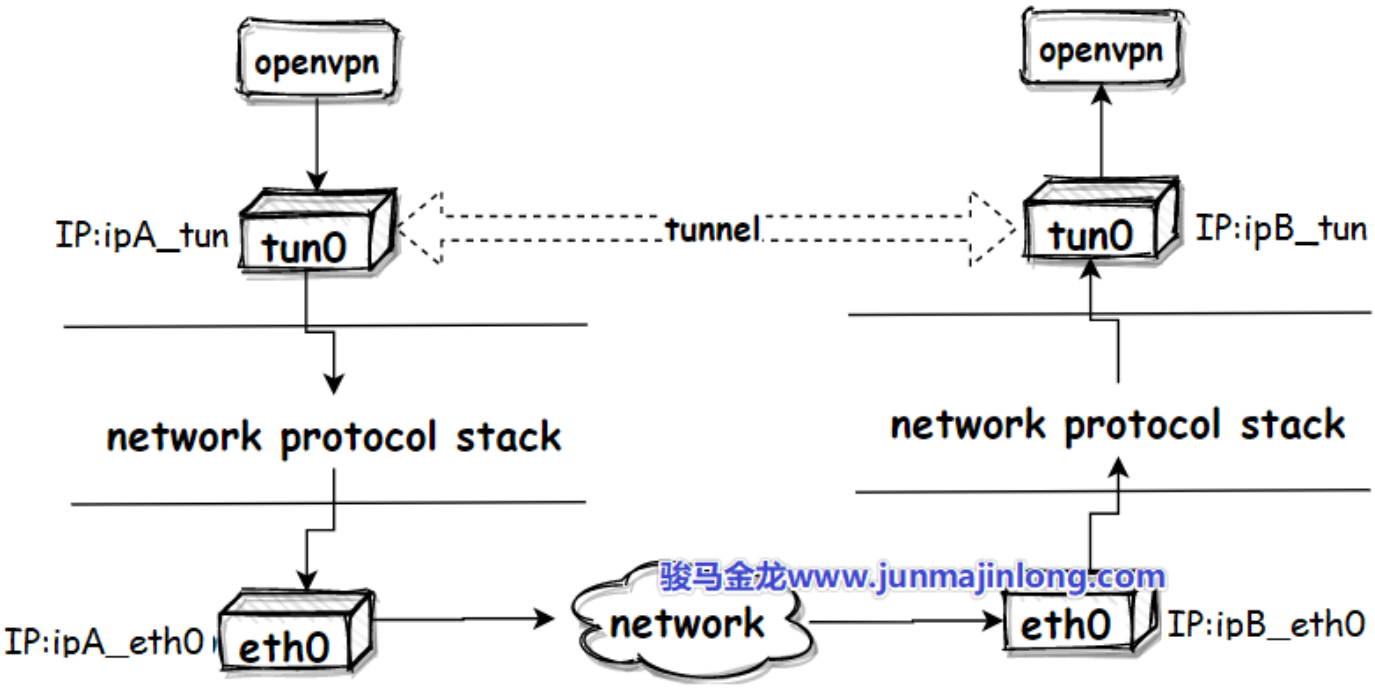
网络虚拟化系列文章

通过openvpn分析tun实现隧道的数据流程

> 提醒：
>
> 1.网上大多文章在这方面的分析都是错的，认为是openvpn对数据进行额外的封装和解封。
>
> 2.但实际上，在不绕过内核的情况下，**网络数据的封装和解封由内核负责，用户空间的程序无法对数据进行封装和解封。**

以OpenVPN配置ip隧道为例，下图是一个简图(而且是不正确的流程图)，后面会详细分析每一个步骤。

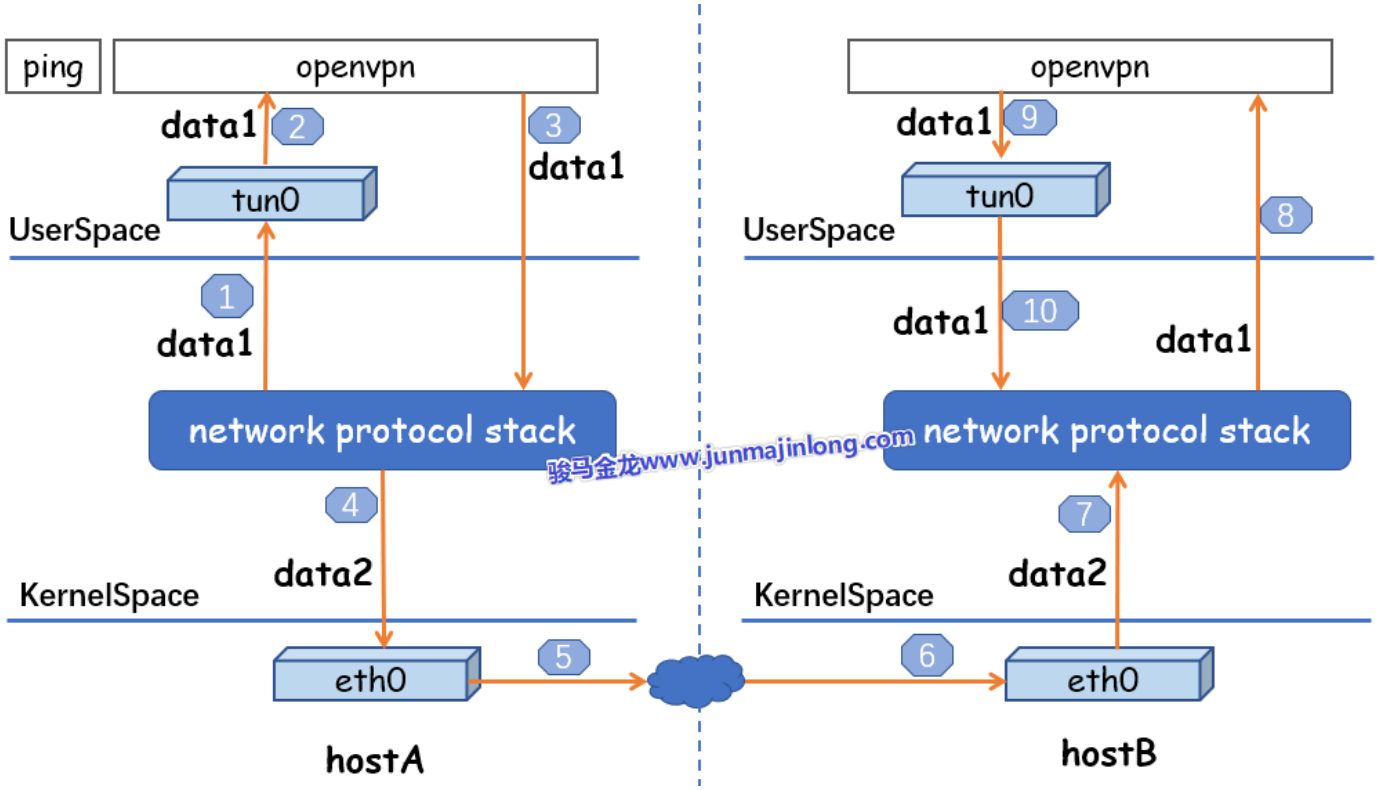


公网上的两个主机节点A、B，物理网卡上配置的IP分别是ipA_eth0和ipB_eth0。

在A、B两个节点上分别运行openvpn的客户端和服务端，它们都会在自己的节点上创建tun设备，且都会读取或写入这个tun设备。

假设这两个tun设备配置的IP地址分别是ipA_tun和ipB_tun，再在A、B节点上分别配置到目标tun IP的路由走本机的tun接口，两者就成功建立了一条能互相通信的隧道。

这里详细分析一下隧道通信的数据流程。以`ping ipB_tun`为例，其整体流程图如下：



其中data1和data2是什么，下面会说明。

通过openvpn发送数据

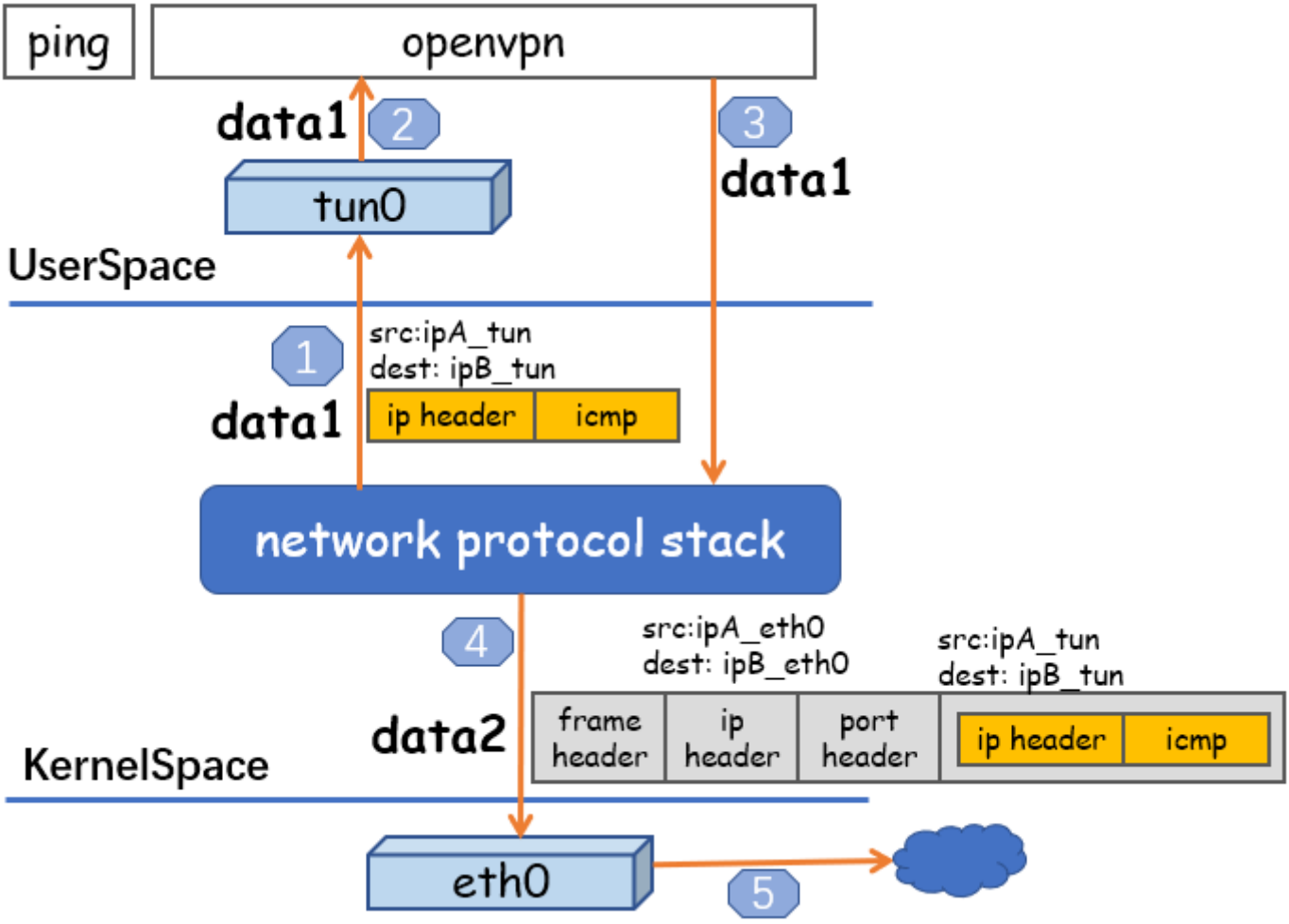
当A节点数据要通过隧道发向B节点时：

- 用户空间执行`ping ipB_tun`，ping程序会请求内核网络协议栈构建icmp协议请求数据
- 经过路由决策，该icmp协议数据要走tun0接口，于是内核将数据从网络协议栈写入tun0设备
  - 写入tun0设备之前，网络协议栈会对icmp请求数据进行封装，假设封装后的数据称为data1
  - tun是三层设备，所以data1中只封装IP头，不封装以太网帧头，其源和目标IP分别是ipA_tun、ipB_tun

    - 注意，data1中没有封装以太网帧头

- OpenVPN读取tun0设备数据，将读取到的data1数据当作普通数据发向B节点的eth0地址ipB_eth0，于是data1写入到网络协议栈
    - 用户程序OpenVPN从虚拟网卡读取的数据是原封不动地data1，其中包含了内核已经封装过的IP头，OpenVPN是无法对数据进行解封的
    - OpenVPN请求内核将data1作为普通数据发送出去，于是包含IP头的data1被写入内核网络协议栈
    - 内核经过路由决策，data1数据要从本机的eth0接口流出
    - 所以网络协议栈会对data1进行封装，假设封装后的数据称为data2
    - data2中封装的内容包括：
        - OpenVPN的源和目标端口(因为OpenVPN是用户服务程序)
        - 两个节点的eth0的IP地址
        - 以太网帧头(因为数据要从物理层的eth0设备出去，所以要从四层封装到二层)
    - 注意data2中有两层IP头，内层的IP头即data1中的IP头是tun设备的IP，外层的IP头是物理网卡eth0的IP
- data2最终通过A的物理网卡eth0发送出去到达B节点的物理网卡eth0

以上完整流程如下图：

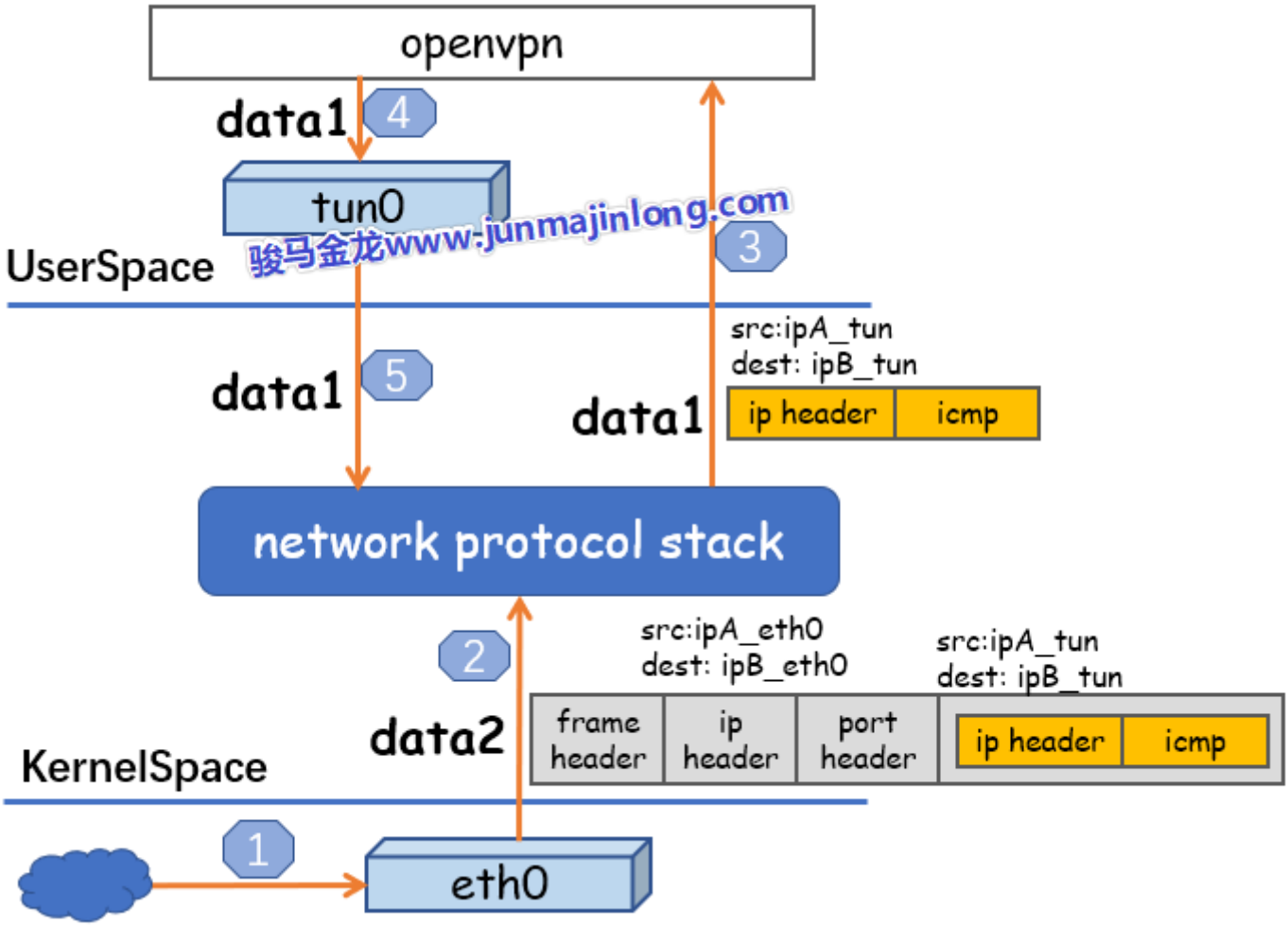


通过openvpn接收收据

A节点通过eth0发送的数据经由网络最终会到达B节点的eth0接口。

- 当B节点的物理网卡eth0收到数据后，对比特流进行解析，得到data2数据，写入内核网络协议栈：
  - 网络协议栈会对data2解封，将以太网帧头、外层IP头和端口层剥掉，最终得到data1
  - 因为目标端口号是OpenVPN程序所监听的，所以解封后的data1数据交给OpenVPN，即内核将data1拷贝到用户空间
  - 注意，data1中完整地包含了A、B两节点中两个tun设备的源和目标IP
- OpenVPN程序得到data1后，发现目标IP是tun0设备的(虽然openvpn无法解封数据，但却可以分析数据)，于是将data1数据从用户空间写入tun0设备(就像外界数据流入物理网卡一样)，data1最终传输到tun0的另一端即内核的网络协议栈中

以上完整流程如下图：



- B节点的网络协议栈对data1数据解封得到最内层的ICMP协议请求数据，同时内核发现目的IP是配置在本机tun0设备上的地址，于是响应它而非丢弃该数据。B构建响应数据的过程类似于A节点构建ping请求时的流程
  - ICMP协议位于tcp/ip协议栈中，不涉及应用层，所以直接由内核构建ping的响应数据
  - 因为解封data1时的源IP是A节点的ipA_tun，所以构建的响应目标是ipA_tun
  - 经过路由决策，该响应数据要从tun0流出，tun0是3层设备，所以只封装IP头(源IP和目标IP分别是ipB_tun、ipA_tun)，而不封装帧头
- 内核将响应数据写入tun0后，openvpn从中读取，读取后将其作为普通数据发送给ipA_eth0，于是数据写入网络协议栈，内核协议栈再次对其封装，包括端口号、两个eth0的IP地址以及以太网帧头，最终通过物理网卡eth0发送出去到达A节点

文章作者：骏马金龙

文章链接：

https://www.junmajinlong.com/virtual/network/data_flow_about_openvpn/