neighbour (watch the US spelling!). When a neighbour does not understand multicast, this command is used to specify the neighbours. In many cases, not all routers are capable of understanding multicast (the packets are sent to an address specifying a group of receivers). In a situation where the neighbour does not process multicast packets, it is necessary to establish a direct link between the routers. The `neighbor` command allows the network administrator to specify a particular router as a RIP neighbour.
- `no network neighbor A.B.C.D`

- Disables the node with IP address A.B.C.D as a RIP neighbour.
- `passive-interface NAME_OF_NETWORK_INTERFACE`
 - This command configures the specified interface (`NAME_OF_NETWORK_INTERFACE`) in passive mode. When the interface is configured in passive mode, all packets received are processed as normal and the router does not send RIP packets (multicast nor unicast) except to the neighbours specified in the command `neighbor`. By default all interfaces are configured as passive.
- `no passive-interface NAME_OF_NETWORK_INTERFACE`
 - Configures the specified interface (`NAME_OF_NETWORK_INTERFACE`) as no passive.
- `version VERSION`
 - Configures the RIP protocol version for sending and receiving. The version (`VERSION`) can be "1" (RIPv1) or "2" (RIPv2). The default version is "2".
- `redistribute kernel`
 - Enables the *kernel* routes redistribution (those that have been created by an internal process of the router's Operating System) in the RIP table (this is the table that handles internally the RIP routing protocol, that is, the one that includes the Distance-Vectors and sends and updates the received RIP messages).
- `redistribute kernel metric <0-16>`
 - Same as the preceding, but specifying the value of the metric used to advertise these routes.
- `no redistribute kernel`
 - Disables the *kernel* route redistribution in the RIP table.
- `redistribute static`
 - Enables the redistribution of static routes in the RIP table.
- `redistribute static metric <0-16>`
 - Same as the preceding, but specifying the value of the metric used to advertise these routes.
- `no redistribute static`
 - Disables the redistribution of static routes in the RIP table.
- `redistribute connected`
 - Enables the redistribution of the directly connected routes of those interfaces with RIP disabled in the RIP table. The redistribution of the directly connected routes of the interfaces with RIP enabled is active by default.
- `redistribute connected metric <0-16>`
 - Same as the preceding, but specifying the value of the metric used to advertise these routes.
- `no redistribute connected`
 - Disables the redistribution of the directly connected routes in the RIP table.
- `redistribute ospf`
 - Enables the redistribution of the routes learned using the OSPF routing protocol in the RIP table.
- `redistribute ospf metric <0-16>`
 - Same as the preceding, but specifying the value of the metric used to advertise these routes.
- `no redistribute ospf`
 - Disables the redistribution of the routes learned using OSPF in the RIP table.
- `redistribute bgp`
 - Enables the redistribution of the routes learned using the BGP routing protocol in the RIP table.
- `redistribute bgp metric <0-16>`
 - Same as the preceding, but specifying the value of the metric used to advertise these routes.
- `no redistribute bgp`
 - Disables the redistribution of the routes learned using BGP in the RIP table.
- `default-metric <1-16>`
 - This command modifies the default metric for the distributed routes. The default value is 1. This command does not affect the metric of the directly connected routes even if they are redistributed by the command `redistribute connected`. To modify the value of the metric of a directly connected route, it is necessary to use the command `redistribute connected metric`.
- `distance <1-255>`
 - Configures to a specific value the value of the administrative distance used by the router for those routes learned using RIP. By default, RIP's administrative distance is 120.
- `no distance <1-255>`

- Restores the default value for RIP's administrative distance.
- `distance <1-255> A.B.C.D/M`
 - Configures to a specific value the value of RIP's administrative distance for those routes learned with RIP within the range A.B.C.D/M.
- `no distance <1-255> A.B.C.D/M`
 - Restores the default value of RIP's administrative distance for those routes learned with RIP within the range A.B.C.D/M.
- `timers basic update timeout garbage`
 - Configures the value of the timers used by RIP: `update` `timeout` `garbage`. The default value of those timers is the one specified by the standard, that is:
 - ◆ `update`. Default value: 30 seconds. When the timer reaches 30 seconds an update is produced, the RIP process wakes up and sends an unsolicited response message containing the complete routing table to all the neighbour routers.
 - ◆ `timeout`. Default value: 180 seconds. If no route updates have been received in this time, the route is considered not valid. However, the route is kept for a certain period of time, allowing notifying the neighbours that the route has been eliminated.
 - ◆ `garbage`. Default value: 120 seconds. Once this time has passed after labelling the route as not valid, the route is actually erased from the routing table.
- `no timers basic`
 - Resets the value of the timers to their default values (30, 180, 120).

Besides the configuration commands accessible from the configuration sub-menu of the RIP protocol, some parameters are configurable from the interface configuration sub-menu (recall that is possible to access the interface configuration sub-menu by typing `interface NAME_OF_NETWORK_INTERFACE` from the configuration mode):

- `ip rip send version VERSION`
 - This configuration command overwrites (that is, has preference over) the global value established by the RIP protocol configuration command `rip version`. This command enables the selected network interface to send packets with RIP version 1, RIP version 2, or both. The parameter `VERSION` can be "1", "2" or "1 2". In case of choosing "1 2", the packets are sent by broadcast and multicast. By default, the sent packets are RIPv2.
- `ip rip receive version VERSION`
 - This configuration command overwrites (that is, has preference over) the global value established by the RIP protocol configuration command `rip version`. This command enables the selected network interface to receive packets with RIP version 1, RIP version 2, or both. The parameter `VERSION` can be "1", "2" or "1 2". In case of choosing "1 2", the received packets can be of either version (RIPv1 and RIPv2). By default, the received packets can be of either version.
- `ip split-horizon`
 - Enables *split-horizon* in the selected network interface. By default, *split-horizon* is active.
- `no ip split-horizon`
 - Disables *split-horizon* in the selected network interface.
- `ip split-horizon poisoned-reverse`
 - Enables *split-horizon with poisoned-reverse* in the selected network interface.
- `no ip split-horizon poisoned-reverse`
 - Disables *split-horizon with poisoned-reverse* in the selected network interface.
- `ip rip authentication mode md5`
 - Configures the selected network interface to use authentication of the MD5 type. Note: if some type of authentication is activated in the router, it will continue to answer to packets of type RIPv1 REQUEST (if the reception of RIPv1 is active), but only messages RIPv2 appropriately authenticated can update the router's RIP routing table.
- `no ip rip authentication mode md5`
 - Disables the use of MD5 type authentication in the selected network interface.
- `ip rip authentication mode text`
 - Configures the selected network interface to use authentication of the "plain text" type, using a simple password (sent in plain text).
- `no ip rip authentication mode text`

- Disables the use of “plain text” authentication type in the selected network interface.
- `ip rip authentication string PASSWORD`
 - Configures the password (PASSWORD) when the “plain text” RIP authentication method is used. The PASSWORD text length must be shorter than 16 characters.
- `no ip rip authentication string CONTRASEÑA`
 - Erases the used password (PASSWORD) when the RIP “plain text” authentication mechanism is used.
- `ip rip authentication key-chain MD5_CHAIN`
 - Configures the MD5 chain (MD5_CHAIN) when the RIP MD5 authentication type is used.
- `no ip rip authentication key-chain CADENA_MD5`
 - Erases the used password (MD5_CHAIN) when the RIP MD5 authentication type is used.

The terminal mode (the start-up mode of the router’s console, identified by the *prompt* “router#”), allows finding information about the performance and state of the RIP protocol (besides the information that is shown when consulting the router’s global routing table). Especially interesting are the following commands:

- `show ip rip`
 - Shows the RIP routes (the ones contained in the RIP routing table). For routes received through RIP, this command will show the time when the packet was sent and the label information. This command will also show this information for the routes redistributed by RIP. For example:

```
router# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface
```

	Network	Next Hop	Metric From	Tag	Time
C(i)	10.0.0.0/24	0.0.0.0	1 self	0	
R(n)	10.0.2.0/26	10.0.2.174	3 10.0.2.174	0	02:45
R(n)	10.0.2.64/26	10.0.2.174	3 10.0.2.174	0	02:45
R(n)	10.0.2.128/27	10.0.2.174	2 10.0.2.174	0	02:45
R(n)	10.0.2.160/30	10.0.2.174	3 10.0.2.174	0	02:45
R(n)	10.0.2.164/30	10.0.2.174	2 10.0.2.174	0	02:45
R(n)	10.0.2.168/30	10.0.2.174	2 10.0.2.174	0	02:45
C(i)	10.0.2.172/30	0.0.0.0	1 self	0	
- `show ip rip status`
 - The command shows RIP’s present state. This includes RIP timers, version, enabled network interfaces and RIP neighbours information. For example:

```
router# show ip rip status
Routing Protocol is "rip"
  Sending updates every 30 seconds with +/-50%, next due in 9 seconds
  Timeout after 180 seconds, garbage collect after 120 seconds
  Outgoing update filter list for all interface is not set
  Incoming update filter list for all interface is not set
  Default redistribution metric is 1
  Redistributing:
  Default version control: send version 2, receive any version
    Interface      Send  Recv  Key-chain
    eth0.2         2    1 2
  Routing for Networks:
    eth0.2
  Routing Information Sources:
    Gateway          BadPackets BadRoutes  Distance Last Update
  Distance: (default is 120)
```

References

[1] Quagga manual, <http://www.quagga.net/docs.php>