Linux 虚拟网络设备之 TUN/TAP 设备


StrokMit...
公众号：闲话操作系统

[Linux 内核文档]关于 TUN/TAP 设备描述：

> TUN/TAP provides packet reception and transmission for user space programs. It can be seen as a simple Point-to-Point or Ethernet device, which, instead of receiving packets from physical media, receives them from user space program and instead of sending packets via physical media writes them to the user space program.

一、TAP/TUN 是什么

在计算机网络中，TUN 与 TAP 是操作系统内核中的**虚拟网络设备**。不同于普通靠硬件网路板卡实现的设备，这些虚拟的网络设备全部由软件实现，并向运行于操作系统上的软件提供与硬件的网络设备完全相同的功能。 TAP 等同于一个以太网设备，它操作第二层数据包如以太网数据帧。TUN 模拟了网络层设备，操作第三层数据包比如 IP 数据封包。

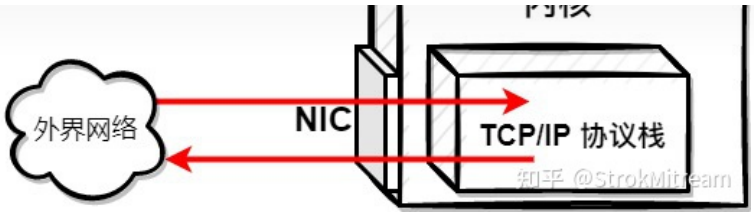
操作系统通过 TUN/TAP 设备向绑定该设备的用户空间的程序发送数据，反之，用户空间的程序也可以像操作硬件网络设备那样，通过 TUN/TAP 设备发送数据。在后种情况下，TUN/TAP 设备向操作系统的网络栈投递（或"注入"）数据包，从而模拟从外部接受数据的过程。

二、物理网卡收发数据的流程

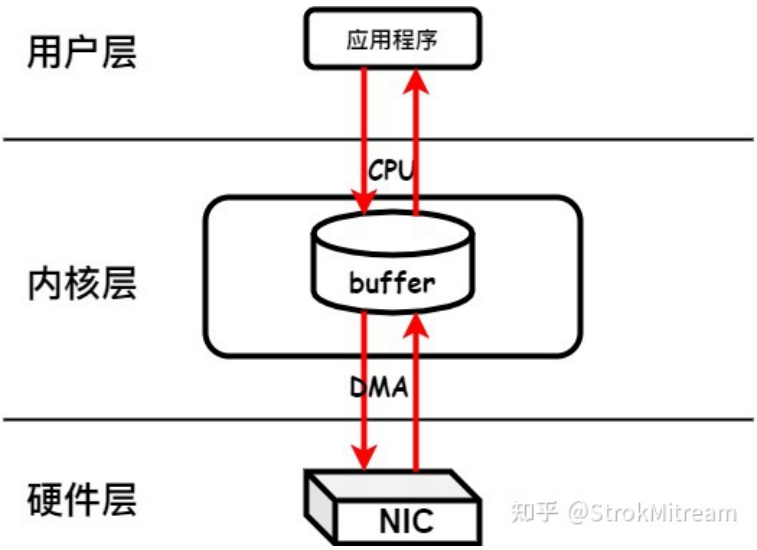
在了解虚拟网卡设备之前，我们先来看一下物理网卡是怎么工作的。

物理网卡是这样收发数据的： **收：**外部网络发送给主机的数据，通过物理网卡接收进来，并传输给内核协议栈处理 **发：**本地主机对外发送数据，将在内核协议栈中封装好数据包，最终通过网卡将数据发送出去

物理网卡，它的一端是**内核空间的网络协议栈**，另一端是**外界网络**，物理网卡就是连接这两者，以 `01` 形式的比特流收发数据的硬件设备。



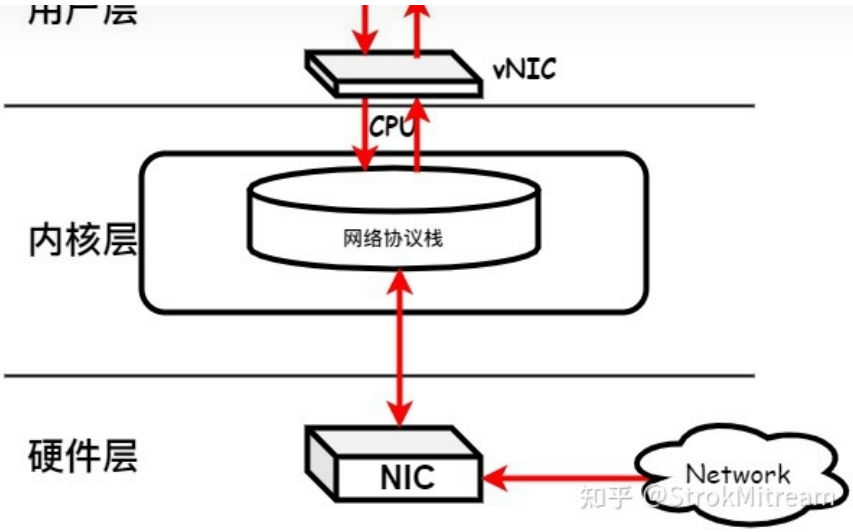
当用户进程的数据要发送出去时，数据从用户空间写入内核的网络协议栈，再从网络协议栈传输到网卡，最后发送出去；当用户进程等待外界响应数据时，数据从网卡流入，传输至内核的网络协议栈，最后数据写入用户空间被用户进程读取。 在这些过程中，内核和用户空间的数据传输，由内核占用CPU来完成；内核和网卡之间的数据，传输由网卡的DMA来完成，不需要占用过多的CPU。



三、虚拟网卡设备

物理网卡需要通过网卡驱动在内核中注册后才能工作，它在内核网络协议栈和外界网络之间传递数据，用户可以为物理网卡配置网卡接口属性，比如 IP 地址，这些属性都配置在内核的网络协议栈中。 内核也可以直接创建虚拟的网卡，只要为虚拟网卡提供网卡驱动程序，使其在内核中可以注册成为网卡设备，它就可以工作。 相比于物理网卡负责内核网络协议栈和外界网络之间的数据传输，虚拟网卡的两端则是**内核网络协议栈和用户空间，它负责在内核网络协议栈和用户空间的程序之间传递数据：**

发送到虚拟网卡的数据来自于用户空间，然后被内核读取到网络协议栈中；内核写入虚拟网卡，准备通过该网卡发送的数据，目的地是用户空间。



四、虚拟网卡和物理网卡的对比

与物理网卡对比一下，物理网卡是硬件设备，位于硬件层；虚拟网卡则可以看作是用户空间的网卡。

物理网卡和虚拟网卡唯一的不同点在于，物理网卡本身的硬件功能：**物理网卡以比特流的方式传输数据。**

也就是说，内核会公平对待物理网卡和虚拟网卡，物理网卡能做的配置，虚拟网卡也能做。比如可以为虚拟网卡接口配置IP地址、设置子网掩码，可以将虚拟网卡接入网桥等等。

只有在数据流经物理网卡和虚拟网卡的那一刻，才会体现出它们的不同，即传输数据的方式不同：物理网卡以**比特流**的方式传输数据，虚拟网卡则直接在内存中拷贝数据(即，在内核之间和读写虚拟网卡的程序之间传输)。

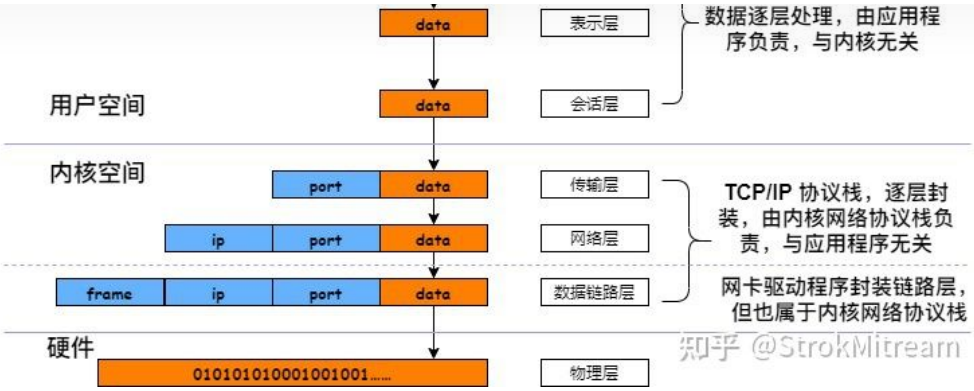
正因为虚拟网卡不具备物理网卡以比特流方式传输数据的硬件功能，所以，绝不可能通过虚拟网卡向外界发送数据，外界数据也不可能直接发送到虚拟网卡上。能够直接收发外界数据的，只能是物理设备。

虽然虚拟网卡无法将数据传输到外界网络，但却：

·可以将数据传输到本机的另一个网卡(虚拟网卡或物理网卡)或其它虚拟设备(如虚拟交换机)上；·可以在用户空间运行一个可读写虚拟网卡的程序，该程序可将流经虚拟网卡的数据包进行处理，比如OpenVPN 程序。

很多人会误解这样的用户空间程序，认为它们可以对数据进行封装。比如认为OpenVPN可以在数据包的基础上再封装一层隧道IP首部，但这种理解是错的。

一定请注意，用户空间的程序是无法对数据包做任何封装和解封操作的，所有的封装和解封都只能由内核的网络协议栈来完成。



使用OpenVPN之所以可以对数据再封装一层隧道IP层，是因为OpenVPN可以读取已经封装过一次IP首部的数据，并将包含IP首部的数据作为普通数据通过虚拟网卡再次传输给内核。 因为内核接收到的是来自虚拟网卡的数据，所以内核会将其当作普通数据（即应用层数据），从头开始封装(从四层封装到二层封装)。当数据从网络协议栈流出时，就有了两层IP首部的封装。

换句话说，每一次看似由用户空间程序进行的额外封装，都意味着数据要从内核空间到用户空间，再到内核空间。以OpenVPN为例：

```
tcp/ip stack --> tun --> OpenVPN --> tcp/ip stack --> Phyical NIC
```

其中tun是OpenVPN创建的一个三层虚拟网卡，tun设备在用户空间和内核空间之间传递数据。

具体的 OpenVPN 数据封装和数据流向的细节，参考更详细的通过 OpenVPN 分析tun实现隧道的数据流程。

五、TUN与TAP的区别

tun和tap都是虚拟网卡设备，但是：

对比项	tun 设备	tap 设备
协议栈层次	三层设备	二层设备
处理的数据包类型	IP 数据包	以太网数据包
是否有 MAC 地址	没有 MAC 地址	有 MAC 地址

tap比tun更接近于物理网卡，可以认为，tap设备等价于去掉了硬件功能的物理网卡

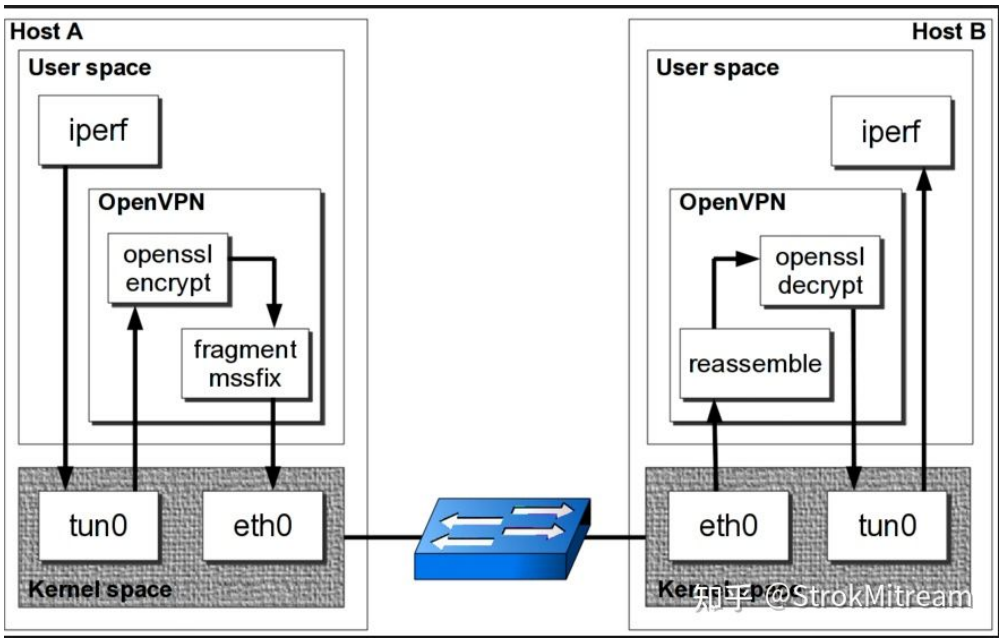
总结一下，虚拟网卡的两个主要功能是：

·1.连接其它设备(虚拟网卡或物理网卡)和虚拟交换机(bridge)·2.提供用户空间程序去收发虚拟网卡上的数据

基于这两个功能，tap设备通常用来连接其它网络设备(它更像网卡)，tun设备通常用来结合用户空间程序实现再次封装。换句话说，tap设备通常接入到虚拟交换机(bridge)上作为局域网的一个节

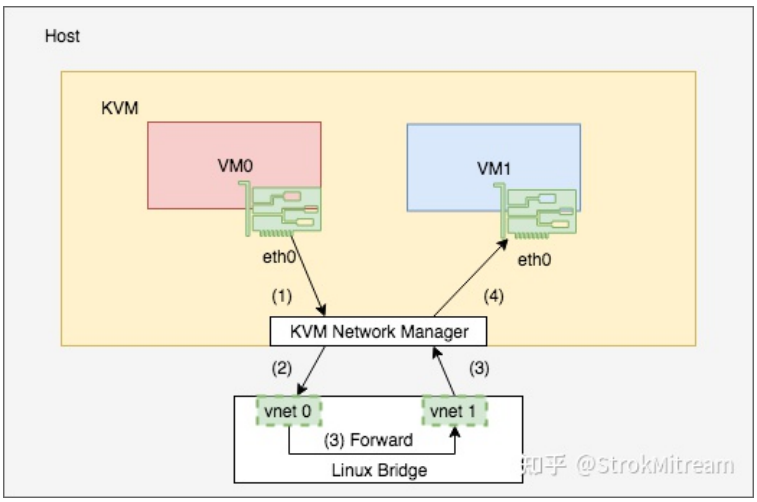
七、TUN 典型使用场景

典型使用场景为 IP 隧道。 举个例子，应用程序发送的数据，从 OpenVPN TUN 接口 tun0 接收，先进行加密处理，再发给物理网卡 eth0 发出去。 对端的 OpenVPN 客户端从物理网卡 eth0 收到加密数据，通过 OpenVPN 进行解密处理之后，再经由 tun0 将数据转发到应用程序。 也就是说，OpenVPN 就像是一个工作在虚拟接口 tun0 与物理接口 eth0 之间的代理，从而实现在物理网络上构建一条加密隧道。



八、TAP 典型使用场景

TAP 接口的典型应用场景是在虚拟化网络中。例如，我们通过KVM创建的多个 VM（虚拟机），以 LinuxBridge（桥接网络）互通；实际上即是通过像 vnet0 这样的 TAP 接口来接入 LinuxBridge 的。 在这种场景下，KVM程序 就是向 TAP 接口读写数据的**用户空间程序**。当 VM0 向本机的 eth0 接口发送数据，KVM 会将数据发送到 TAP 接口 vnet0 ，再通过 LinuxBridge 将数据转发到 vnet1 上。然后，KVM 将数据发送到 VM1 的 eth0 口。



1.hechao.li/2018/05/21/Tu... 2.backreference.org/2010/... 3.segmentfault.com/a/1190...
4.github.com/xgfone/snipp... 5.zhaohuabing.com/post/20... 6.blog.liu-kevin.com/2020...
7.kernel.org/doc/Document...

编辑于 2021-07-11 23:44

网络协议 协议栈 网络虚拟化

7 条评论 切换为时间排序

写下你的评论...

1111 IP 属地河北 · 07-29
文件很的很好，有个小疑问，数据从tcp/ip->tun这一步，如何判断哪些什么会流入tun?还是所有数据都会流入?
赞

StrokMitream (作者) 回复 **1111** IP 属地广东 · 8 小时前
这个问题问得很好!
赞

StrokMitream (作者) 回复 **1111** IP 属地广东 · 8 小时前
在实际配置 tun 隧道中，是需要配置路由规则，把流量指 tun 这张虚拟网卡设备的。

详细的 数据收发包流程，可以参考下这篇：Linux虚拟网络设备之tun/tap - SegmentFault 思否
赞

小白鼠 IP 属地上海 · 06-15
写得太棒了!
赞

StrokMitream (作者) 回复 **小白鼠** IP 属地广东 · 8 小时前
感谢!
赞

Cocyber 04-13
感谢楼主，写的真棒！解答了心中的困惑!
赞

StrokMitream (作者) 回复 **Cocyber** 04-13
感谢支持!
赞

闲话操作系统
OS 原理和内核设计开发