



网络虚拟化系列文章



理解物理网卡、网卡接口、内核、IP等属性的关系

我想很多人都对它们产生过迷惑，物理网卡明明是一块小板子，为什么和eth0(或Windows下的本地连接)是对应的呢，为什么还可以给物理网卡配置IP地址呢，为什么还有那么多属性呢，等等谜之问题。

一个类比

我先以人来做一個类比，我想这样应该会有助于理解，具体细节留待后文解释。

人对应物理网卡，人的姓名对应网卡接口名称，人的档案信息(保存在公安机关)对应网卡接口的配置信息，公安机关对应内核。

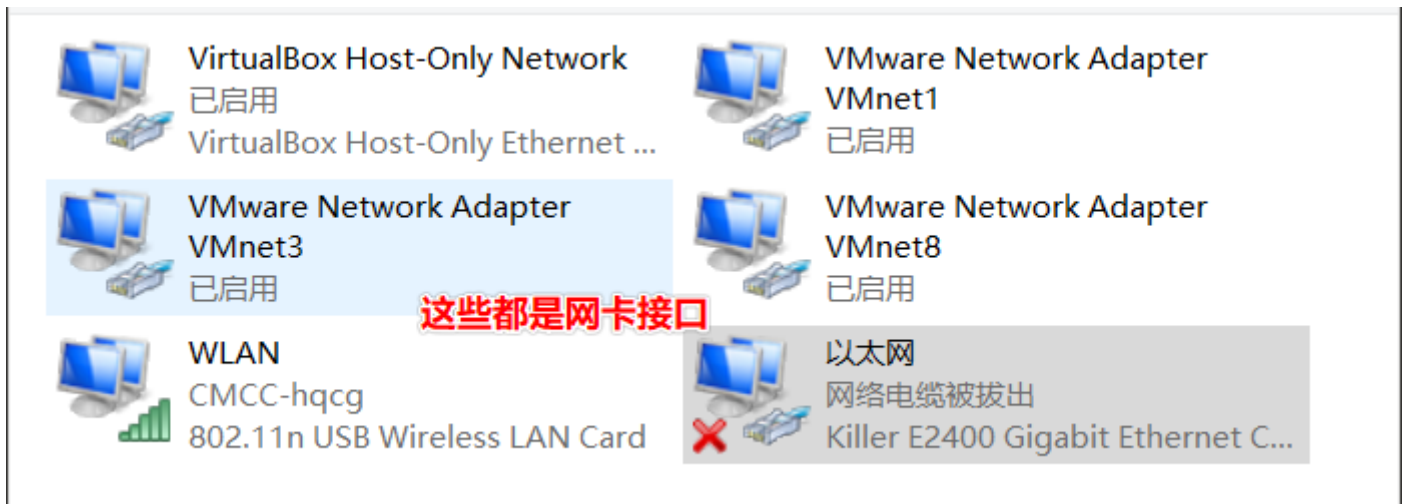
你的朋友知道这个名字对应的就是你，但是对于派出所或某些机构来说，不能依赖你的名字来识别你，因为人可能会改名，也可能会和别人重名，所以人都有一个身份证号码。派出所或某些机构办事的时候，会根据你的档案信息做出对应处理，比如户籍是杭州，某些事就得交给杭州办。

网卡也一样，用户可以通过网卡接口名称识别网卡，但网卡接口名称是不可靠的，接口名称只是显示给用户看的，对于内核来说，内核是通过为该网卡接口分配的UUID属性来识别网卡的。内核和网卡交互时，内核需要根据网卡接口的配置信息做出决策。

网卡和网卡接口

物理网卡是一个硬件设备，硬件设备要在操作系统上工作，需要有对应的驱动程序。物理网卡的驱动程序向操作系统内核注册该网卡设备，从而让内核识别该物理网卡。

网卡注册成功后，用户就可以看到该物理网卡对应的网卡接口(如eth0、本地连接等名称)，再经过一番配置，就可以启动该网卡设备使其工作。



但是需要注意，网卡注册成功后，并非一定会显示对应的接口名称，这取决于设备管理程序的行为。换句话说，即使看不到网卡接口，但可能它对应的网卡设备已经可以工作。

🔗 网卡接口配置属性

另外，还可以对网卡做一些配置，比如配置IP地址、配置DNS、配置是否使用DHCP动态获取IP地址等等。

```
Bash
1 $ cat /etc/sysconfig/network-scripts/ifcfg-ens160
2 TYPE=Ethernet
3 PROXY_METHOD=none
4 BROWSER_ONLY=no
5 BOOTPROTO=dhcp
6 DEFROUTE=yes
7 IPV4_FAILURE_FATAL=no
8 IPV6INIT=yes
9 IPV6_AUTOCONF=yes
10 IPV6_DEFROUTE=yes
11 IPV6_FAILURE_FATAL=no
12 IPV6_ADDR_GEN_MODE=stable-privacy
13 NAME=ens160
14 UUID=97c84984-8b8e-4686-af19-1cb2f5f89db6
15 DEVICE=ens160
16 ONBOOT=yes
```

你可能从一开始就认为这些属性都是配置在网卡上的，之所以会这样认为大概是因为这些属性都是直接控制对应网卡的。

但实际上，**这些属性都配置在内核中**，严格来说一部分信息保存在内核的网络协议栈中。更严格地说，**这些属性都应该称之为网卡接口的属性，而不是网卡属性**，所谓的网卡属性是那些网卡出厂设置的硬件属性，比如MAC地址。网卡接口属性一部分来自于网卡属性，这些硬件属性有些甚至可以修改（比如修改MAC地址），只要内核能识别它们即可。

所以，**IP地址、网关等网卡配置信息，都属于内核**。下图中所有的信息都属于内核。



当为某网卡接口配置属性后，在内核的网络协议栈中，该网卡就有了对应的配置属性（档案信息），以后内核需要和该网卡交互时，就可以基于这些属性来工作。

比如，为网卡接口eth0配置了开机自启动，那么在系统启动的时候，内核就会自动启动该网卡设备。

比如，为eth0配置了IP地址10.10.10.10后，那么在内核协议栈中，该网卡接口就对应了该IP地址（注意，IP等网卡接口属性属于内核）。当本机收到目标地址是10.10.10.10的数据包后，内核知道该地址是为本机某个网卡接口所配置的，于是内核收下该数据包，而不会丢弃或转发。而且，还可以为同一个网络接口配置多个IP地址（因为它们属于内核），内核能辨别即可。

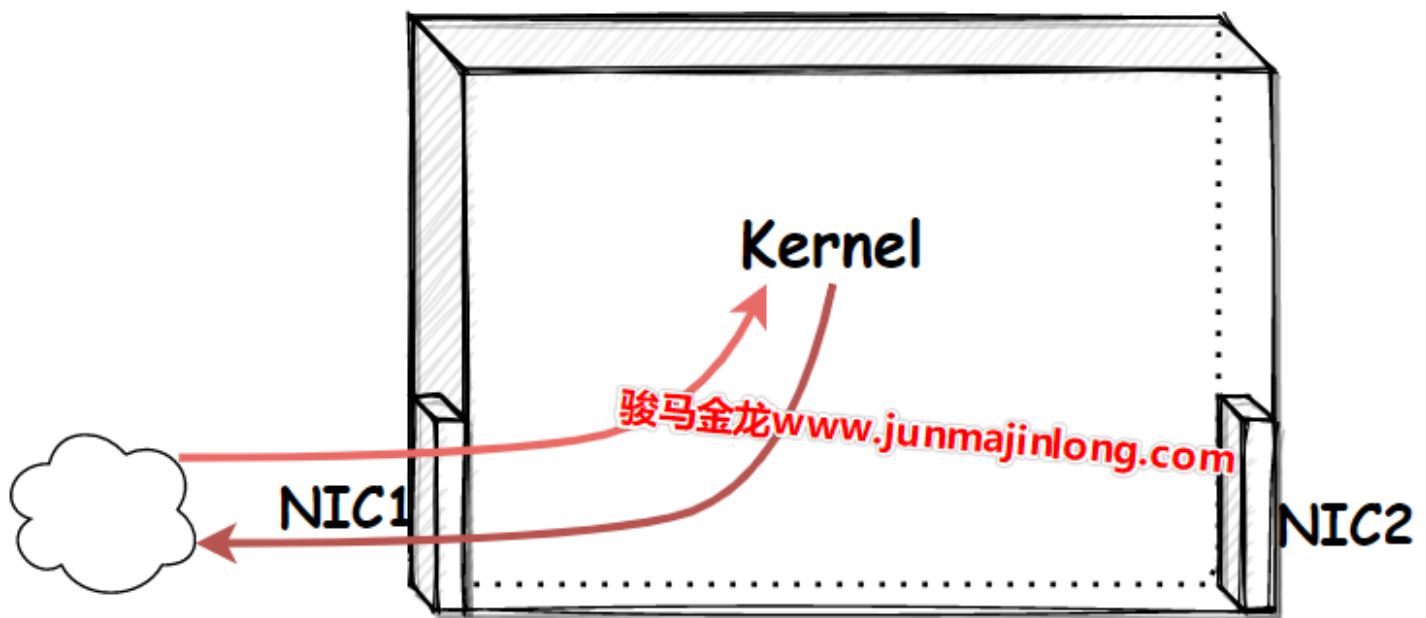
比如，为eth0配置了子网掩码后，基于该子网掩码，内核可以知道某个IP地址是否和该网卡接口在同一个网段中。

总之，**这些接口配置信息都是为内核配置的，内核会基于这些配置信息做出处理。**

🔗 内核和网卡的关系

网卡工作在OSI七层模型的最底层（物理层）。它最基本的功能是以比特流的方式接收和发送数据（可以通过介质传输），这是硬件自身的功能。

仅就内核和网卡的关系来考虑，内核就像一个封闭的加工厂，网卡就像加工厂的大门。所以，对于内核来说，网卡的作用就是接收外界流入的数据和发送内部流出的数据。



正如上图，每个物理网卡都像门一样，都连接着两端：**一端是内核（网络协议栈），另一端是外界网络。**

当本机要通过该网卡向外发送数据时，数据从内核协议栈写入到该网卡，然后网卡发送出去。

当该网卡接收到外界传来的数据时，网卡需要中断通知内核有数据来临(因为只有内核才具有读写硬件数据的特权)，数据将从网卡读取到内核协议栈。这就像是带有门卫系统的门一样，有人来了先报告一声。

同时，网卡是一个带有芯片的硬件设备，带有芯片意味着除了硬件自身的基础功能外，还可以通过一些代码(网卡驱动程序)来控制网卡的工作，比如中断通知内核，比如DMA，等。

如果网卡没有DMA功能，网络协议栈和网卡之间的数据传输都将由内核完成读或写。网卡具备DMA功能时，网卡和网络协议栈之间的数据传输主要由网卡DMA完成，DMA传输完数据后，DMA控制器中断通知内核，表示数据已写入目标内存地址。

此外，物理网卡也有缓存。当内核想要通过某网卡发送数据时，该网卡可以将内核写入该网卡的数据缓存起来，然后由网卡自身来决定何时发送数据。同理，网卡接收数据时也可以缓存一部分后再通知内核来读取网卡中的数据。

总结

- ① 物理网卡是硬件设备，工作在OSI最底层，其硬件功能是以比特流的方式接收和发送数据
- ② 物理网卡需网卡驱动程序向内核注册后方可工作，注册后一般会显示对应的网卡接口，网卡接口名称是给用户看的，内核不以接口名称来识别网卡
- ③ 为网卡接口做的配置属于内核而非网卡，内核中有些接口属性来自硬件网卡，而且有些属性可以修改。另外，可以为一个接口配置多个IP地址
- ④ 内核在和某网卡交互时，会基于内核中属于该网卡(接口)的配置属性做出决策和对应的处理
- ⑤ 物理网卡一端是内核(网络协议栈)，另一端是外界网络
- ⑥ 物理网卡收发数据(先不考虑DMA方式):
 - 发：本机通过某物理网卡发送数据时，内核将网络协议栈中的数据写入该网卡，该网卡将其发送出去
 - 收：网卡收到数据后，中断通知内核，内核从网卡读取数据并放入网络协议栈
- ⑦ 但网卡具有DMA功能，所以网卡和网络协议栈之间的数据传输主要由网卡负责，DMA传输完毕后DMA控制器中断通知内核已传输完毕

文章作者： 骏马金龙

文章链接: https://www.junmajinlong.com/virtual/network/kernel_nic_ip/

版权声明: 本博客所有文章除特别声明外, 均采用 [CC BY-NC-SA 4.0](#) 许可协议。转载请注明来自 骏马金龙!

