NAME

inet6_rth_space, **inet6_rth_init**, **inet6_rth_add**, **inet6_rth_reverse**, **inet6_rth_segments**, **inet6_rth_getaddr** - IPv6 Routing Header Options manipulation

SYNOPSIS

#include <netinet/in.h>

socklen_t
inet6_rth_space(*int*, *int*);

*void **
inet6_rth_init(*void **, *socklen_t*, *int*, *int*);

int
inet6_rth_add(*void **, *const struct in6_addr **);

int
inet6_rth_reverse(*const void **, *void **);

int
inet6_rth_segments(*const void **);

*struct in6_addr **
inet6_rth_getaddr(*const void **, *int*);

DESCRIPTION

The IPv6 Advanced API, RFC 3542, defines the functions that an application calls to build and examine IPv6 Routing headers. Routing headers are used to perform source routing in IPv6 networks. The RFC uses the word "segments" to describe addresses and that is the term used here as well. All of the functions are defined in the *<netinet/in.h>* header file. The functions described in this manual page all operate on routing header structures which are defined in *<netinet/ip6.h>* but which should not need to be modified outside the use of this API. The size and shape of the route header structures may change, so using the APIs is a more portable, long term, solution.

The functions in the API are split into two groups, those that build a routing header and those that parse a received routing header. We will describe the builder functions followed by the parser functions.

inet6_rth_space

The **inet6_rth_space**() function returns the number of bytes required to hold a Routing Header of the type, specified in the *type* argument and containing the number of addresses specified in the *segments*

argument. When the type is IPV6_RTHDR_TYPE_0 the number of segments must be from 0 through 127. Routing headers of type IPV6_RTHDR_TYPE_2 contain only one segment, and are only used with Mobile IPv6. The return value from this function is the number of bytes required to store the routing header. If the value 0 is returned then either the route header type was not recognized or another error occurred.

inet6_rth_init

The **inet6_rth_init**() function initializes the pre-allocated buffer pointed to by *bp* to contain a routing header of the specified type. The *bp_len* argument is used to verify that the buffer is large enough. The caller must allocate the buffer pointed to by bp. The necessary buffer size should be determined by calling **inet6_rth_space**() described in the previous sections.

The **inet6_rth_init**() function returns a pointer to *bp* on success and NULL when there is an error.

inet6_rth_add

The **inet6_rth_add**() function adds the IPv6 address pointed to by *addr* to the end of the routing header being constructed.

A successful addition results in the function returning 0, otherwise -1 is returned.

inet6_rth_reverse

The **inet6_rth_reverse**() function takes a routing header, pointed to by the argument *in*, and writes a new routing header into the argument pointed to by *out*. The routing header at that sends datagrams along the reverse of that route. Both arguments are allowed to point to the same buffer meaning that the reversal can occur in place.

The return value of the function is 0 on success, or -1 when there is an error.

The next set of functions operate on a routing header that the application wants to parse. In the usual case such a routing header is received from the network, although these functions can also be used with routing headers that the application itself created.

inet6_rth_segments

The **inet6_rth_segments**() function returns the number of segments contained in the routing header pointed to by *bp*. The return value is the number of segments contained in the routing header, or -1 if an error occurred. It is not an error for 0 to be returned as a routing header may contain 0 segments.

inet6_rth_getaddr

The **inet6_rth_getaddr**() function is used to retrieve a single address from a routing header. The *index* is the location in the routing header from which the application wants to retrieve an address. The *index*

parameter must have a value between 0 and one less than the number of segments present in the routing header. The **inet6_rth_segments**() function, described in the last section, should be used to determine the total number of segments in the routing header. The **inet6_rth_getaddr**() function returns a pointer to an IPv6 address on success or NULL when an error has occurred.

EXAMPLES

RFC 3542 gives extensive examples in Section 21, Appendix B.

KAME also provides examples in the advapitest directory of its kit.

DIAGNOSTICS

The **inet6_rth_space**() and **inet6_rth_getaddr**() functions return 0 on errors.

The **inet6_rthdr_init**() function returns NULL on error. The **inet6_rth_add**() and **inet6_rth_reverse**() functions return 0 on success, or -1 upon an error.

SEE ALSO

W. Stevens, M. Thomas, E. Nordmark, and T. Jinmei, *Advanced Sockets API for IPv6*, RFC 3542, May 2003.

S. Deering and R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, RFC2460, December 1998.

HISTORY

The implementation first appeared in KAME advanced networking kit.