Linux Tun/Tap

Linux Tun/Tap 介绍

Posted by "Huabing Zhao" on Monday, February 24, 2020

什么是Tun/Tap

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用软件实现，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。

应用程序如何操作Tun/Tap

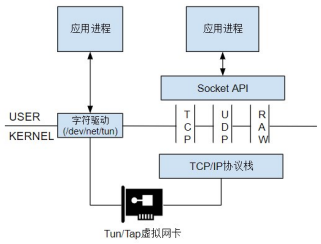
Linux Tun/Tap驱动程序为应用程序提供了两种交互方式：虚拟网络接口和字符设备/dev/net/tun。写入字符设备/dev/net/tun的数据会发送到虚拟网络接口中；发送到虚拟网络接口中的数据也会出现在该字符设备上。

应用程序可以通过标准的Socket API向Tun/Tap接口发送IP数据包，就好像对一个真实的网卡进行操作一样。除了应用程序以外，操作系统也会根据TCP/IP协议栈的处理向Tun/Tap接口发送IP数据包或者以太网数据包，例如ARP或者ICMP数据包。Tun/Tap驱动程序会将Tun/Tap接口收到的数据包原样写入到/dev/net/tun字符设备上，处理Tun/Tap数据的应用程序如VPN程序可以从该设备上读取到数据包，以进行相应处理。

应用程序也可以通过/dev/net/tun字符设备写入数据包，这种情况下该字符设备上写入的数据包会被发送到Tun/Tap虚拟接口上，进入操作系统的TCP/IP协议栈进行相应处理，就像从物理网卡进入操作系统的数据一样。

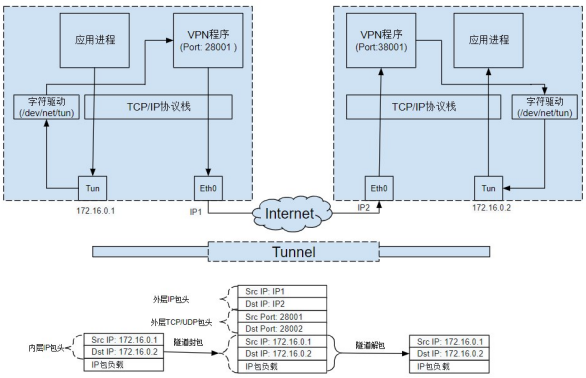
Tun虚拟设备和物理网卡的区别是Tun虚拟设备是IP层设备，从/dev/net/tun字符设备上读取的是IP数据包，写入的也只能是IP数据包，因此不能进行二层操作，如发送ARP请求和以太网广播。与之相对的是，Tap虚拟设备是以太网设备，处理的是二层以太网数据帧，从/dev/net/tun字符设备上读取的是以太网数据帧，写入的也只能是以太网数据帧。从这点来看，Tap虚拟设备和真实的物理网卡的能力更接近。

下图描述了Tap/Tun的工作原理：

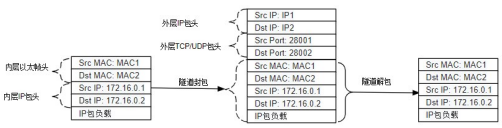


使用Tun/Tap创建点对点隧道

通过应用程序从/dev/net/tun字符设备中读取或者写入数据看上去并没有太大用处，但通过将Tun/Tap结合物理网络设备使用,我们可以创建一个点对点的隧道。如下图所示，左边主机上应用程序发送到Tun虚拟设备上的IP数据包被VPN程序通过字符设备接收，然后再通过一个TCP或者UDP隧道发送到右端的VPN服务器上，VPN服务器将隧道负载中的原始IP数据包写入字符设备，这些IP包就会出现在右侧的Tun虚拟设备上，最后通过操作系统协议栈和socket接口发送到右侧的应用程序上。

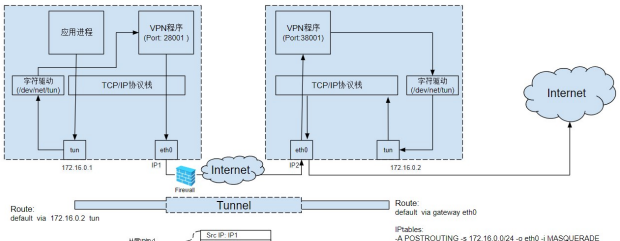


上图中的隧道也可以采用Tap虚拟设备实现。使用Tap的话，隧道的负载将是以太数据帧而不是IP数据包，而且还会传递ARP等广播数据包。



使用Tun/Tap隧道绕过防火墙

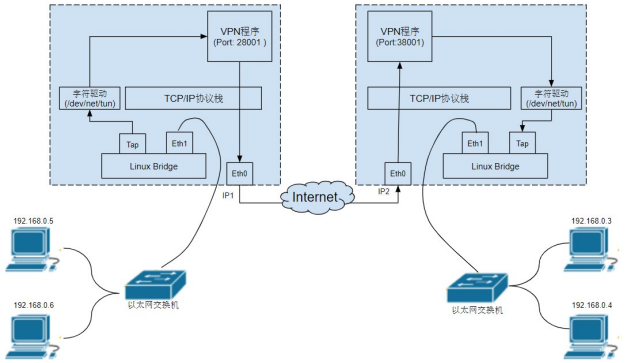
结合路由规则和IPTables规则，可以将VPN服务器端的主机作为连接外部网络的网关，以绕过防火墙对客户端的一些外部网络访问限制。如下图所示，防火墙规则允许客户端访问主机IP2，而禁止访问其他Internet上的节点。通过采用Tun隧道，从防火墙角度只能看到被封装后的数据包，因此防火墙认为客户端只是在访问IP2，会对数据进行放行。而VPN服务端在解包得到真实的访问目的后，会通过路由规则和IPTables规则将请求转发到真正的访问目的地上，然后再将真实目的地的响应IP数据包封装进隧道后原路返回给客户端，从而达到绕过防火墙限制的目的。



使用Tap隧道桥接两个远程站点

如下图所示，可以使用tap建立二层隧道将两个远程站点桥接起来，组成一个局域网。对于两边站点中的主机来说，访问对方站点的主机和本地站点的主机的方式没有区别，都处于一个局域网192.168.0.0/24中。

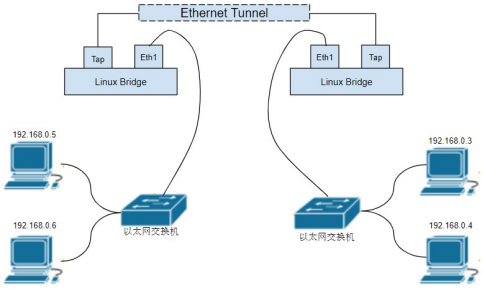
VPN主机上有两个物理网卡，其中Eth0用于和对方站点的VPN主机进行通信，建立隧道。Eth1在通过网线连接到以太网交换机的同时也被则加入了Linux Bridge，这相当于用一条网线将Linux Bridge上的一个端口（Eth1）连接到了本地站点的以太网交换机上，Eth1上收到的所有数据包都会被发送到Linux Bridge上，Linux Bridge发给Eth1的数据包也会被发送到以太网交换机上。Linux Bridge上还有一个Tap虚拟网卡，用于VPN程序接收从Linux Bridge上收到的数据包。



假设192.168.0.5发出了一个对192.168.0.3的ARP请求，该ARP请求在网络中经过的路径如下：

1. 192.168.0.5发出ARP请求，询问192.168.0.3的MAC地址。
2. 该ARP请求将被发送到以太网交换机上。
3. 以太网交换机对该请求进行泛洪，发送到其包括Eth1在内的所有端口上。
4. 由于Eth1被加入了VPN主机上的Linux Bridge，因此Linux Bridge收到该ARP请求。
5. Linux Bridge对该ARP请求进行泛洪，发送到连到其上面的Tap虚拟网卡上。
6. VPN程序通过/dev/net/tun字符设备读取到该ARP请求，然后封装到TCP/UDP包中，发送到对端站点的VPN主机。
7. 对端站点的VPN程序通过监听TCP/UDP端口接收到封装的ARP请求，将ARP请求通过/dev/net/tun字符设备写入到Tap设备中。
8. Linux Bridge泛洪，将ARP请求发送往Eth1，由于Eth1连接到了以太网交换机上，以太网交换机接收到了该ARP请求。
9. 以太网交换机进行泛洪，将ARP请求发送给了包括192.168.0.3的所有主机。
10. 192.168.0.3收到了APR请求，判断IP地址和自己相同，对此请求进行响应。
11. 同理，ARP响应包也可以按照该路径返回到图左边包括192.168.0.5在内的站点中。

从站点主机的角度来看，上面图中两个VPN主机之间的远程连接可以看作一条虚拟的网线，这条网线将两个Linux Bridge连接起来。这两个Linux Bridge和两个以太网交换机一起将左右两个站点的主机连接在一起，形成了一个局域网。



参考资料

- Universal TUN/TAP device driver
- Universal TUN/TAP device driver Frequently Asked Question
- Tun/Tap interface tutorial
- A simplistic, simple-minded, naive tunnelling program using tun/tap interfaces and TCP

← PREVIOUS POST

NEXT POST →

FEATURED TAGS

aeraki aeraki-mesh ambient-mesh api-gateway bitcoin blockchain cryptocurrency dubbo envoy envoy-gateway istio kubernetes metaprotocol microservice onap service-mesh