Computer Science and Application 计算机科学与应用, 2023, 13(5), 1055-1064
Published Online May 2023 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2023.135103

Hans汉斯

基于TUN设备的P2PVPN设计

张百川，康晓凤，蔡超萍，王　可，杨雪艳

徐州工程学院信息工程学院(大数据学院)，江苏　徐州

收稿日期：2023年4月20日；录用日期：2023年5月19日；发布日期：2023年5月29日

摘　要

在大数据时代背景下，隐私保护和网络安全问题受到广泛关注。为应对挑战，本文提出了基于TUN设备的点对点虚拟专用网络(P2PVPN)设计方案，利用TUN设备构建虚拟网络接口，并采用分布式路由表管理节点间通信。该方案采用了基于ed25519非对称加密的去中心化网络节点结构，提高数据传输安全性，具备跨平台运行能力，实现高通用性。实验验证表明，该设计在性能和安全性方面优异。与传统VPN相比，P2PVPN允许用户在无需第三方服务器情况下进行点对点通信，因此可以完全抵御DoS攻击，使数据传输更加安全稳定，为用户带来灵活、可靠、高效的P2PVPN服务体验。

关键词

虚拟私人网络，Tun/Tap设备，网络安全，Linux网络协议栈

Exploration of P2PVPN Ideas Based on TUN/TAP Technology

Baichuan Zhang, Xiaofeng Kang, Chaoping Cai, Ke Wang, Xueyan Yang

College of Information Engineering (Big Data College), Xuzhou University of Technology, Xuzhou Jiangsu

Received: Apr. 20th, 2023; accepted: May 19th, 2023; published: May 29th, 2023

Abstract

Privacy protection and network security concerns have received widespread attention in the context of the big data era. This paper proposes a design scheme for a point-to-point virtual private network (P2PVPN) based on TUN devices to address these challenges. The scheme employs TUN devices to create virtual network interfaces and utilizes distributed routing tables to manage communication between nodes. In addition, the proposed solution adopts a decentralized network node structure that is based on ed25519 asymmetric encryption to enhance data transmis-

文章引用: 张百川, 康晓凤, 蔡超萍, 王可, 杨雪艳. 基于TUN设备的P2PVPN设计[J]. 计算机科学与应用, 2023, 13(5): 1055-1064. DOI: 10.12677/csa.2023.135103

sion security. The design offers cross-platform operability and boasts high versatility. Experimental verification indicates that the proposed P2PVPN design exhibits excellent performance and security. Furthermore, in contrast to traditional VPNs, P2PVPN enables peer-to-peer communication without the need for third-party servers, rendering it impervious to DoS attacks. As a result, data transmission becomes more secure and stable, providing users with a flexible, reliable, and efficient P2PVPN service experience.

Keywords

Virtual Private Network, Tun/Tap Device, Cyber Security, Linux Network Protocol Stack

1. 引言

近年来，随着互联网技术的快速发展，网络安全和隐私保护逐渐成为了人们关注的焦点。虽然虚拟专用网络(VPN)作为一种常用的网络安全技术，能够加密用户数据并保护网络隐私，但传统的 VPN [1]在部署、性能和抗攻击能力方面仍存在一定局限性。点对点虚拟专用网络(P2PVPN)作为一种新型的 VPN 技术，虽然在某些方面取得了显著的优势[2]，但在实际应用中仍然面临着一些问题。

现有的 P2PVPN 技术如在性能、安全性和跨平台兼容性等方面存在一定的不足。首先，在性能方面，n2n 方案的数据传输速度和网络稳定性尚待进一步提高[3]。其次，在安全性方面，Wireguard 在加密算法和身份验证机制上仍存在可提升的空间[4]。最后，在跨平台兼容性方面，NordVPN 在不同操作系统和设备上的运行效果不尽如人意[5]。

针对现有 P2PVPN 技术存在的问题，本文提出了一种基于 TUN 设备的 P2PVPN 设计方案，以探究并尝试解决上述问题。首先，本文将通过优化网络架构和通信协议，提高 P2PVPN 的性能表现，包括数据传输速度和网络稳定性[6]。其次，为了增强 P2PVPN 的安全性，将采用基于 ed25519 非对称加密算法的公私钥模型的身份验证机制[7]，从而确保用户数据的机密性和完整性。本设计方案的创新之处在于采用了 ed25519 非对称加密算法，提高了数据传输的安全性。同时，该方案具备跨多个平台运行的能力，实现了高度的通用性。

本文的结构安排如下：第一部分为引言，简要介绍了研究背景、现有 P2PVPN 技术存在的问题以及本文的主要研究内；第二部分将阐述基于 TUN 设备的 P2PVPN 设计方案的基本概念、架构以及与传统 VPN 的比较；第三部分将详细描述实验过程和结果，展示本设计方案在性能和安全性方面的表现优势；最后，第四部分总结全文，回顾本设计方案的主要贡献，并探讨后续工作的可能方向。

2. 点对点虚拟私人网络简介

P2PVPN 是一种点对点(P2P)架构的 VPN 技术，它利用加密协议在两个设备之间建立安全、加密的通信隧道，以保护它们之间的通信，适用于多种应用场景，如远程办公、加密通信、绕过特定防火墙限制等。

下面从五个方面比较 P2PVPN 与传统 VPN：

1) 连接方式：传统 VPN 通常采用中央服务器来协调连接，而 P2PVPN 则是直接在客户端之间建立

点对点的连接，不需要依赖中央服务器。

2) 数据流量：传统 VPN 数据流量通常需要通过 VPN 服务器进行转发，而 P2PVPN 则是直接在客户端之间传输，因此在一些特殊情况下，P2PVPN 可以更加快速和高效。

3) 安全性：传统 VPN 和 P2PVPN 都可以提供安全、加密的通信隧道，以保护通信数据的安全性。不过，P2PVPN 在一些场景下拥有更强大的抗攻击能力，如流量攻击场景下 P2PVPN 的抗攻击能力更加强大。

4) 部署成本：传统 VPN 需要中央服务器来协调连接，因此在部署和维护方面可能需要更多的成本和资源。而 P2PVPN 则是直接在客户端之间建立连接，因此部署和维护成本相对较低。

5) 灵活性：传统 VPN 需要中央服务器来协调连接，因此可能会受到网络拓扑和配置限制。而 P2PVPN 则更加灵活自由，可以适应各种不同的网络环境和拓扑结构。

3. 点对点虚拟私人网络的设计

3.1. Linux 下数据包的接收过程

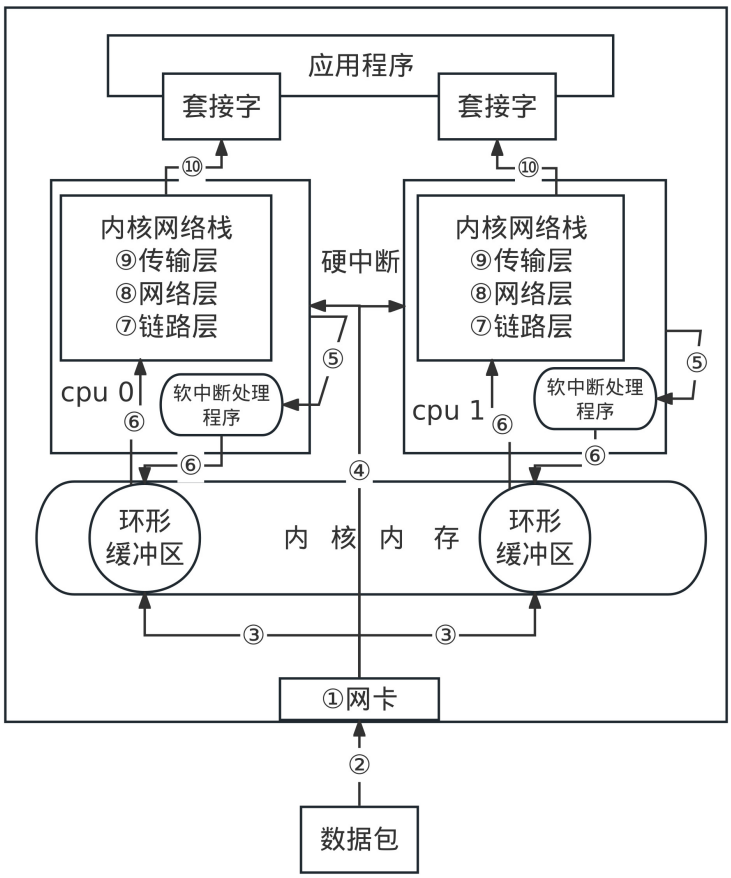

Figure 1. Processofpacketreceiving
图 1. 数据包接收过程

Linux 下数据包的接收过程如图 1 所示。步骤① 进行网卡的初始化操作，步骤② 当一个数据包从网络上到达网络接口卡(NIC)时，步骤③ NIC 会将数据包的信息(具体数据包格式由网卡规定)通过 DMA (Direct Memory Access)机制直接写入接收环形缓冲区中，步骤④ 并向主机的处理器发送一个硬件中断请

求(IRQ)，以通知主机有数据包到达，步骤⑤ 硬件中断程序会屏蔽该 CPU 上其他的硬件中断，为了减小对其他中断事件的影响，处理少量信息后将调用软中断处理程序(ksoftirq)释放对于硬件中断的屏蔽，将数据包的处理推迟到软中断的上下文中，接下来的操作都在软中断内执行，步骤⑥ 软中断处理程序(ksoftirq)会将数据包从接收环形缓冲区中取出，构建一个 sk_buff 数据结构，并将其提交到协议栈中。在 Linux 内核协议栈中，数据包将依次经过各个层次的处理，如步骤⑦ 数据链路层、步骤⑧ 网络层(IPv4/IPv6)、步骤⑨ 传输层(TCP/UDP)等，每一层都会处理数据包的相关信息，并将其传递到下一层或者丢弃/转发数据包，直到到达应用层。步骤⑩ 当数据包到达应用层时，它将被传递给相应的套接字(Socket)，以供应用程序使用。

由于虚拟私人网络与 IP 层的路由查询密切相关，因此接下来详细介绍图 1 中 IP 层的具体处理流程(步骤⑧)。

3.2. IP 层的处理流程

1) IP 层路由选择(IP Layer Routing Selection)

为了将数据包正确地转发到目标地址，IP 包会通过路由表按照正确的方向进行跳转。要实现这一目标，数据包必须包含源地址和目的地址。IP 协议支持数据分片功能。当数据过长时，IP 会将其分成多个分片并逐个发送。到达目的地时，这些分片的顺序可能与发送顺序不同，因此 IP 需要将这些分片重新组装成原始数据段，IP 数据报文格式如图 2 所示。

Figure 2. Format of IP packet
图 2. IP 数据包格式

IP 层接收数据包的目的地有两种情况：

其一为目标 IP 为本机，可以顺利通过本层协议栈，并交由运输层协议栈进行处理。

其二为目标 IP 不是本机，如果本机转发选项关闭，则将丢弃数据包，否则进行路由选择和转发。

IP 层发送数据包的目的地有两种情况：

其一为目标 IP 位于发送计算机所连接的本地网络之一。

其二为地理位置遥远、未连接到本地网络且只能通过网关访问的计算机。

以 Linux 5.10 内核为例，在讨论内核中数据包路由相关的 IP 层代码之前，首先需要了解一个内核网络相关的非常重要的数据结构：sk_buff。

2) sk_buff 结构体(sk_buff structure)

sk_buff 结构体用于封装网络数据包。每当一个网络数据包进入 Linux 内核协议栈时，都会被封装成一个 sk_buff 结构体，然后在协议栈内传递和处理。该结构体中与路由相关的成员为_skb_refdst。

这个成员存储一个指针，该指针指向一个称为目的入口(dst_entry)的结构体。该结构体可以决定 sk_buff 数据包的最终流向——对于已经接收的 sk_buff 数据包而言，可以决定数据包是否应该被转发，还是交由下一层进行处理，对于将要发送的 sk_buff 数据包而言，可以决定数据包应该从哪个设备发出。

dst_entry 结构中有两个重要的成员，分别为 input 和 output，这两个成员都是函数指针，它们分别处理传入和传出数据包的函数指针。当网络堆栈接收到数据包并需要进行处理时，调用 input 所指向的函数。当网络堆栈需要传输数据包时，调用 output 所指向的函数。在 dst_entry 结构中，input 和 output 函数指针通常在创建新的目标缓存条目时初始化。input 函数指针的初始化选项如表 1 所示，output 函数指针的初始化如表 2 所示。

Table 1. Callback function of input function pointer table
表 1. Input 函数指针回调函数表

函数	描述
ip_local_deliver	将数据包传递至本地的运输层
ip_forward	转发一个单播数据包
ip_mc_input	转发一个多播数据包
ip_error	处理一个无法到达的地址

Table 2. Callback function of output function pointer table
表 2. Output 函数指针回调函数表

函数	描述
ip_output	将数据包传递至链路层
ip_mc_output	将多播的数据包传递至链路层
ip_rt_bug	打印警告，因为这个函数无意义
dst_discard_out	丢掉该数据包

分配给这些指针的具体函数取决于正在创建的目标缓存条目的类型和所使用的网络协议。并且，如果 sk_buff 有相同的目的地址，则会共享相同的 det_entry。

3) IP 层数据接收(Receiving Data at the IP Layer)

IP 层中数据流的接收和发送的具体函数调用流程是首先 IP 层利用 ip_rcv 函数接收数据，该函数所需参数为一个 sk_buff 结构体，该函数中间过程会对数据包进行筛选，并查询路由表来设置数据包(sk_buff)的_skb_refdst 成员，图 3 中的 skb_dst_set 函数使其指向正确的目的入口(dst_entry)，最终由 dst_input 函数将数据发送到网络内核栈的下一层。dst_input 函数仅由一条语句构成：

return skb_dst(skb)->input(skb)；

skb_dst 函数将数据包(sk_buff)中的_skb_refdst 成员取出，该成员为指向 dst_entry 结构体的指针，dst_entry 结构体在被初始化的时候，其 input 成员根据目的地址被注册指定的接收数据函数。具体函数调用如图 3 所示。

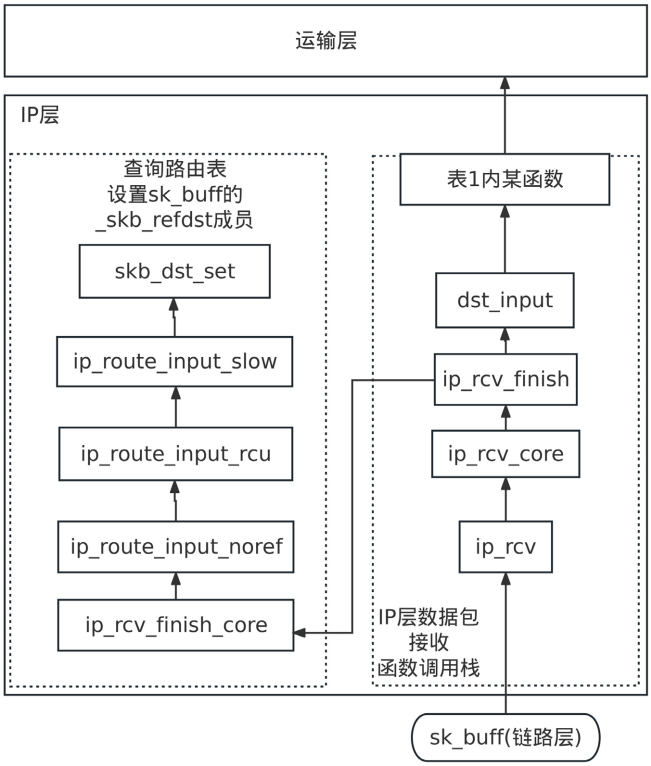

Figure 3. The process of calling IP layer data reception functions
图 3. IP 层数据接收函数调用过程

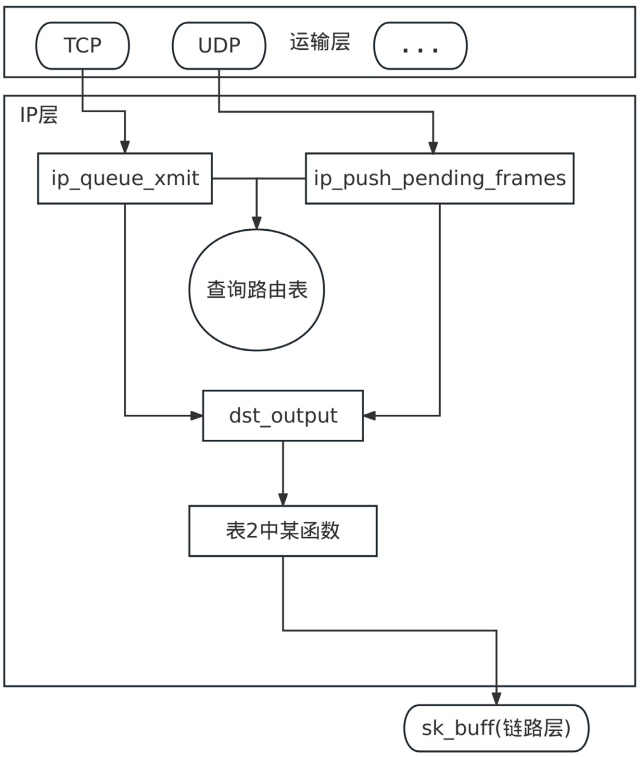

Figure 4. The process of calling IP layer data transmission functions
图 4. IP 层数据发送函数调用过程

4) IP 层数据发送(Sending Data at the IP layer)

在 IP 层中，数据的发送与接收类似，但是数据的发送需要从传输层的多个模块获取数据，具体如图 4 所示。在应用发送数据的过程中，数据包从传输层的不同模块传递到 IP 层，然后数据包(sk_buf)经过路由设置，最后通过 dst_output 函数来将数据包从网络设备中发送出去。dst_output 函数由一句代码组成，如下：

return skb_dst(skb)->output(net, sk, skb)；

skb_dst 函数将数据包(sk_buff)中的_skb_refdst 成员取出，该成员为指向 dst_entry 结构体的指针，dst_entry 结构体在被初始化的时候，其 output 成员根据目的地址被注册为指定发送函数。

3.3. TUN 设备

TUN 设备是一个三层的虚拟网络设备[6]，它的一端连接着 Linux 内核网络协议栈，另一端连接着用户空间，用户可以通过读写该虚拟网络设备，直接获取该设备内原始的 IP 报文，或者向该设备写入指定格式的信息，该信息会进入内核网络协议栈。将图 1 简化，并添加 TUN 设备后如图 5 所示。

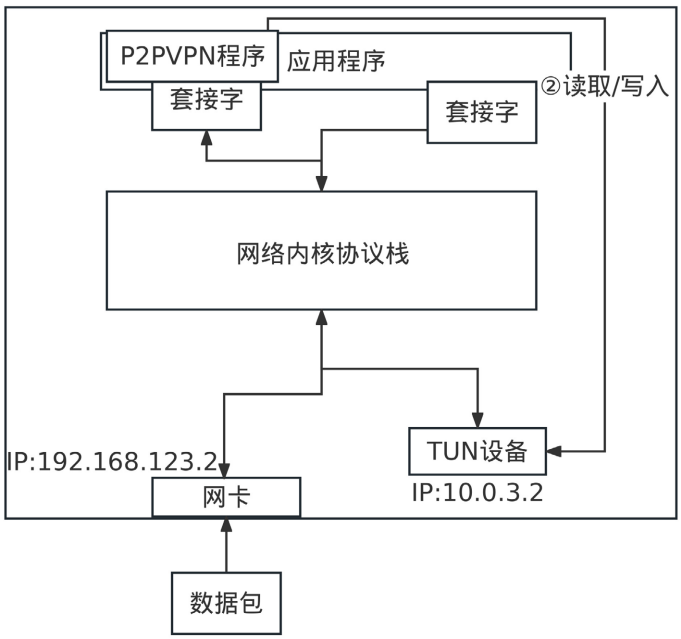

Figure 5. Adding TUN device to the kernel
图 5. 内核添加 TUN 设备

要在 Linux 上使用 TUN 设备，程序必须打开/dev/net/tun 并发出相应的 ioctl()以向内核注册网络设备[7]。网络设备将显示为 tunXX，具体取决于所选的选项。当程序关闭文件描述符时，网络设备和所有相应的路由将消失。

3.4. P2PVPN 的实现流程

首先，建立一个公共索引节点[4]，它包含所有连接到虚拟网络的设备信息。该索引节点用于辅助设备在复杂网络环境下使用技术手段进行网络穿透[8]，从而建立 P2P 隧道。

然后，P2PVPN 客户端通过 open("/dev/net/tun", O_RDWR)创建一个 TUN 设备[9]，利用 system 函数调用 ifconfig 工具来设置 TUN 设备的虚拟 IP 及掩码，如图 5 所示。并在使用 execvp 函数调用 IP 工具在

主机路由表内添加路由项，该路由项的目的网络为 TUN 设备的虚拟 IP 所在的网络 10.0.3.0/24，接口为 TUN 设备。

最后，当应用程序向虚拟私人网络发送数据时，数据包会经过路由选择进入 TUN 设备，其具体流程如 3.2 所述。P2PVPN 客户端通过读取进入 TUN 设备的数据包，对整个数据包进行自定义加密。加密后的数据将重新经过协议栈进行封装，并且新封装的数据包的目的地址变为 P2P 隧道的对端设备的真实 IP，最终通过物理网卡向外发出，如图 6 所示。

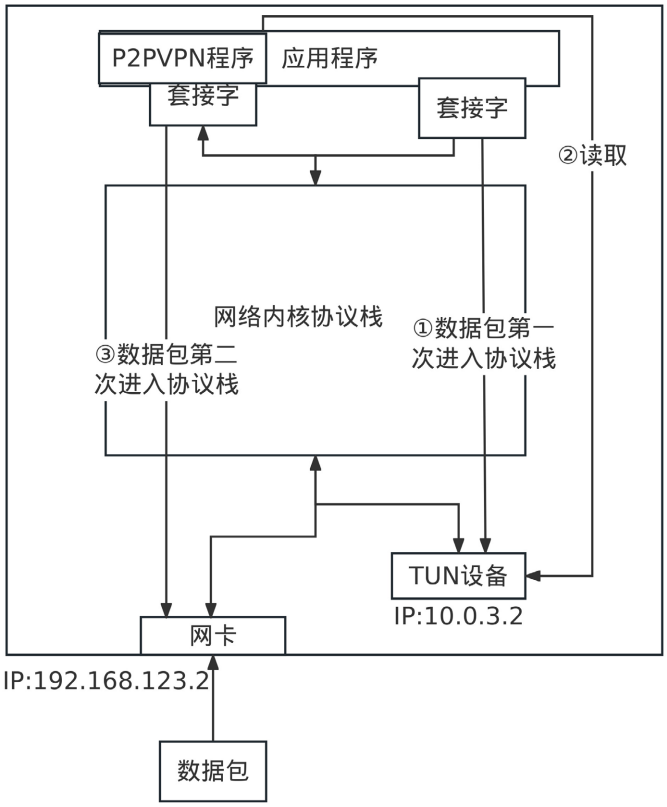

Figure 6. P2PVPN receive packet
图 6. P2PVPN 数据包接收

在接收数据时，物理网卡接收到数据包后，数据包经过路由选择后，判断数据包的目的地址为本机之后，将数据包向传输层呈递，直至等待 Socket 队列。P2PVPN 通过 Socket 接收到该数据包，数据包的应用数据实际为加被加密后的 IP 层数据。P2PVPN 客户端使用解密算法解密数据后，将获得一个带有虚拟 IP 地址的数据包，将该数据包写入 TUN 设备后，带有虚拟私人网络 IP 的数据包将进入内核协议栈，并最终被应用程序获取，如图 7 所示。

3.5. P2PVPN 的性能比较实验

在本章节中，我们将通过不同流量下对 vpn 服务器进行 Dos 攻击中的性能测试来评估所提出的基于 TUN 设备的 P2PVPN 方案，并将其与其他常见 VPN 方案进行比较。我们选择了 OpenVPN，IPsec(AES-GCM) 两种常见的 VPN 方案进行性能对比，测试设备为同一网络环境下的两台 Intel Core i5-8400 CPU，8 GB RAM，1 Gbps 本地网络连接，测试操作系统为 Ubuntu 20.04 LTS，每种 VPN 方案分别进行 10 次测试，取平均值，结果如表 3 所示：

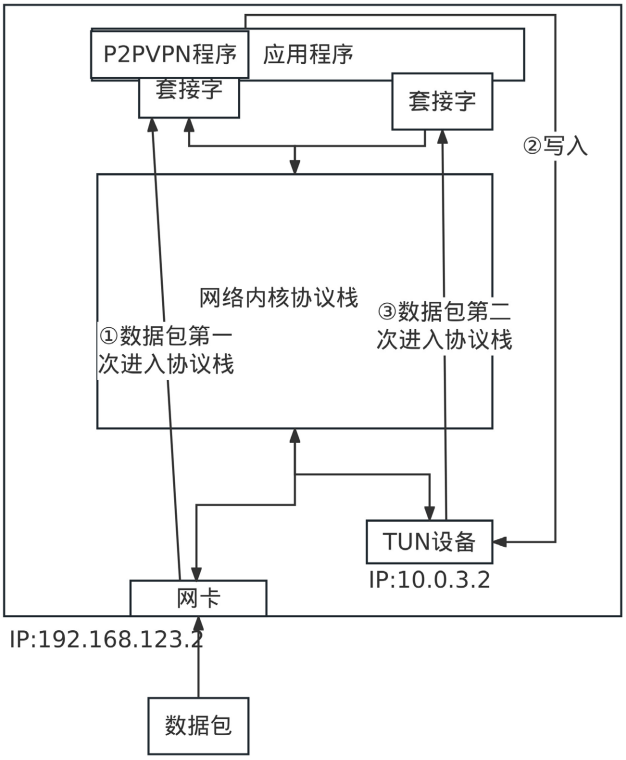

Figure 7. P2PVPN send packet
图 7. P2PVPN 数据包接收

Table 3. Table for VPN performance
表 3. Dos 攻击下 VPN 性能比较表格

vpn方案	延迟(ms)	吞吐量(Mbps)	Dos攻击流量(Mbps)
IPsec (AES-GCM)	0.503	881	100
OpenVPN	1.542	258	100
P2pVPN	0.487	684	100
IPsec (AES-GCM)	2.424	541	500
OpenVPN	7.123	131	500
P2pVPN	0.509	634	500
IPsec (AES-GCM)	N/A	N/A	1000
OpenVPN	N/A	N/A	1000
P2pVPN	0.544	628	1000

综合不同 DoS 攻击情况下的延迟和吞吐量测试的结果，由于基于 TUN 设备的 P2PVPN 方案的去中心化特性，攻击流量无法到达分布式节点，所以在高 DoS 流量攻击的情况下均表现优于其他常见 VPN 方案。这表明该方案能够在保证数据安全的同时，提供更高效的传输和解密处理，从而为用户带来更好的网络连接体验。

4. 结论

本文针对现有 VPN 技术的不足，提出了一种基于 TUN 设备的 P2P 架构 VPN 设计方案，通过优化网

络架构、通信协议和引入 ed25519 非对称加密算法以及公私钥身份验证机制，实现了更高的性能、安全性和跨平台兼容性。实验验证表明，本方案在网络安全和隐私保护领域具有显著的实用性和创新价值。特别是在抵御 DoS 攻击方面，本文提出的 P2PVPN 方案表现出强大的抵抗能力，为未来进一步研究和优化 P2PVPN 技术提供了新思路。

致　谢

本文为徐州工程学院大学生创新创业训练计划项目(xcx2022190, xcx2022192)的阶段性成果之一。

参考文献

[1] 何国彪. 去中心化可信互联网基础设施关键技术研究[D]: [博士学位论文]. 北京: 北京交通大学, 2021.

[2] Kerravala, Z. (2022) What Is a VPN? A Secure Network over the Internet. *Network World* (*Online*), **12**, 12-16.

[3] 马潇潇, 蒋诚智, 赵鑫, 等. 基于零信任理念的 VPN 访问控制技术[J]. 集成电路应用, 2022, 39(12): 114-115.

[4] 杨波. 内网穿透技术在远程访问中的研究与实现[J]. 长江信息通信, 2022, 35(7): 102-105.

[5] 谢小峰. VPN 技术在局域网中的组网的应用探讨[J]. 自动化应用, 2022(5): 68-70.

[6] 黄嘉煜. P2P 网络穿透策略与优化方法研究[D]: [硕士学位论文]. 哈尔滨: 哈尔滨工业大, 2019.

[7] 陈金莲. 依托 tap/tun 设备分析 openstack 中的网络虚拟化[J]. 黄冈职业技术学院学报, 2019, 21(4): 146-147.

[8] 欧炜滨. 基于 NAT 穿越的 P2P 系统的研究和实现[D]: [硕士学位论文]. 广州: 华南理工大学, 2021.

[9] 谭海涛. Linux 的 TUN/TAP 程序设计[C]//中国通信学会, 北方工业大学. 2007 通信理论与技术新发展——第十二届全国青年通信学术会议论文集(下册). 北京: 电子工业出版社, 2007: 988-992.