

# Linux

## Deprecated Linux networking commands and their replacements

In my [article](#) detailing the command line utilities available for configuring and troubleshooting network properties on Windows and Linux, I mentioned some Linux tools that, while still included and functional in many Linux distributions, are actually considered [deprecated](#) and therefore should be phased out in favor of more modern replacements.

Specifically, the deprecated Linux networking commands in question are: **arp**, **ifconfig**, **iptunnel**, **iwconfig**, **nameif**, **netstat**, and **route**. These programs (except **iwconfig**) are included in the [net-tools](#) package that has been unmaintained for years. The functionality provided by several of these utilities has been reproduced and improved in the new [iproute2](#) suite, primarily by using its new **ip** command. The **iproute2** software code is available from [Kernel.org](#). **iproute2** documentation is available from [the Linux Foundation](#) and [PolicyRouting.org](#).

### Deprecated command

**arp**

**ifconfig**

**iptunnel**

**iwconfig**

**nameif**

**netstat**

**route**

### Replacement command

**ip n (ip neighbor)**

**ip a (ip addr), ip link, ip -s**

**ip tunnel**

**iw**

**ip link, ifrename**

**ss, ip route (for netstat-r),**

**netstat-g)**

**ip r (ip route)**

Now let's take a closer look at these deprecated commands and their replacements.

This article will not focus on **iproute2** or the **ip** command in detail; instead it will simply give one-to-one mappings between the deprecated commands and their new counterparts. For replacement commands that are listed as 'not apparent', please [contact me](#) if you know otherwise.

### Jump to:

- [Arp](#)
- [Ifconfig](#)
- [Iptunnel](#)
- [Iwconfig](#)
- [Nameif](#)
- [Netstat](#)
- [Route](#)
- [Discussion](#)
- [Recommended reading](#)

Please note that **nslookup** and **dig** are covered separately [here](#).

## Arp

### Deprecated arp commands

**arp -a [host] or --all [host]**

Shows the entries of the specified host name or IP address. If the **[host]** parameter is not used, all entries will be displayed.

**arp -d [ip\_addr] or --delete [ip\_addr]**

### Replacement

**ip n (or ip neighbor), or ip**

**ip n del [ip\_addr]** (this "inv

# Linux

Removes the ARP cache entry for the specified host.

**arp -D** or **--use-device**

**ip n f [ip\_addr]** (or **ip n flu**)

Not apparent

Uses the hardware address associated with the specified interface.

**arp -e**

Not apparent

Shows the entries in default (Linux) style.

**arp -f [filename]** or **--file [filename]**

Not apparent

Similar to the **-s** option, only this time the address info is taken from the file that **[filename]** set up. If no **[filename]** is specified, `/etc/ethers` is used as default.

**arp -H** or **--hw-type [type]** or **-t [type]**

Not apparent

When setting or reading the ARP cache, this optional parameter tells **arp** which class of entries it should check for. The default value of this parameter is **ether** (i.e. hardware code 0x01 for IEEE 802.3 10Mbps Ethernet).

**arp -i [int]** or **--device [int]**

**ip n [add | chg | del | repl]**

Selects an interface. When dumping the ARP cache only entries matching the specified interface will be printed. For example, **arp -i eth0 -s 10.21.31.41 A321.ABCF.321A** creates a static ARP entry associating IP address 10.21.31.41 with MAC address A321.ABCF.321A on **eth0**.

**arp -n** or **--numeric**

Not apparent

Shows IP addresses instead of trying to determine domain names.

**arp -s [ip\_addr] [hw\_addr]** or **--set [ip\_addr]**

**ip n add [ip\_addr] lladdr [hw\_addr] [nud\_state]** (see example below)

Manually creates a static ARP address mapping entry for host **[ip\_addr]** with the hardware address set to **[hw\_addr]**.

**arp -v**

**ip -s n** (or **ip -stats n**)

Uses verbose mode to provide more details.

Some **ip neighbor** examples are as follows:

```
# ip n del 10.1.2.3 dev eth0
```

Invalidates the ARP cache entry for host 10.1.2.3 on device **eth0**.

```
# ip neighbor show dev eth0
```

Shows the ARP cache for interface **eth0**.

```
# ip n add 10.1.2.3 lladdr 1:2:3:4:5:6 dev eth0 nud perm
```

Adds a “permanent” ARP cache entry for host 10.1.2.3 device **eth0**. The Neighbor Unreachability Detection (**nud**) state can be one of the following:

- **noarp** – entry is valid. No attempts to validate this entry will be made but it can be removed when its lifetime expires.
- **permanent** – entry is valid forever and can be only be removed administratively.
- **reachable** – entry is valid until the reachability timeout expires.
- **stale** – entry is valid but suspicious.

## Ifconfig

# Linux

## Deprecated ifconfig commands

### ifconfig

Displays details on all network interfaces.

**ifconfig [interface]**

The name of the interface. This is usually a driver name followed by a unit number; for example, **eth0** for the first Ethernet interface. **Eth0** will usually be a PC's primary network interface card (NIC).

**ifconfig [address\_family]**

To enable the interpretation of differing naming schemes used by various protocols, **[address\_family]** is used for decoding and displaying all protocol addresses. Currently supported address families include **inet** (TCP/IP, default), **inet6** (IPv6), **ax25** (AMPR Packet Radio), **ddp** (Appletalk Phase 2), **ipx** (Novell IPX) and **netrom** (AMPR Packet radio).

**ifconfig [interface] add [address/prefixlength]**

Adds an IPv6 address to the **[interface]**.

**ifconfig [interface] address [address]**

Assigns the specified IP **[address]** to the specified **[interface]**.

**ifconfig [interface] allmulti** or **-allmulti**

Enables or disables all-multicast mode. If selected, all multicast packets on the network will be received by the **[interface]** specified. This enables or disables the sending of incoming frames to the kernel's network layer.

**ifconfig [interface] arp** or **-arp**

Enables or disables the use of the ARP protocol on this **[interface]**.

**ifconfig [interface] broadcast [address]**

Specifies the address to use to use for broadcast transmissions. By default, the broadcast address for a subnet is the IP address with all ones in the host portion of the subnet address (i.e., a.b.c.255 for a /24 subnet).

**ifconfig [interface] del [address/prefixlength]**

Removes an IPv6 address from the **[interface]**, such as **eth0**.

**ifconfig [interface] down**

Disables the **[interface]**, such as **eth0**.

**ifconfig [interface] hw [class] [address]**

Sets the hardware (MAC) address of this **[interface]**, if the device driver supports this operation. The keyword must be followed by the name of the hardware **[class]** and the printable ASCII equivalent of the hardware address. Hardware classes currently supported include **ether** (Ethernet), **ax25** (AMPR AX.25), **ARCnet** and **netrom** (AMPR NET/ROM).

**ifconfig [interface] io\_addr [address]**

Sets the start **[address]** in [I/O space](#) for this device.

**ifconfig [interface] irq [address]**

Sets the [interrupt line](#) used by the network interface.

**ifconfig [interface] mem\_start [address]**

Sets the start address for [shared memory](#) of the interface.

**ifconfig [interface] media [type]**

## Replacement

**ip a** (or **ip addr**)

**ip a show dev [interface]**

**ip -f [family] a**

**[family]** can be **inet** (IPv4), and **-6** = **-f inet6**.

**ip a add [ip\_addr/mask] d**

**ip a add [ip\_addr/mask] d**

**ip mr iif [name]** or **ip mro**  
which multicast packets are

**ip link set arp on** or **arp of**

**ip a add broadcast [ip\_ad**

**ip link set dev [interface]**  
broadcast address)

**ip a del [ipv6\_addr** or **ipv4**

**ip link set dev [interface]**

**ip link set dev [interface]**

Not apparent; possibly **eth0**

Not apparent; possibly **eth0**

Not apparent; possibly **eth0**

Not apparent; possibly **eth0**

# Linux

Sets physical port or medium type. Examples of **[type]** are **10baseT**, **10base2**, and **AUI**. A **[type]** value of **auto** will tell the interface driver to automatically determine the media type (driver support for this command varies).

**ifconfig [interface] mtu [n]**

**ip link set dev [interface]**

Sets the Maximum Transfer Unit ([MTU](#)) of an interface to **[n]**.

**ifconfig [interface] multicast**

**ip link set dev [interface]**

Sets the [multicast flag](#) on the interface (should not normally be needed as the drivers set the flag correctly themselves).

**ifconfig [interface] netmask [mask\_address]**

Not apparent

Sets the subnet mask (not the IP address) for this **[interface]**. This value defaults to the standard Class A, B, or C subnet masks (based on the interface IP address) but can be changed with this command.

**ifconfig [interface] pointopoint** or **-pointopoint**

not apparent; possibly **ipppe** specifies the address of the

Enables or disables [point-to-point](#) mode on this **[interface]**.

**ifconfig [interface] promisc** or **-promisc**

**ip link set dev [interface]**

Enables or disables [promiscuous mode](#) on the **[interface]**.

**ifconfig [interface] txqueuelen [n]**

**ip link set dev [interface]**

Sets the [transmit queue length](#) on the **[interface]**. Smaller values are recommended for connections with high latency (i.e., dial-up modems, ISDN, etc).

**ifconfig [interface] tunnel [address]**

**ip tunnel mode sit** (other p

Creates a Simple Internet Transition (IPv6-in-IPv4) device which tunnels to the IPv4 **[address]** provided.

**ifconfig [interface] up**

**ip link set [interface] up**

Activates (enables) the **[interface]** specified.

Some examples illustrating the **ip** command are as follows; using the table above you should be able to figure out what they do.

```
# ip link show dev eth0
# ip a add 10.11.12.13/8 dev eth0
# ip link set dev eth0 up
# ip link set dev eth0 mtu 1500
# ip link set dev eth0 address 00:70:b7:d6:cd:ef
```

## Iptunnel

### Deprecated iptunnel commands

**iptunnel [add | change | del | show]**

### Replacement

**ip tunnel a** or **add**  
**ip tunnel chg** or **change**

# Linux

**iptunnel add [name] [mode {ipip | gre | sit} ] remote [remote\_addr]  
local [local\_addr]  
iptunnel -V or --version**

**ip tunnel d or del  
ip tunnel ls or show  
ip tunnel add [name] [mo  
}] remote [remote\_addr]  
not apparent**

The syntax between **iptunnel** and **ip tunnel** is very similar as these examples show.

```
# [iptunnel | ip tunnel] add ipip-tunl1 mode ipip remote 83.240.67.86 (ipip-tunl1 is the name of the  
tunnel, 83.240.67.86 is the IP address of the remote endpoint).  
# [iptunnel | ip tunnel] add ipi-tunl2 mode ipip remote 104.137.4.160 local 104.137.4.150 ttl 1  
# [iptunnel | ip tunnel] add gre-tunl1 mode gre remote 192.168.22.17 local 192.168.10.21 ttl 255
```

**Iptunnel** is covered in more depth [here](#).

## Iwconfig

**Iwconfig's** successor, **iw**, is still in development. Official documentation for **iw** is available [here](#) and [here](#).

### Deprecated iwconfig commands

#### **iwconfig**

Displays basic details about wireless interfaces, such as supported protocols ([802.11a/b/g/n](#)), Extended Service Set ID ([ESSID](#)), mode, and access point. To view these details about a particular interface, use **iwconfig [interface]** where the interface is the device name, such as **wlan0**.

**iwconfig [interface] ap [address]**

**Replacement  
iw dev [interface] link**

Not apparent

Forces the wireless adapter to register with the access point given by the **[address]**, if possible. This address is the cell identity of the access point (as reported by wireless scanning) which may be different from its MAC address.

**iwconfig commit**

Not apparent

Some wireless adapters may not apply changes immediately (they may wait to aggregate the changes, or apply them only when the card is brought up via **ifconfig**). This command (when available) forces the adapter to immediately apply all pending changes.

**iwconfig [interface] essid [name]**

**iw [interface] connect [na**

Connects to the WLAN with the ESSID **[name]** provided. With some wireless adapters, you can disable the ESSID checking (ESSID promiscuous) with **off** or **any** (and **on** to re-enable it).

**iwconfig [interface] frag [num]**

Not apparent

Sets the maximum fragment size which is always lower than the maximum packet size. This parameter may also control Frame Bursting available on some wireless adapters (the ability to send multiple IP packets together). This mechanism would be enabled if the fragment size is larger than the maximum packet size. Other valid frag parameters to **auto**, **on**, and **off**.

**iwconfig [interface] [freq | channel]**

**iw dev [interface] set fre**

Sets the operating frequency or channel on the wireless device. A value below 1000 indicates a channel number, a value greater than 1000 is a frequency in

**iw dev [interface] set cha**

# Linux

Hz. You can append the suffix **k**, **M** or **G** to the value (for example, "2.46G" for 2.46 GHz frequency). You may also use **off** or **auto** to let the adapter pick up the best channel (when supported).

**iwconfig [interface] key [key] [mode] [on | off]**

**iw [interface] connect [name]**

To set the current encryption **[key]**, just enter the key in hex digits as XXXX-XXXX-XXXX-XXXX or XXXXXXXX. You can also enter the key as an ASCII string by using the **s:** prefix. **On** and **off** re=enable and disable 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.

To connect to an AP with WPA2-PSK:

**iwconfig [interface] mode [mode]**

Not apparent

Sets the operating mode of the wireless device. The **[mode]** can be **Ad-Hoc**, **Auto**, **Managed**, **Master**, **Monitor**, **Repeater**, or **Secondary**.

**Ad-Hoc:** the network is composed of only one cell and without an access point.

**Managed:** the wireless node connects to a network composed of many access points, with roaming.

**Master:** the wireless node is the synchronization master, or it acts as an access point.

**Monitor:** the wireless node is not associated with any cell and passively monitors all packets on the frequency.

**Repeater:** the wireless node forwards packets between other wireless nodes.

**Secondary:** the wireless node acts as a backup master/repeater.

**iwconfig [interface] modu [modulation]**

Not apparent

Forces the wireless adapter to use a specific set of modulations. Modern adapters support various modulations, such as 802.11b or 802.11g. The list of available modulations depends on the adapter/driver and can be displayed using **iwlist modulation**. Some options are **11g**, **CCK OFDMA**, and **auto**.

**iwconfig [interface] nick [name]**

Not apparent

Sets the nick name (or station name).

**iwconfig [interface] nwid [name]**

Not apparent

Sets the Network ID for the WLAN. This parameter is only used for pre-802.11 hardware as the 802.11 protocol uses the ESSID and access point address for this function. With some wireless adapters, you can disable the Network ID checking (NWID promiscuous) with **off** (and **on** to re-enable it).

**iwconfig [interface] power [option]**

Not apparent; some power c

**iwconfig [interface] power min | max [secondsu | secondsm]**

**iwconfig [interface] power mode [mode]**

**iw dev [interface] set pow**

**iwconfig [interface] power on | off**

**iw dev [interface] get pow**

Configures the power management scheme and mode. Valid **[options]** are: **period [value]** (sets the period between wake ups), **timeout [value]** (sets the timeout before going back to sleep), **saving [value]** (sets the generic level of power saving).

The **min** and **max** modifiers are in seconds by default, but append the suffices **m** or **u** to specify values in milliseconds or microseconds.

Valid **[mode]** options are: **all** (receive all packets), **unicast** (receive unicast packets only, discard multicast and broadcast) and **multicast** (receive multicast and broadcast only, discard unicast packets).

**On** and **off** re-enable or disable power management.

**iwconfig [interface] rate/bit [rate]**

**iw [interface] set bitrates**

Sets the bit rate in bits per second for cards supporting multiple bit rates. The bit-

# Linux

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. Suffixes **k**, **M** or **G** can be added to the numeric **[rate]** (decimal multiplier :  $10^3$ ,  $10^6$  and  $10^9$  b/s), or add '0' for enough. The **[rate]** can also be **auto** to select automatic bit-rate mode (fallback to lower rate on noisy channels), or **fixed** to revert back to fixed setting. If you specify a bit-rate numeric value and append **auto**, the driver will use all bit-rates lower and equal than this value.

**iwconfig [interface] retry [option] [value]**

Not apparent

To set the maximum number of retries (MAC retransmissions), enter **limit [value]**. To set the maximum length of time the MAC should retry, enter **lifetime [value]**. By default, this value is in seconds; append the suffices **m** or **u** to specify values in milliseconds or microseconds. You can also add the **short**, **long**, **min** and **max** modifiers.

**iwconfig [interface] rts [threshold]**

Not apparent

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 the threshold parameter to **auto**, **fixed** or **off**.

**iwconfig [interface] sens [threshold]**

Not apparent

Sets the sensitivity threshold (defines how sensitive the wireless adapter is to poor operating conditions such as low signal, signal interference, etc). Modern adapter designs seem to control these thresholds automatically.

**iwconfig [interface] txpower [value]**

**iw dev [interface] set txp**

For adapters supporting multiple transmit powers, this sets the transmit power in dBm. If **W** is the power in Watt, the power in dBm is  $P = 30 + 10 \cdot \log(W)$ . 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).

**iwconfig --help**

**iw help**

Displays the iwconfig help message.

**iwconfig --version**

**iw --version**

Displays the version of iwconfig installed.

Some examples of the **iw** command syntax are as follows.

```
# iw dev wlan0 link
# iw wlan0 connect CoffeeShopWLAN
# iw wlan0 connect HomeWLAN keys 0:abcde d:1:0011223344 (for WEP)
```

## Nameif

### Deprecated nameif commands

**nameif [name] [mac\_address]**

### Replacement

**ip link set dev [interface]**

If no name and MAC address are provided, it attempts to read addresses from `/etc/mactab`. Each line of mactab should contain an interface name and MAC address (or comments starting with #).

**ifrename -i [interface] -n**

**nameif -c [config\_file]**

**ifrename -c [config\_file]**

# Linux

Reads from **[config\_file]** instead of /etc/mactab.

**nameif -s**

Not apparent

Error messages are sent to the syslog.

## Netstat

### Deprecated netstat commands

**netstat -a** or **--all**

### Replacement

**ss -a** or **--all**

Shows both listening and non-listening [sockets](#).

**netstat -A [family]** or **--protocol=[family]**

**ss -f [family]** or **-family=[family]**

Specifies the address families for which connections are to be shown. **[family]** is a comma separated list of address family keywords like **inet**, **unix**, **ipx**, **ax25**, **netrom**, and **ddp**. This has the same effect as using the **--inet**, **--unix (-x)**, **--ipx**, **--ax25**, **--netrom**, and **--ddp** options.

**netstat -c** or **--continuous**

Not apparent

Configures **netstat** to refresh the displayed information every second until stopped.

**netstat -C**

**ip route list cache**

Prints routing information from the route cache.

**netstat -e** or **--extend**

**ss -e** or **--extended**

Displays an increased level of detail. Can be entered as twice (as **--ee**) for maximum details.

**netstat -F**

Not apparent

Prints routing information from the forward information database ([FIB](#)).

**netstat -g** or **--groups**

**ip maddr**, **ip maddr show**

Displays multicast group membership information for IPv4 and IPv6.

**netstat -i** or **--interface=[name]**

**ip -s link**

Displays a table of all network interfaces, or the specified **[name]**.

**netstat -l** or **--listening**

**ss -l** or **--listening**

Shows only listening sockets (which are omitted by **netstat** by default).

**netstat -M** or **--masquerade**

Not apparent

Displays a list of [masqueraded](#) connections (connections being altered by Network Address Translation).

**netstat -n** or **--numeric**

**ss -n** or **--numeric**

Show numerical addresses instead of trying to determine symbolic host, port or user names (skips DNS translation).

**netstat --numeric-hosts**

Not apparent

Shows numerical host addresses but does not affect the resolution of port or user names.

**netstat --numeric ports**

Not apparent

Shows numerical port numbers but does not affect the resolution of host or user

# Linux

names.

**netstat --numeric-users**

Not apparent

Shows numerical user IDs but does not affect the resolution of host or port names.

**netstat -N** or **--symbolic**

**ss -r** or **--resolve**

Displays the symbolic host, port, or user names instead of numerical representations. **Netstat** does this by default.

**netstat -o** or **--timers**

**ss -o** or **--options**

Includes information related to networking timers.

**netstat -p** or **--program**

**ss -p**

Shows the process ID (PID) and name of the program to which each socket belongs.

**netstat -r** or **--route**

**ip route**, **ip route show** all

Shows the kernel routing tables.

**netstat -s** or **--statistics**

**ss -s**

Displays summary statistics for each protocol.

**netstat -t** or **--tcp**

**ss -t** or **--tcp**

Filters results to display TCP only.

**netstat -T** or **--notrim**

Not apparent

Stops trimming long addresses.

**netstat -u** or **--udp**

**ss -u** or **--udp**

Filters results to display UDP only.

**netstat -v** or **--verbose**

Not apparent

Produces verbose output.

**netstat -w** or **--raw**

**ss -w** or **--raw**

Filter results to display [raw sockets](#) only.

**netstat -Z** or **--context**

Not apparent

Prints the [SELinux](#) context if SELinux is enabled. On hosts running SELinux, all processes and files are labeled in a way that represents security-relevant information. This information is called the SELinux context.

## Route

### Deprecated route commands

**route**

### Replacement

**ip route**

Displays the host's routing tables.

**route -A [family] [add]** or **route --[family] [add]**

**ip -f [family] route**

Uses the specified address family with **add** or **del**. Valid families are **inet** (DARPA Internet), **inet6** (IPv6), **ax25** (AMPR AX.25), **netrom** (AMPR NET/ROM), **ipx** (Novell IPX), **ddp** (Appletalk DDP), and **x25** (CCITT X.25).

**route -C** or **--cache**

**[family]** can be **inet** (IP), **inet6**, or **ax25**.  
= **-f inet6**.

Not apparent; **ip route show**

Operates on the kernel's routing cache instead of the forwarding information

# Linux

base ([FIB](#)) routing table.

**route -e** or **-ee**

**ip route show**

Uses the **netstat-r** format to display the routing table. **-ee** will generate a very long line with all parameters from the routing table.

**route -F** or **--fib**

Not apparent

Operates on the kernel's Forwarding Information Base ([FIB](#)) routing table (default behavior).

**route -h** or **--help**

**ip route help**

Prints the help message.

**route -n**

Not apparent

Shows numerical IP addresses and bypass host name resolution.

**route -v** or **--verbose**

**ip -s route**

Enables verbose command output.

**route -V** or **--version**

**ip -V**

Displays the version of **net-tools** and the **route** command.

**route add** or **del**

**ip route [add | chg | repl | ...]**

Adds or delete a route in the routing table.

**route [add or del] dev [interface]**

**ip route [add | chg | repl | ...]**

Associates a route with a specific device. If **dev [interface]** is the last option on the command line, the word dev may be omitted.

**route [add or del] [default] gw [gw]**

**ip route add default via [ ... ]**

Routes packets through the specified gateway IP address.

**route [add or del] -host**

Not apparent

Specifies that the target is a host (not a network).

**route [add or del] -irrt [n]**

Not apparent; **ip route [add | chg | repl | ...] estimate; rttvar [number]**

Sets the initial round trip time ([IRTT](#)) for TCP connections over this route to **[n]** milliseconds (1-12000). This is typically only used on AX.25 networks. If omitted the [RFC 1122](#) default of 300ms is used.

**route [add or del] -net**

Not apparent

Specifies that the target is a network (not a host).

**route [add or del] [-host or -net] netmask [mask]**

Not apparent

Sets the subnet **[mask]**.

**route [add or del] metric [n]**

**ip route [add | chg | repl | ...]**

Sets the metric field in the routing table (used by routing daemons) to the value of **[n]**.

**route [add or del] mod, dyn, or reinstate**

Not apparent

Install a dynamic or modified route. These flags are for diagnostic purposes, and are generally only set by routing daemons.

**route [add or del] mss [bytes]**

**ip route [add | chg | repl | ...] these destinations when estab**

Sets the TCP Maximum Segment Size ([MSS](#)) for connections over this route to the number of **[bytes]** specified.

**route [add or del] reject**

**ip route add prohibit [net ...]**

# Linux

Installs a blocking route, which will force a route lookup to fail. This is used to mask out networks before using the default route. This is not intended to provide firewall functionality.

**route [add or del] window [n]**

**ip route [add | chg | repl | del]**

Set the [TCP window size](#) for connections over this route to the value of **[n]** bytes. This is typically only used on AX.25 networks and with drivers unable to handle back-to-back frames.

Some examples of **ip route** command syntax are as follows.

```
# ip route add 10.23.30.0/24 via 192.168.8.50
# ip route del 10.28.0.0/16 via 192.168.10.50 dev eth0
# ip route chg default via 192.168.25.110 dev eth1
# ip route get [ip_address] (shows the interface and gateway that would be used to reach a remote host. This command would be especially useful for troubleshooting routing issues on hosts with large routing tables and/or with multiple network interfaces).
```

Quelle: [Doug Vitale Tech Blog](#)

Eindeutige ID: #1082

Verfasser: n/a

Letzte Änderung: 2018-09-29 21:26