NAME

ip-neighbour − neighbour/arp tables management.

SYNOPSIS

ip [*OPTIONS*] **neigh** { *COMMAND* | **help** }

ip neigh { **add** | **del** | **change** | **replace** } { *ADDR* [**lladdr** *LLADDR*] [**nud** *STATE*] | **proxy** *ADDR* } [**dev** *DEV*] [**router**] [**extern_learn**]

ip neigh { **show** | **flush** } [**proxy**] [**to** *PREFIX*] [**dev** *DEV*] [**nud** *STATE*] [**vrf** *NAME*]

ip neigh get *ADDR* **dev** *DEV*

STATE := { **permanent** | **noarp** | **stale** | **reachable** | **none** | **incomplete** | **delay** | **probe** | **failed** }

DESCRIPTION

The **ip neigh** command manipulates *neighbour* objects that establish bindings between protocol addresses and link layer addresses for hosts sharing the same link. Neighbour entries are organized into tables. The IPv4 neighbour table is also known by another name - the ARP table.

The corresponding commands display neighbour bindings and their properties, add new neighbour entries and delete old ones.

ip neighbour add
    add a new neighbour entry

ip neighbour change
    change an existing entry

ip neighbour replace
    add a new entry or change an existing one

    These commands create new neighbour records or update existing ones.

    to *ADDRESS* **(default)**
        the protocol address of the neighbour. It is either an IPv4 or IPv6 address.

    dev *NAME*
        the interface to which this neighbour is attached.

    proxy indicates whether we are proxying for this neigbour entry

    router indicates whether neigbour is a router

    extern_learn
        this neigh entry was learned externally. This option can be used to indicate to the kernel that this is a controller learnt dynamic entry. Kernel will not gc such an entry.

    lladdr *LLADDRESS*
        the link layer address of the neighbour. *LLADDRESS* can also be **null**.

nud *STATE*

the state of the neighbour entry. **nud** is an abbreviation for 'Neighbour Unreachability Detection'. The state can take one of the following values:

permanent
the neighbour entry is valid forever and can be only be removed administratively.

noarp the neighbour entry is valid. No attempts to validate this entry will be made but it can be removed when its lifetime expires.

reachable
the neighbour entry is valid until the reachability timeout expires.

stale the neighbour entry is valid but suspicious. This option to **ip neigh** does not change the neighbour state if it was valid and the address is not changed by this command.

none this is a pseudo state used when initially creating a neighbour entry or after trying to remove it before it becomes free to do so.

incomplete
the neighbour entry has not (yet) been validated/resolved.

delay neighbor entry validation is currently delayed.

probe neighbor is being probed.

failed max number of probes exceeded without success, neighbor validation has ultimately failed.

ip neighbour delete

delete a neighbour entry

The arguments are the same as with **ip neigh add**, except that **lladdr** and **nud** are ignored.

Warning: Attempts to delete or manually change a **noarp** entry created by the kernel may result in unpredictable behaviour. Particularly, the kernel may try to resolve this address even on a **NOARP** interface or if the address is multicast or broadcast.

ip neighbour show

list neighbour entries

to *ADDRESS* **(default)**
the prefix selecting the neighbours to list.

dev *NAME*
only list the neighbours attached to this device.

vrf *NAME*
only list the neighbours for given VRF.

proxy list neighbour proxies.

unused only list neighbours which are not currently in use.

nud *STATE*

only list neighbour entries in this state. *NUD_STATE* takes values listed below or the special value **all** which means all states. This option may occur more than once. If this option is absent, **ip** lists all entries except for **none** and **noarp**.

ip neighbour flush

flush neighbour entries
This command has the same arguments as **show.** The differences are that it does not run when no arguments are given, and that the default neighbour states to be flushed do not include **permanent** and **noarp**.

With the **-statistics** option, the command becomes verbose. It prints out the number of deleted neighbours and the number of rounds made to flush the neighbour table. If the option is given twice, **ip neigh flush** also dumps all the deleted neighbours.

ip neigh get

lookup a neighbour entry to a destination given a device

proxy indicates whether we should lookup a proxy neigbour entry

to *ADDRESS* **(default)**

the prefix selecting the neighbour to query.

dev *NAME*

get neighbour entry attached to this device.

EXAMPLES

ip neighbour

Shows the current neighbour table in kernel.

ip neigh flush dev eth0

Removes entries in the neighbour table on device eth0.

ip neigh get 10.0.1.10 dev eth0

Performs a neighbour lookup in the kernel and returns a neighbour entry.

SEE ALSO

ip(8)

AUTHOR

Original Manpage by Michail Litvak <mci@owl.openwall.com>