**linux操作系统的组播设置**

2009年10月25日 08:40:00 [brookson](https://me.csdn.net/brookson) 阅读数：1786

我的安装linux操作系统的工控机无法进行组播，提示socket error 101  
network is unreached。我在工控机输入route -n，没有显示组播的信息，于是我就加入route add -net 224.0.0.0 netmask 240.0.0.0 dev eth1(eth1要根据设备而定)。问题解决。

**linux下route路由设置命令详解**

2017年05月22日 09:38:43 [不才b\_d](https://me.csdn.net/sinat_36184075) 阅读数：9263 标签： [路由](http://so.csdn.net/so/search/s.do?q=%E8%B7%AF%E7%94%B1&t=blog)[网络](http://so.csdn.net/so/search/s.do?q=%E7%BD%91%E7%BB%9C&t=blog) 更多

个人分类： [Linux/Unix](https://blog.csdn.net/sinat_36184075/article/category/6681176)

版权声明：本文为博主不才b\_d原创文章，未经允许不得转载。 || 博客地址：blog.csdn.net/sinat\_36184075 https://blog.csdn.net/sinat\_36184075/article/details/72621289

# route -n                                                  // 显示路由表,哪条在前就用哪条,都没有就用default

# route add -net 224.0.0.0 netmask 240.0.0.0 dev eth0       // 向路由表中增加1条路由

# route add -net 224.0.0.0 netmask 240.0.0.0 reject         // 屏蔽1条路由

# route del -net 224.0.0.0 netmask 240.0.0.0                // 删除1条路由

# route del default gw 192.168.120.240                      // 删除和添加设置默认网关

# route [add|del] [-net|-host] target [netmask Nm] [gw Gw] [[dev] If]

其中：

- add : 添加一条路由规则

- del : 删除一条路由规则

- net : 目的地址是一个网络

- host : 目的地址是一个主机

- target : 目的网络或主机

- netmask : 目的地址的网络掩码

- gw : 路由数据包通过的网关

- dev : 为路由指定的网络接口

添加到主机的路由

# route add -host 192.168.1.2 dev eth0:0

# route add -host 10.20.30.148 gw 10.20.30.40

添加到网络的路由

# route add -net 10.20.30.40 netmask 255.255.255.248 eth0

# route add -net 10.20.30.48 netmask 255.255.255.248 gw 10.20.30.41

# route add -net 192.168.1.0/24 eth1

添加默认路由

# route add default gw 192.168.1.1

删除路由

# route del -host 192.168.1.2 dev eth0:0

# route del -host 10.20.30.148 gw 10.20.30.40

# route del -net 10.20.30.40 netmask 255.255.255.248 eth0

# route del -net 10.20.30.48 netmask 255.255.255.248 gw 10.20.30.41

# route del -net 192.168.1.0/24 eth1

# route del default gw 192.168.1.1

设置包转发

开启 Linux 的路由功能可以通过调整内核的网络参数来实现。要配置和调整内核参数可以使用 sysctl 命令。例如：要开启 Linux 内核的数据包转发功能可以使用如下的命令。

# sysctl -w net.ipv4.ip\_forward=1

这样设置之后，当前系统就能实现包转发，但下次启动计算机时将失效。为了使在下次启动计算机时仍然有效，需要将下面的行写入配置文件/etc/sysctl.conf。

# vi /etc/sysctl.conf

net.ipv4.ip\_forward = 1

用户还可以使用如下的命令查看当前系统是否支持包转发。

# sysctl net.ipv4.ip\_forward

<http://blog.csdn.net/eastonwoo/article/details/16358723>

> ipconfig /all    // 查看windows下网卡信息，linux下是 ifconfig

> route print      // 查看路右边信息

路由表信息解释

  1）名词解释：

Active Routes：活动的路由

Network destination ：目的网段

Netmask：子网掩码

Gateway：网关，又称下一跳路由器。在发送IP数据包时，网关定义了针对特定的网络目的地址，数据包发送到的下一跳服务器。如果是本地计算机直接连接到的网络，网关通常是本地计算机对应的网络接口，但是此时接口必须和网关一致；如果是远程网络或默认路由，网关通常是本地计算机所连接到的网络上的某个服务器或路由器。

Interface：接口，接口定义了针对特定的网络目的地址，本地计算机用于发送数据包的网络接口。网关必须位于和接口相同的子网（默认网关除外），否则造成在使用此路由项时需调用其他路由项，从而可能会导致路由死锁。

Metric：跳数，跳数用于指出路由的成本，通常情况下代表到达目标地址所需要经过的跳跃数量，一个跳数代表经过一个路由器。跳数越低，代表路由成本越低，优先级越高。

Persistent Routes：手动配置的静态固化路由

  2）第一条路由信息：**缺省路由**

  当系统接收到一个目的地址不在路由表中的数据包时，系统会将该数据包通过192.168.99.8(PC-ip)这个接口发送到缺省网关192.168.99.1(PC-网关)。

  3）第二条路由信息：**本地环路**

  当系统接收到一个发往目标网段127.0.0.0的数据包时，系统将接收发送给该网段的所有数据包。

  4）第三条路由信息：**直连网段的路由记录**

  当系统接收到一个发往目的网段192.168.99.0/24的数据包时，系统会将该数据包通过192.168.99.8这个接口发送出去。

  5）第四条路由信息：**本地主机路由**

  当系统接收到一个目标ip地址为本地网卡ip地址的数据包时，系统会将该数据包收下。

  6）第五条路由信息：**本地广播路由**

  当系统接收到一个发给直连网段的本地广播数据包时，系统会将该数据包从192.168.99.8这个接口以广播的形式发送出去。

  7）第六条路由信息：**组播路由**

  当系统接收到一个组播数据包时，系统会将该数据包从192.168.99.8这个接口以组播的形式发送出去。

  8）第七条路由信息：**广播路由**

  在系统接收到一个绝对广播数据包时，系统会将该数据包通过192.168.99.8这个接口发送出去。

  9）Default Gateway（**缺省网关**）

二、windows路由表操作

windows路由表的设置主要通过route命令，ROUTE命令格式如下：

ROUTE [-f] [-p] [command [destination] [MASK netmask] [gateway] [METRIC metric] [IF interface]

其中 **–f** 参数用于清除路由表，**-p** 参数用于永久保留某条路由（即在系统重启时不会丢失路由）。

Command主要有**PRINT**（打印）、**ADD**（添加）、**DELETE**（删除）、**CHANGE**（修改）共4个命令。

Destination代表所要达到的目标IP地址。

MASK是子网掩码的关键字。Netmask代表具体的子网掩码，如果不加说明，默认是255.255.255.255（单机IP地址）。如果代表全部出口子网掩码可用0.0.0.0。

Gateway代表出口网关。

其他interface和metric分别代表特殊路由的接口数目和到达目标地址的跳数，一般默认。

【路由器route设定】————鸟哥私房菜

ip分享器

[root@linux ~]# ifconfig [device] [ IP ] netmask [netmask ip] [up|down]

[root@linux ~]# ifconfig eth0:0 192.168.0.100 netmask 255.255.255.0 up

来建立一个虚拟的网络接口，这样就可以立刻连接上 IP 分享器了。

如何在开机的时候就启动 IP alias 呢？有两个简单的方法可以使用：

• 透过 /etc/rc.d/rc.local：

将『ifconfig eth0:0 192.168.0.100 netmask 255.255.255.0 up』的指令写入 /etc/rc.d/rc.local 当中，这样开机的时候就能够启动这个虚拟接口， 不过这方法有个弱点，就是当使用类似『 /etc/init.d/network restart 』的指令时，该接口可能就会被取消。

• 透过 /etc/sysconfig/network-scripts/ifcfg-eth0:0：

举例来说，你可以透过底下这个方法来建立一个虚拟装置的设定档案：

[root@linux ~]# cd /etc/sysconfig/network-scripts

[root@linux network-scripts]# vi ifcfg-eth0:0

DEVICE=eth0:0 <==相当重要！一定要与文件名相同的装置代号！

ONBOOT=yes

BOOTPROTO=static

IPADDR=192.168.0.100

NETMASK=255.255.255.0

NETWORK=192.168.0.0

BROADCAST=192.168.0.255

[root@linux network-scripts]# ifup eth0:0

[root@linux network-scripts]# ifdown eth0:0

路由器的主要功能就是： 『转发网络报文』 也就是说，路由器会分析来源端封包的 IP 表头，找出目标的 IP 后，透过路由器本身的路由表 (routing table) 将这个封包向下一个目标 (next hop) 传送。这就是路由器的功能。

路由表是由 Linux 的核心功能所提供的，这个转递封包的能力也是 Linux 核心所提供， 那如何启动这个封包转递呢？很简单啊，只要这样做即可：

[root@linux ~]# echo "1" > /proc/sys/net/ipv4/ip\_forward

上面这个动作就在打开 Linux 核心的封包转递能力。你可以将上述的指令写入 /etc/rc.d/rc.local 当中， 以使 Linux 可以在开机的时候就启动封包转递的功能， 也可以透过修改 /etc/sysctl.conf 来达成开机启动封包转递：

[root@linux ~]# vi /etc/sysctl.conf

# 将底下这个设定值修改正确即可！

net.ipv4.ip\_forward = 1

[root@linux ~]# sysctl -p <==立刻让该设定生效

sysctl 这个指令是在核心工作时用来直接修改核心参数的一个指令，更多的功能可以参考 man sysctl 查询。 不要怀疑！只要这个动作，你的 Linux 就具有最简单的路由器功能了。

NAT (Network Address Translation, 网络地址转译)。Linux 的 NAT 主机可以透过修改封包的 IP 表头数据之来源或目标 IP ，让来自私有 IP 的封包可以转成 NAT 主机的公共 IP ，就可以连上 Internet ！

Linux Router配置方法：

PC1[192.168.0.20] ————Linux Router———— PC2[192.168.10.30]

首先linux Router需要有两张网卡。

1. 先处理 eth0

[root@linux ~]# vi /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0

BOOTPROTO=static

BROADCAST=192.168.10.255

IPADDR=192.168.10.254

NETMASK=255.255.255.0

NETWORK=192.168.10.0

ONBOOT=yes

2. 再先处理 eth1

[root@linux ~]# vi /etc/sysconfig/network-scripts/ifcfg-eth1

DEVICE=eth1

BOOTPROTO=static

BROADCAST=192.168.0.255

IPADDR=192.168.0.100

NETMASK=255.255.255.0

NETWORK=192.168.0.0

GATEWAY=192.168.0.254 <==这个设定值很重要喔！

ONBOOT=yes

3. 启动 IP 转递

[root@linux ~]# echo "1" > /proc/sys/net/ipv4/ip\_forward

# 上述指令如果没有问题，将他加入 /etc/rc.d/rc.local 当中去！

4. 重新启动网络，并且观察路由

[root@linux ~]# /etc/init.d/network restart

[root@linux ~]# route -n

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

192.168.0.0 0.0.0.0 255.255.255.0 U 0 0 0 eth1

192.168.10.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0

127.0.0.0 0.0.0.0 255.0.0.0 U 0 0 0 lo

0.0.0.0 192.168.0.254 0.0.0.0 UG 0 0 0 eth1

• 请问您如何将您的 eth0 这个接口修改成为 192.168.100.2 在网域 192.168.100.0/25 之内的网络参数内容？

因为 192.168.100.0/25 的 netmask 为 255.255.255.128 ，所以可以这样做：

ifconfig eth0 192.168.100.2 netmask 255.255.255.128 up

这样即可！如果尚须其它的参数，则需要以档案形式来下达，如 vi /etc/sysconfig/network-scripts/ifcfg-eth0，并修改为：

DEVICE=eth0

ONBOOT=yes

BOOTPROTO=static

IPADDR=192.168.100.2

NETMASK=255.255.255.128

NETWORK=192.168.100.0

BROADCAST=192.168.100.127

• 请手动设定 eth0:1 这个虚拟接口，使成为网络参数： 192.168.200.2, 网域在 192.168.200.0/24。

  ifconfig eth0:1 192.168.200.2 up

• 如何观察路由表？

  route -n 即可查阅！注意到 0.0.0.0 那个目标(default gateway)。

• 如何启动 Linux 的 IP Forward 功能？

  直接以『echo "1" > /proc/sys/net/ipv4/ip\_forward 』即可！

• 假设你想要连接到 168.95.1.1 ，那么你该如何判断你经过『多少个』节点？

  可以使用 traceroute 168.95.1.1 来分析每个节点的传送信息，也可以透过 ping 168.95.1.1 所回传的那个 ttl 值判断节点数量。

• 万一您的网络有点停顿，发现可能是网络上某个节点出现问题，您应该如何确认是哪一部 Router出问题？

  就利用 traceroute 吧！