NAME

route6d - RIP6 Routing Daemon

SYNOPSIS

route6d [**-adDhlnqsS**] [**-R** *routelog*] [**-A** *prefix/preflen,if1[,if2...]*] [**-L** *prefix/preflen,if1[,if2...]*] [**-N** *if1[,if2...]*] [**-O** *prefix/preflen,if1[,if2...]*] [**-P** *number*] [**-p** *pidfile*] [**-Q** *number*] [**-T** *if1[,if2...]*] [**-t** *tag*]

DESCRIPTION

The **route6d** utility is a routing daemon which supports RIP over IPv6.

Options are:

-a Enables aging of the statically defined routes. With this option, any statically defined routes will be removed unless corresponding updates arrive as if the routes are received at the startup of **route6d**.

-R *routelog*

This option makes the **route6d** to log the route change (add/delete) to the file *routelog*.

-A *prefix/preflen,if1[,if2...]*

This option is used for aggregating routes. *prefix/preflen* specifies the prefix and the prefix length of the aggregated route. When advertising routes, **route6d** filters specific routes covered by the aggregate, and advertises the aggregated route *prefix/preflen*, to the interfaces specified in the comma-separated interface list, *if1[,if2...]*. The characters "*", "?", and "[" in the interface list will be interpreted as shell-style pattern. The **route6d** utility creates a static route to *prefix/preflen* with RTF_REJECT flag, into the kernel routing table.

-d Enables output of debugging message. This option also instructs **route6d** to run in foreground mode (does not become daemon).

-D Enables extensive output of debugging message. This option also instructs **route6d** to run in foreground mode (does not become daemon).

-h Disables the split horizon processing.

-l By default, **route6d** will not exchange site local routes for safety reasons. This is because semantics of site local address space is rather vague (specification is still in being worked), and there is no good way to define site local boundary. With **-l** option, **route6d** will exchange site local routes as well. It must not be used on site boundary routers, since **-l** option assumes that all interfaces are in the same site.

-L *prefix/preflen,if1[,if2...]*

Filter incoming routes from interfaces *if1,[if2...]*. The **route6d** utility will accept incoming routes that are in *prefix/preflen*. If multiple **-L** options are specified, any routes that match one of the options is accepted. ::/0 is treated specially as default route, not "any route that has longer prefix length than, or equal to 0". If you would like to accept any route, specify no **-L** option. For example, with "**-L** 2001:db8::/16,if1 **-L** ::/0,if1" **route6d** will accept default route and routes in 6bone test address, but no others.

-n Do not update the kernel routing table.

-N *if1[,if2...]*

Do not listen to, or advertise, route from/to interfaces specified by *if1,[if2...]*.

-O *prefix/preflen,if1[,if2...]*

Restrict route advertisement toward interfaces specified by *if1,[if2...]*. With this option **route6d** will only advertise routes that matches *prefix/preflen*.

-P *number*

Specifies routes to be ignored in calculation of expiration timer. The *number* must be 1, 2, or 3 and it means route flags of RTF_PROTO1, RTF_PROTO2, or RTF_PROTO3. When 1 is specified, routes with RTF_PROTO1 will never expire.

-p *pidfile*

Specifies an alternative file in which to store the process ID. The default is */var/run/route6d.pid*.

-Q *number*

Specifies flag which will be used for routes added by RIP protocol. The default is 2 (RTF_PROTO2).

-q Makes **route6d** in listen-only mode. No advertisement is sent.

-s Makes **route6d** to advertise the statically defined routes which exist in the kernel routing table when **route6d** invoked. Announcements obey the regular split horizon rule.

-S This option is the same as **-s** option except that no split horizon rule does apply.

-T *if1[,if2...]*

Advertise only default route, toward *if1,[if2...]*.

-t *tag* Attach route tag *tag* to originated route entries. *tag* can be decimal, octal prefixed by 0, or hexadecimal prefixed by 0x.

Upon receipt of signal SIGINT or SIGUSR1, **route6d** will dump the current internal state into */var/run/route6d_dump*.

FILES

/var/run/route6d_dump dumps internal state on SIGINT or SIGUSR1

SEE ALSO

G. Malkin and R. Minnear, *RIPng for IPv6*, RFC2080, January 1997.

NOTE

The **route6d** utility uses IPv6 advanced API, defined in RFC2292, for communicating with peers using link-local addresses.

Internally **route6d** embeds interface identifier into bit 32 to 63 of link-local addresses (fe80::xx and ff02::xx) so they will be visible on internal state dump file (*/var/run/route6d_dump*).

Routing table manipulation differs from IPv6 implementation to implementation. Currently **route6d** obeys WIDE Hydrangea/KAME IPv6 kernel, and will not be able to run on other platforms.

Current **route6d** does not reduce the rate of the triggered updates when consecutive updates arrive.