

IPv6 multicast sendto() 'operation not supported'

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/net/2004-02/0314.html>

From: Donald McLachlan (*don_at_mainframe.dgrc.crc.ca*)

Date: 02/26/04

Date: Thu, 26 Feb 2004 11:27:31 -0500 (EST)

To: freebsd-net@freebsd.org

Hello,

[I'm new to freebsd. This seems like the most related mailing list.
If there is a better mailing list I should post to, please point me
in the right direction.]

In preparation for writing an IPv6 multicast application I wrote a little
test program (shown below). This program worked on linux (RedHat), but
when I try it on a FreeBSD box (5.0, running zebra router and pim6dd)
sendto() fails with "Operation not supported" ... ???

To verify my app was OK, I installed a solaris box on the LAN beside the
linux box and the app compiled and worked fine.

Thinking it might be a bug in 5.0 or an interaction with zebra/pim6dd I
installed a FreeBSD 5.2 box on the lan beside the other 2 app boxes and
I get the same error again. Anyone know what is missing/wrong in my test app?

[I'm guessing FreeBSD wants some extra socket options set, but I don't know
which ones.]

Thanks,
Don

```
/*
```

```
    mcast6.c
```

```
*/
```

```
#include <stdio.h>  
#include <sys/types.h>  
#include <sys/socket.h>  
#include <sys/time.h>  
#include <netinet/in.h>  
#include <arpa/inet.h>  
#include <netdb.h>  
#include <string.h>
```

freebsd-net: IPv6 multicast sendto() 'operation not supported'

```
#include <strings.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>
#include <net/if.h>

extern int errno;

void start_sender(int sock, struct sockaddr_in6 *sin);
void start_listener(int sock);

int main(int argc, char *argv[])
{
    int sock, hops;
    unsigned int ifindex;
    struct sockaddr_in6 sin;
    struct ipv6_mreq mreq;
    char ifname[IF_NAMESIZE];

    static unsigned char the_addr[16];

    inet_pton(AF_INET6, "ff05::abcd", the_addr);

    if ((sock=socket(AF_INET6, SOCK_DGRAM, 0)) < 0) /* connect to socket */
    {
        perror("socket");
        exit(3);
    }

    bzero((char *)&sin, (int)sizeof(sin)); /* setup address info */
    bcopy(the_addr, (char *)&sin.sin6_addr, sizeof(the_addr));
    sin.sin6_family = AF_INET6;
    sin.sin6_port = htons(3000);
    /* bind addr to sock */
    if(bind(sock, (struct sockaddr *)&sin, sizeof(sin)) < 0)
    {
        perror("bind");
        exit(errno);
    }

    hops = 255;
    if(setsockopt(sock, IPPROTO_IPV6, IPV6_MULTICAST_HOPS, &hops, sizeof(hops)) < 0)
    {
        perror("setsockopt(IPV6_MULTICAST_HOPS)");
        exit(errno);
    }

    ifindex = 0;
#ifdef ORIG
    strcpy(ifname, "xl1");
    ifindex = if_nametoindex(ifname);
#endif
}
```

IPv6 multicast sendto() 'operation not supported'

freebsd-net: IPv6 multicast sendto() 'operation not supported'

```
if(ifindex == 0)
{
    perror("if_nametoindex()");
    exit(errno);
}

if(setsockopt(sock, IPPROTO_IPV6, IPV6_MULTICAST_IF, &ifindex, sizeof(ifindex)) < 0)
{
    perror("setsockopt(IPV6_MULTICAST_IF)");
    exit(errno);
}
#endif

/* setup group info */
bcopy(the_addr, (char *)&mreq.ipv6mr_multiaddr, sizeof(the_addr));
mreq.ipv6mr_interface = ifindex; /* and I/F index */

/* join group */
if(setsockopt(sock, IPPROTO_IPV6, IPV6_JOIN_GROUP, (char *)&mreq, sizeof(mreq)) < 0)
{
    perror("setsockopt(IPV6_JOIN_GROUP)");
    exit(21);
}

start_sender(sock, &sin);
start_listener(sock);

while(wait((int *)0) != -1); /* wait for all children to die */

return(0);
}

void start_sender(int sock, struct sockaddr_in6 *sin)
{
    char bitbucket[2000];
    pid_t pid;

    pid = fork();
    switch(pid)
    {
        case 0: /* child */
            while(1)
            { /* TX mcast PDU */
                strcpy(bitbucket, "How now brown cow?");

                if(sendto(sock, bitbucket, strlen(bitbucket)+1, 0, (struct sockaddr *)sin, sizeof(*sin)) < 0)
                {
                    perror("sendto(sender)");
                    exit(23);
                }
                sleep(10);
            }
        }
    }
}
```

IPv6 multicast sendto() 'operation not supported'

freebsd-net: IPv6 multicast sendto() 'operation not supported'

```
break;

case -1: /* parent fork failed */
    printf("fork(sender): failed\n");
    fflush(stdout);
    break;

default: /* parent fork passed */
    printf("sender PID = %d\n", (int)pid);
    fflush(stdout);
    break;
}
}

void start_listener(int sock)
{
    pid_t pid;
    unsigned int fromlen;
    struct sockaddr_in6 from;
    char bitbucket[2000], addrstr[INET6_ADDRSTRLEN];

    pid = fork();
    switch(pid)
    {
        case 0: /* child */
            while(1)
            {
                bitbucket[0] = '\0';
                fromlen = sizeof(from);
                if(recvfrom(sock, bitbucket, sizeof(bitbucket),
                    0, (struct sockaddr *)&from,
                    &fromlen) < 0)
                {
                    perror("listener: recvfrom()");
                    exit(21);
                }

                inet_ntop(AF_INET6, &from.sin6_addr, addrstr,
                    sizeof(addrstr));
                printf("< %s : \"%s\"\n", addrstr, bitbucket);
                fflush(stdout);
            }
            break;

        case -1: /* parent fork failed */
            printf("fork(listener): failed\n");
            fflush(stdout);
            break;

        default: /* parent fork passed */
            printf("listener PID = %d\n", (int)pid);
    }
}
```

IPv6 multicast sendto() 'operation not supported'

freebsd-net: IPv6 multicast sendto() 'operation not supported'

```
    fflush(stdout);  
    break;  
}  
}
```

freebsd-net@freebsd.org mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-net>

To unsubscribe, send any mail to "freebsd-net-unsubscribe@freebsd.org"