NAME

route - manually manipulate the routing tables

SYNOPSIS

route [**-j** *jail*] [**-dnqtv**] *command* [[*modifiers*] *args*]

DESCRIPTION

The **route** utility is used to manually manipulate the network routing tables. It normally is not needed, as a system routing table management daemon, such as routed(8), should tend to this task.

The **route** utility supports a limited number of general options, but a rich command language, enabling the user to specify any arbitrary request that could be delivered via the programmatic interface discussed in route(4).

The following options are available:

-4 Specify **inet** address family as family hint for subcommands.

-6 Specify **inet6** address family as family hint for subcommands.

-d Run in debug-only mode, i.e., do not actually modify the routing table.

-n Bypass attempts to print host and network names symbolically when reporting actions. (The process of translating between symbolic names and numerical equivalents can be quite time consuming, and may require correct operation of the network; thus it may be expedient to forget this, especially when attempting to repair networking operations).

-t Run in test-only mode. */dev/null* is used instead of a socket.

-v (verbose) Print additional details.

-q Suppress all output from the **add**, **change**, **delete**, and **flush** commands.

-j *jail* Run inside a jail.

The **route** utility provides the following commands:

add Add a route.
flush Remove all routes.
delete Delete a specific route.

del Another name for the **delete** command.
change Change aspects of a route (such as its gateway).
get Lookup and display the route for a destination.
monitor Continuously report any changes to the routing information base, routing lookup misses, or suspected network partitionings.
show Another name for the **get** command.

The monitor command has the syntax:

route [**-n**] **monitor** [**-fib** *number*]

The flush command has the syntax:

route [**-n**] **flush** [*family*] [**-fib** *number*]

If the **flush** command is specified, **route** will ''flush'' the routing tables of all gateway entries. When the address family may is specified by any of the **-inet6**, or **-inet** modifiers, only routes having destinations with addresses in the delineated family will be deleted. Additionally, **-4** or **-6** can be used as aliases for **-inet** and **-inet6** modifiers. When a **-fib** option is specified, the operation will be applied to the specified FIB (routing table).

The add command has the following syntax:

route [**-n**] **add** [**-net** | **-host**] *destination gateway* [*netmask*] [**-fib** *number*]

and the other commands have the following syntax:

route [**-n**] *command* [**-net** | **-host**] *destination* [*gateway* [*netmask*]] [**-fib** *number*]

where *destination* is the destination host or network, *gateway* is the next-hop intermediary via which packets should be routed. Routes to a particular host may be distinguished from those to a network by interpreting the Internet address specified as the *destination* argument. The optional modifiers **-net** and **-host** force the destination to be interpreted as a network or a host, respectively. Otherwise, if the *destination* has a "local address part" of INADDR_ANY (0.0.0.0), or if the *destination* is the symbolic name of a network, then the route is assumed to be to a network; otherwise, it is presumed to be a route to a host. Optionally, the *destination* could also be specified in the *net*/*bits* format.

For example, 128.32 is interpreted as **-host** 128.0.0.32; 128.32.130 is interpreted as **-host** 128.32.0.130; **-net** 128.32 is interpreted as 128.32.0.0; **-net** 128.32.130 is interpreted as 128.32.130.0; and 192.168.64/20 is interpreted as **-net** 192.168.64 **-netmask** 255.255.240.0.

A *destination* of *default* is a synonym for the default route. For IPv4 it is **-net -inet** 0.0.0.0, and for IPv6 it is **-net -inet6** ::.

If the destination is directly reachable via an interface requiring no intermediary system to act as a gateway, the **-interface** modifier should be specified; the gateway given is the address of this host on the common network, indicating the interface to be used for transmission. Alternately, if the interface is point to point the name of the interface itself may be given, in which case the route remains valid even if the local or remote addresses change.

The optional **-netmask** modifier is intended to achieve the effect of an OSI ESIS redirect with the netmask option, or to manually add subnet routes with netmasks different from that of the implied network interface (as would otherwise be communicated using the OSPF or ISIS routing protocols). One specifies an additional ensuing address parameter (to be interpreted as a network mask). The implicit network mask generated in the AF_INET case can be overridden by making sure this option follows the destination parameter.

For AF_INET6, the **-prefixlen** qualifier is available instead of the **-mask** qualifier because non-continuous masks are not allowed in IPv6. For example, **-prefixlen** 32 specifies that a network mask of ffff:ffff:0000:0000:0000:0000:0000:0000 will be used. The default prefixlen is 64. However, it is assumed to be 0 if **default** is specified for *destination*. Note that the qualifier works only for AF_INET6 address family.

Routes have associated flags which influence operation of the protocols when sending to destinations matched by the routes. These flags may be set (or sometimes cleared) by indicating the following corresponding modifiers:

-xresolve RTF_XRESOLVE - emit mesg on use (for external lookup)
-iface ~RTF_GATEWAY - destination is directly reachable
-static RTF_STATIC - manually added route
-nostatic ~RTF_STATIC - pretend route added by kernel or daemon
-reject RTF_REJECT - emit an ICMP unreachable when matched
-blackhole RTF_BLACKHOLE - silently discard pkts (during updates)
-proto1 RTF_PROTO1 - set protocol specific routing flag #1
-proto2 RTF_PROTO2 - set protocol specific routing flag #2

The optional modifiers **-rtt**, **-rttvar**, **-sendpipe**, **-recvpipe**, **-mtu**, **-hopcount**, **-expire**, and **-ssthresh** provide initial values to quantities maintained in the routing entry by transport level protocols, such as TCP or TP4. These may be individually locked by preceding each such modifier to be locked by the **-lock** meta-modifier, or one can specify that all ensuing metrics may be locked by the **-lockrest** meta-modifier.

Note that **-expire** accepts expiration time of the route as the number of seconds since the Epoch (see time(3)). When the first character of the number is "+" or "-", it is interpreted as a value relative to the current time.

The optional modifier **-fib** *number* specifies that the command will be applied to a non-default FIB. The *number* must be smaller than the *net.fibs* sysctl(8) MIB. When this modifier is not specified, or a negative number is specified, the default FIB shown in the *net.my_fibnum* sysctl(8) MIB will be used.

The *number* allows multiple FIBs by a comma-separeted list and/or range specification. The "**-fib** 2,4,6" means the FIB number 2, 4, and 6. The "**-fib** 1,3-5,6" means the 1, 3, 4, 5, and 6.

In a **change** or **add** command where the destination and gateway are not sufficient to specify the route (as in the ISO case where several interfaces may have the same address), the **-ifp** or **-ifa** modifiers may be used to determine the interface or interface address.

All symbolic names specified for a *destination* or *gateway* are looked up first as a host name using gethostbyname(3). If this lookup fails, getnetbyname(3) is then used to interpret the name as that of a network.

The **route** utility uses a routing socket and the new message types RTM_ADD, RTM_DELETE, RTM_GET, and RTM_CHANGE. As such, only the super-user may modify the routing tables.

FreeBSD provides support for scalable multipath routing. It is activated by default, but can be turned off by setting the *net.route.multipath* sysctl(8) MIB to 0.

There are multiple route lookup algorithms available. They can be configured by setting *net.route.algo.inet.algo* for IPv4 and *net.route.algo.inet6.algo* for IPv6 sysctl(8) MIBs.

A list of available algorithms can be obtained by accessing the following sysctl(8) MIBs *net.route.algo.inet.algo_list* for IPv4 and *net.route.algo.inet6.algo_list* for IPv6.

The following algorithms are available:

radix Base system radix backend.

bsearch Lockless binary search in a special IP array, tailored for a small FIB with <16 routes. This algorithm is only available for IPv4.

radix_lockless Lockless immutable radix, re-created on every rtable change, tailored for a small FIB with <1000 routes.

dpdk_lpm DPDK DIR24-8-based lookups, lockless datastructure, optimized for large FIBs. DIR24-8 relies on a large flat lookup table (64 MB with IPv4) which is directly indexed by the more significant portion of the lookup key. In order to use the dpdk_lpm algorithm one or both of the following kernel modules must be loaded via loader.conf(5):
dpdk_lpm4.ko DPDK implementation for IPv4.
dpdk_lpm6.ko DPDK implementation for IPv6.

dxr IPv4 only, lockless, compressed lookup structure (below 2.5 Bytes per IPv4 prefix for large BGP FIBs) which easily fits into modern CPU cache hierarchies, lookup throughput scales linearly with CPU cores. Loadable as a kernel module at runtime or via loader.conf(5):
fib_dxr.ko

The algorithms are selected automatically based on the size of the routing table of the system. They can be changed, but not every algorithm performs best for every FIB size.

EXIT STATUS

The **route** utility exits 0 on success, and >0 if an error occurs.

EXAMPLES

Add a default route to the network routing table. This will send all packets for destinations not available in the routing table to the default gateway at 192.168.1.1:

```
route add -net 0.0.0.0/0 192.168.1.1
```

A shorter version of adding a default route can also be written as:

```
route add default 192.168.1.1
```

Add a static route to the 172.16.10.0/24 network via the 172.16.1.1 gateway:

```
route add -net 172.16.10.0/24 172.16.1.1
```

Change the gateway of an already established static route in the routing table:

```
route change -net 172.16.10.0/24 172.16.1.2
```

Display the route for a destination network:

route show 172.16.10.0

Delete a static route from the routing table:

route delete -net 172.16.10.0/24 172.16.1.2

Remove all routes from the routing table:

route flush

The routing table can be listed with netstat(1).

DIAGNOSTICS

add [host | network] %s: gateway %s flags %x The specified route is being added to the tables. The values printed are from the routing table entry supplied in the ioctl(2) call. If the gateway address used was not the primary address of the gateway (the first one returned by gethostbyname(3)), the gateway address is printed numerically as well as symbolically.

delete [host | network] %s: gateway %s flags %x As above, but when deleting an entry.

%s %s done When the **flush** command is specified, each routing table entry deleted is indicated with a message of this form.

Network is unreachable An attempt to add a route failed because the gateway listed was not on a directly-connected network. The next-hop gateway must be given.

not in table A delete operation was attempted for an entry which was not present in the tables.

routing table overflow An add operation was attempted, but the system was low on resources and was unable to allocate memory to create the new entry.

gateway uses the same route A **change** operation resulted in a route whose gateway uses the same route as the one being changed. The next-hop gateway should be reachable through a different route.

SEE ALSO

netstat(1), netintro(4), route(4), loader.conf(5), arp(8), routed(8)

HISTORY

The **route** utility appeared in 4.2BSD.

BUGS

The first paragraph may have slightly exaggerated routed(8)'s abilities.

Currently, routes with the RTF_BLACKHOLE flag set need to have the gateway set to an instance of the lo(4) driver, using the **-iface** option, for the flag to have any effect; unless IP fast forwarding is enabled, in which case the meaning of the flag will always be honored.