Forum: CentOS 6 - Software Support

Topic: Getting Multicast to work on a VLAN interface

Subject: Getting Multicast to work on a VLAN interface

Posted by: Jiro

Posted on: 2012/7/9 12:46:27

Hello I am trying to get IP Multicast to work on a VLAN interface.

My code looks like this:

```
 int getInterfaceInfo(const char *ifc, uint32_t *ipaddr, uint8_t *mac) { # define MAX_IFC_SETS 16
printf("get InterfaceInfo ifc: <%s>, len: %in", ifc,strlen(ifc));    struct ifconf ifr_conf;   struct ifreq
set[MAX_IFC_SETS];   int s;   unsigned i;   struct sockaddr_in *addr;    if ((s = socket(AF_INET,
SOCK_DGRAM, 0)) < 0){     printf("Error Socket!n");     return -1;   }    ifr_conf.ifc_len = sizeof(struct
ifreq) * MAX_IFC_SETS;   ifr_conf.ifc_req = set;    if (ioctl(s, SIOCGIFCONF, & ifr_conf) < 0) {
printf("Error SIOCGICONF!n");     return -1;   }    for (i = 0; i < ifr_conf.ifc_len / sizeof(struct ifreq); i++){
   if ((!ifc[0] && strncmp(ifr_conf.ifc_req[i].ifr_name, "eth0", IFNAMSIZ) == 0) ||          (ifc[0] &&
strncmp(ifr_conf.ifc_req[i].ifr_name, ifc, IFNAMSIZ) == 0)) {        addr = (struct sockaddr_in *) &
ifr_conf.ifc_req[i].ifr_addr;        *ipaddr =  addr->sin_addr.s_addr;        // return interface index
printf("Our Interface: index: %i, name: %s, ip:
%sn",i,ifr_conf.ifc_req[i].ifr_name,inet_ntoa(addr->sin_addr));         // get also mac        if (ioctl(s,
SIOCGIFHWADDR, &ifr_conf.ifc_req[i])  < 0)          return -1;        memcpy(mac,
ifr_conf.ifc_req[i].ifr_hwaddr.sa_data, 6);        close(s);        return i;     }   }   close(s);    return -1; }
MCPort::MCPort(const PortAddress_t mcaddr, const uint16_t port, const char *laddr) {    struct
sockaddr_in ssock;   struct ip_mreq mreq;   int one = 1;   int zero = 0;   int ttl = 64;    m_port = port;
uint8_t mac[6];   // initialize    memset(&ssock, 0, sizeof(ssock));   ssock.sin_family = AF_INET;
ssock.sin_addr.s_addr = INADDR_ANY;    // create datagram socket   if ((m_sock = socket(PF_INET,
SOCK_DGRAM, 0)) < 0)     return;      setsockopt(m_sock, SOL_SOCKET, SO_REUSEADDR, (char
*)&one, sizeof(one));    // bind to port   ssock.sin_port = htons(port);   if (bind(m_sock, (struct
sockaddr *)&ssock,            sizeof(struct sockaddr_in)) < 0){     m_sock = -1;     return;   }    // to be
able to receive multicast packets
memcpy(&mreq.imr_multiaddr.s_addr,&mcaddr.addr,sizeof(mreq.imr_multiaddr.s_addr));   uint32_t
ipaddr;   int index = getInterfaceInfo(laddr, &ipaddr, mac);   mreq.imr_interface.s_addr = ipaddr;    if
(setsockopt(m_sock, IPPROTO_IP, IP_ADD_MEMBERSHIP, (char *)&mreq, sizeof(mreq)) < 0) {
m_sock = -3;     return;   }   // prevent from receiving own multicast packet   if (setsockopt(m_sock,
IPPROTO_IP, IP_MULTICAST_LOOP, (char *)&zero, sizeof(zero)) < 0) {     m_sock = -4;     return;
}    // to be able to send multicast packets from the right interface   struct in_addr localInterface;
localInterface.s_addr = ipaddr;   printf("sockopt ip: %sn",inet_ntoa(localInterface));   if
(setsockopt(m_sock, IPPROTO_IP, IP_MULTICAST_IF, (char*)&localInterface,
sizeof(localInterface)) < 0)   {      printf("Error %i: %sn",errno,strerror(errno));      m_sock = -6;
return;   } }
```

Network configuration is as follows:

```
 [root@localhost ~]# ifconfig eth0      Link encap:Ethernet  HWaddr 08:00:27:78:EC:49          inet
addr:192.168.102.166  Bcast:192.168.103.255  Mask:255.255.254.0          inet6 addr:
```

fe80::a00:27ff:fe78:ec49/64 Scope:Link UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:827107 errors:0 dropped:0 overruns:0 frame:0 TX packets:678985 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:419623378 (400.1 MiB) TX bytes:547112945 (521.7 MiB) eth0.57 Link encap:Ethernet HWaddr 08:00:27:78:EC:49 inet addr:11.0.0.66 Bcast:11.255.255.255 Mask:255.0.0.0 inet6 addr: fe80::a00:27ff:fe78:ec49/64 Scope:Link UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:0 errors:0 dropped:0 overruns:0 frame:0 TX packets:33 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:0 RX bytes:0 (0.0 b) TX bytes:4177 (4.0 KiB) lo Link encap:Local Loopback inet addr:127.0.0.1 Mask:255.0.0.0 inet6 addr: ::1/128 Scope:Host UP LOOPBACK RUNNING MTU:16436 Metric:1 RX packets:0 errors:0 dropped:0 overruns:0 frame:0 TX packets:0 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:0 RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)

Kernel Info:
[root@localhost ~]# uname -a Linux LiveLinVM 2.6.32-220.el6.i686 #1 SMP Tue Dec 6 16:15:40 GMT 2011 i686 i686 i386 GNU/Linux

My code is called with the following parameters:

mcaddr is "xE0x00x00x0a"
port 200
laddr eth0 or eth0.57

What happens is this:
When laddr is "eth0" all is fine and working as expected. I can receive UDP-Pakets sent to the Multicast-Address and sent UDP-Pakets go out on the correct interface

When laddr is "eth0.57" the "setsockopt IP_MULTICAST_IF" call fails with EINVAL.

Can anybody point me in the right direction where my mistake is?

BR
Jiro