NAME

iwconfig − configure a wireless network interface

SYNOPSIS

iwconfig [*interface*]
iwconfig *interface* **[essid** *X***] [nwid** *N***] [mode** *M***] [freq** *F***]**
  [channel *C***]***[sens* **S** *]***[ap** *A* **]***[nick* **NN]**
  [rate *R***] [rts** *RT***] [frag** *FT***] [txpower** *T***]**
  [enc *E***] [key** *K***] [power** *P***] [retry** *R***]**
  [modu *M***] [commit]**
iwconfig --help
iwconfig --version

DESCRIPTION

Iwconfig is similar to *ifconfig*(8), but is dedicated to the wireless interfaces. It is used to set the parameters of the network interface which are specific to the wireless operation (for example : the frequency). **Iwconfig** may also be used to display those parameters, and the wireless statistics (extracted from */proc/net/wireless*).

All these parameters and statistics are device dependent. Each driver will provide only some of them depending on hardware support, and the range of values may change. Please refer to the man page of each device for details.

PARAMETERS

essid Set the ESSID (or Network Name - in some products it may also be called Domain ID). The ESSID is used to identify cells which are part of the same virtual network.
As opposed to the AP Address or NWID which define a single cell, the ESSID defines a group of cells connected via repeaters or infrastructure, where the user may roam transparently.
With some cards, you may disable the ESSID checking (ESSID promiscuous) with *off* or *any* (and *on* to reenable it).
If the ESSID of your network is one of the special keywords (*off*, *on* or *any*), you should use -- to escape it.
Examples :
iwconfig eth0 essid any
iwconfig eth0 essid "My Network"
iwconfig eth0 essid -- "ANY"

nwid Set the Network ID. As all adjacent wireless networks share the same medium, this parameter is used to differentiate them (create logical colocated networks) and identify nodes belonging to the same cell.
This parameter is only used for pre-802.11 hardware, the 802.11 protocol uses the ESSID and AP Address for this function.
With some cards, you may disable the Network ID checking (NWID promiscuous) with *off* (and *on* to reenable it).
Examples :
iwconfig eth0 nwid AB34
iwconfig eth0 nwid off

nick[name]
Set the nickname, or the station name. Some 802.11 products do define it, but this is not used as far as the protocols (MAC, IP, TCP) are concerned and completely useless as far as configuration goes. Only some wireless diagnostic tools may use it.
Example :
iwconfig eth0 nickname "My Linux Node"

mode Set the operating mode of the device, which depends on the network topology. The mode can be *Ad-Hoc* (network composed of only one cell and without Access Point), *Managed* (node connects to a network composed of many Access Points, with roaming), *Master* (the node is the synchronisation master or acts as an Access Point), *Repeater* (the node forwards packets between other

wireless nodes), *Secondary* (the node acts as a backup master/repeater), *Monitor* (the node is not associated with any cell and passively monitor all packets on the frequency) or *Auto*.
Example :

iwconfig eth0 mode Managed
iwconfig eth0 mode Ad-Hoc

freq/channel

Set the operating frequency or channel in the device. A value below 1000 indicates a channel number, a value greater than 1000 is a frequency in Hz. You may append the suffix k, M or G to the value (for example, "2.46G" for 2.46 GHz frequency), or add enough '0'.
Channels are usually numbered starting at 1, and you may use *iwlist*(8) to get the total number of channels, list the available frequencies, and display the current frequency as a channel. Depending on regulations, some frequencies/channels may not be available.
When using Managed mode, most often the Access Point dictates the channel and the driver may refuse the setting of the frequency. In Ad-Hoc mode, the frequency setting may only be used at initial cell creation, and may be ignored when joining an existing cell.
You may also use *off* or *auto* to let the card pick up the best channel (when supported).
Examples :

iwconfig eth0 freq 2422000000
iwconfig eth0 freq 2.422G
iwconfig eth0 channel 3
iwconfig eth0 channel auto

ap Force the card to register to the Access Point given by the address, if it is possible. This address is the cell identity of the Access Point, as reported by wireless scanning, which may be different from its network MAC address. If the wireless link is point to point, set the address of the other end of the link. If the link is ad-hoc, set the cell identity of the ad-hoc network.
When the quality of the connection goes too low, the driver may revert back to automatic mode (the card selects the best Access Point in range).
You may also use *off* to re-enable automatic mode without changing the current Access Point, or you may use *any* or *auto* to force the card to reassociate with the currently best Access Point.
Example :

iwconfig eth0 ap 00:60:1D:01:23:45
iwconfig eth0 ap any
iwconfig eth0 ap off

rate/bit[rate]

For cards supporting multiple bit rates, set the bit-rate in b/s. The bit-rate is the speed at which bits are transmitted over the medium, the user speed of the link is lower due to medium sharing and various overhead.
You may append the suffix k, M or G to the value (decimal multiplier : 10^3, 10^6 and 10^9 b/s), or add enough '0'. Values below 1000 are card specific, usually an index in the bit-rate list. Use *auto* to select automatic bit-rate mode (fallback to lower rate on noisy channels), which is the default for most cards, and *fixed* to revert back to fixed setting. If you specify a bit-rate value and append *auto*, the driver will use all bit-rates lower and equal than this value.
Examples :

iwconfig eth0 rate 11M
iwconfig eth0 rate auto
iwconfig eth0 rate 5.5M auto

txpower

For cards supporting multiple transmit powers, sets the transmit power in dBm. If *W* is the power in Watt, the power in dBm is *P = 30 + 10.log(W)*. If the value is postfixed by *mW*, it will be automatically converted to dBm.
In addition, *on* and *off* enable and disable the radio, and *auto* and *fixed* enable and disable power control (if those features are available).
Examples :

iwconfig eth0 txpower 15
iwconfig eth0 txpower 30mW
iwconfig eth0 txpower auto
iwconfig eth0 txpower off

sens Set the sensitivity threshold. This define how sensitive is the card to poor operating conditions (low signal, interference). Positive values are assumed to be the raw value used by the hardware or a percentage, negative values are assumed to be dBm. Depending on the hardware implementation, this parameter may control various functions.
On modern cards, this parameter usually control handover/roaming threshold, the lowest signal level for which the hardware remains associated with the current Access Point. When the signal level goes below this threshold the card starts looking for a new/better Access Point. Some cards may use the number of missed beacons to trigger this. For high density of Access Points, a higher threshold make sure the card is always associated with the best AP, for low density of APs, a lower threshold minimise the number of failed handoffs.
On more ancient card this parameter usually controls the defer threshold, the lowest signal level for which the hardware considers the channel busy. Signal levels above this threshold make the hardware inhibits its own transmission whereas signals weaker than this are ignored and the hardware is free to transmit. This is usually strongly linked to the receive threshold, the lowest signal level for which the hardware attempts packet reception. Proper setting of these thresholds prevent the card to waste time on background noise while still receiving weak transmissions. Modern designs seems to control those thresholds automatically.
Example :
iwconfig eth0 sens -80
iwconfig eth0 sens 2

retry Most cards have MAC retransmissions, and some allow to set the behaviour of the retry mechanism.
To set the maximum number of retries, enter *limit 'value'*. This is an absolute value (without unit), and the default (when nothing is specified). To set the maximum length of time the MAC should retry, enter *lifetime 'value'*. By defaults, this value is in seconds, append the suffix m or u to specify values in milliseconds or microseconds.
You can also add the *short*, *long*, *min* and *max* modifiers. If the card supports automatic mode, they define the bounds of the limit or lifetime. Some other cards define different values depending on packet size, for example in 802.11 *min limit* is the short retry limit (non RTS/CTS packets).
Examples :
iwconfig eth0 retry 16
iwconfig eth0 retry lifetime 300m
iwconfig eth0 retry short 12
iwconfig eth0 retry min limit 8

rts[_threshold]
RTS/CTS adds a handshake before each packet transmission to make sure that the channel is clear. This adds overhead, but increases performance in case of hidden nodes or a large number of active nodes. This parameter sets the size of the smallest packet for which the node sends RTS ; a value equal to the maximum packet size disables the mechanism. You may also set this parameter to *auto*, *fixed* or *off*.
Examples :
iwconfig eth0 rts 250
iwconfig eth0 rts off

frag[mentation_threshold]
Fragmentation allows to split an IP packet in a burst of smaller fragments transmitted on the medium. In most cases this adds overhead, but in a very noisy environment this reduces the error penalty and allow packets to get through interference bursts. This parameter sets the maximum fragment size which is always lower than the maximum packet size.
This parameter may also control Frame Bursting available on some cards, the ability to send

multiple IP packets together. This mechanism would be enabled if the fragment size is larger than the maximum packet size.
You may also set this parameter to *auto*, *fixed* or *off*.
Examples :

iwconfig eth0 frag 512
iwconfig eth0 frag off

key/enc[ryption]

Used to manipulate encryption or scrambling keys and security mode.
To set the current encryption key, just enter the key in hex digits as *XXXX-XXXX-XXXX-XXXX* or *XXXXXXXX*. To set a key other than the current key, prepend or append *[index]* to the key itself (this won't change which is the active key). You can also enter the key as an ASCII string by using the *s:* prefix. Passphrase is currently not supported.
To change which key is the currently active key, just enter *[index]* (without entering any key value).
off and *on* disable and reenable encryption.
The security mode may be *open* or *restricted*, and its meaning depends on the card used. With most cards, in *open* mode no authentication is used and the card may also accept non-encrypted sessions, whereas in *restricted* mode only encrypted sessions are accepted and the card will use authentication if available.
If you need to set multiple keys, or set a key and change the active key, you need to use multiple **key** directives. Arguments can be put in any order, the last one will take precedence.
Examples :

iwconfig eth0 key 0123-4567-89
iwconfig eth0 key [3] 0123-4567-89
iwconfig eth0 key s:password [2]
iwconfig eth0 key [2]
iwconfig eth0 key open
iwconfig eth0 key off
iwconfig eth0 key restricted [3] 0123456789
iwconfig eth0 key 01-23 key 45-67 [4] key [4]

power Used to manipulate power management scheme parameters and mode.
To set the period between wake ups, enter *period 'value'*. To set the timeout before going back to sleep, enter *timeout 'value'*. To set the generic level of power saving, enter *saving 'value'*. You can also add the *min* and *max* modifiers. By default, those values are in seconds, append the suffix m or u to specify values in milliseconds or microseconds. Sometimes, those values are without units (number of beacon periods, dwell, percentage or similar).
off and *on* disable and reenable power management. Finally, you may set the power management mode to *all* (receive all packets), *unicast* (receive unicast packets only, discard multicast and broadcast) and *multicast* (receive multicast and broadcast only, discard unicast packets).
Examples :

iwconfig eth0 power period 2
iwconfig eth0 power 500m unicast
iwconfig eth0 power timeout 300u all
iwconfig eth0 power saving 3
iwconfig eth0 power off
iwconfig eth0 power min period 2 power max period 4

modu[lation]

Force the card to use a specific set of modulations. Modern cards support various modulations, some which are standard, such as 802.11b or 802.11g, and some proprietary. This command force the card to only use the specific set of modulations listed on the command line. This can be used to fix interoperability issues.
The list of available modulations depend on the card/driver and can be displayed using *iwlist modulation*. Note that some card/driver may not be able to select each modulation listed

independently, some may come as a group. You may also set this parameter to *auto* let the card/driver do its best.

Examples :

iwconfig eth0 modu 11g
iwconfig eth0 modu CCK OFDMa
iwconfig eth0 modu auto

commit

Some cards may not apply changes done through Wireless Extensions immediately (they may wait to aggregate the changes or apply it only when the card is brought up via *ifconfig*). This command (when available) forces the card to apply all pending changes.
This is normally not needed, because the card will eventually apply the changes, but can be useful for debugging.

DISPLAY

For each device which supports wireless extensions, *iwconfig* will display the name of the **MAC protocol** used (name of device for proprietary protocols), the **ESSID** (Network Name), the **NWID**, the **frequency** (or channel), the **sensitivity**, the **mode** of operation, the **Access Point** address, the **bit-rate**, the **RTS threshold**, the **fragmentation threshold**, the **encryption key** and the **power management** settings (depending on availability).

The parameters displayed have the same meaning and values as the parameters you can set, please refer to the previous part for a detailed explanation of them.
Some parameters are only displayed in short/abbreviated form (such as encryption). You may use *iwlist*(8) to get all the details.
Some parameters have two modes (such as bitrate). If the value is prefixed by '**=**', it means that the parameter is fixed and forced to that value, if it is prefixed by '**:**', the parameter is in automatic mode and the current value is shown (and may change).

Access Point/Cell

An address equal to 00:00:00:00:00:00 means that the card failed to associate with an Access Point (most likely a configuration issue). The **Access Point** parameter will be shown as **Cell** in ad-hoc mode (for obvious reasons), but otherwise works the same.

If */proc/net/wireless* exists, *iwconfig* will also display its content. Note that those values will depend on the driver and the hardware specifics, so you need to refer to your driver documentation for proper interpretation of those values.

Link quality

Overall quality of the link. May be based on the level of contention or interference, the bit or frame error rate, how good the received signal is, some timing synchronisation, or other hardware metric. This is an aggregate value, and depends totally on the driver and hardware.

Signal level

Received signal strength (RSSI - how strong the received signal is). May be arbitrary units or dBm, *iwconfig* uses driver meta information to interpret the raw value given by */proc/net/wireless* and display the proper unit or maximum value (using 8 bit arithmetic). In *Ad-Hoc* mode, this may be undefined and you should use *iwspy*.

Noise level

Background noise level (when no packet is transmitted). Similar comments as for **Signal level**.

Rx invalid nwid

Number of packets received with a different NWID or ESSID. Used to detect configuration problems or adjacent network existence (on the same frequency).

Rx invalid crypt

Number of packets that the hardware was unable to decrypt. This can be used to detect invalid encryption settings.

Rx invalid frag
Number of packets for which the hardware was not able to properly re-assemble the link layer fragments (most likely one was missing).

Tx excessive retries
Number of packets that the hardware failed to deliver. Most MAC protocols will retry the packet a number of times before giving up.

Invalid misc
Other packets lost in relation with specific wireless operations.

Missed beacon
Number of periodic beacons from the Cell or the Access Point we have missed. Beacons are sent at regular intervals to maintain the cell coordination, failure to receive them usually indicates that the card is out of range.

AUTHOR

Jean Tourrilhes – jt@hpl.hp.com

FILES

/proc/net/wireless

SEE ALSO

ifconfig(8), **iwspy**(8), **iwlist**(8), **iwevent**(8), **iwpriv**(8), **wireless**(7).