硕士学位论文

Android 移动终端 Wi-Fi 安全接入

关键技术的研究与实现

RESEARCH AND IMPLEMENTATION OF KEY TECHNOLOGY ON WI-FI SECURE ACCESS FOR ANDROID MOBILE TERMINALS

傅春乐

哈尔滨工业大学

2016 年 6 月

国内图书分类号：TP393.2　　学校代码：10213
国际图书分类号：004.7　　密级：公开

工程硕士学位论文

Android 移动终端 Wi-Fi 安全接入关键技术的研究与实现

硕 士 研 究 生：傅春乐
导　　师：何清刚 副教授
申 请 学 位：工程硕士
学　　科：计算机技术
所 在 单 位：计算机科学与技术学院
答 辩 日 期：2016 年 6 月
授予学位单位：哈尔滨工业大学

Classified Index: TP393.2
U.D.C: 004.7

Dissertation for the Master Degree in Engineering

RESEARCH AND IMPLEMENTATION OF KEY TECHNOLOGY ON WI-FI SECURE ACCESS FOR ANDROID MOBILE TERMINALS

Candidate： Fu Chunle
Supervisor： A. P. He Qinggang
Academic Degree Applied for： Master of Engineering
Speciality： Computer Technology
Affiliation： School of Computer Science and Technology
Date of Defence： June, 2016
Degree-Conferring-Institution： Harbin Institute of Technology

摘 要

Wi-Fi 热点为移动用户提供了快速便捷的网络接入服务，同时也容易被黑客非法利用，成为其窃取用户上网的个人账号和隐私数据的网络犯罪工具。为了保证移动用户在不可信无线网络环境下安全接入互联网，本文对移动VPN(Mobile VPN, MVPN)技术展开研究，改进了该技术在传输速度和通信连接稳定性方面的不足以更好的适用于移动终端。

首先，本文对 MVPN 的虚拟隧道技术展开研究，给出了该技术的分类标准、基本原理、评价指标和应用场景。通过研究 Android 平台的隧道构建方式和工作方式，本文实现了简单隧道 MVPN。功能测试结果表明该隧道能够作为移动终端所有流量的载体接收、发送数据，且不会影响终端设备的连网速度。

为了提高虚拟隧道的安全性，本文对 MVPN 的安全通信协议展开研究。研究工作分析了现有 MVPN 协议的不足，借鉴了 OpenVPN 协议设计思想，并提出了一种快速传输隧道协议。该协议将握手阶段与传输阶段分层实现，有助于简化隧道协议的复杂度，并提高通信隧道的传输效率。性能测试结果表明，基于该协议实现的快速传输 MVPN 较 OpenVPN 相比具有更快的传输速度和更低的传输时延，并且能保证虚拟隧道的安全通信。

为了提高安全隧道的稳定性，本文对 MVPN 通信连接的稳定性展开研究。研究工作阐明了现有 MVPN 应用稳定性不足的根本原因，提出了一种通用的 MVPN 通信保障机制，并使用形式化语言和有限状态机模型对该机制的基本原理分别进行抽象定义和数学建模。随后本文基于该机制设计了 MVPN 的通信保障模型并给出了该模型的原型实现—通信保障 MVPN。稳定性测试结果表明，通信保障模型的稳定性达到 96.67%。同时，该模型还具有一定的通用性和可扩展性。

最后，本文将上述实现的安全、快速、稳定的 MVPN 与 Wi-Fi 安全接入的应用场景有机结合，设计了 Android 移动终端 Wi-Fi 安全接入系统。系统测试结果表明，该系统功能能够满足移动终端的 Wi-Fi 安全接入的需求，整体性能满足预期目标，但耗电量有待进一步优化。

关键词： Wi-Fi；安全接入；移动 VPN；快速传输隧道协议；通信保障模型

Abstract

Wi-Fi hotspots provide mobile users with rapid and convenient network access service. Meanwhile, they are easy to be illegally utilized as tools of cyber crimes to steal users' personal accounts and private datum by hackers. In order to guarantee mobile users' secure access to Internet in untrusted wireless network environments, this paper researches on technology of Mobile Virtual Private Network (MVPN) and improves its shortage on transmission speed and connection stability to better cater for mobile terminals.

First of all, virtual tunnel technology of MVPN is researched with presenting its taxonomy criterions, basic principles, evaluations and application scenarios. By studing the establishment and working way of tunnels on Android platform, simple tunnel MVPN is implemented. Functional test results show that the tunnel is able to receive and transmit all the mobile traffic without limiting network access speed.

To enhance the security of virtual tunnels, this paper studies on secure communication protocols of MVPN. The shortages of existing MVPN protocols are analyzed and fast transmission tunnel protocol is raised based on the design concept of OpenVPN's protocol. Handshake stage and transmission stage of this protocol are implemented hierarchically for simplifying the complexity of tunnel protocols and improving the transmission efficiency of communication tunnels. Performance test results verify fast transmission MVPN implmented with the proposed protocol has faster transmission speed and lower transmission delay than OpenVPN. It can ensure secure communication of virtual tunnels as well.

To improve the stability of secure tunnels, this paper researches on the stability of MVPN connections. Our study clarifies the underlying reasons for insufficient stability of existing MVPN applications and proposes a generic communication supportable mechanism for MVPN. The basic principle of this mechanism is abstracted in formal languages and its mathematical models are constructed with Finite State Machine models. Then, a communication supportable model of MVPN is put forward in view of the mechanism above and communication suppotable MVPN is implemented as a prototype of this model. Stability test results prove the stability of communication supportable MVPN reaches 96.67%. Meanwhile, the

communication supportable model is superior in its generality and scalability.

Finally, this paper combines the secure, fast and stable MVPN implemented above with application scenarios of Wi-Fi secure access to design Wi-Fi secure access system for Android mobile terminals. System test results testify that the system functions can meet the requirements of Wi-Fi secure access on mobile terminals and the system performance appeals to our expectation. However, the power consumption is to be optimized further.

Key words: Wi-Fi, secure access, Mobile VPN, fast transmission tunnel protocol, communication supportable model

目 录

第1章 绪论

本章介绍了课题研究背景和意义，分析了课题相关技术的研究现状，并据此引出了本文研究内容，最后给出了本文组织结构。

1.1 课题研究背景和意义

互联网安全接入的需求是针对当下迅速膨胀的互联网接入需求以及日益增长的网络恶意攻击行为的背景提出的，Wi-Fi 无线网络的环境相对开放，接入设备的身份相对模糊，用户群体的构成相对复杂，导致 Wi-Fi 环境下的网络安全接入问题显得尤为突出，主要表现在以下三个方面：

第一，在用户需求方面，Wi-Fi 热点的接入需求巨大。随着智能城市理念的广泛推广和无线基础设施的全面建设，无论是在家庭、企业还是在公共场所，Wi-Fi 热点已遍布于我们生活的各个角落。据中国互联网络信息中心(China Internet Network Information Center，CNNIC)2016 年 1 月最新发布的《第 37 次中国互联网发展状况统计报告》指出，截止到 2015 年 12 月，在网民规模和结构方面，我国手机网民规模已达 6.20 亿，90.1%的网民通过手机上网；在网络连接方式方面，91.8%的网民通过 Wi-Fi 无线网络接入互联网[1]。由此可见，移动终端贡献了绝大多数的 Wi-Fi 接入总量。

第二，在网络安全方面，传统 Wi-Fi 的接入方式存在很多安全隐患，无论是在家庭 Wi-Fi、企业 Wi-Fi 还是公共 Wi-Fi 中，都存在被暴力破解 Wi-Fi 密码和被中间人攻击窃取流量隐私、盗取账号信息的风险。例如，乌云漏洞平台在 2015 年 5 月和 7 月，分别曝光北京首都机场 T1 和 T3 航站楼 Wi-Fi 热点存在认证漏洞和后台管理漏洞。又如，2015 年 7 月，360 天巡实验室发布《2015 企业无线网络安全报告》，指出超过 45%的企业 Wi-Fi 密码被分享公开，超过 90%的企业 Wi-Fi 使用了不安全的纯数字密码，钓鱼 Wi-Fi 成为黑客入侵企业网络的重要途径[2]。因此，在 Wi-Fi 的开放环境下，接入用户可能悄无声息地遭到上网流量窃听、数据隐私泄露、账号口令失窃等网络攻击，这对用户的个人数据隐私造成严重的威胁。

第三，在社会舆论方面，Wi-Fi 的安全问题正引起社会各方的关注。2014 年 9 月，腾讯公司宣布与众多国内商家和知名 Wi-Fi 提供商一起成立“腾讯 Wi-Fi 安全联盟”，期望为用户提供快速便捷、安全稳定的 Wi-Fi 网络。但对

于移动终端而言，可信的 Wi-Fi 热点并不一定意味着安全的 Wi-Fi 接入。攻击者依然可能通过钓鱼 Wi-Fi 的攻击方式迫使用户在不知情的情况下接入恶意的 Wi-Fi 热点[3]。2015 年央视 315 晚会现场便演示了黑客利用免费钓鱼 Wi-Fi 网络窃取移动设备中用户社交账号信息、邮箱账号和密码等个人隐私的过程，提醒移动用户警惕防御钓鱼 Wi-Fi。

因此，本课题将围绕 Wi-Fi 无线网络环境下移动终端安全接入问题展开研究，研究一种适用于 Android 移动终端的 Wi-Fi 安全接入技术，以确保移动终端无论在何时，何地，无论接入家庭 Wi-Fi、企业 Wi-Fi 还是公共 Wi-Fi 等何种类型的 Wi-Fi，无论 Wi-Fi 的接入环境是否存在恶意攻击者，都能提供移动用户安全、快速、稳定的互联网接入的解决方案。

1.2 课题相关技术的研究现状

本节对课题相关的背景技术和关键技术展开研究，通过实践调研、理论研究和技术分析相结合的方法，确定本文研究的核心技术和值得进一步探索的研究内容。

1.2.1 恶意 Wi-Fi 攻击方式研究

根据恶意 Wi-Fi 攻击的发起者不同，将恶意 Wi-Fi 的类型分成傀儡 Wi-Fi 和钓鱼 Wi-Fi 两大类，其中，恶意 Wi-Fi 的网络攻击细分为五种具体方式，如表 1-1 所示。

表 1-1 恶意 Wi-Fi 攻击方式对比表

Wi-Fi 类型	攻击方式	攻击特点	适用范围
傀儡 Wi-Fi	被动式攻击	长期固定监听，但实施复杂	家庭、企业 Wi-Fi
钓鱼 Wi-Fi	被动式攻击	实施简单，但攻击效率低下	公共 Wi-Fi
	主动式攻击	攻击效果好，但容易被检测	公共 Wi-Fi
	Karma 攻击	隐蔽性好，但需要满足条件	公共 Wi-Fi
	中间人攻击	攻击效果好，但容易被检测	公共 Wi-Fi

（1）傀儡 Wi-Fi

Wi-Fi 接入点(Access Point, AP)本身不是恶意的，但遭到黑客攻击、控制后，对接入 Wi-Fi AP 的终端设备进行流量窃听、内容篡改、设备扫描甚至恶意控制。傀儡 Wi-Fi 的攻击方式需要先攻破 AP 后台管理系统，注入恶意程序控制 AP，再对接入的终端设备进行网络攻击。因此，这类攻击实施相对复杂，攻击周期较长，但是黑客一旦控制了 AP 节点，便可以长时间监听终端设备的数据流量，因此常用于长期控制和监听家庭 Wi-Fi 和企业 Wi-Fi 等用户相对固

定的终端设备流量。

（2）钓鱼 Wi-Fi

Wi-Fi AP 本身就是恶意的，通常是由黑客在真实 Wi-Fi AP 附近环境中部署的高仿 AP 节点。之所以称之为高仿 AP 节点，是因为恶意 AP 与真实 AP 具有相同的服务集标识符(Service Set Identifier, SSID)，旨在隐瞒用户该 Wi-Fi AP 的真实性。钓鱼 Wi-Fi AP 往往具有较大的工作功率和较强的无线信号，以便终端设备选择 Wi-Fi AP 信号接入时取得竞争优势[4]。根据钓鱼 Wi-Fi 的攻击原理的不同，其攻击方式大致有三种。第一种，被动式攻击，钓鱼 Wi-Fi AP 等待终端设备接入并进行流量监听。第二种，主动式攻击，钓鱼 Wi-Fi AP 采取无线阻塞的攻击手段，迫使终端设备与真实 AP 断开并接入当前钓鱼 Wi-Fi AP, 之后再监听终端设备的数据流量[5]。第三种，Karma 攻击，这种攻击方式具有特定的应用条件—终端设备需开启无线网络的自动扫描功能[6]。开启无线网络自动扫描功能的终端设备会周期性地广播曾连接的 Wi-Fi AP 信息，以确认当前网络环境中是否存在历史连接的 Wi-Fi 热点。钓鱼 Wi-Fi AP 监听到这种特殊的报文后，给予对方虚假响应，欺骗终端设备曾经接入过该 Wi-Fi 热点，从而诱使终端设备的接入。除此之外，钓鱼 Wi-Fi 还有一种中间人的攻击方式，该方式无须仿造环境节点中存在的 Wi-Fi AP。钓鱼 Wi-Fi AP 通过 ARP 欺骗的方式成为真实 Wi-Fi AP 和终端设备之间的中间人，持续接收和转发双方流量，从而达到秘密监听用户流量的目的[7]。

1.2.2 移动应用通信现状调研

为了更好地研究 Android 移动终端的 Wi-Fi 安全接入技术，本课题首先对 Android 应用的通信安全现状展开调研。样本集由 89 个 360 应用市场的 Android 应用和 62 个 Google Play 应用市场的 Android 应用组成，国内外应用总数共计 151 个。样本集中应用的类型广泛，涉及系统安全、通讯社交、影音视听、新闻阅读、购物电商等十余种类，调研的具体工作介绍如下：

（1） 目的

调研目的是为了统计 Android 应用数据通信的加密情况和承载协议，以此分析 Android 应用数据通信的安全现状。

（2） 方法

在测试机上安装样本集中应用的 apk，用户试用应用软件，使用 Wireshark 工具抓取应用产生的数据流量并进行协议分析，观察并分析数据流量的加密情况和承载协议，并将应用的加密程度分成三类：明文应用（数据流量均未加

密）、部分加密应用（仅对涉及用户名、密码的流量进行加密）和全部加密应用（所有数据流量均加密）。

（3）分析

图 1-1 统计了样本集中所有应用的加密情况。从纵向整体来看，151 个 Android 应用包含明文应用 46 个，部分加密应用 67 个，全部加密应用 38 个，分别占样本数的 30.46%，44.37%和 25.17%。从横向对比来看，样本集中国内 360 应用市场仍存在 44.94%的明文应用，全部加密应用个数仅占 15.73%。而样本集中国外 Google Play 应用市场中，仅有 11.54%的明文应用。

图 1-2 统计了样本集中加密应用的承载协议情况。从整体情况来看，样本集中所有应用均采用 SSL 或 TLS 协议加密数据流量。统计结果表明 SSL 协议、TLSv1 协议和 TLSv2 协议分别占加密协议的 6%，59%，35%。

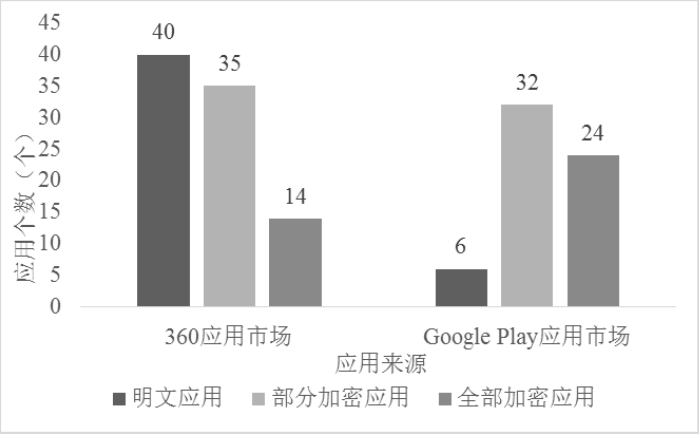


图 1-1 应用加密情况统计

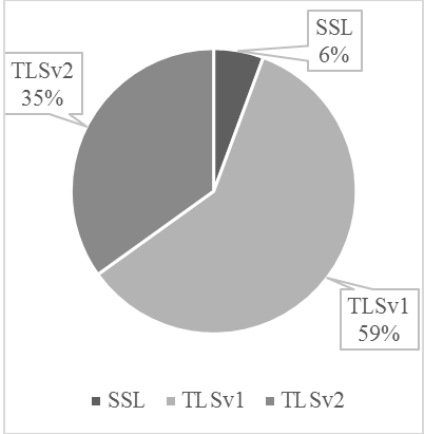


图 1-2 安全协议类型统计

（4）结论

1) 国内仍有相当大比例的 Android 应用使用明文通信，一旦接入 Wi-Fi 热点的移动终端遭到中间人攻击，这些应用的明文流量将直接暴露用户隐私。
2) 样本应用中，占绝大多数比例的是部分加密应用，这种方式看似保护了用户关键的隐私数据，但实际上仅仅保护了用户的用户名和密码，用户的访问记录依然是透明。一旦用户访问记录和用户个人信息被关联成功，用户的个人隐私又将处于不利的环境之中。
3) 样本应用中，所有涉及加密的流量均采用 SSL 或 TLS 协议进行传输。尽管 TLS 协议在理论研究中被证明是安全的，但在实际应用中并非是 100%安全的。相关研究已经证明 TLS 协议在实际应用中经常存在漏洞，攻击者一旦找到这些漏洞便可破解用户的数据流量[8-10]。因此，仅依靠应用开发者提供的 SSL 协议或者 TLS 协议并非能确保移动通信的安全。

1.2.3 网络安全接入技术研究

现有的网络安全接入技术主要包括虚拟专用网络(Virtual Private Network, VPN)技术、安全代理技术和匿名通信技术，三种技术的实现原理不尽相同，但在应用于终端设备时具有统一的目标：通过网络安全协议承载数据流量，对终端用户进行身份认证，对终端设备流量进行安全加密，保障终端设备安全接入互联网，保障终端用户的网络安全和数据隐私。表 1-2 对这三种网络安全接入技术进行了详细地对比总结。

表 1-2 网络安全接入技术对比表

网络安全接入技术	核心技术	通信方式	技术特点	应用
VPN	虚拟网卡技术 认证加密技术 路由转发技术 接入控制技术	客户端与服务端构建安全隧道，所有流量承载于安全协议，通过加密隧道转发和接收。	实现方式灵活 协议选择广泛 所有数据加密	OpenVPN ExpressVPN GlobalVPN
安全代理	认证加密技术 代理转发技术 负载均衡技术	客户端与目的地址通常使用单跳非直接连接的加密方式通信。	协议选择单一 网页代理较多 全局代理较少	Shadowsocks ProxyDroid
匿名通信	认证加密技术 重路由技术 组播广播技术	发送者与接收者使用多跳转发方式进行加密通信。	隐藏发送身份 隐藏接收身份 隐藏收发关系	Tor Orbot

（1）VPN 技术

VPN 是一种在公共网络线路中搭建的虚拟的，个人或群体专用的通信隧道的技术，因此 VPN 隧道具有虚拟性和专用性两个特征。虚拟性体现的是 VPN 隧道并不是真正意义上人为搭建的网络线路，而专用性则体现 VPN 隧道使用安全加密的通信协议传输网络数据。VPN 由客户端和服务端两端组成，通信双方通过虚拟网卡技术构建虚拟安全隧道。通过配置系统路由，客户端的所有发送流量导入虚拟网卡进行加密、压缩、二次协议封装后经过 VPN 隧道发送到服务端，服务端将接收的流量进行协议解封、解压、解密后，发往目的地址。VPN 技术的优势在于实现方式灵活，协议选择广泛，加密压缩方式也可以用户自行选择，并且能够保障客户端的所有数据流量的安全[11]。

（2）安全代理技术

代理技术和 VPN 技术在宏观组成上有一定的相似性，都由客户端和服务端两部分组成。但代理技术的实现方式完全不同，无须构造通信隧道，直接使用报文转发技术将客户端请求移至服务端完成，实现客户端和目的地址的间接访问。安全代理又名加密代理，相比于基于 HTTP 协议实现的透明代理，加密代理具有更好的隐私保护能力和更强的抗干扰能力。加密代理的通信协议选择

余地不多，常见的加密代理有 HTTPS 和 SOCKS 协议可供选择[12,13]。代理技术的类型可以分为局部代理和全局代理。局部代理最为常见，通常用于浏览器的网页代理，以插件或者配置文件的形式加载客户端的配置，仅对浏览器产生的流量进行代理转发；而全局代理与 VPN 技术的功能比较相似，将客户端所有的数据流量都通过代理进行转发接收，其中比较有代表性的全局代理就是 Shadowsocks 代理。安全代理技术隐藏了用户的真实 IP，并提供了用户流量的加密，是一种可靠安全的网络接入方法。

（3）匿名通信技术

匿名通信技术可以理解为是代理技术在问题规模上的升级，即发送者与接受者的数据通信需要经过多台中继服务端的转发控制，其目的在于隐藏数据发送者的身份，隐藏数据接收方的身份和隐藏数据收发双方的通信关系，使得攻击者无法通过网络嗅探对数据包进行溯源，无法对用户的通信进行持续跟踪。较为有代表性的匿名通信项目是开源项目洋葱路由器（The Onion Router, Tor），该项目的目的已由最初保护军事情报通信发展为保护用户的网络隐私。一个简单的 Tor 网络由一个接入节点、多个中继节点和一个出口节点组成。发送方通过使用 Tor 浏览器接入 Tor 网络，最终与接收者通信。发送者的数据包经 Tor 浏览器的本地代理加密后发往 Tor 网络的接入节点，经中继节点的层层传递，最终在出口节点解密后发往接收方[14]。目前，Tor 浏览器和其他相关工具已经在 PC 终端和 Android 移动终端上广泛应用，以局部加密代理的形式保护用户的网页访问隐私。

从技术实现的角度来看，VPN 技术比安全代理技术更加灵活简单，安全协议的选择和加密算法的选择也更为广泛；从通信效率的角度来看，VPN 技术没有匿名访问技术那么复杂的多层路由选择，而是通信双方直接建立隧道通信，更加直接稳定；从适用范畴的角度来看，VPN 技术能够通过安全隧道保护移动终端的所有数据流量，更适用于保护移动终端 Wi-Fi 接入的所有流量，而大多数安全代理技术和匿名通信技术侧重于局部流量的保护。因此，本节将对移动 VPN（Mobile VPN, MVPN）技术现状进一步研究和分析，以保障移动终端在不可信的 Wi-Fi 环境中安全快速地接入互联网。

1.2.4 MVPN 技术研究现状分析

MVPN 技术源于传统 VPN 技术，因此 MVPN 借鉴了传统 VPN 的核心技术，包括虚拟隧道技术、身份认证技术、密钥交换技术和数据加密技术。其中，虚拟隧道技术的实现大多基于虚拟网卡的构建和系统路由的配置；而身份认证

技术、密钥交换技术和数据加密技术的实现则依赖于一整套安全通信协议的体系框架。二者的关系是相辅相成、缺一不可的。前者是后者的基础，后者为前者提供安全保证，所以现有 MVPN 的核心技术也可以概括为虚拟隧道技术和安全协议技术两部分。

当前 MVPN 的研究工作主要是围绕这两点展开的，安全协议技术的研究工作偏重于学术理论研究，致力于对现有安全协议的改进和优化；虚拟隧道技术的研究工作则偏重于工程实践应用，实际应用过程中通常以传统 VPN 协议作为虚拟隧道的安全保证。无论是在学术科研领域还是在工程实践领域，MVPN 技术均有广泛的研究与应用。因此，下面将从将研究性工作和应用性工作两个角度对 MVPN 的研究现状进行介绍。

（1）研究性工作

在学术科研领域，MVPN 的研究性工作主要集中在对其通信模型的研究和通信协议的改进两个方面。

在通信模型的研究方面，Shu 等人[15]提出一种基于 SOCKS 协议和 SSL 协议相结合的 MVPN 通信模型。该模型将 MVPN 通信隧道的网络层和应用层分离，网络层采用 SOCKS 端口转发技术接收或发送数据报文，应用层使用 SSL 协议对数据进行安全加密，实验表明该模型安全可靠且简单易用。Uskov [16]对基于 IPSec 协议的 MVPN 技术进行长期研究，提出了 MVPN 概念模型和设计方案，为 MVPN 系统管理员和用户提供细致的 IPSec MVPN 配置方案和使用方案。另外，Uskov[17] 还对 IPSec 密钥交换阶段的认证算法和数据传输阶段的加密算法进行全面的比对测试，得出了不同算法组合对 IPSec 性能影响，并给出了不同背景下，IPSec MVPN 在通信安全和传输效率之间的折衷选择方案。Liyanage 和 Gurtov[18]提出了分别基于 IKEv2 协议和基于主机标识协议(Host Identity Protocol, HIP)的两种 IPSec MVPN 通信模型以满足长期演进(Long Term Evolution，LTE)网络的安全需求。Lakbabi 等人[19]和 Mao 等人[20]都对基于 IPSec 协议和基于 SSL 协议的两种不同通信模型的 MVPN 进行研究，对比分析了两种 MVPN 在通信特征、密钥管理、身份认证、接入控制等多方面的不同，最终给出了不同应用场景下 MVPN 的选择方案。

在通信协议改进方面，Yu[21]等人提出一种改进的预共享密钥（Pre-Shared Key, PSK）的密钥交换方式以简化 IPSec 密钥交换协议(Internet Exchange Key, IKE)的通信流程，实验表明简化的 IKE 协议能够保证安全的密钥交换并提高通信效率。为了提高实时网络语音通话服务(Voice over Internet Protocol, VoIP)传输效率，Kim 和 Yoo[22]提出一种基于 PPTP 接入集中器(PPTP Access

Concentrator , PAC) 和 PPTP 网络服务器(PPTP network server, PNS)的 PPTP MVPN 架设方案。实验证明这种 MVPN 的部署方案能够降低客户端计算开销和传输负载，可以满足 VoIP 服务实时性需求。Zuquete 等人[23]对开源 OpenVPN 的传输协议字段进行修改，增加了会话 ID 字段，以会话的概念维持通信双方的连接，以保证 MVPN 在网络连接发生改变时维持通信连接的稳定性。Oya 等人[24]研究发现 IPSec 的移动互联网密钥交换(IKEv2 Mobility and Multihoming Protocol, MOBIKE)并不能为大范围移动的移动终端提供 VPN 连接维持，因此提出了优化路由的 MOBIKE 协议以增加客户端对服务端的最优选路支持，实验结果表明改进后的 MOBIKE 能提供移动终端更好的稳定性。Liu 等人[25]针对应用的实时传输需求提出一种基于 SIP 协议的 MVPN 解决方案，实验证明这种 SIP-MVPN 比国际互联网工程任务组(Internet Engineering Task Force, IETF)提出的 MVPN 解决方案具有更低的传输时延，因此适用于实时应用的数据传输。

（2）应用性工作

在工程实践领域，MVPN 应用性工作涉及到移动终端的安全通信、远程办公、智能家居、流量监测、恶意检测等各个领域。

Yu 等人[26]实现了基于 SSL MVPN 的移动安全接入系统，使用 SSL 通道保障移动终端安全接入互联网。针对企业远程办公的安全性问题，Hong 和 Kim[27]开发了智能手机的 PPTP MVPN 客户端应用以保证远程接入企业内网的数据安全。为了保障智能家居中无线传感器的数据在互联网中的安全传输，Liao 等人[28]实现了智能家庭的 MVPN 控制传输系统。Kilinc 等人[29]将 MVPN 技术应用于构建移动终端的虚拟防火墙，移动云提供云端的数据分析与检测，这种 MVPN 技术与移动云技术有机结合方案被认为是一种增强型的移动安全模型。Anastasia 等人[30]使用 MVPN 实时监控移动设备的网络流量以检测移动终端的隐私数据是否泄漏，保障用户的通信安全和个人隐私。Choi 等人[31]将 MVPN 技术应用于移动僵尸网络的检测与防御，将 MVPN 作为获取移动终端所有流量的工具，并提取流量特征进一步分析僵尸网络的特征。

1.2.5 相关技术现状总结与分析

恶意 Wi-Fi 攻击方式层出不穷，但最终目的都是为了监听终端用户的数据流量，窃取用户的隐私数据。因此，如果能够保证移动终端的数据流量均以安全可靠的通信方式进行传输，便无需担心黑客的网络嗅探。但现实的情况是：在调研的 Android 应用样本集中，有 30.46%的应用采用明文传输数据，有

44.37%的应用仅对部分数据加密，仅有 25.17%的应用对全部数据加密。由于所有涉及加密的流量均采用 SSL 或 TLS 协议进行传输，而 TLS 协议在实际应用中一旦存在漏洞便容易遭到攻击，所以仅依靠 SSL/TLS 保障数据的安全性是不够的。为了防止黑客的网络嗅探，保护移动用户的数据隐私，本节进一步研究了网络安全接入技术，分析了 VPN 技术、安全代理技术和匿名通信技术的通信方式和技术特点。经过对比分析，VPN 技术由于其安全隧道的通信特征和灵活多变的技术实现，更适用于保障端到端的安全通信。

因此，本课题选择 MVPN 技术作为移动终端 Wi-Fi 安全接入的关键技术展开研究。纵观国内外学者的研究内容，MVPN 研究与应用的重点聚焦在通信模型设计、传输协议改进和应用场景延伸三个方面。其中，协议模型设计和应用场景延伸方面的研究成果相对突出，并且展现出以理论研究带动实际应用态势。相比之下，传输协议改进方面的工作对 MVPN 性能优化效果相对不足且无典型应用实例，所以 MVPN 的传输性能问题有待进一步研究。除了当前 MVPN 的主流研究内容，MVPN 在广泛的实际应用中暴露出更多的问题，如 MVPN 客户端的稳定性不足、占用系统资源多、耗电量大等问题接二连三的被 MVPN 用户抛出，这些问题也值得进一步探索。

综上所述，现有 MVPN 技术可以视为是一种满足移动终端安全通信需求的传统 VPN 的“优化平行移植”。优化移植指的是现有工作对传统 VPN 通信模型和通信协议改进优化，提升传统 VPN 协议在 MVPN 通信中的安全性和交互性，降低计算开销和传输时延，以更好地适用于移动终端。平行移植的概念则是指出现有 MVPN 的不足，只是横向的移植优化了 VPN 技术，并未纵向的考虑到移动终端有限的系统资源和移动终端所处的开放式网络环境，导致了 MVPN 通信效率和稳定性一直不尽如人意。

1.3 论文研究内容和结构安排

本节概括地介绍了论文研究内容，并给出全文的结构安排。

1.3.1 论文研究内容

基于课题相关技术现状的总结和分析，本课题选取了 MVPN 技术作为 Android 移动终端 Wi-Fi 安全接入的关键技术，主要研究 MVPN 技术的三个方面：虚拟隧道技术、安全通信协议和通信保障模型，以解决现有 MVPN 技术在数据传输速度和连接稳定性上的不足。虚拟隧道技术是 MVPN 技术的基础，安全通信协议的研究目的是在保障虚拟隧道的安全性的前提下，尽可能提升

MVPN的数据传输速度。通信保障模型的研究目的则是进一步提高安全隧道的稳定性，更好地提高 MVPN 客户端的可用性和友好性。论文的整体研究框架如图 1-3 所示。

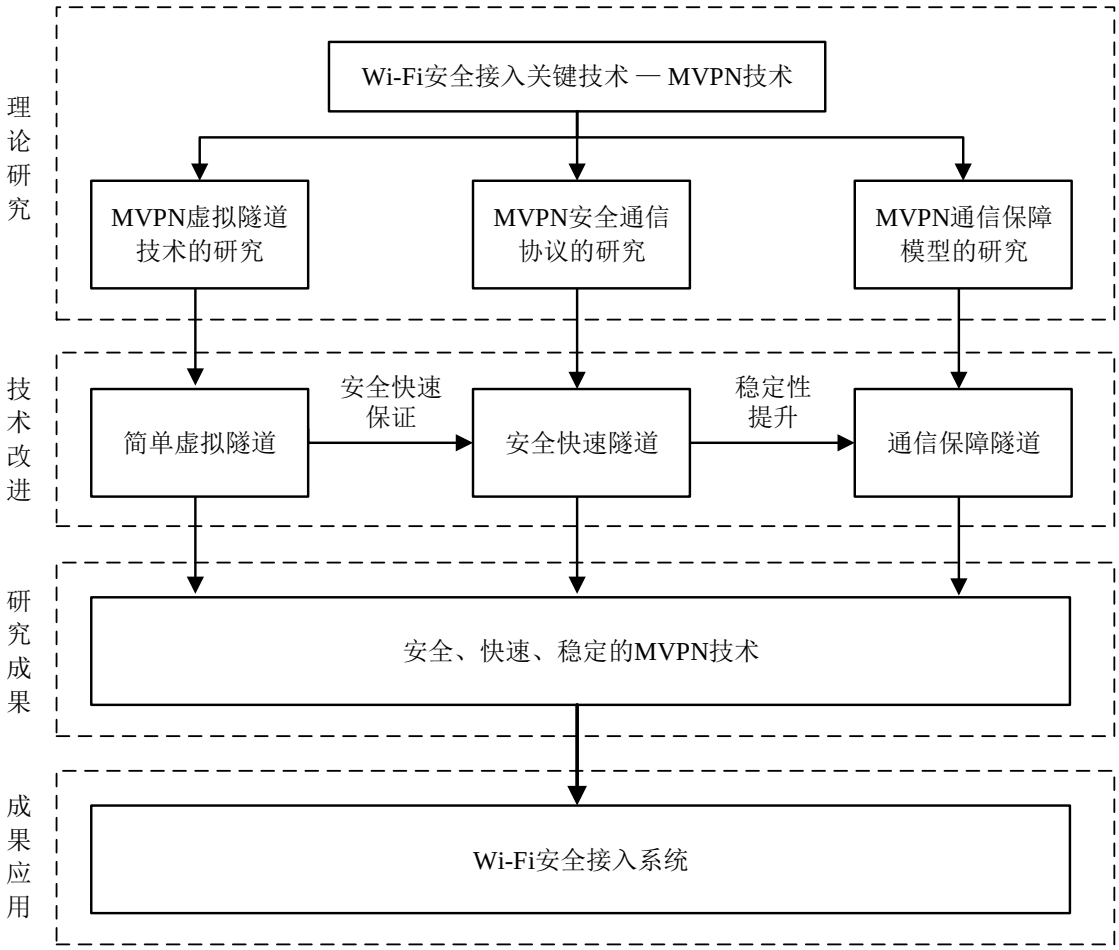


图 1-3 论文整体研究框架

因此，本文的研究内容主要包括以下四点：

第一，对 Android 平台 MVPN 的虚拟隧道技术展开研究。较其他的网络安全接入技术，虚拟隧道技术是 VPN 技术的最大特色，同时也是 VPN 技术的基础，因此本课题将其放置于首要位置。MVPN 隧道技术的研究重点在于掌握 Android 平台下隧道的创建方式和隧道的工作方式，并解决 MVPN 如何在客户端的 Android 操作系统和服务端的 Linux 操作系统中分别创建虚拟隧道并实现二者的互连互通。

第二，对 MVPN 的安全通信协议展开研究。在实现 Android MVPN 客户端与 Linux VPN 服务端的隧道通信后，隧道的安全通信是本课题的一个研究重点。因此，该项研究工作将对现有 MVPN 通信协议进行对比分析，指出存在

问题并提出一种快速传输隧道协议。该协议是一种基于对称加密体制的分层通信协议，将认证密钥交换过程与认证加密传输过程分离，前者由应用层标准 SSL 安全通道保障握手安全，后者由网络层私有 MVPN 加密隧道保障传输安全并提高通信效率。最后本文基于快速传输隧道协议设计了快速传输 MVPN，测试了该 MVPN 的性能表现，并给出了该 MVPN 的安全评价。

第三，对 MVPN 通信连接的稳定性展开研究。在实现 Android MVPN 客户端与 Linux VPN 服务端的安全通信后，通信连接的稳定性是本课题的另一个研究重点。针对现有 MVPN 在移动网络环境中普遍存在稳定性不足的现象，分析 MVPN 与移动网络存在不相适应的根本原因，并提出一种移动网络环境下 MVPN 通信保障机制。随后使用形式化语言对其基本原理进行抽象定义，使用有限状态机模型对抽象概念进行数学建模，最后提出了一种通用的、可扩展的 MVPN 通信保障模型，设计了通信保障 MVPN 并对该 MVPN 的稳定性进行测试。

第四，研究 MVPN 技术与 Wi-Fi 安全接入的有机结合方案。该项工作是对前三项 MVPN 技术研究成果的具体应用，给出基于 MVPN 技术的移动终端 Wi-Fi 安全接入系统的整体架构设计、核心组件设计和具体模块设计方案，最终实现该原型系统并给出该系统的测试结果与测试评价。

综上所述，本课题对 Wi-Fi 安全接入的关键技术—MVPN 技术的研究采用的是一种递进式和迭代式的研究方法，即从 MVPN 的隧道基础开始研究，到 MVPN 安全协议的研究，再到 MVPN 稳定性的研究，最后将上述安全、快速、稳定的 MVPN 技术的研究成果应用于移动终端 Wi-Fi 安全接入需求。

1.3.2 论文结构安排

根据本文的研究内容和方法，将论文组织结构安排如图 1-4 所示，具体内容描述如下。

第 1 章：绪论，介绍课题研究背景和意义，介绍相关技术的研究现状，总结分析现有问题和值得探究的研究内容，给出本文的研究内容和组织结构。

第 2 章：MVPN 虚拟隧道技术的研究，概述 MVPN 的虚拟隧道技术，研究 Android 平台虚拟隧道的创建方式和工作方式，最后实现简单隧道 MVPN 的 Android 客户端和 Linux 服务端并给出功能测试结果。

第 3 章：MVPN 安全通信协议的研究，分析现有 MVPN 协议的存在问题，结合安全协议的设计目标给出协议设计的新思路，给出基于该思路的新协议的设计方案。最后介绍基于该协议实现快速传输 MVPN 原型，并通过理论分析

和性能测试证明该协议具有较好的安全性和传输效率。

第 4 章：MVPN 通信保障模型的研究，分析 MVPN 通信连接稳定性不足的原因，并结合 MVPN 的应用现状给出一种通用的 MVPN 通信保障方法。随后，给出基于该方法的 MVPN 通信保障模型设计和原型实现，并通过稳定性测试实验验证该模型的稳定性。

第 5 章：原型系统设计与实现，介绍移动终端 Wi-Fi 安全接入系统的设计与实现，实验证明该系统能够保障移动终端的通信安全和较好的系统性能。

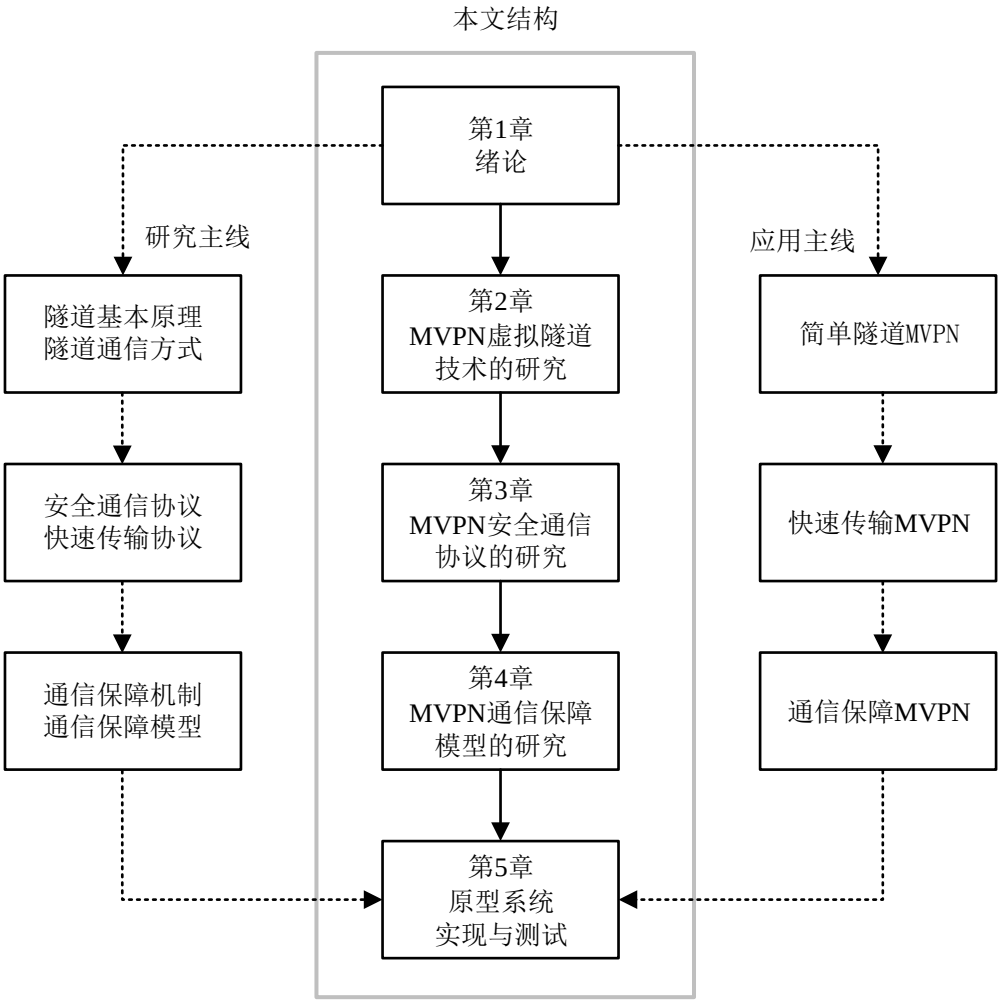


图 1-4 本文组织结构图

第2章 MVPN 虚拟隧道技术的研究

本章对 MVPN 技术的基础—虚拟隧道技术展开研究，首先总结分析了现有的 MVPN 虚拟隧道技术。接下来对 Android 平台下隧道创建方式和工作方式进行研究，最后介绍基于 Android API 实现的简单隧道 MVPN，并对其进行测试以验证虚拟隧道功能。

2.1 MVPN 虚拟隧道技术概述

本节首先分析了两种不同类型的 MVPN 隧道技术的利弊，随后选取其中适用于本课题研究范畴的一种 MVPN 隧道技术进一步研究。

2.1.1 分类标准

根据 MVPN 隧道的创建者不同，可以将 MVPN 隧道的类型分成主动创建隧道和被动创建隧道 [32]。主动建立隧道指工作在移动终端的 MVPN 客户端直接向 VPN 服务端的安全网关发起通信请求，所以 MVPN 客户端是隧道的创建者，移动终端是隧道的起点，安全网关是隧道的终点。这种隧道的建立方式通常用于个人和企业级别的安全接入和可靠通信。被动建立隧道指移动终端本身不参与隧道的创建，移动终端只与接入网关进行身份认证，接着由部署在接入网关的 VPN 客户端与出口网关的 VPN 服务器创建 VPN 通信隧道，所以运行在接入网关的 VPN 客户端是隧道的创建者，接入网关是隧道的起点，出口网关是隧道的终点。这种隧道的建立方式常应用于运营商级别的远程接入和安全通信。

这两种不同 MVPN 隧道的实现各有利弊。前者以移动终端为起点，安全网关为终点的安全隧道保证了两者端到端的安全通信，保障所有终端设备数据流量的安全。但与此同时，MVPN 客户端增加了移动终端处理数据包封装和加密解密的计算开销，增加了移动终端的电量消耗，并在一定程度上降低移动终端的性能。相比而言，后者的实现方式对于移动终端无需增加额外的计算开销，只需要与接入网关进行身份认证即可，所以移动终端的性能和电量都不会受到影响。但在通信过程中，由于移动终端与接入网关之间的通信链路并未受到 VPN 安全隧道的保护，因此可能会遭到中间人攻击。

本课题所研究的 Wi-Fi 安全接入问题需要借助 MVPN 的安全隧道技术保障端到端的可靠通信，因此属于主动创建隧道的 MVPN。接下来，本节将详细介

绍该类型隧道的基本原理、评价指标和应用背景。

2.1.2 基本原理

MVPN隧道的主动创建需用通过建立虚拟网卡和配置系统路由实现。虚拟网卡由Tun/Tap设备和字符驱动设备两部分组成。通过配置系统路由，Tun/Tap设备直接与TCP/IP协议栈进行交互，负责从协议栈读取或写入数据包。Tun和Tap 设备工作原理相同，只是处理的数据包类型不同。Tun 设备负责网络层数据包的处理而 Tap 设备负责链路层数据包的处理。字符驱动设备用于实现数据包在系统用户态和内核态之间的转换，包括接收数据时将数据包从 MVPN 进程写入 Tun/Tap 设备，和发送数据时将数据包从 Tap/Tap 设备写回 MVPN 进程。图 2-1 描绘了 MVPN 虚拟网卡的工作原理。

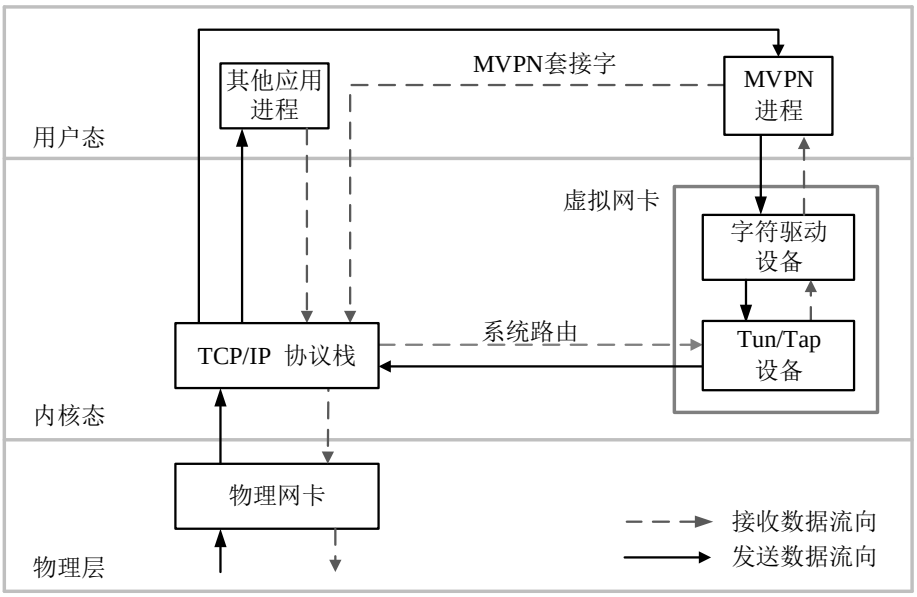


图 2-1 虚拟网卡的工作原理图

接收数据包时，MVPN进程监听数据包的到来，在解密解封数据包后，通过字符驱动设备将原始数据包写入 Tun/Tap 设备。该设备读取数据包后根据系统路由将数据包发往目标应用进程。发送数据包时，根据系统路由的配置，应用进程的原始数据包将写入 Tun/Tap 设备，字符驱动设备将 Tun/Tap 设备处的数据包读取到用户态的 MVPN 进程。MVPN 进程对数据包加密、封装，并使用受保护的 MVPN 套接字将数据包发往物理网卡。

2.1.3 评价指标

传统 VPN 隧道通常是从通信安全和通信速度两个方面进行评价，MVPN

隧道在考虑安全、快速两个指标的同时，需要兼顾移动终端的移动性、网络连接的易变性和电量资源受限的特点，因此稳定性和耗电量也被纳入 MVPN 隧道的评价指标。因此，MVPN隧道的评价指标包括通信安全、传输速度、稳定性和耗电量四个方面。

（1） 通信安全

MVPN 隧道应当基于安全的通信协议实现，保证 VPN 连接从创建到最终断开的整个阶段，提供安全的身份认证、密钥交换和数据加解密，满足传输过程中数据的机密性和完整性。

（2） 传输速度

MVPN隧道应当具有较低的延时，并尽可能的降低对移动终端网络带宽的影响，尽可能的保证移动用户在使用 MVPN 时的网络接入速度和上传速度与未使用 MVPN 时保持一致，甚至提供更优的连网速度。

（3） 稳定性

移动终端的移动性和网络连接的易变性对 MVPN 的稳定性提出了更大的挑战。MVPN隧道应当具备连接维持和移动支持的能力。当移动终端的网络连接发生切换或中断时，MVPN应当能够感知网络接入的变化并采取相应的措施维持连接或暂停连接；当移动终端的网络连接恢复时，MVPN应当能够感知网络的恢复并恢复 VPN 连接。

（4） 耗电量

移动终端的电量资源非常宝贵，MVPN隧道增加了数据包封装和数据加解密的计算开销，相应地增加了移动终端的电量消耗。为了尽可能降低电耗，通常的解决方案有两种。一是采用低开销的加密算法，这就需要 MVPN 的加密算法在加密强度和加密开销上折衷选择；二是只对移动终端的敏感数据进行加密，这就需要对 MVPN 隧道流量先分类后加密。与此同时，如何设定数据敏感的分类标准，如何快速有效的进行流量分类又成为一个棘手的问题。

2.1.4 应用场景

MVPN隧道技术在工作、生活、娱乐等各方面具有广泛的应用，根据不同需求主要可以分为四类应用背景，不同应用背景的网络拓扑结构如图 2-2 所示。

第一类，用于安全访问外网资源。MVPN的客户端和服务端分别部署在移动终端和安全代理服务器上，移动用户通过安全代理类型的 VPN 服务接入互联网目标资源。第二类，用于端到端的安全通信。MVPN提供移动用户安全通信服务，包括文字、图片等“轻媒体”聊天服务和 VoIP 服务、视频等“重媒

体”通信服务。第三类，用于远程访问内网资源。这种方式的服务通常在企业比较常见，VPN 服务端部署在企业的出口网关，在外地的企业员工通过 MVPN 接入内网资源以满足远程移动办公的需求。第四类，用于安全接入分布式私有云。如今许多大型互联网公司都提供公有云服务，许多的云用户群体通常拥有多个公有云提供商的云资源，图 2-2 给出了一种基于公有云的私有云构建方式，VPN 是这种构建方式的核心技术。借助这种方式，移动用户可以安全有效的管理、接入、配置私有云。

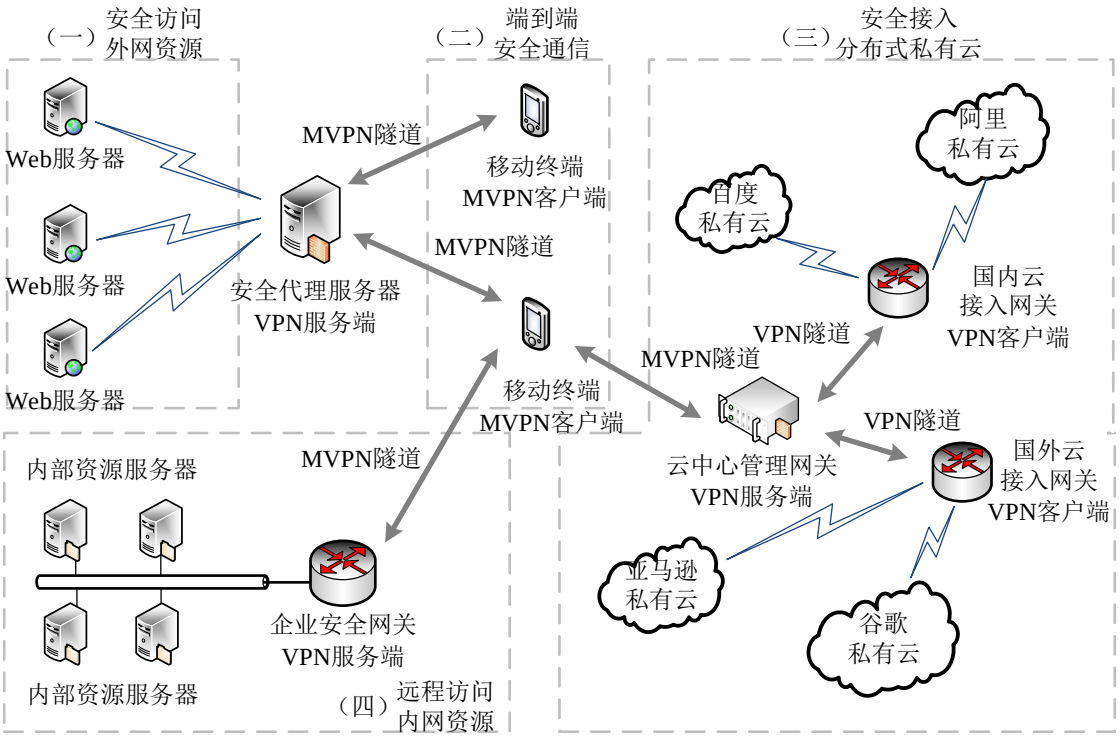


图 2-2 MVPN 不同应用背景的网络拓扑示意图

2.2 Android MVPN 虚拟隧道技术介绍

Android MVPN 隧道技术主要由隧道创建方式、工作方式和其他实现细节三部分组成。隧道的创建和工作方式是实现其他细节的基础，主要涉及的问题是如何在 Android 操作系统的底层 Linux 内核中创建虚拟网卡、配置系统路由并获取终端的所有原始流量。整个过程与本文 2.1.2 节介绍的虚拟网卡工作原理基本一致。除此之外，隧道的工作方式还涉及到隧道在 Android MVPN 客户端的运行机制。隧道的实现细节则关系到隧道协议的通信安全、传输速度和稳定性。本节主要关注隧道的创建和其工作方式，关于隧道的实现细节将在本文后续章节进一步研究。

2.2.1 隧道创建方式

Android 平台常见的隧道创建方式主要有四种。第一种方式是调用 Android Shell 相关的 API，在底层 Linux 操作系统的内核空间直接创建 Tun/Tap 设备，并配置系统路由以实现虚拟网卡相应的功能。第二种方式是调用 Android 非官方 API 的 Profile 类实现虚拟网卡的创建，Profile 类提供的函数接口实际上是对第一种创建方式的普遍封装，因此这两种隧道建立方式实质上是一样的。第三种则是通过 Android JNI（Java Native Interface, Java 本地接口）技术，使用 Java 语言直接调用 C 或 C++实现的 VPN 源程序，在底层 Linux 系统的内核空间创建虚拟网卡。这三种隧道的创建方式早在 Android 操作系统 2.0 版本被提出使用，以满足 Android 移动用户的 MVPN 使用需求。但借助这三种方式实现的 MVPN 客户端需要以获得 Android 移动终端的 Root 权限为前提，因此在早期 Android MVPN 的用户量一直不大，Android MVPN 技术发展缓慢。随着 Android 操作系统 4.0 版本的发布，Google 公司首次提供了 Android MVPN 的官方 API — VpnService 类, 开发者可以调用 VpnService 类在用户空间创建虚拟网卡，设置系统路由、DNS、MTU 等网络参数。因此，开发者可以将开发的重点放在隧道协议的实现，用户也不必再为了使用 MVPN 而 Root 手机，一切都变得简单方便。所以基于 VpnService API 实现的 MVPN 逐渐成为 Android 平台主流的 MVPN 解决方案。

2.2.2 隧道工作方式

隧道经配置和创建后开始工作，Android 系统通知栏会显示实时流经 MVPN 隧道的数据包信息。使用 adb shell 脚本工具连接 Android 移动终端，输入相应的 Linux 指令查看便能发现系统网卡设备增加了 Tun/Tap 设备，系统路由表增加了 Tun 设备的一条规则，分别如图 2-3 和 2-4 所示。通过配置一条指向 tun0 设备的 0.0.0.0/1 系统路由，虚拟网卡能够成功获取移动终端所有流量。

```
shell@android:/ $ netcfg
lo       UP                                  127.0.0.1/8   0x00000049 00:00:00:00:00:00
sit0     DOWN                                  0.0.0.0/0   0x00000080 00:00:00:00:00:00
tun0     UP                                 10.8.0.14/32   0x00000051 00:00:00:00:00:00
p2p0     UP                                    0.0.0.0/0   0x00001003 ae:f7:f3:a9:8d:ba
rmnet_usb0 DOWN                                0.0.0.0/0   0x00000000 00:00:00:00:00:00
rmnet_usb2 DOWN                                0.0.0.0/0   0x00000000 00:00:00:00:00:00
rmnet_usb1 DOWN                                0.0.0.0/0   0x00000000 00:00:00:00:00:00
rmnet_usb3 DOWN                                0.0.0.0/0   0x00000000 00:00:00:00:00:00
wlan0    UP                           192.168.31.103/24   0x00001043 ac:f7:f3:a9:8d:ba
```

图 2-3 隧道建立后 Android 终端的网卡设备变化

```
shell@android:/ $ ip route
0.0.0.0/1 dev tun0  scope link
default via 192.168.31.1 dev wlan0
128.0.0.0/1 dev tun0  scope link
192.168.31.0/24 dev wlan0  proto kernel  scope link  src 192.168.31.103  metric 318
192.168.31.1 dev wlan0  scope link
```

图 2-4 隧道建立后 Android 终端的路由表变化

VpnService 类的实现继承了 Android 四大核心组件之一的 Service，所以具有 Service 组件具有较高优先级的特性。不同于工作在 Android UI 前端的 Activity 进程，Service 进程工作在系统的后台。同样是工作在用户空间的 Activity 进程和 Service 进程， Service 进程在系统中拥有更高的服务优先级。当系统出现内存不足等异常情况时，Android 运行机制会优先杀掉多余的 Activity 进程，为 Service 进程保留足够的系统资源。即便 Service 进程被意外杀死，只要开发者将 Service 进程配置成 START_STICKY 模式，Service 进程就可以在程序剩余资源允许的情况下重启进程。因此，基于 VpnService 创建的 MVPN 隧道能够保障在非用户旨意的应用异常退出后自动恢复。

2.3 Android 简单隧道 MVPN 的实现与测试

通过对 VpnService API 的学习和研究，我们首先实现了 Android 平台的 MVPN Demo — 简单隧道 MVPN（Simple Tunnel MVPN, ST-MVPN）。简单隧道表现在三个方面：第一，该 MVPN 的通信流程简单，只涉及会话初始化、密钥交换、DHCP、网络配置和隧道传输五个阶段；第二，该 MVPN 的通信协议简单，只是对虚拟网卡获取的原始 IP 层数据包进行异或加密后，使用 UDP 协议的固定端口传输数据。第三，应用使用简单，用户无需进行任何隧道配置便可以一键连接通信。ST-MVPN 客户端使用 Java 语言实现，以移动应用的形式部署在 Android 移动终端上。ST-MVPN 服务端使用 C 语言实现，以软件服务的形式部署在 Linux 服务器上。

2.3.1 隧道通信流程

ST-MVPN 通信双方的交互流程分为五个阶段。第一阶段，ST-MVPN 客户端发出会话初始化请求报文，服务端收到请求报文后给予响应以示服务器工作正常。第二阶段，客户端生成 XOR 加密密钥并发送至服务端，服务端接收后给予响应表示收到密钥。第三阶段，客户端发出 DHCP 请求，服务端收到 DHCP 请求后，从虚拟 IP 的缓冲池中取一个空闲 IP 并将其通过响应报文发送给客户端。第四阶段，客户端进行通信前最后的网络配置，创建虚拟网卡、配

置路由信息、DNS、MTU 等基本网络参数。第五阶段，通信双方使用 MVPN 加密隧道通信。整个通信过程如图 2-5 所示。

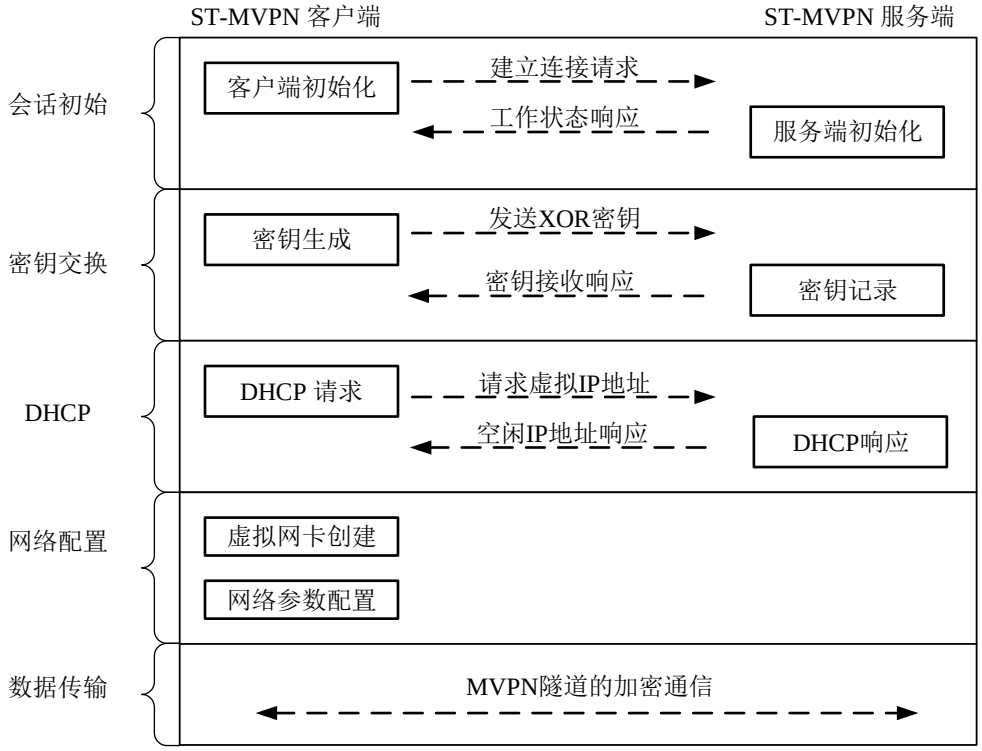


图 2-5 ST-MVPN 通信流程示意图

2.3.2 隧道报文格式

ST-MVPN 提供代理模式的 VPN 服务，即将 ST-MVPN 服务端作为安全通信的第三方，接收和响应客户端的所有加密流量并提供客户端间接的数据请求和数据响应。客户端隧道使用简单的私有协议与服务端加密通信。图 2-6 介绍了客户端发送数据包的整个过程。客户端发送数据时，虚拟网卡通过配置的系统路由获取移动终端的所有数据包，根据 VpnService API 提供的上层接口，所有的数据包均是 IP 层数据包。数据包经客户端 MVPN 进程加密后，通过 UDP 协议再次封装发往服务端的固定端口。服务端进程接收到数据包后，解析 UDP 协议头部信息后解密密文负载，根据解密后原始报文的 IP 层头部信息的目标地址采取不同处理方式。如果原始报文的目标 IP 地址就是服务器，那么整个发送过程到此为止；否则，服务端对原始报文的源 IP 地址进行 SNAT 地址映射后，将其发往该报文的目的地址。客户端接收数据包的过程是该过程的逆过程，因此这里不再赘述。

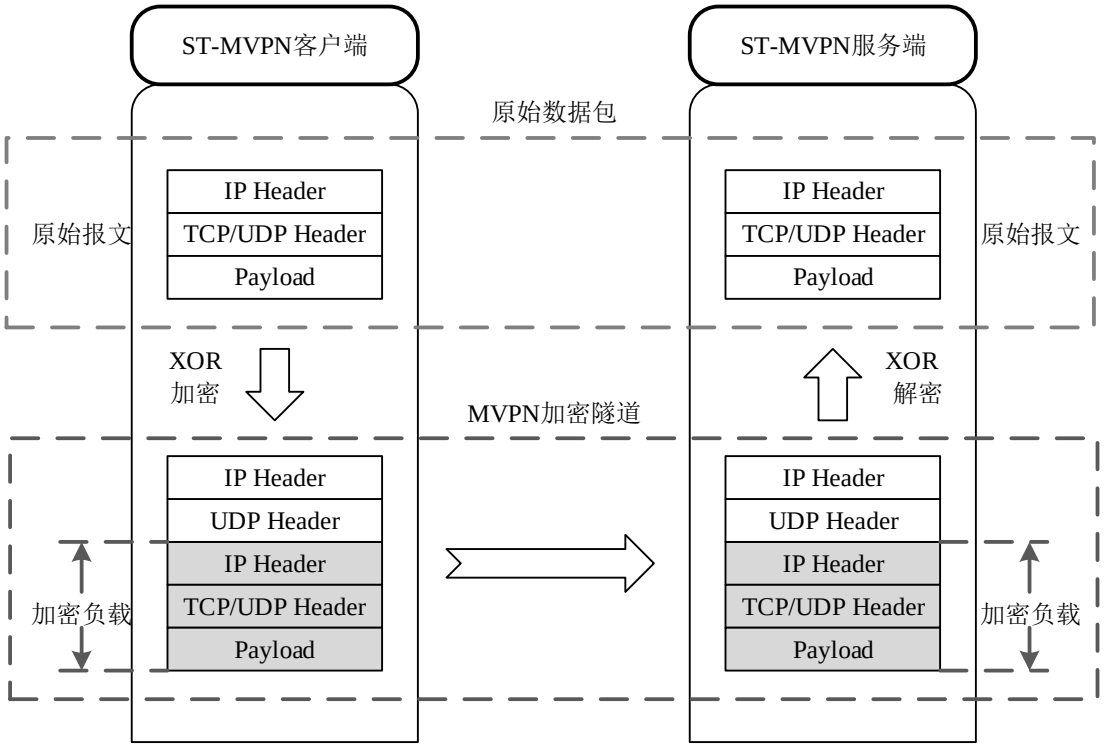


图 2-6 ST-MVPN 报文格式及报文发送流程示意图

2.3.3 隧道功能测试

测试涉及的硬件设备包括一台部署 ST-MVPN 客户端应用的 Android 移动终端和一台部署了 ST-MVPN 服务端程序的 Linux 操作系统的 PC 机。实验涉及的硬件设备配置信息如表 2-1 所示。

表 2-1 MVPN 稳定性实验的硬件设备配置表

设备类型	配置参数	配置描述
移动终端	设备型号	MI 2S
	CPU 型号	Snapdragon APQ8064 Pro 1.7GHz
	RAM 容量	2G
	操作系统	Android 4.1.1
PC 机	CPU 型号	Intel(R) Core(TM) i5-4570 CPU @ 3.20GHz
	内存容量	4G
	操作系统	Ubuntu 14.04.4 LTS

隧道功能测试在局域网进行，Android 移动终端通过 Wi-Fi 热点接入互联网。ST-MVPN 客户端连接服务端成功后，移动终端使用浏览器访问网页正常。图 2-7 是通过 adb logcat 命令导出的 ST-MVPN 客户端流量日志的截图，数据包外层的源 IP 地址已修改为 10.8.0.0/24 的 VPN 虚拟网段的地址，从使用现象和系统日志都表明 ST-MVPN 的隧道功能正常。

```
I/SimpleTunnelMVpnService(14879): IP : 10.8.0.5->119.188.110.15 Protocol=6,HLen=20
I/SimpleTunnelMVpnService(14879): TCP : PSH 17664->40 389824512:1074148284
I/SimpleTunnelMVpnService(14879): IP : 71.140.94.63->58.56.48.53 Protocol=54,HLen=20
I/SimpleTunnelMVpnService(14879): IP : 10.8.0.5->119.188.110.15 Protocol=6,HLen=20
I/SimpleTunnelMVpnService(14879): TCP : PSH 17664->52 389890048:1074148271
I/SimpleTunnelMVpnService(14879): IP : 71.140.94.63->58.56.48.53 Protocol=54,HLen=20
I/SimpleTunnelMVpnService(14879): IP : 10.8.0.5->119.188.110.15 Protocol=6,HLen=20
I/SimpleTunnelMVpnService(14879): TCP : PSH 17664->40 389955584:1074148282
I/SimpleTunnelMVpnService(14879): IP : 71.140.94.63->58.56.48.53 Protocol=54,HLen=20
I/SimpleTunnelMVpnService(14879): IP : 10.8.0.5->119.188.110.15 Protocol=6,HLen=20
I/SimpleTunnelMVpnService(14879): TCP : PSH 17664->40 390021120:1074148281
```

图 2-7 ST-MVPN 客户端数据包发送日志

进一步地，为了测试 ST-MVPN 的使用是否会对移动终端的网络带宽造成影响，分别对使用 ST-MVPN 和不使用 MVPN 直接连网的移动终端网速情况进行测试。测试先后在同一台移动终端进行，为了保障相同时间内终端设备流量的一定，分别统计两种连网方式下播放同一时间长度视频时的网速情况。网速的采样间隔为 10 秒一次，得到两种情况下的网速对比情况，如图 2-8 所示。结果表明，两种连网方式在从第 0 秒到 130 秒左右的时间均将视频加载完毕，这段时间内 ST-MVPN 连网和直接连网的网络速度基本一致，未见使用 MVPN 隧道导致网络带宽减少情况。

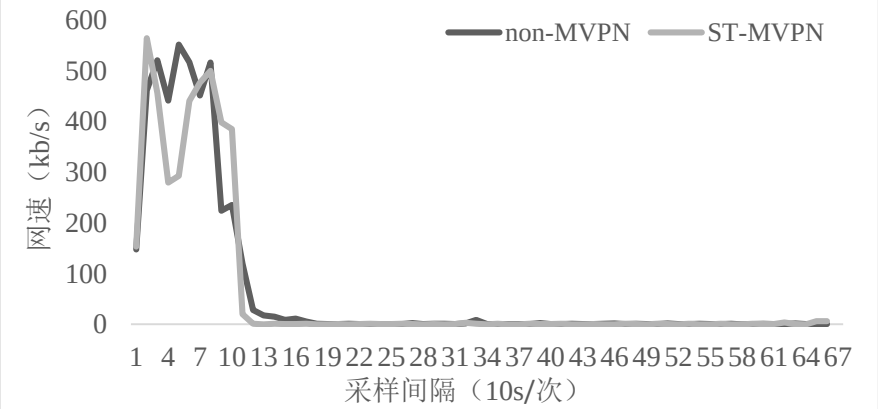


图 2-8 使用 ST-MVPN 和未使用 ST-MVPN 的网速情况对比图

2.4 本章小结

本章对 MVPN 隧道技术展开研究，首先从分类标准、基本原理、评价指标和应用场景对现有 MVPN 隧道技术进行总结和分析。接下来进一步研究了 Android 虚拟隧道的创建方式和工作方式，最后介绍基于 VpnService API 实现的代理服务模式的简单隧道 MVPN，实验表明该 MVPN 隧道能够提供快速稳定的代理访问服务，并且该隧道未对移动终端的网络带宽造成影响。

第3章 MVPN 安全通信协议的研究

本章对 MVPN 安全通信协议展开研究，分析了传统 VPN 协议的存在问题，提出了一种 MVPN 通信协议的新思路并给出该协议的设计方案。随后基于该协议实现了快速传输 MVPN，并将其与 OpenVPN 进行了性能对比实验。最后，本章给出了快速传输 MVPN 的安全评价。

3.1 MVPN 安全通信协议的研究

本节对 MVPN 安全通信协议展开研究，分析了传统 VPN 协议的存在问题，总结了 MVPN 安全通信协议的设计目标，最终提出 MVPN 安全协议的新思路。

3.1.1 传统 VPN 协议的问题分析

传统 VPN 的安全通信协议主要分为公有协议和私有协议两种，公有协议是 IETF 统一制定的互联网通信标准，主要包括 L2TP, PPTP, IPSec 和 SSL 等，表 3-1 总结了这四种协议的特点。私有协议则是开发者根据特定用户的需求设计的加密通信协议。公有协议比较安全可靠，由于广泛的应用因而更加稳定通用。相比之下，私有协议通常是为了满足用户追求传输速度和通信质量的需求，因此私有协议的安全性和复杂性通常低于公有协议。

表 3-1 传统 VPN 安全通信协议特点总结表

协议名称	协议层次	终端认证	用户认证	加密方式	安全漏洞	速度	稳定性
L2TP	链路层	无	有	无	明文隧道	最慢	较差
PPTP	链路层	无	有	RC4	存在明显漏洞	最快	较差
IPSec	网络层	有	有	3DES, AES 等	无明显漏洞	较快	较好
SSL	应用层	有	有	3DES, RC4 等	无明显漏洞	较快	较好

MVPN 安全通信协议借鉴了传统 VPN 协议的特点，但并不是每一种协议都适合 MVPN。根据 2.1.3 节 MVPN 的隧道评价指标，MVPN 安全通信协议应当兼顾通信安全、传输速度、稳定性和耗电量四个方面，因此直接将传统 VPN 协议直接应用于 MVPN 的安全通信协议会带来一系列的不相适应性问题，下面对常见的四种 VPN 安全通信协议和 OpenVPN 私有协议进行详细分析。

（1）L2TP

首先，在通信安全方面，L2TP 本身不加密数据传输隧道，所以 L2TP 通常与 IPSec 共同使用以保证数据传输的机密性和完整性，但这种 L2TP 承载于 IPSec 的实现方式将数据封装两次，增加了处理数据包的开销，因此会对客户

端造成低传输速度和高耗电量。第二，在客户端性能方面，L2TP 的稳定性一直是令人诟病的问题，由于通信端口固定、通信指纹特征明显，L2TP 的数据传输容易遭到旁路设备的阻断和欺骗。第三，L2TP 缺少了可靠的通信保障，但移动终端面临着频繁切换的网络连接方式的问题，这一点对于 L2TP MVPN 来说是致命的缺陷[33]。第四，L2TP 是链路层协议，根据 2.2 节 Android 平台隧道创建方式的介绍，基于 VpnService API 的创建方式只提供网络层和以上的协议接口，所以这种官方的、主流的、不需要 Root Android 移动终端的隧道创建方式并不适用于 Android 平台的 L2TP MVPN。因此，Android L2TP MVPN 的实现需要以 Root Android 操作系统为代价。

（2）PPTP

PPTP 的安全性一直是 VPN 用户担心的问题，由于 PPTP 的 MS-CHAPv2 身份认证协议存在漏洞，导致用户密码可能被暴力破解，接下来的连锁反应就是 VPN 隧道的传输数据遭到破解[34]，因此将 PPTP 应用于 MVPN 同样存在相同的安全隐患。除此之外，PPTP MVPN 还存在上述的 L2TP 第二点、第三点、第四点一样的问题，这里就不再赘述。

（3）IPSec

IPSec 相比于 L2TP 和 PPTP 而言，在理论上被证明是安全的。从本质上而言，IPSec 是一种安全体系，不仅是提供安全协议，更是提供一种安全标准。从协议的角度，IPSec 协议主要包括 IKE 协议、认证首部（Authentication Header, AH）协议和封装安全荷载（Encapsulating Security Payload, ESP）协议；从标准的角度，IPSec 提供了安全策略、密钥协商协议框架和数据安全传输模式的定义。但是，IPSec 这种面面俱到的安全体系优势，也导致了协议本身的复杂性。这种复杂性导致了 IPSec MVPN 在实际应用过程中只满足应用场景需求的相关标准，因此 IPSec MVPN 在实际应用中的兼容性大大降低。另外，这种复杂性还导致 IPSec MVPN 稳定性的缺失。尽管 IPSec MOBIKE 扩展协议致力于解决由于网络连接切换导致的 IPSec MVPN 通信连接异常的问题，但实际应用中鲜有实现。所以，实际应用中的 IPSec MVPN 通常缺少必要的稳定性支持的实现，无法应对通信路由的异常改变，无法实时获取通信连接状态，最终导致 VPN 通信连接的异常。

（4）SSL

SSL 的安全性在理论上也已得到证明，是目前应用最广泛的安全通信协议之一。SSL VPN 通常是基于浏览器实现的内嵌式的应用层加密通道，因此其通道的构建方式与传统基于虚拟网卡的隧道构建方式完全不同。所以 SSL VPN

尽管称为 VPN，但其实质上是一种基于 SSL 协议实现的应用层加密代理。因此，SSL VPN 是一种 B/S 架构的 VPN，通常只针对单个应用提供可靠安全的加密通道，与本课题所研究的 MVPN 不属于同一个范畴。

（5）OpenVPN 的私有协议

OpenVPN 是基于虚拟网卡和 SSL 协议实现的开源 VPN 应用软件。OpenVPN 的私有协议非常复杂，其最大的特色在于将 IPSec 密钥交换思想和 SSL 应用层数据加密有机结合，通信双方先建立 SSL 应用加密隧道再进行通信双方的密钥协商，既保证通信双方的双向身份认证和安全密钥交换，又保证数据加密方式不再依赖于 SSL 的协商结果[35]。这种先建立控制隧道交换密钥，后建立传输隧道加密数据的方式非常灵活。但 OpenVPN 仍然存在性能瓶颈，主要集中在三个方面：第一，OpenVPN 是一种单进程、单线程实现的 VPN，因此虚拟网卡的数据包处理存在瓶颈，应用于移动终端的 OpenVPN 可能会放大 Tun/Tap 的性能问题；第二，OpenVPN 在 TCP/UDP 层和 SSL 层之间实现的可靠传输层违背传统 VPN 的尽可能传输的设计理念，尽管重传机制能够保证传输的稳定性，但也对传输效率造成了影响；第三，OpenVPN 的数据加解密依赖于 OpenSSL 加密库，应用于安全网关、服务器和 PC 机的 OpenVPN 通常配套使用硬件加速卡以提高数据报文的加解密效率，但这种改进方案对于移动终端来说并不适用。

综上所述，直接将传统 VPN 的通信协议应用于移动终端的 MVPN 是不可取的。L2TP 和 PPTP 存在安全隐患，稳定性是较大的问题，且应用于 Android 移动终端还需要以 Root Android 操作系统为代价。IPSec 尚无明显漏洞，但整个体系相当复杂，实际应用过程往往不能兼顾协议的安全、效率和稳定性。SSL 无明显安全漏洞且应用广泛，但现有的 SSL VPN 与本课题研究的 VPN 不属于同一个范畴。OpenVPN 的握手协议设计巧妙，但传输协议复杂冗余，且基于 OpenSSL 加密库的传统加解密算法对于移动终端而言是性能瓶颈。

从目前情况看，SSL 协议与移动终端的适配性最好，可以直接应用于 MVPN 通信双方的安全身份认证和密钥交换，但需要设计私有 MVPN 通信协议承载加密数据，并选择适用于移动终端的加密算法。

3.1.2 MVPN 安全协议设计目标

安全通信协议通过使用加解密技术来解决网络安全通信的问题，主要表现身份的认证性、信息的机密性、信息的可用性、数据的完整性和通信的不可抵赖性五个方面[36]。通信协议的安全性依赖于完备的密码加密体制，现有加密

体制主要包括公钥加密体制和密钥加密体制。前者对服务端的计算开销和性能消耗要求远高于后者，后者的主要难题在于通信双方如何在不可信安全信道中协商对称加密密钥。为了兼顾安全和性能，绝大多数现有安全协议的通信结构采用将认证密钥协商和认证对称加密相结合的方式。

因此，MVPN 安全协议的设计目标包括三部分：可靠身份认证、认证密钥交换和认证数据加密。可靠的身份认证是后续安全通信的基础，MVPN 通信协议首先需要进行双方身份认证，以确保通信双方的真实身份；安全的密钥交换以身份确定的通信双方为前提，在安全信道下可以使用 Diffie-Hellman 密钥交换协议，否则可以使用基于椭圆曲线加密的 Diffie-Hellman 密钥交换协议进行安全密钥交换。带验证的数据加密将加密算法与 MAC 校验机制相结合以保障加密数据的机密性和完整性，确保数据发送过程中未被非法篡改。传统带认证的数据加密方式主要有三种：Encrypt-then-MAC，Encrypt-and-MAC 和 MAC-then-Encrypt，IT 工业界普遍认为第一种方式是最安全的认证加密方式, 即加密时，先加密后计算消息验证码；解密时，先计算消息验证码后解密[37]。

除了上述的安全指标，由于网络通信的实体涉及移动终端，因此 MVPN 安全通信协议还需要满足快速传输、低时延、低电量消耗的特点。

3.1.3 MVPN 协议设计的新思路

通过对传统 VPN 协议的问题分析发现 SSL 协议与 MVPN 的适配性最好，能够提供安全的数据传输通道。综合考虑安全通信协议的设计目标，我们提出了快速传输隧道 (Fast Transmission Tunnel, FTT)协议。FTT 协议借鉴了 OpenVPN 协议的设计思想，将认证密钥交换过程与认证加密传输过程分离，认证密钥交换在应用层的标准 SSL 安全通道环境下进行，认证加密传输则在网络层的 MVPN 加密隧道环境下进行，实现了基于对称加密体制的分层通信协议。

FTT 协议相比于传统 VPN 协议具有以下三点优势：第一，将对称加密体制中认证密钥交换的安全性难题交给标准 SSL 协议处理，大大降低了 FTT 的数据传输协议的复杂性，可以有效降低通信双方数据传输时的报文处理开销；第二，将认证密钥交换与认证加密传输分离使得 MVPN 加密隧道的实现方式更加灵活，可以选择更适用于移动终端 CPU 架构的新型认证加密算法，提高加密效率和传输效率，降低传输时延，有助于提高 MVPN 客户端性能；第三，明确了标准 SSL 通道和私有 MVPN 隧道的分工，前者只负责为后者提供密钥协商与更新，后者只负责加密数据传输，降低了 MVPN 实现难度。

FTT 协议相比于 OpenVPN 协议的优势主要在两个方面：第一，FTT 的数据传输协议的层次更加简单有效，去除了 OpenVPN 冗余的可靠传输层支持，同时简化了 FTT 协议的头部设计，降低了报文处理开销和传输时延；第二，采用新型加密算法库 Libsodium 的加密算法实现 MVPN 隧道数据的加密解密，相比于 OpenVPN 使用的传统加解密算法具有明显的速度优势。

3.2 MVPN 安全通信协议的设计

本节首先将从协议整体架构对 FTT 协议进行初步介绍，随后从基本协议流程对 FTT 协议的通信过程进行详细介绍，最后从协议字段和算法应用的角度对 FTT 协议进行补充说明。

3.2.1 FTT 协议整体架构

FTT 协议的整体架构分为应用层协议和网络层协议两个层次，其结构如图 3-1 所示。FTT 的通信流程包括 FTT 握手和 FTT 传输两个阶段，分别对应于 FTT 协议的两层架构。握手阶段的身份认证和密钥交换的安全性均由标准的应用层 SSL 协议保障，包括 SSL 的握手协议、记录协议和警告协议。握手阶段的最终目标是为传输阶段协商出安全可靠的对称密钥。传输阶段使用握手阶段得到的对称密钥和带认证的对称加密算法对传输数据进行快速加密，加密后的数据承载于私有的网络层 MVPN 隧道协议。

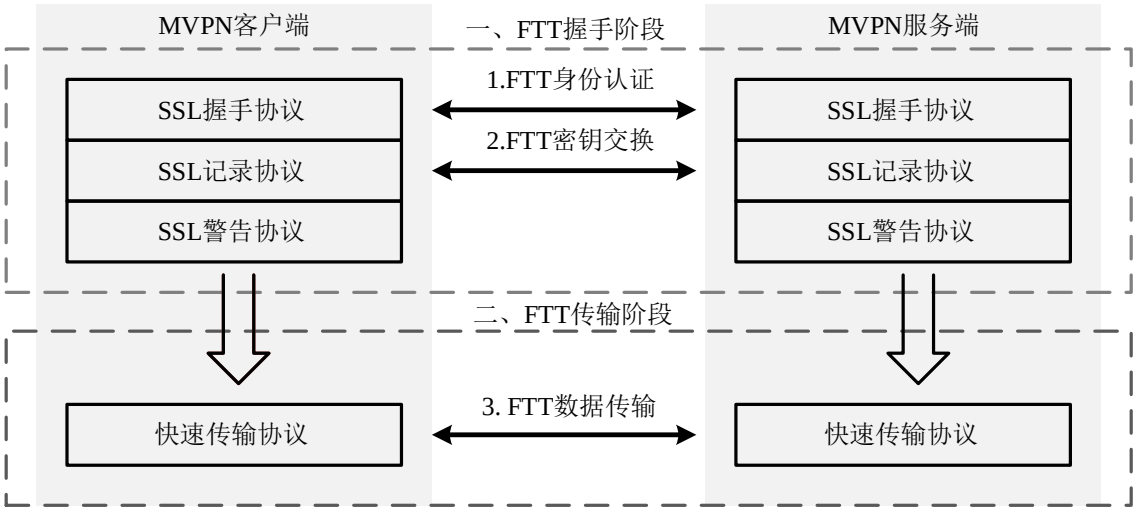


图 3-1 FTT 协议的整体架构示意图

3.2.2 FTT 基本协议流程

FTT 基本协议流程比较复杂，按照协议的通信顺序主要包括 FTT 身份认证

协议、FTT 密钥交换协议和 FTT 快速传输协议，如图 3-2 所示。下面对整个协议流程进行详细描述。

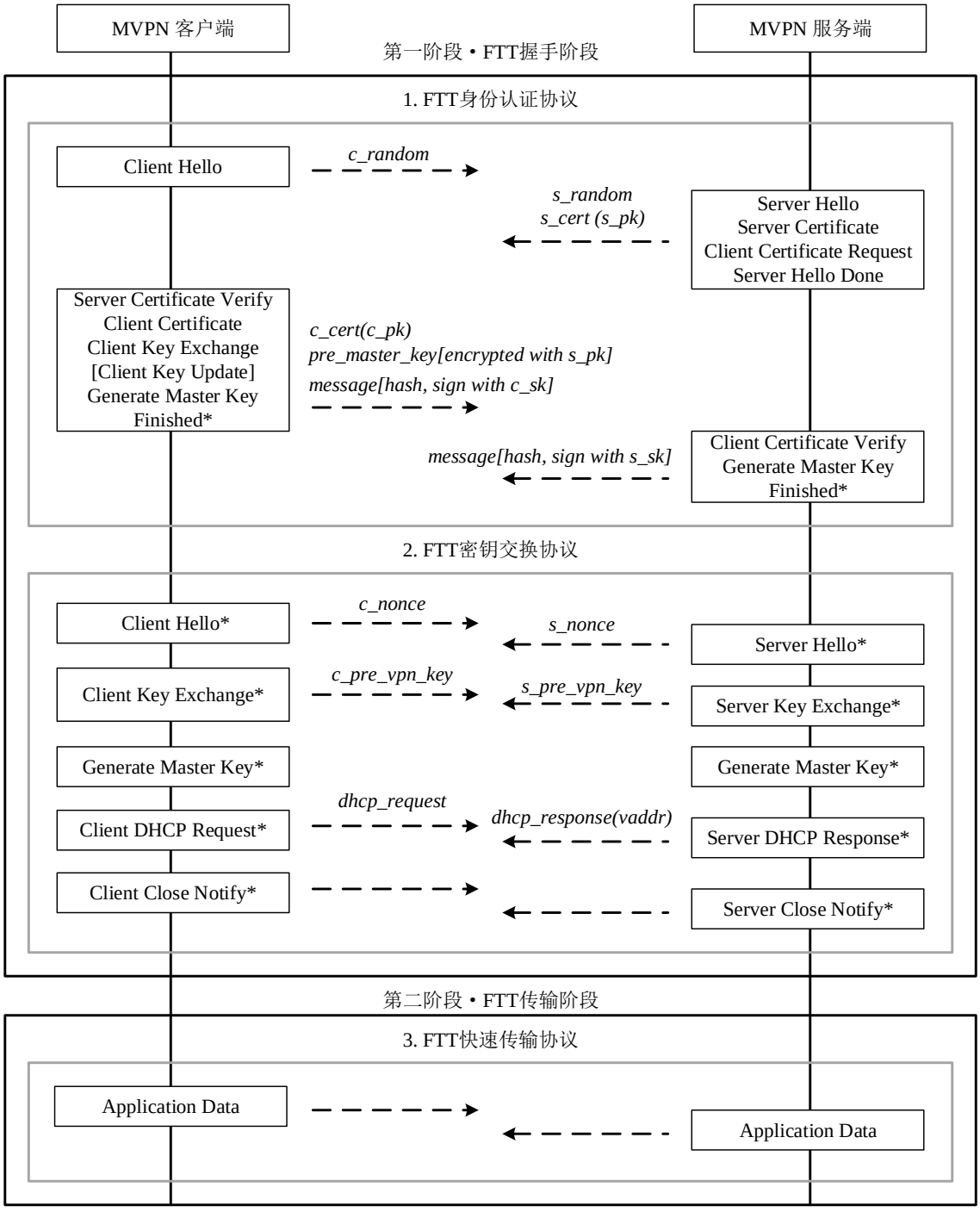


图 3-2 FTT 基本协议流程示意图

首先，进行通信双方的身份认证，该身份认证方式是基于标准 SSL 协议公钥证书的双向认证。

（1）客户端先与服务端三次握手，握手成功后客户端生成随机数

c_random，该随机数将进一步用于密钥推导和防御重放攻击。客户端将 *c_random* 和协议相关算法信息作为 Client Hello 消息发往服务端。

（2）类似地，服务端重复（1）中操作，将 *s_random* 和相关信息发往客户端。另外，服务端还将包含本机公钥 *s_pk* 的证书发往客户端并请求客户端证书。上述操作完成后，服务端初始化完毕并发送 Server Hello Done 消息。此时，服务端拥有客户端随机数 *c_random* 和本地随机数 *s_random*。

（3）客户端收到服务端发来 Server Hello 后提取服务端随机数 *s_random*。客户端接收到服务端的证书后对其进行验证并提取服务端公钥 *s_pk*，验证成功后将包含本机公钥 *c_pk* 的证书发往服务端。此时，客户端拥有服务端随机数 *s_random*, 本地随机数 *c_random* 和服务端公钥 *s_pk*。

（4）类似地，服务端重复（3）中操作，验证客户端证书的真实性并提取客户端公钥 *c_pk*, 服务端拥有客户端随机数 *c_random*, 本地随机数 *s_random* 和客户端公钥 *c_pk*。

（5）客户端生成随机数作为预主密钥 *pre_master_key*，并使用服务端公钥 *s_pk* 加密后发往服务端，服务端收到后使用本地私钥 *s_sk* 对加密的预主密钥进行解密，得到 *pre_master_key*。

（6）客户端和服务端根据已知的 *pre_master_key*, *c_random* 和 *s_random* 生成主密钥 *master_key*。

（7）客户端将之前所有握手信息计算 MAC，使用客户端私钥 *c_sk* 对 MAC 签名，使用 *master_key* 对握手信息和 MAC 加密，以 Finished 消息形式发往服务端。服务端收到消息后，先使用 *master_key* 解密，再检查签名和 MAC。

（8）类似的，服务端重复（7）中操作发送 Finished 消息至客户端。

至此，FTT 协议完成通信双方的身份认证，该过程与标准 SSL 的握手协议保持一致，因此 SSL 安全通信隧道也已建立完毕。接下来进行通信双方的密钥交换，由于整个过程对应于标准 SSL 记录协议的数据传输，因为密钥交换在可靠信道中传输，所以直接采用 Diffie-Hellman 密钥交换协议生成最终快速传输协议的对称密钥。接下来对 FTT 密钥交换协议进行详细描述。

（9）客户端生成随机数 *c_nonce* 并通过 SSL 通道发送至服务端，一方面用于客户端密钥交换的通信初始，另一方面，用于后续数据传输的数据认证，防止重放攻击。

（10）类似地，服务端重复（9）中操作发送 *s_nonce* 至客户端。

（11）客户端与服务端事先约定好 Diffie-Hellman 两个非机密素数 p 和原根 g, 客户端随机生成一个秘密大整数 a, 计算 $A = g^a \bmod p$, 并将 A 通过 SSL

通道发往服务端。类似地，服务端随机生成一个秘密大整数 b, 计算 $B = g^b \ mod \ p$,并将 B 通过 SSL 通道发往客户端。

（12）客户端收到 B 后计算对称加密的主密钥 $vpn\_key = B^a \ mod \ p$，同理服务端收到 A 后计算 $vpn\_key = A^b \ mod \ p$ 得到相同的主密钥。

（13）客户端发送 DHCP 请求至服务端。服务端收到 DHCP 请求后响应一个空闲的虚拟 IP 地址 *vaddr*。

（14）客户端向服务端发出 Close Notify 消息即将断开 SSL 连接，服务端收到后回复 Close Notify 消息，最后通信双方通过 TCP 四次握手结束 SSL 安全通道。

至此，FTT 协议的密钥交换完成，该阶段使用了标准 SSL 的记录协议承载通信双方的随机数交互和 Diffie-Hellman 密钥交换，密钥交换结束后使用了标准的 SSL 警告协议断开 SSL 连接。接下来进行 FTT 的快速传输。

（15）客户端配置虚拟网卡和路由信息等基本网络参数，根据 FTT 握手阶段协商出的 *vpn_key* 密钥和随机数，进行带认证的对称加密传输。

至此，MVPN 通信双方建立了安全的通信隧道，通信双方使用简单的 VPN 私有协议快速传输数据。另外，通信过程中密钥更新的安全需要借助重新创建安全的 SSL 通道来保证。

3.2.3 FTT 协议报文格式

由于 FTT 协议主要分 FTT 握手和 FTT 传输两个不同阶段，因此 FTT 协议报文类型也包括握手报文和传输报文两种，其报文格式如 3-3 所示。

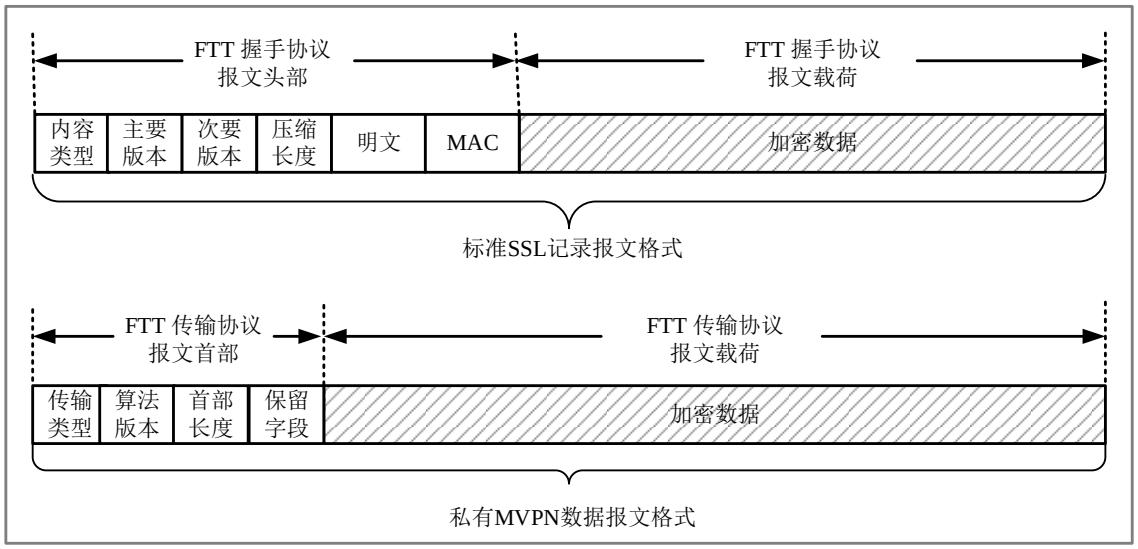


图 3-3 FTT 协议报文格式说明

握手阶段使用标准的SSL通道进行身份认证和密钥交换，因此该阶段的报文格式与标准 SSL 记录报文格式完全相同，故这里不再赘述。本文 3.2.2 节 FTT 协议通信流程（9）至（14）阶段涉及随机数、密钥交换参数和 DHCP 参数均位于该报文格式的加密数据字段，受到 SSL 协议的保护。

传输阶段的数据报文字段设计简单，FTT 传输协议首部由传输类型、算法版本、首部长度和保留字段四部分组成，共占四个字节。传输类型包括快速传输和安全断开两种方式。算法版本则指名数据加密的算法版本号。首部长度标记报文头部大小，通信双方收到包后会首先校验包长以防传输过程出错。保留字段暂未投入使用。

3.2.4 FTT 协议算法应用

FTT 握手阶段涉及的身份认证算法和密钥交换算法均基于标准 SSL 协议使用的算法实现，这里只介绍 FTT 快速传输协议所使用的 ChaCha20-Poly1305 认证加密算法。该算法是由密码学学术权威的 Daniel J. Bernstein 教授针对移动互联网加解密算法性能低下问题提出的新型加密算法。在非硬件加速环境和同等密钥长度的前提下，ChaCha20-Poly1305 算法的加解密速度已被证明是传统 AES-128-GCM 认证加密算法的 3 倍[38]。Google 公司已将该算法应用于旗下所有移动应用的流量加密。

ChaCha20-Poly1305 认证加密算法包括 ChaCha20 加密算法和 Poly1305 加密消息认证码，是一种带关联数据的认证加密算法 (Authenticated Encryption with Associated Data, AEAD)。该算法解决了传统认证加密算法对于数据加密和数据认证的先后顺序的争执问题，将二者集成在算法内部，仅向外提供接口，开发者无须关心 AEAD 的实现细节。目前已有许多加密库支持 ChaCha20-Poly1305 认证加密算法，Libsodium 开源加密库便是其中的一个轻便易用的加密库。该加密库基于 C 语言实现，提供加密解密、数字签名、密码哈希等一系列简单易用的函数接口，目前已被移植到其他众多平台被广泛应用。

本课题通过 Android Java 本地接口(Java Native Interface, JNI)技术实现了 Libsodium 加密库在 Android 平台的移植，并使用 AEAD 的 ChaCha20-Poly1305 加密算法接口对 FTT 的快速传输协议的承载数据进行认证加密。

3.3 基于 FTT 协议的 FT-MVPN 的实现与测试

本节首先从通信模型和工作模块两方面介绍基于 FTT 协议实现的快速传输 MVPN(Fast Transport MVPN, FT-MVPN)，随后对 FT-MVPN 和 OpenVPN 进

行了性能对比测试，最后给出了 FT-MVPN 的安全评价。

3.3.1 FT-MVPN 通信模型

基于 3.2.1 节和 3.2.2 节 FTT 协议的整体架构设计和基本流程设计，将 FT-MVPN 的通信模型设计如图 3-4。FT-MVPN 的通信模型包括标准 SSL 安全通道和 FT-MVPN 安全隧道两个阶段。

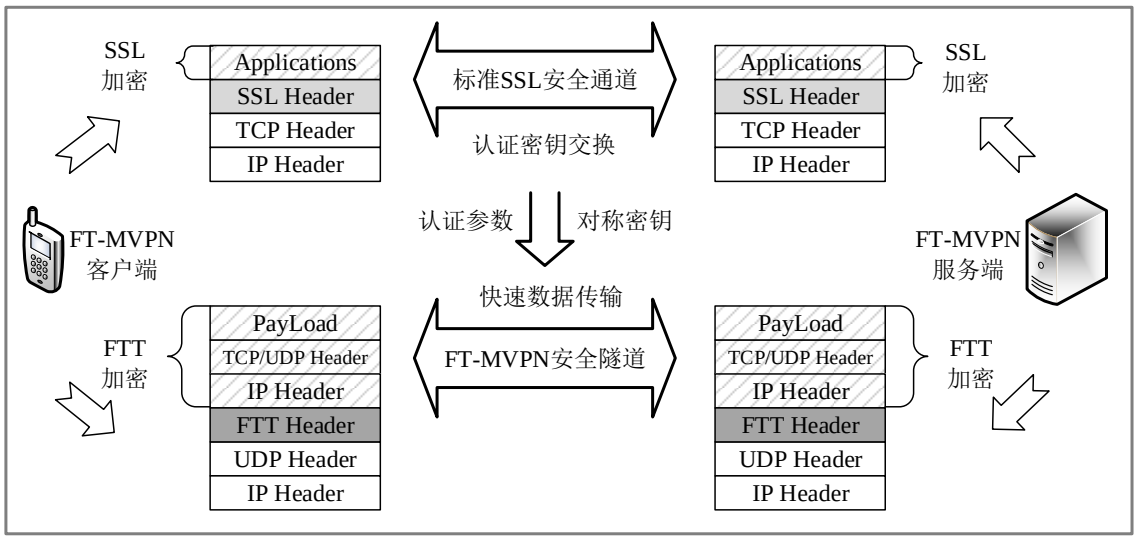


图 3-4 FT-MVPN 通信模型示意图

第一阶段，FT-MVPN 的双方通信首先会进行标准 SSL 握手以进行必要的双向身份认证，该身份认证的过程同样是为 MVPN 提供双向认证。完成身份认证后通信双方进行标准 SSL 协议的密钥交换，生成的主密钥用于对称加密承载数据的 SSL 记录协议。标准 SSL 记录协议启动后，标准 SSL 的安全通道建立完毕，此时通信双方的所有数据传输均受到标准 SSL 对称加密算法的保护。在标准 SSL 通道的保护下，FT-MVPN 开始进行 Diffie-Hellman 密钥交换和其他加密参数的协商，密钥交换后生成的主密钥用于 FT-MVPN 安全隧道的加密。除此之外，FT-MVPN 客户端向服务端发出 DHCP 请求以获得 VPN 安全隧道的虚拟 IP 地址。上述所有步骤完成后，通信双方断开标准 SSL 通道，准备建立 FT-MVPN 的安全隧道。该阶段涉及 VPN 安全隧道的参数协商均通过 SSL 记录协议的加密，在标准 SSL 通道中传输。

第二阶段，FT-MVPN 通信双方使用第一阶段协商出的对称密钥和 AEAD 的 ChaCha20-Poly1305 认证加密算法对数据加密传输。发送的原始数据包经过虚拟网卡的字符驱动设备，由内核态转向用户态的 FT-MVPN 进程处理。FT-MVPN 进程按照 FTT 传输协议的报文格式对数据包进行再次组装,并使用 UDP

固定端口发往服务端。接收的数据包则是上述说明的逆向过程，故不再赘述。

3.3.2 FT-MVPN 模块设计

FT-MVPN 客户端和服务端均主要由身份认证模块、密钥交换模块、DHCP 模块、虚拟隧道模块和快速传输模块组成，但通信两端的模块实现方式和具体功能存在不同之处，如图 3-5 所示。

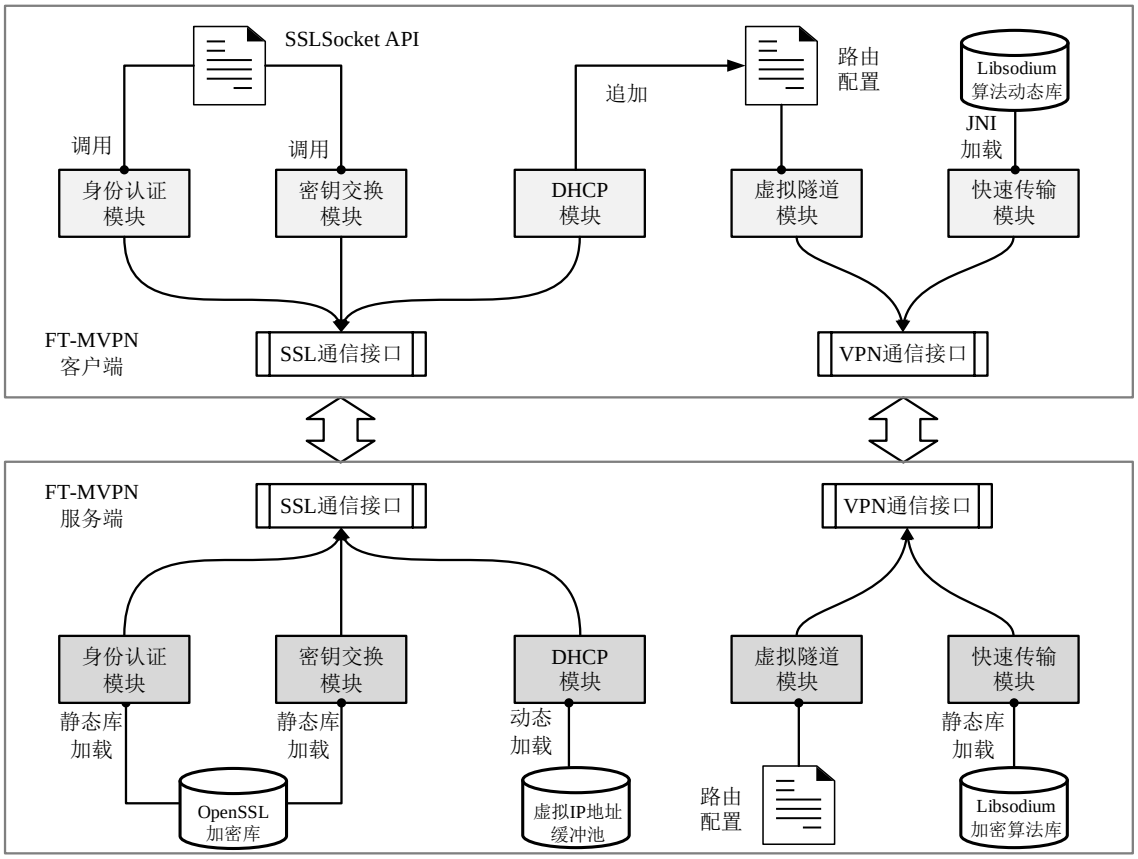


图 3-5 FT-MVPN 通信两端的模块设计图

前三个模块的双方通信基于 SSL 通信接口实现，其中客户端的 SSL 通信均通过 Android SSLSocket API 实现，服务端的 SSL 通信则是基于 OpenSSL 加密库 C 语言实现。客户端 DHCP 模块需要主动发起受 SSL 保护的 DHCP 请求，服务端 DHCP 模块接收解密后，从 FT-MVPN 虚拟 IP 地址缓冲池取出一个空闲虚拟 IP，并通过 SSL 通信接口发往客户端，客户端接收解密后，保留该虚拟 IP 并准备用于配置客户端虚拟网卡。

后两个模块的双方通信基于 VPN 通信接口实现，客户端虚拟隧道模块首先基于 Android VpnService API 创建虚拟网卡和配置系统路由，使用 DHCP 获得的虚拟 IP 地址与服务端通信，之后通信双方的所有数据流量均承载于 FT-

MVPN 私有协议，并使用 VPN 通信接口发送和接收。其中，客户端快速传输模块使用 Android JNI 方式加载 Libsodium 算法动态库中的 ChaCha20-Poly1305 认证加密算法，服务端则采用加载静态库的方式调用该算法。

3.3.3 FT-MVPN 性能测试

测试涉及的硬件设备包括一台 Android 移动终端和一台 Linux 服务器，具体硬件配置参数如表 3-2 所示。其中，Android 移动终端部署了 FT-MVPN 客户端应用和 OpenVPN 应用，Linux 服务器部署了 FT-MVPN 服务端和 OpenVPN 服务端。

表 3-2 FT-MVPN 实验的硬件设备配置表

设备类型	配置参数	配置描述
移动终端	设备型号	MI 2S
	CPU 型号	Snapdragon APQ8064 Pro 1.7GHz
	RAM 容量	2G
	操作系统	Android 4.1.1
服务器	CPU 型号	Intel® Xeon® CPU E5-2630 0 @ 2.30GHz
	内存容量	1G
	操作系统	CentOS Linux release 7.0.1406

实验选择 FT-MVPN 和 OpenVPN 进行性能对比测试，重点考察 FTT 协议对 MVPN 客户端整体性能的改进成果，因此将对两者的传输时延、传输速度、CPU 占用率、内存占用率和耗电量进行重点测试。

（1）传输时延测试

传输时延测试借助 Android 底层 Linux 系统的 ping 工具，分别测试 FT-MVPN 和 OpenVPN 客户端 ping 相同 IP 地址的延时情况，测试 10 次延时结果，统计如图 3-6 所示。

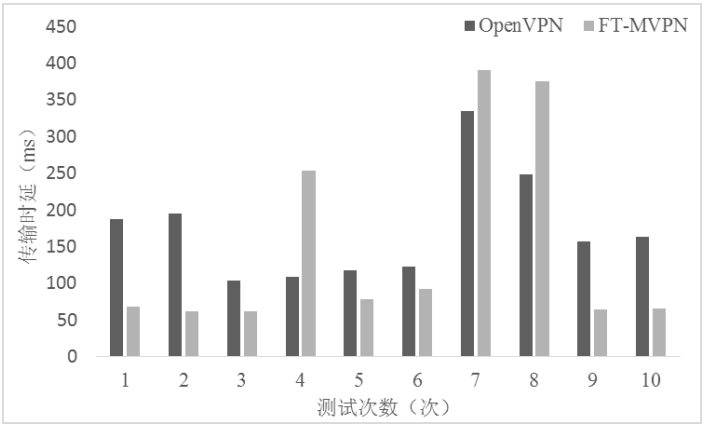


图 3-6 传输时延对比结果

在 10 次传输时延测试中，FT-MVPN 有 3 次传输时延高于 OpenVPN，有 7 次传输时延低于 OpenVPN。进一步地对二者平均传输时延进行统计，FT-MVPN 的平均传输时延为 150.92 毫秒, OpenVPN 的平均传输时延为 174 毫秒。无论从传输时延的总体情况看，还是从平均传输时延看，FT-MVPN 的传输时延均低于 OpenVPN 的传输时延。

（2）传输速度测试

传输速度测试考察 FT-MVPN 客户端与 OpenVPN 客户端在相同流量背景下的网速情况。测试分三组进行，分别选取时间长度为 6 分 16 秒，10 分 31 秒和 25 分 37 秒的视频作为相同背景流量的保证，每组测试选择 FT-MVPN 与 OpenVPN 进行对照实验，网速的采样间隔为 10 秒一次，分别记录两种 MVPN 的不同网速情况，如图 3-7 至图 3-9 所示。

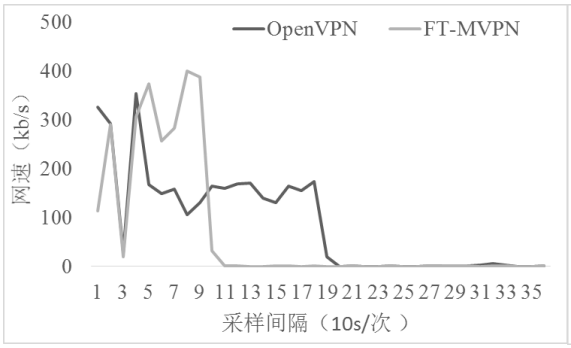


图 3-7 第一组测试网速对比结果

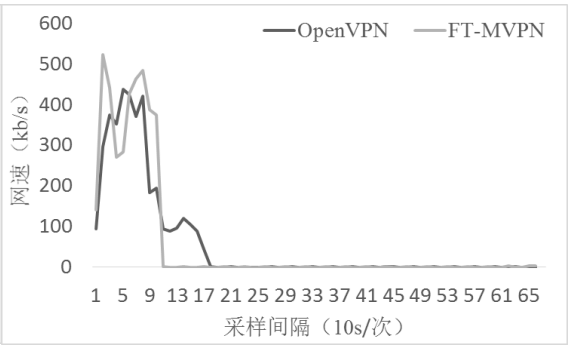


图 3-8 第二组测试网速对比结果

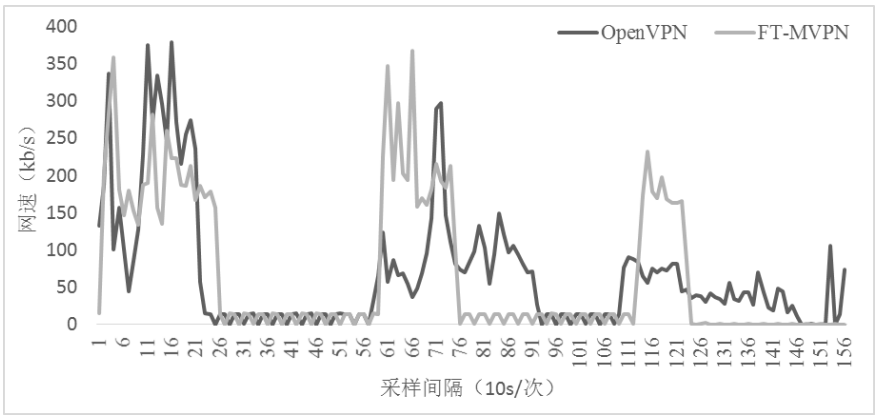


图 3-9 第三组测试网速对比结果

第一组视频长度为 6 分 16 秒的测试集中，FT-MVPN 大约在 110 秒加载完毕，OpenVPN 的加载时长约为 200 秒。在视频整体加载的时间区间内，FT-MVPN 的网速峰值约为 400kb/s, OpenVPN 的网速峰值约为 350kb/s。

第二组视频长度为 10 分 31 秒的测试集中，FT-MVPN 大约在 105 秒加载完毕，OpenVPN 的加载时约为 180 秒。在视频整体加载的时间区间，FT-

MVPN 的网速峰值约为 520kb/s, 而 OpenVPN 的网速峰值约为 420kb/s。

第三组视频长度为 25 分 37 秒的测试集中，由于视频较长，浏览器对视频进行分段缓冲。使用 FT-MVPN 时，视频实际缓冲的时间区间分别为[0, 260], [570, 780]和[1080,1260], 实际缓冲时长约为 650 秒。使用 OpenVPN 时，视频实际缓冲的时间区间分别为[0,240], [570,920]和[1080,1470], 实际缓冲时长约为 980 秒。在视频缓冲时间内，两者网速峰值相差无几，均为 370kb/s 左右。

纵观三组对照实验，FT-MVPN 的视频缓冲时间分别比 OpenVPN 的视频缓冲时间减少 45% ,41.67%和 33.67%。在相同流量的背景下，FT-MVPN 的平均传输速度具有明显优势，二者传输速度峰值基本持平。

（3）CPU 占用情况测试

CPU 占用情况测试考察 FT-MVPN 客户端与 OpenVPN 客户端在相同流量背景下的 CPU 占用率情况。同时考虑到较长时间的测试可以提供更有价值的对比参考意见，因此选取上述第三组视频长度为 25 分 37 秒的测试集，对两种客户端的 CPU 占用情况进行统计，统计结果如图 3-10 所示。

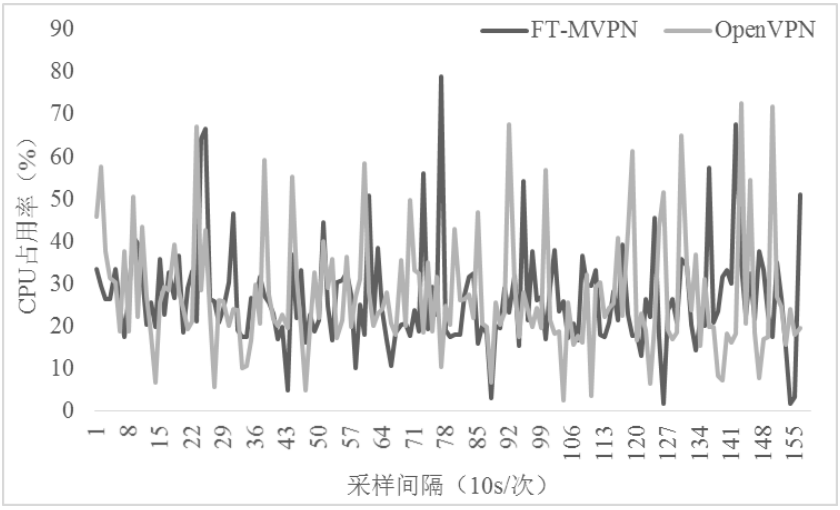


图 3-10 CPU 占用情况对比结果

在相同时间、相同流量背景下，FT-MVPN 客户端和 OpenVPN 客户端的 CPU 占用率基本持平，平均 CPU 占用率分别为 26.58%和 27.08%。

（4）内存占用情况测试

内存占用情况测试选取测试集的要求与 CPU 占用情况测试基本一致，不再赘述，直接给出在视频长度为 25 分 37 秒的测试集情况下，两种客户端的内存占用率对比结果，如图 3-11 所示。其中，FT-MVPN 的内存占用率明显高于 OpenVPN 的内存占用率。前者的平均内存占用率为 51.25%, 后者的平均内存占用率为 47.39%。因此在相同时间、相同流量的背景下，FT-MVPN 的内存占

用率明显高于 OpenVPN，有待进一步优化。

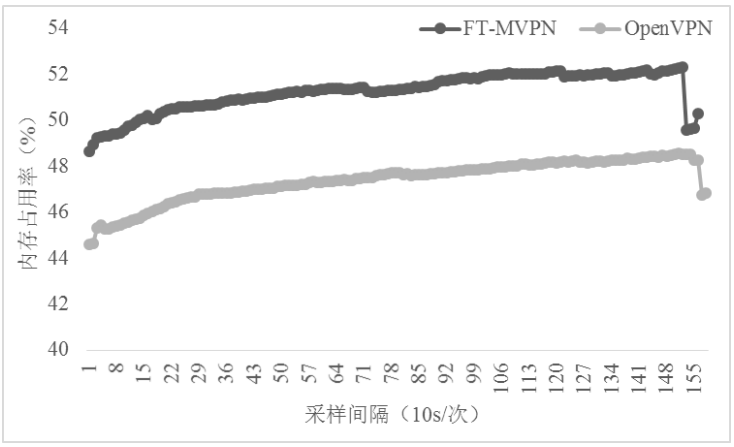
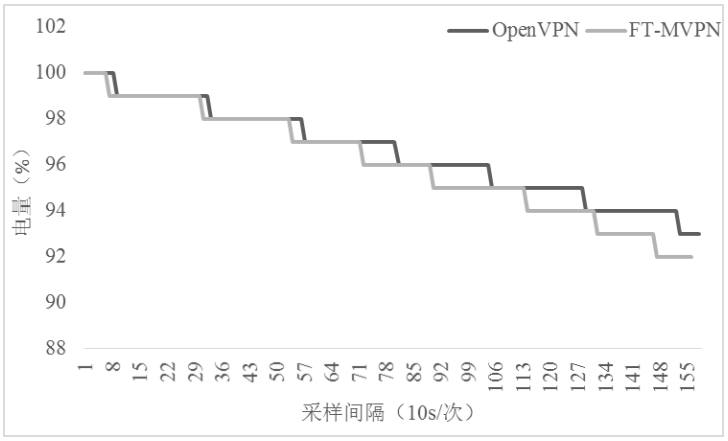


图 3-11 内存占用情况对比结果

（5） 电量消耗情况测试

同测试（3）和（4），电量消耗情况测试的测试集选取要求与其基本一致，这里不再赘述，直接给出在视频长度为 25 分 37 秒的测试集情况下，两种客户端的电量消耗对比结果，如图 3-12 所示。



3-12 电量消耗对比结果

在 1550 秒的测试时间内，使用 FT-MVPN 时，移动终端的电量从 100%下降到 93%；使用 OpenVPN 时，移动终端的电量从 100%下降到 92%。因此，在相同时间，相同流量背景下，FT-MVPN 与 OpenVPN 电量消耗情况相当，前者比后者多出 1%的电量剩余。

综上，性能测试结果表明 FT-MVPN 相比于 OpenVPN 在传输时延和传输速度两项计算性能指标上均有明显优势，在 CPU 占用率和耗电量两项系统性能指标上基本持平，在内存占用情况上具有劣势。内存占用率偏高，主要原因在于 MVPN 客户端代码优化不足，与 FTT 协议本身关系不大。因此，FTT 协

议的整体传输性能优于 OpenVPN。

3.3.4 FT-MVPN 安全评价

FT-MVPN 通信安全依赖于 FTT 协议的保障， FTT 协议包括身份认证协议、密钥交换协议和快速传输协议，因此从这三个方面对 FTT 协议进行安全评价：从身份认证的安全性角度看，FTT 基于标准的 SSL 协议进行了双向身份认证，解决了中间人攻击的问题；从密钥交换的安全性角度看，FTT 基于 SSL 安全可信通道进行了 Diffie-Hellman 密钥协商，保障了安全的密钥衍生和密钥交换过程。同时密钥的定期更新保证了 FTT 协议的向前安全性；从数据传输的安全性角度看，FTT 协议基于公认安全的 AEAD ChaCha20-Poly1305 认证加密算法实现，保障了数据传输的机密性和完整性，并且将通信双方交互的随机数作为 AEAD 参数进行数据验证巧妙的解决了重放攻击的隐患。

综上所述，FTT 能够解决通信双方身份的认证性、数据的机密性和数据的完整性，能够抵抗常见的中间人攻击和重放攻击，并能够保证向前安全性，所以 FT-MVPN 隧道的通信安全值得信赖。

3.4 本章小结

本章分析了传统 VPN 通信协议的存在问题，总结了 MVPN 安全协议的设计目标，根据上述两点提出了快速传输隧道协议 FTT，并给出了该协议的详细设计。该协议将认证密钥交换过程与认证加密传输过程分离，认证密钥交换在应用层的标准 SSL 安全通道环境下进行，认证加密传输则在网络层的 MVPN 加密隧道环境下进行，实现了一种基于对称加密体制的分层通信协议。随后，基于 FTT 提出了快速传输 MVPN 的原型 FT-MVPN，性能测试结果表明 FT-MVPN 相比于 OpenVPN 具有更低传输时延和更高的传输速度，CPU 占用率和耗电量指标也均与后者持平，但内存占用情况相比于后者略显不足。FT-MVPN 客户端代码有待进一步优化从而降低内存消耗。从整体情况看，FTT 能够在保障通信安全的前提下提高 MVPN 客户端的传输效率。

第4章 MVPN 通信保障模型的研究

本章对 MVPN 通信连接的稳定性情况进行研究，针对现有 MVPN 技术和应用分别进行理论分析和实践调研。为了保障稳定的 MVPN 通信连接，本章提出了 MVPN 的通信保障模型，实验表明该模型能够显著的提高 MVPN 的稳定性，并具有一定的通用性和可扩展性。

4.1 MVPN 通信连接稳定性的研究

本节对 MVPN 的稳定性问题来源进行理论性分析，结合 MVPN 实际应用的调研、测试、分析，最终给出本课题的 MVPN 稳定性优化方案。

4.1.1 MVPN 的稳定性问题来源

传统的 VPN 技术以软件服务的形式工作在个人电脑、路由网关、服务器等固定装置上，其工作的网络环境相对稳定，因此传统 VPN 通常只采用客户端和服务器端之间定期发送、接收心跳包的方式检测网络异常，所以可以为用户提供“端到端”稳定可靠的安全通信[39]。较传统 VPN 而言，MVPN 通常工作在相对复杂的移动网络环境中，移动设备的网络接入方式随时可能由于人为因素或非人为因素发生改变。人为因素包括用户主动开启网络、关闭网络和切换网络。非人为因素通常是外部环境所主导，主要包括蜂窝移动网络信号的中断、无线网络信号的衰减或中断等导致的网络连接方式改变。这些移动终端的网络中断和切换容易造成 MVPN 的连接丢失，甚至出现应用异常的情况[40]。为了解决 MVPN 与移动网络环境之间存在的不相适应性，本节进一步调研 MVPN 应用的稳定性现状，分析现有 MVPN 技术与移动终端存在鸿沟的根本原因。

4.1.2 MVPN 应用的稳定性现状

本课题对 Google Play 应用商店的 60 个 MVPN 和 GitHub 的 20 个开源 MVPN 项目的稳定性进行调研。调研工作包括可用性测试与协议分析和应用逆向分析两个方面。

首先，我们对样本中的 Android 应用进行可用性测试与协议分析。可用性测试的目的在于研究移动终端在面临移动网络信号的中断、切换和恢复时，MVPN 的工作状态、可用性情况和性能表现。协议分析的目的在于帮助了解

MVPN 的稳定性与承载协议的关系。我们对样本集中所有的 MVPN 进行试用并抓包分析，在试用时人为的切换或中断移动终端的网络接入类型，待移动设备网络恢复后观察 MVPN 的功能是否正常，测试结果和协议分析结果如图 4-1 所示。其中，不稳定的表现包括应用异常、连接丢失和数据阻塞。相对稳定的表现指的是移动 MVPN 在上述不稳定现象发生后一定时间恢复正常工作。综合考虑 Google Play 应用商店和 GitHub 开源项目，仅 18.75%的 MVPN 能够在变化的移动网络中相对稳定的工作，其余的 MVPN 都存在稳定性不足的问题，出现问题的根本原因主要有两点：其一，移动网络的复杂性和易变性。其二，MVPN 没有针对这些网络异常情况采取应有的处理，传统 VPN 的工作机制在应用于移动终端时需要采取一定的改进。分别考虑不同协议的 MVPN 的稳定性，所有的 PPTP 和 L2TP MVPN 均表现不稳定，其他 MVPN 均有少部分表现相对稳定。

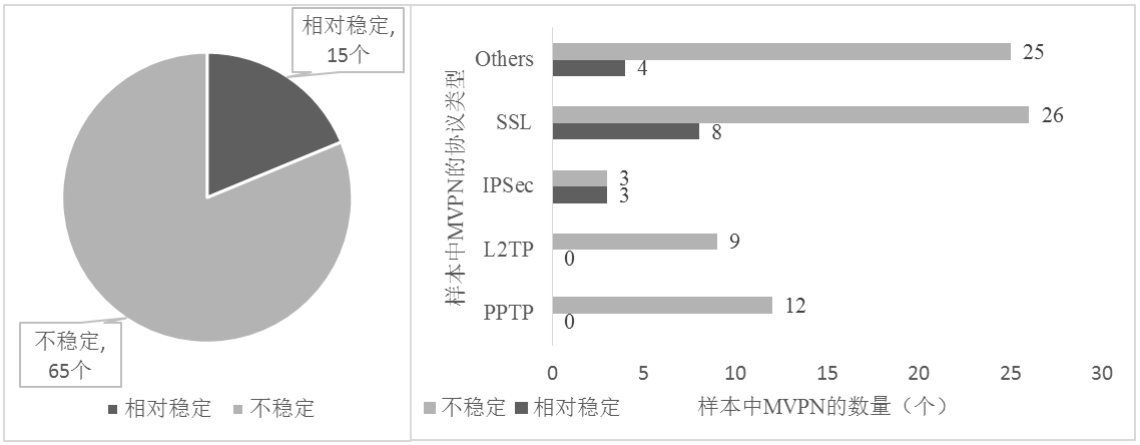


图 4-1 样本中 MVPN 应用的稳定性情况统计

为了进一步探究部分 MVPN 的表现相对稳定的原因，接下来我们对这部分样本应用进行逆向分析。我们使用 dex2jar 工具得到 MVPN 应用的 jar 包，并使用 jdgui 工具查看应用的源码。尽管反编译得到的大部分核心源码已经被混淆处理，但是我们依然取得了一部分有价值的结果。结果表明稳定性较好的 15 个 MVPN 样本在面临网络异常状况主要有两种处理方式。第一种方式，增加了对受保护的 VPN 套接字的异常处理以及时处理网络异常状况；第二种方式，定时检测移动终端的网络接入方式是否发生改变，并及时重启 MVPN 以恢复安全连接。这两种方式能在一定程度上能解决网络改变造成的 MVPN 连接异常情况，但不具有通用性和可扩展性，只有综合考虑 MVPN 的工作状态和移动设备的网络连接变化，才能有效地解决 MVPN 与移动网络环境不相适应的问题。

4.1.3 MVPN 稳定性的优化方案

MVPN 稳定性问题来源的分析结果和稳定性现状的调研结果表明：现有MVPN技术与移动网络环境之间存在的不相适应性，未能考虑移动设备网络连接的易变性对 MVPN 通信连接的影响。尽管现有从协议角度解决这种不相适应性的解决方案，如针对 IPSec 的移动终端扩展协议 MOBIKE 增加了网络参数改变后的密钥更新和 NAT 配置更新以保证通信的恢复[41,42]。又如将 SIP 协议与 SSL/TLS 协议相结合的 MVPN 解决方案，将 VPN 通信的连接维持从网络层提高到会话层，因此网络异常导致的客户端 IP 地址改变不会导致 MVPN 的异常[43]。所以现有的稳定性的优化方案仅是针对具体协议提供的解决方案，不能满足其他公有协议和私有协议 MVPN 的通信保障，因此不具有通用性和可扩展性。

因此，我们将充分考虑移动终端的移动性和网络连接的易变性特点，重新定义一种适应于复杂网络环境的 MVPN 工作机制，设计一种通用 MVPN 的通信保障机制，并据此提出具有可扩展性的 MVPN 通信保障模型，使得该模型在移动网络复杂环境下能保证 VPN 连接的“礼貌中断，智能恢复”。

4.2 MVPN 通信保障模型的研究与原型实现

本节首先以形式化语言的形式对提出的“礼貌中断，智能恢复”通信保障机制的基本原理进行描述。接下来，使用基于有限状态机的接收器模型和 Mealy 机模型分别对 MVPN 的工作状态和网络状态建模，得到 MVPN 的工作状态模型和网络状态模型。除此之外，本章节还介绍了一种通信保障机制和MVPN的关联机制，以更好的将两个数学模型结合使用。最后给出了通信保障模型在 Android 平台的设计方案与原型实现。

4.2.1 通信保障机制的基本原理

“礼貌中断，智能恢复”的通信保障机制重新定义了一种适用于移动终端的 MVPN 的工作机制，即从 MVPN 的工作状态和网络连接两个方面入手，充分考虑 MVPN 在网络变化环境下的不同应对机制。

首先，在 MVPN 的工作状态方面。传统的 VPN 通常只有“运行”，“终止”两种工作状态，因此用集合 *tvpnState* 表示传统 VPN 的工作状态：

$$tvpnState = \{start, stop\} \tag{4-1}$$

为了能有效地应对网络接入方式的变化并保证 MVPN 的正常工作，该机

制增加“暂停”和“异常”两种工作状态，因此我们使用集合 *mvpnState* 表示 MVPN 的工作状态，如公式(4-2)中：

$$mvpnState = \{start, pause, stop, exception\} \tag{4-2}$$

其中“暂停”状态是针对网络连接的易变性提出的，为了让 MVPN 在面临网络异常时“礼貌中断”。“异常”状态则是专门针对 MVPN 在网络配置、物理链路异常或其他程序异常提出的。因此追加定义 MVPN 的工作事件集合 *mvpnEvent* 和异常事件集合 *exceptionEvent*，分别用公式(4-3)和(4-4)表示：

$$mvpnEvent = \{startVpn, pauseVpn, stopVpn, restartVpn\} \tag{4-3}$$

$$exceptionEvent = \{configFailure, linkError, progException\} \tag{4-4}$$

第二，在 MVPN 的网络连接方面。移动网络类型相对复杂，为了使得 MVPN 能够在网络恢复时“智能恢复”，我们对移动终端的网络接入类型详细分类。移动终端网络状态定义为无网络接入、蜂窝移动网络和无线网络, 分别用集合 *noNet*, *cmNet* 和 *wNet* 表示, 如在公式(4-5) 至 (4-7)中：

$$noNet = \{NET\_NO\} \tag{4-5}$$

$$cmNet = \{NET\_2G, NET\_3G, NET\_4G\} \tag{4-6}$$

$$wNet = \{NET\_WiFi\} \tag{4-7}$$

同时，移动终端的网络接入状态可以标记为集合 *netState*, 网络变化事件定义为集合 *netEvent*, 网络变化事件应当考虑网络中断事件（使用集合 *interruptEvent* 表示）、网络切换事件（使用集合 *switchEvent* 表示）和网络恢复事件（使用集合 *recoverEvent* 表示）的三种情况，公式(4-8)至 (4-12)对上述集合进行定义。

$$netState = noNet \cup cmNet \cup wNet \tag{4-8}$$

$$netEvent = interruptEvent \cup switchEvent \cup recoverEvent \tag{4-9}$$

$$interruptEvent = \{(x, y) \mid x \in cmNet \cup wNet, y \in noNet\} \tag{4-10}$$

$$switchEvent = \{(x, y) \mid x, y \in cmNet \cup wNet, x \neq y\} \tag{4-11}$$

$$recoverEvent = \{(x, y) \mid x \in noNet, y \in cmNet \cup wNet\} \tag{4-12}$$

其中，MVPN 的网络变化事件集合 *netEvent* 中的元素均表示为二元组*(x，y), x* 表示 MVPN 早前的网络状态，*y* 表示 MVPN 当前的网络状态，通过二元组我们可以轻易的定义 MVPN 经历了哪一种网络事件。例如*(NET_4G, NET_NO)*表示 MVPN 的网络状态从原先的 4G 网络连接变为了无网络连接，该网络事件为网络中断。又如*(NET_4G, NET_WiFi)* 表示 MVPN 的网络状态从 4G 网络连接变为了无线网络连接，该网络事件为网络切换事件。再如*(NET_NO, NET_WiFi)*表示 MVPN 的网络状态从无网络连接变为了无线网路连接，该网络

事件为网络恢复事件。

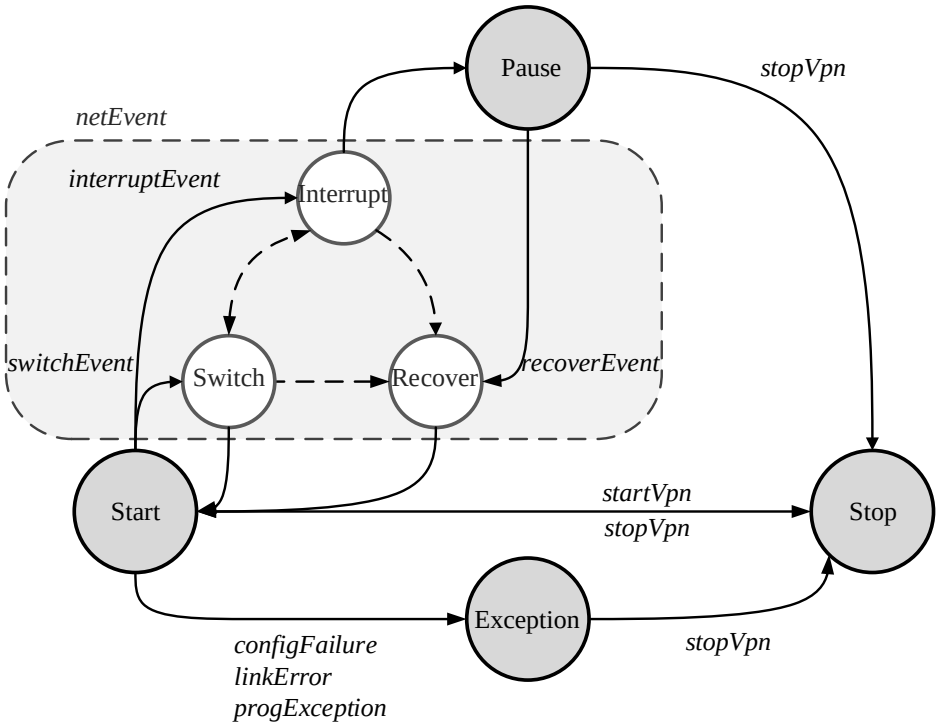


图 4-2 通信保障机制的基本原理图

通过上述的形式化语言描述，MVPN 稳定性的问题被转化为处理集合 *mvpnState* 状态和集合 *netEvent* 状态矛盾的问题，简单的说，如何在 MVPN 工作时应对网络中断、切换事件，在网络恢复时如何智能恢复，即“礼貌中断、智能恢复”的通信保障机制，其原理图如图 4-2。因此，该通信保障机制的基本原理就在 MVPN 运行时实时监控 MVPN 的工作状态和网络状态，并在网络连接改变或异常事件发生时，采取相应的调整措施以保障 MVPN 的“礼貌中断，智能恢复”。

4.2.2 通信保障机制的数学模型

由图 4-2 不难发现该通信保障机制的核心组件是状态和事件，因此我们使用有限状态自动机（Finite State Machine, FSM）模型为 MVPN 的工作状态和网络状态分别构建工作状态模型和网络状态模型。

（1）MVPN 的工作状态模型

MVPN 工作状态可以使用一个五元组的接收器有限状态机(Acceptor FSM, AFSM)数学模型描述，即 $(\Sigma, S, s_0, \delta, F)$，$\Sigma$ 是输入字母表，S 是状态的非空有限集合，s_0 是初始状态集合，δ 是状态转移函数，F 是最终状态集合。那么将 MVPN 的工作状态模型定义为公式(4-13)至(4-17)，其状态转换图如图 4-3。

$$\sum = m\text{vpn}Event \tag{4-13}$$

$$S = mvpnState \tag{4-14}$$

$$s_0 = \{start\} \tag{4-15}$$

$$\delta : S \times \sum \to S \tag{4-16}$$

$$F = \{stop\} \tag{4-17}$$

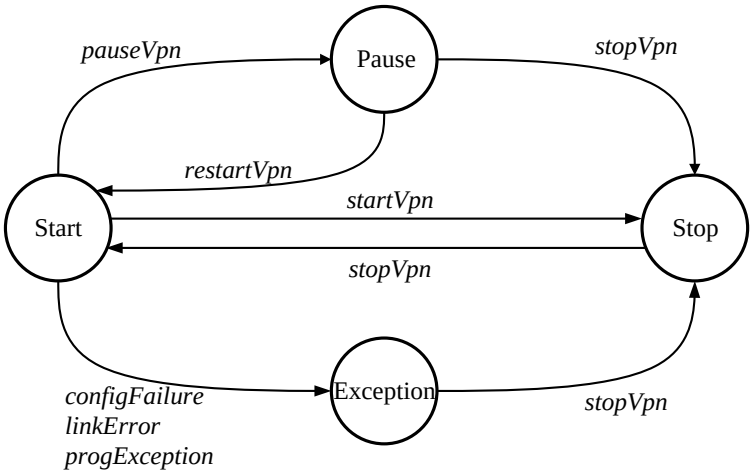


图 4-3 MVPN 工作状态模型的状态转换图

MVPN 的工作状态包括{开启，暂停，终止，异常}四种状态，“开启”状态和“终止”状态的相互切换依靠用户打开或关闭 VPN 连接的事件触发。当 MVPN 处于“开启”状态时，移动终端网络连接发生改变，MVPN 便会接受到暂停 VPN 的信号，并同时将 VPN 的工作状态置为“暂停”。处于“暂停”状态的 MVPN 待移动终端网络恢复，收到重启 VPN 信号时，MVPN 恢复正常的“开启”状态。若处于“暂停”状态的 VPN 接收到用户的关闭 VPN 连接事件信号，则直接将 MVPN 关闭。除此之外，当“开启”的 VPN 发生网络配置异常、物理链路异常或程序工作异常时，直接将 VPN 的状态置为“异常”，待程序记录异常日志后，自动关闭 MVPN。

（2）MVPN 的网络状态模型

MVPN 网络状态可以定义为一个六元组的变换器有限自动状态机(Transducer FSM, TFSM)数学模型 $(\sum, \Gamma, S, s_0, \delta, \omega)$，其中 $\sum$ 是输入字母表，Γ 是输出字母表，S 是状态的非空有限集合，s_0 是初始状态集合，δ 是状态转移函数，ω 是输出函数。由于输出函数 ω 的输出结果依赖于当前状态 S 和输入选项 $\sum$，因此该有限状态机模型属于 Mealy 机。因此，将 MVPN 的网络状态模型定义为公式(4-18)至(4-23)，其状态转换图如图 4-4。

$$\sum = netEvent \tag{4-18}$$

$$\Gamma = mvpnEvent \tag{4-19}$$

$$S = netState = noNet \cup cmNet \cup wNet \tag{4-20}$$

$$s_0 = S \tag{4-21}$$

$$\delta : S \times \sum \to S \tag{4-22}$$

$$\omega : S \times \sum \to \Gamma \tag{4-23}$$

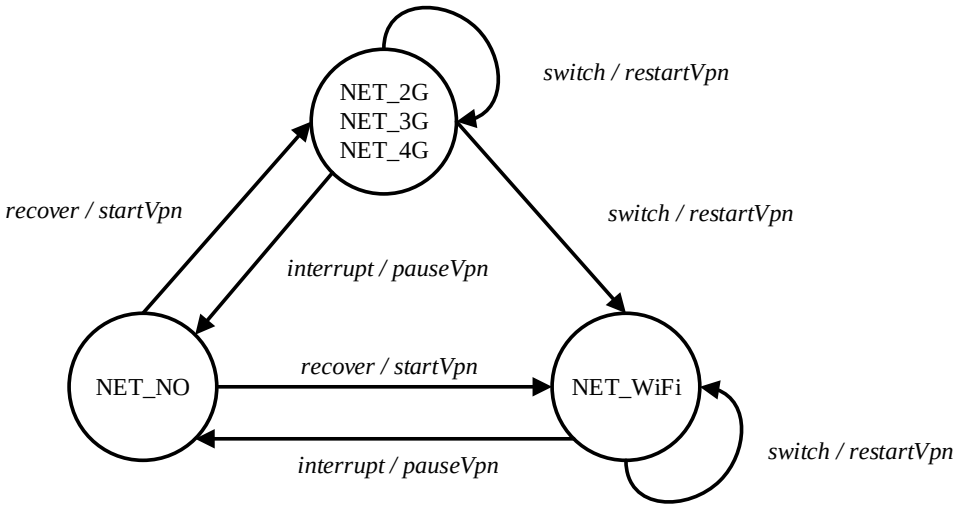


图 4-4 MVPN 网络状态模型的状态转换图

MVPN 网络状态包括{NET_NO, NET_2G, NET_3G, NET_4G, NET_WiFi}五种。图 4-4 将其按公式(4-5)至(4-7)分成了无网络连接，蜂窝移动网络和无线网络三大类。从“NET_NO”状态向其他任意一种网络连接状态的改变需要移动终端网络恢复事件的触发，并输出启动 VPN 的信号。反之，从其他任意一种有网络连接的状态向“NET_NO”状态的改变需要移动终端网络中断事件的触发，并输出暂停 VPN 的信号。其他网络切换事件的触发，如蜂窝移动网络“NET_2G”，“NET_3G”，“NET_4G”和无线网络“NET_WiFi”之间的任意切换，需要移动终端网络切换事件的触发，并输出重启 VPN 信号。

4.2.3 通信保障机制与 MVPN 的关联策略

为了使通信保障机制与 MVPN 协同工作，需要设计一种关联机制将该通将二者有机结合。根据通信保障机制的工作状态模型和网络状态模型定义，两个模型可以分别设计成两个状态监控器，一个用于实时监控 MVPN 工作状态，另一个用于实时监控移动终端的网络连接状态。图 4-5 描绘了 MVPN 与工作状态监控器、网络状态监控器的关联机制和三者之间的数据流向关系。

工作状态监控器由网络配置控制器，网络连接控制器和工作状态控制器三

部分组成。网络配置控制器记录 MVPN 的工作事件(*mvpnEvent*)与异常事件(*exceptionEvent*), 网络连接控制器从网络状态接收器实时获取网络连接类型的变化事件(*netEvent*),并将移动设备现有的连接状态(*netState*)反馈给工作状态控制器。工作状态控制器将当前的网络状态与较早前得网络状态进行比较，并决策 MVPN 下一步工作事件(*mvpnEvent*)。网络状态监控器由一个网络状态接收器组成，感知网络连接是否改变，并将当前的网络事件(*netEvent*)反馈给网络连接控制器。

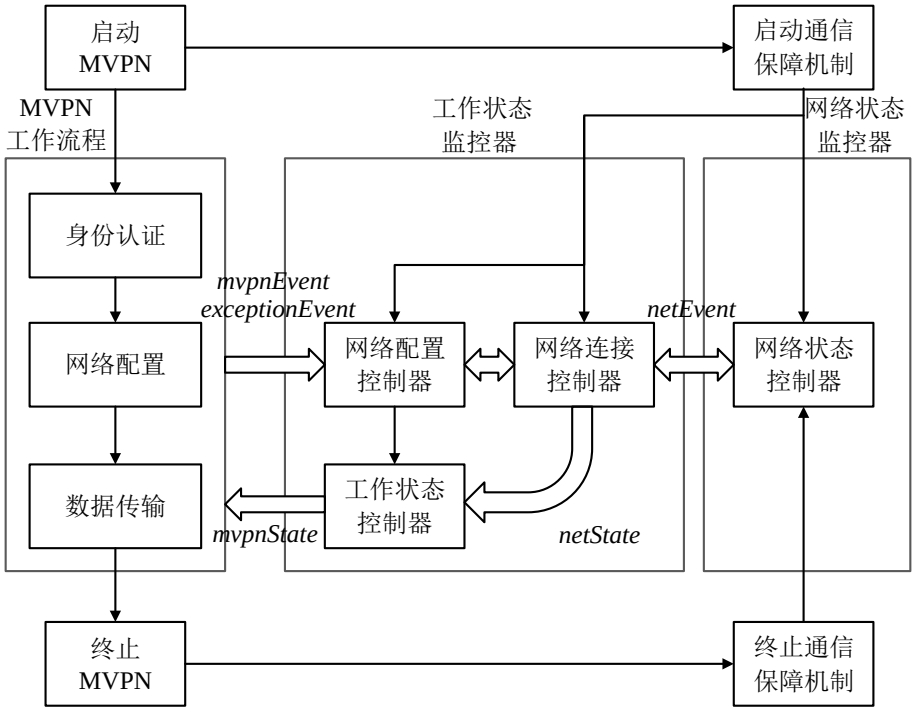


图 4-5 通信保障机制与 MVPN 的关联策略示意图

4.2.4 通信保障模型的设计与原型实现

根据提出的通信保障机制以及该机制和 MVPN 的关联策略，我们进一步设计了 MVPN 通信保障模型。图 4-6 给出通信保障模型在 Android 平台的设计方案。

（1）应用程序层：MVPN 用户界面，提供程序与用户交互接口。

（2）应用程序框架层：MVPN 的核心逻辑设计，主要由 Android 的 Activity，Service 和 Broadcast 三大核心组件组成。该模型调用 Android VpnService 和 ConnectivityManager API 分别实现 MVPN 和网络动态监测。 因此，该模型有由两个重要的后台服务进程组成，即可扩展的 MVPN Service 和 Network Service, 分别用于实现通信保障机制中的 MVPN 工作状态监控器和网

络状态监控器。二者的关系为 Network Service 的生命周期依赖于 MVPN Service 的生命周期，因此设计为 Android Service 机制中的绑定关系。简而言之，MVPN 工作时，网络监控器便启动系统 Broadcast Receiver, 用于动态监测系统的网络变化事件，并将网络连接状态实时反馈给工作状态监控器，以及时控制 MVPN 的工作状态。

（3）系统运行库层：包括系统库，Android 运行执行周期的核心库和 Dalvik 虚拟机，向上层提供 Android VpnService 和 ConnectivityManager API 的支持。

（4）Linux 内核层：应用框架层通过 API 为 MVPN 创建的虚拟网卡、配置的路由信息在该层运行。

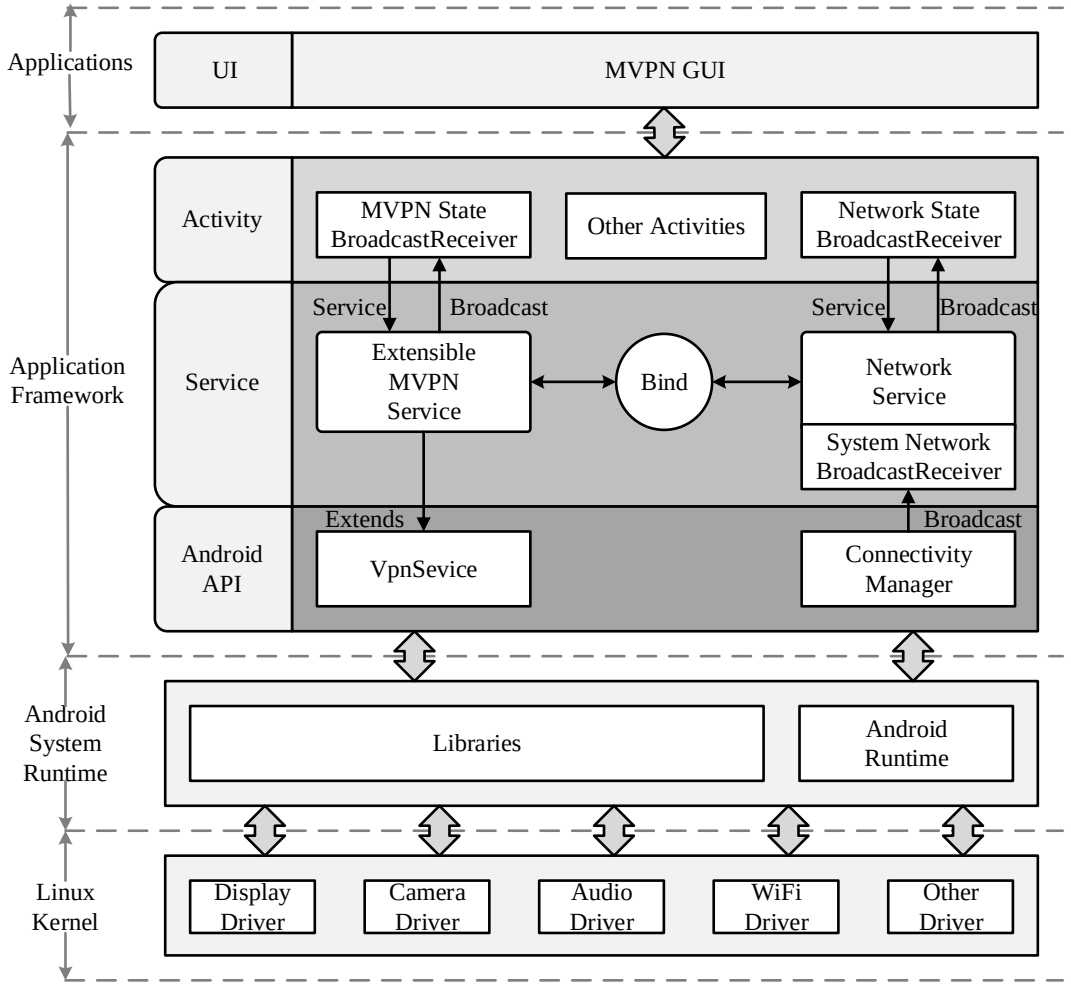


图 4-6 Android 平台的 MVPN 通信保障模型示意图

该通信保障模型不仅适用于 Android 平台的 MVPN，还适用于 IOS 和 Windows 等其他移动平台。只要根据提出的通信保障机制和相应平台的 API 实

现 MVPN 的工作状态监控器和网络状态监控器，该模型可以简单地在其他平台移植和使用，因此该模型具有一定的通用性和可扩展性。

基于 3.3 节实现的 FT-MVPN, 我们实现了通信保障模型的原型系统—通信保障 MVPN(Communication Supportable MVPN, CS-MVPN)。图 4-7 描述了 CS-MVPN 系统模块设计。

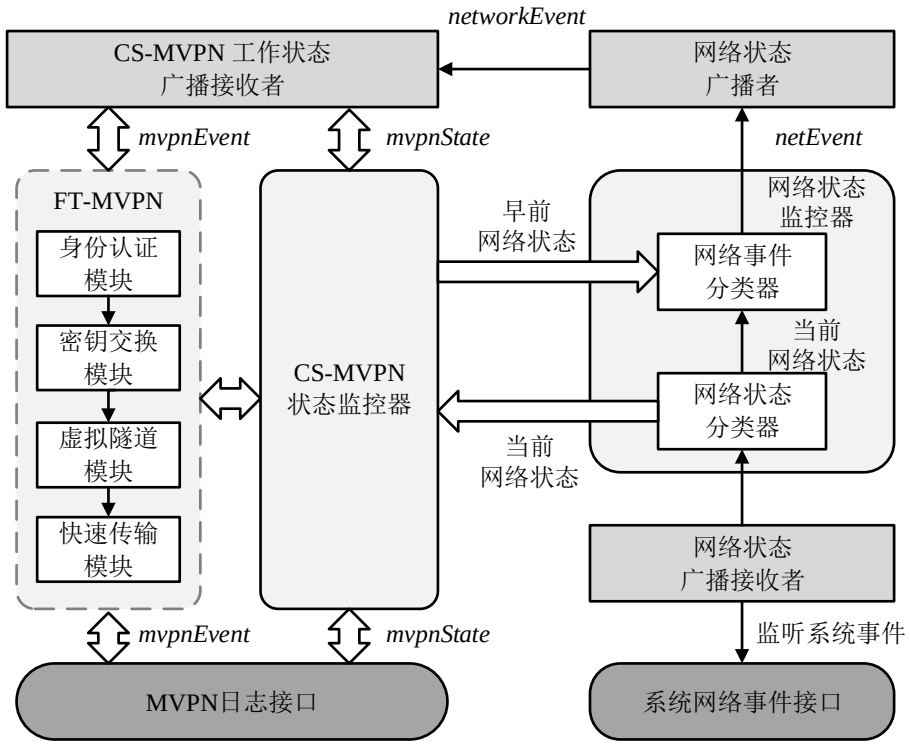


图 4-7 CS-MVPN 系统模块设计图

CS-MVPN 主要由三部分组成：FT-MVPN 的核心模块、CS-MVPN 状态监控器和网络状态控制器。CS-MVPN 继承了 FT-MVPN 的核心模块，包括身份认证模块、密钥交换模块、虚拟隧道模块和快速传输模块。CS-MVPN 状态监控器用于从工作状态广播接受者获取 VPN 的实时工作状态。网络状态控制器则包括网络状态分类器和网络事件分类器。前者动态监测系统的网络变化并分析当前网络连接的类型。通过比对当前的网络状态和早前的网络状态，后者确定移动终端发生的网络事件，并将其通过网络状态广播者发送给工作状态广播接收者。另外，CS-MVPN 利用实现的 MVPN 日志接口将最新的工作状态和网络状态发送给服务器端以便于实验数据的收集。

4.3 MVPN 通信连接的稳定性测试

在完成 Android 平台的 FT-MVPN 和 CS-MVPN 两个 MVPN 原型系统后，

我们希望通过在易变的网络环境中的 MVPN 可用性测试以证明基于通信保障模型实现的 MVPN 具有更好的稳定性。实验选取了三个 MVPN 客户端进行对照实验，包括一款源自 Google Play 应用商店的 MVPN（在后续的测试中，我们用 X-MVPN 进行描述）、基于 VpnService 的 FT-MVPN 和基于通信保障模型的 CS-MVPN。

4.3.1 实验环境

实验环境包括一台 Android 移动终端和一台 Linux 服务器，具体配置参数信息如表 4-1 所示。

表 4-1 MVPN 稳定性实验的硬件设备配置表

设备类型	配置参数	配置描述
移动终端	设备型号	MI 2S
	CPU 型号	Snapdragon APQ8064 Pro 1.7GHz
	RAM 容量	2G
	操作系统	Android 4.1.1
服务器	CPU 型号	Intel® Xeon® CPU E5-2630 0 @ 2.30GHz
	内存容量	1G
	操作系统	CentOS Linux release 7.0.1406

其中，移动终端安装了 X-MVPN 应用和已实现的 FT-MVPN 客户端及 CS-MVPN 客户端，服务器端部署了 FT-MVPN 和 CS-MVPN 的服务端程序。

4.3.2 实验过程

由于测试对象有三个 MVPN 客户端，因此实验分三组进行。每组需要分多轮实验进行，分别对 MVPN 经历在网络中断事件、网络切换事件和网络恢复事件后可用性情况进行人为鉴定。根据公式(4-11)-(4-12)的网络事件定义，测试覆盖所有的网络事件需要进行 21 轮。每轮实验将重复 10 次，以尽量保证 MVPN 在当前测试集中的稳定性表现不具有随机性。

MVPN 应用开启 VPN 连接并正常工作后，实验正式开始。对于每组可用性测试的每一轮实验，人为改变移动终端的网络连接以模拟该组当轮实验的网络事件，观察网络连接变化后 MVPN 的工作状态是否异常。每轮重复 10 次后，统计 MVPN 的本轮异常次数。最终统计的三组 MVPN 异常数据如表 4-2 所示。在网络中断事件发生后，X-MVPN、FT-MVPN 和 CS-MVPN 均未发生应用崩溃等异常行为，只是 VPN 连接由于移动终端网络连接的中断而中断；在网络切换事件发生后，X-MVPN、FT-MVPN 和 CS-MVPN 分别在 130 次测试中出现 53 次，105 次和 6 次异常情况；在网络恢复事件发生后，X-MVPN 在四组

共计 40 次实验中，出现了 11 次不可使用的异常情况，FT-MVPN 出现了 24 次不可使用的异常情况，而 CS-MVPN 仅出现 1 次不可使用的情况。

表 4-2 网络事件中 MVPN 可用性测试结果统计表

网络事件类型	网络事件描述	10 次试验中 MVPN 表现异常的次数		
		X-MVPN	FT-MVPN	CS-MVPN
网络中断事件	(NET_2G, NET_NO)	0	0	0
	(NET_3G, NET_NO)	0	0	0
	(NET_4G, NET_NO)	0	0	0
	(NET_WiFi, NET_NO)	0	0	0
网络切换事件	(NET_2G, NET_3G)	4	7	0
	(NET_2G, NET_4G)	4	4	0
	(NET_2G, NET_WiFi)	6	10	0
	(NET_3G, NET_2G)	4	8	2
	(NET_3G, NET_4G)	1	3	0
	(NET_3G, NET_WiFi)	5	10	0
	(NET_4G, NET_2G)	4	7	1
	(NET_4G, NET_3G)	2	6	0
	(NET_4G, NET_WiFi)	5	10	0
	(NET_WiFi, NET_2G)	6	10	3
	(NET_WiFi, NET_3G)	4	10	0
	(NET_WiFi, NET_4G)	2	10	0
	(NET_WiFi, NET_WiFi)	6	10	0
网络恢复事件	(NET_ NO, NET_2G)	2	8	1
	(NET_ NO, NET_3G)	4	7	0
	(NET_ NO, NET_4G)	2	5	0
	(NET_ NO, NET_WiFi)	3	4	0

将表格中对照实验的每轮实验的异常情况进行统计，得到三组 MVPN 更直观的异常波动情况，如图 4-8 所示。

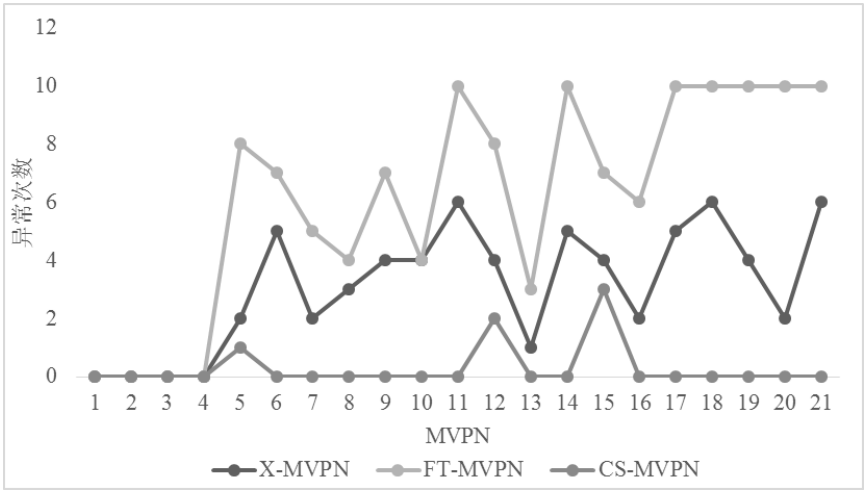


图 4-8 每轮实验的 MVPN 异常次数统计图

从三种 MVPN 的异常波动折线不难发现 CS-MVPN 折线的波动幅度最小，FT-MVPN 折线的波动幅度最大，X-MVPN 折线的波动幅度位于二者之间。因

此，从直观的角度来看，基于通信保障模型的 CS-MVPN 具有更好的稳定性。

4.3.3 结果分析

据统计，在每组进行的 210 次实验中，CS-MVPN 发生了 7 次异常工作情况，FT-MVPN 发生了 129 次工作异常，X-MVPN 发生了 64 次工作异常。为了进一步使用实验数据描述三组 MVPN 的稳定性，定义公式(4-24):

$$P(stability)=(1-\frac{N(exception)}{N(trial)})\times 100\% \tag{4-24}$$

其中，*N(exception)*表示在测试过程中出现总的 MVPN 异常次数，*N(trial)*表示总得测试次数，*P(stability)*表示 MVPN 正常工作次数占总实验次数的百分比。所以 *P(stability)*可以被用来衡量 MVPN 的稳定性，百分比越高，则稳定性越高。因此不难算出三种 MVPN 在网络中断事件、网络恢复事件和网络切换事件的三种不同测试集时表现出的稳定性，如图 4-9 所示。更进一步地，我们可以计算出 CS-MVPN、FT-MVPN 和 X-MVPN 的整体稳定性分别为 96.67%，38.57%和 69.52%。

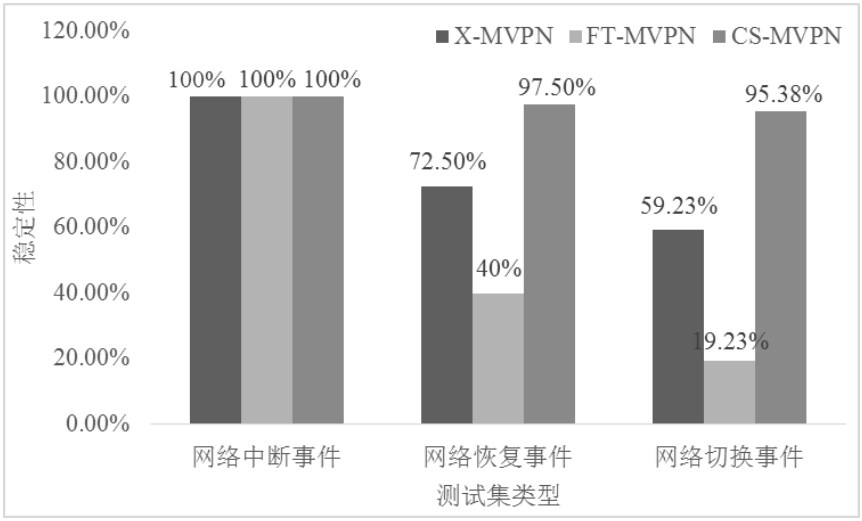


图 4-9 不同测试集 MVPN 稳定性情况

从图 4-9 中不难发现，CS-MVPN 较 FT-MVPN 和 X-MVPN 在实验中均表现出更好的稳定性。然而实验结果表明基于通信保障模型的 CS-MVPN 仍然存在一定的不稳定性。从表 4-2 中不难发现每次异常情况均发生在 2G 网络的恢复或切换中，我们通过查询程序日志，发现 CS-MVPN 恢复失败的原因在于 VPN 客户端与服务器握手超时。我们推测是 2G 网络不稳定或速度较慢导致的 VPN 连接恢复失败。于是我们增加了 VPN 的握手超时阈值，发现此时的 CS-MVPN 在 2G 网络切换事件发生时，能够在用户可忍耐的时间范围内，恢复

VPN 连接。因此，尽管 CS-MVPN 在实验中未表现出 100%的稳定性，但实验结果证明通信保障模型为 MVPN 通信连接提供更好的稳定性。

4.4 本章小结

本章对 MVPN 通信连接的稳定性问题展开研究，通过对 Android MVPN 应用的稳定性调研，我们分析了现有 MVPN 技术与移动终端存在鸿沟的原因。根据调研的统计分析结果，我们提出了一种保障 MVPN 在复杂网络环境下稳定通信的新思路，即“礼貌中断，智能恢复”的通信保障机制。该机制重新定义了 MVPN 的工作机制，增加移动网络的接入控制以适应于移动网络的复杂环境。接下来我们对 MVPN 的工作状态、网络连接状态、网络变化事件等新思路中的几个关键点使用形式化语言进行抽象描述，并通过有限状态机模型对 MVPN 的工作状态和网络状态进行数学建模。在此基础之上，我们进一步提出了 MVPN 的工作状态模型和网络状态模型。为了更好的将通信保障机制与 MVPN 协同工作，我们设计了通信保障机制和 MVPN 的关联机制。最后，我们基于上述提到的通信保障机制和关联机制，提出了适用于 MVPN 通信保障模型。实验结果证明，该模型能够在移动网络复杂环境下保证 MVPN 通信的“礼貌中断，智能恢复”，并且具有较好的通用性和可扩展性。

第5章 原型系统设计与实现

在本课题研究基础和实现成果的支撑下，本章提出了基于 MVPN 技术的 Android 移动终端 Wi-Fi 安全接入的解决方案，并给出该解决方案的系统原型—Wi-Fi 安全接入系统(Wi-Fi Secure Access System, WSAS)。本章将从系统整体架构和系统各实体的模块实现对本系统进行详细介绍，最后给出本系统的测试过程和测试结果。

5.1 系统架构设计

Wi-Fi 安全接入系统共包括移动终端、管理服务器、代理服务器、Mob 短信验证平台和短信服务商五个实体，实体之间的工作关系和主要交互功能如图 5-1 所示。

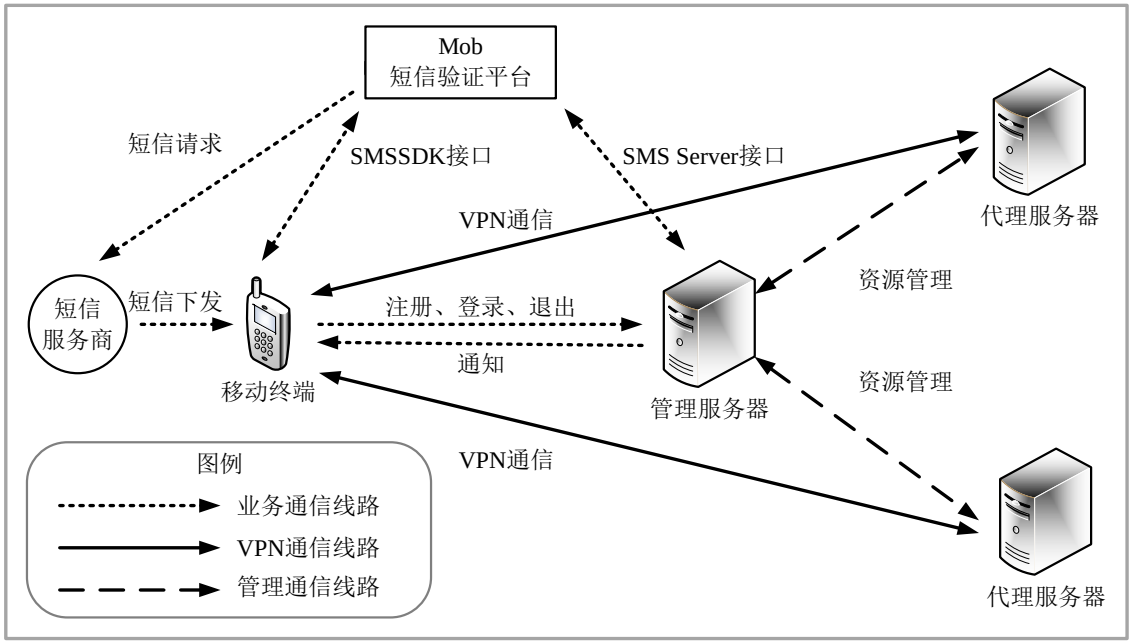


图 5-1 Wi-Fi 安全接入系统的整体架构图

每个实体在该原型系统中扮演的角色描述如下：

（1）移动终端

Wi-Fi 安全接入系统的客户端以 Android apk 的形式工作在 Android 移动终端。用户通过客户端完成本系统的注册和登录后，便可以免费使用 VPN 通信业务在非受信的 Wi-Fi 环境中安全接入互联网。移动终端在本系统中扮演的角色最为复杂，与剩余的其他四个实体均有通信关联：需要与管理服务器进行

本系统的各项业务通信；需要与代理服务器直接建立 VPN 安全隧道进行安全通信；需要调用 Mob 短信验证平台的短信开发接口（SMSSDK），向 Mob 第三方服务器请求发送系统服务的验证码；需要处理短信服务商下发的验证码短信通知。

（2）管理服务器

管理服务器部署了本系统业务通信服务端程序，除了负责配合移动终端完成注册、登录、退出等基本业务服务，还涉及到与 Mob 短信验证平台共同完成移动终端的短信注册码验证。除此之外，管理服务器与代理服务器集群共同构成内部管理网络，负责代理服务器的资源调度和状态管理。

（3）代理服务器

代理服务器部署了本系统 VPN 通信服务端程序，负责与客户端 MVPN 通信模块建立安全加密隧道并保障移动用户数据的机密性和完整性，并以代理模式接收和转发移动终端的数据流量。另外，代理服务器还需要周期性地向管理服务器汇报当前服务端的系统状态和网络状态。

（4）Mob 短信验证平台

Mob短信验证平台为本系统提供第三方的短信验证码发放请求和内容验证服务。前者是接收移动终端的短信验证码请求，并向短信服务商发出短信下发通知。后者则是帮助管理服务端验证移动终端接收到的验证码与短信下发内容是否一致。

（5）短信提供商

短信提供商负责为本系统提供短信下发服务，短信下发请求则是本系统通过调用第三方短信验证平台提供的 SMSSDK，实现的间接请求发送。

5.2 客户端设计与实现

本节首先介绍客户端的核心组件的层次结构，随后将深入每个组件内部结构分别介绍其模块设计，最后介绍了客户端的业务通信模块的主要功能。

5.2.1 核心组件结构

如图 5-2 所示，客户端的核心组件包括 Wi-Fi 感知组件和 MVPN 通信组件。前两个组件均在移动应用服务与终端网卡设备之间的层次工作，WSAS 业务模块与其他移动应用服务处于同一层次。终端网卡设备位于最底层，是移动终端数据流量接收的起点和发送的终端。Wi-Fi 感知组件工作在移动网卡设备之上，对移动终端接入 Wi-Fi 的安全性进行感知。对于用户受信的 Wi-Fi 热点，

该组件决策允许移动应用服务直接接入 Wi-Fi 热点；对于未受信的 Wi-Fi 热点，该组件决策移动应用服务调用 MVPN 通信组件，先建立 VPN 安全加密隧道作为 Wi-Fi 安全通道，再发送和接收移动应用服务的数据流量。MVPN 通信组件位于 Wi-Fi 感知组件之上，根据 Wi-Fi 感知组件的决策结果决定是否启用。移动应用服务位于整体结构的最上层，应用根据其业务逻辑和用户行为与其他实体通信，WSAS 业务模块便工作于此，处理用户注册、登录、退出等系统基本业务需要。

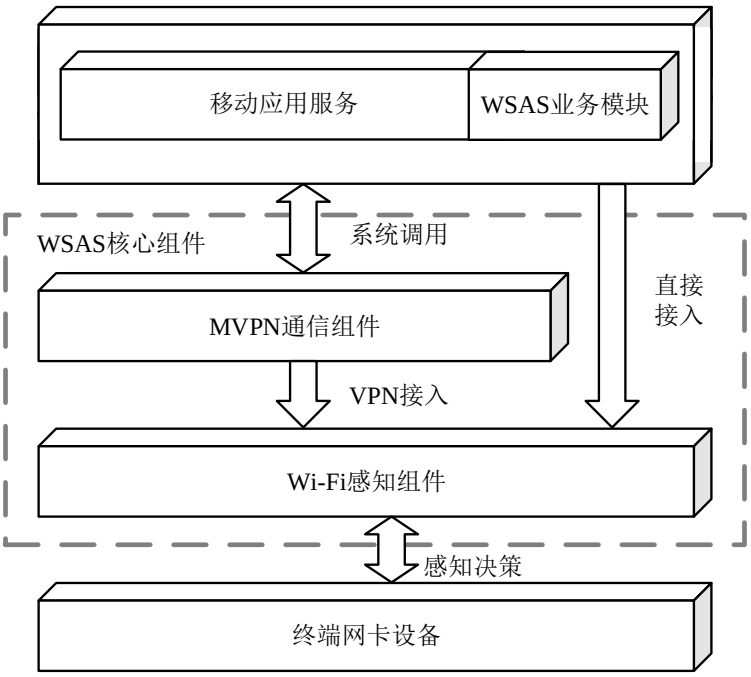


图 5-2 客户端核心组件层次结构图

Wi-Fi 感知组件在本系统中扮演的是 MVPN 组件开关的角色，可以独立地用于 Android 移动终端的 Wi-Fi 管理器。类似地，MVPN 通信组件同样可以作为 Android MVPN 应用单独存在。

5.2.2 Wi-Fi 感知组件

Wi-Fi 感知组件主要由 Wi-Fi 识别模块、Wi-Fi 过滤模块、Wi-Fi 控制模块三个模块以及若干配置文件组成。图 5-3 描述了各模块之间的工作关系，各模块具体功能描述如下：

（1）Wi-Fi 识别模块

该模块实时监控 Wi-Fi 信号的接入，根据启动配置文件规定的 Wi-Fi 参数的规格需求，提取 Wi-Fi 信号对应的参数信息并将其回传 Wi-Fi 过滤模块。在默认情况下，配置文件包含 Wi-Fi 热点的 SSID 和 MAC 二元组信息。该模块提

取接入 Wi-Fi 信号的二元组在过滤模块做比对。

（2）Wi-Fi 过滤模块

该模块将识别阶段获得的 SSID 和 MAC 二元组与 Wi-Fi 白名单中的二元组进行比对。白名单匹配成功，移动应用直接接入该受信 Wi-Fi 热点；匹配失败则认为该 Wi-Fi 热点未受信，将启动 Wi-Fi 控制模块。白名单每行存储受信 Wi-Fi 热点的相关参数，参数的个数和属性与启动配置文件配置的规格相同。白名单由用户自行配置，用户将受信的 Wi-Fi 热点配入白名单中，如家庭 Wi-Fi 等。

（3）Wi-Fi 控制模块

该模块以 MVPN 通信组件的开关形式存在决定是否启用 MVPN 通信组件。当接收到 Wi-Fi 过滤模块的非白名单信号后，主动开启 MVPN 通信组件；另外，使用移动网络连网的用户可以通过该开关开启 MVPN 通信组件。

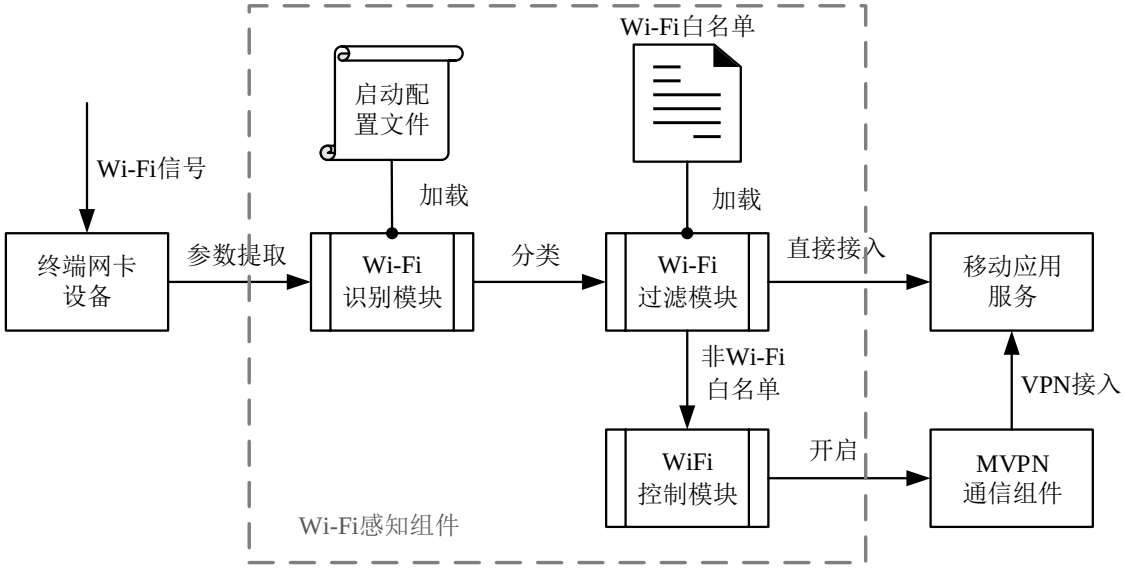


图 5-3 Wi-Fi 感知组件工作流程示意图

5.2.3 MVPN 通信组件

MVPN 是本课题研究的核心技术，本文第 2 章、第 3 章、第 4 章已迭代地实现了安全、快速、稳定的 CS-MVPN，其模块设计已在本文 4.2.4 节图 4-7 给出并附有详细说明，因此这里不再赘述。

5.2.4 业务通信模块

业务通信模块设计如图 5-4 所示，将该模块划分为前端业务、中间件和通信接口三个子模块，下面对其分别介绍。

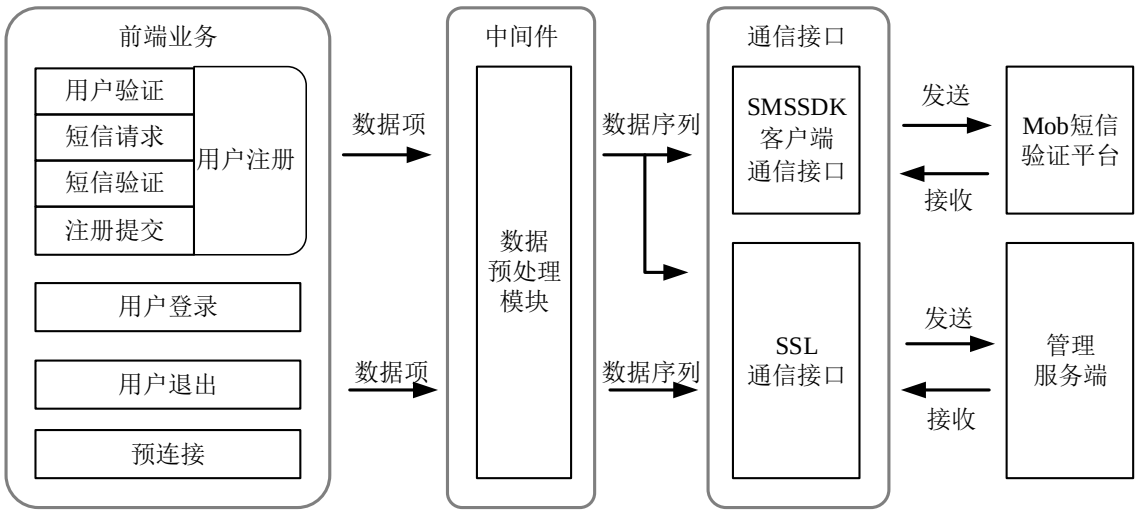


图 5-4 业务通信模块工作流程示意图

（1）前端业务

前端业务包括用户注册、用户登录、用户退出和预连接四个功能的逻辑处理。预连接指客户端连接代理服务端前，向管理服务器请求获取将要连接的代理服务器 IP 地址。由于代理服务器资源由管理服务端统一管理，因此每次客户端连接 VPN 前都需要进行预连接。在这四项前端业务中，用户注册功能步骤最复杂，交互实体最多且采用不同的通信接口，图 5-5 对用户注册流程进行详细描述。

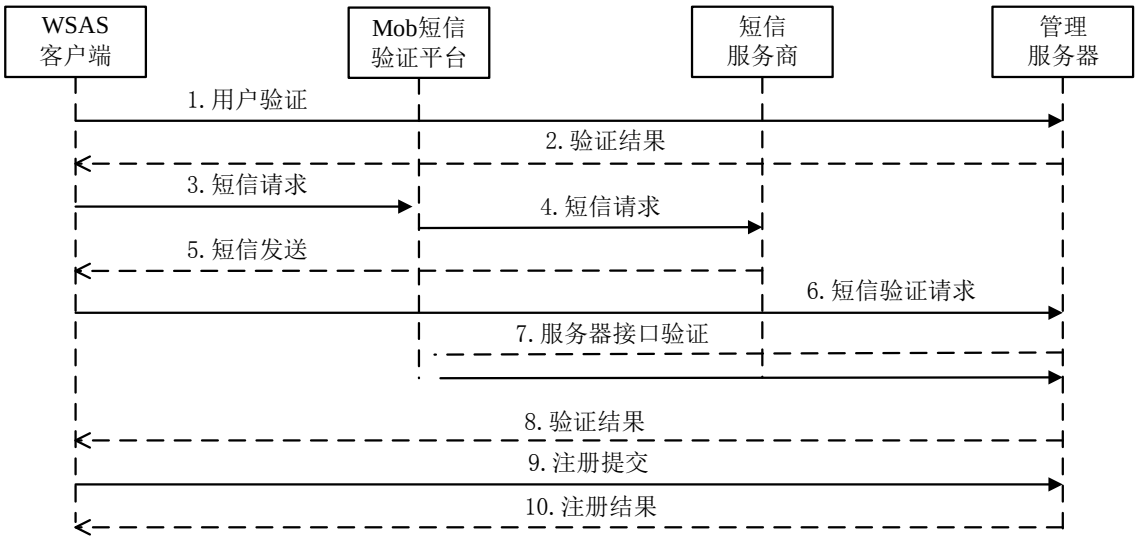


图 5-5 用户注册业务时序图

首先，客户端进行用户验证以确定当前用户未注册。接着，客户端向 Mob 短信验证平台短信请求，由其代向短信服务商请求下发验证码短信。客户端等待接收短信后，将验证码发往管理服务器，管理服务器使用 Mob 短信验证平

台第三方接口将数据发向该平台服务器验证。等待平台返回验证结果，管理服务端将验证结果回传至客户端完成验证。最后，客户端提交用户注册信息完成注册。

（2）中间件

中间件是主要是数据预处理模块，该模块处理前端业务的数据项并将其拼接成数据序列后，转交至通信接口。图 5-6 以用户注册交互事件为例给出相应的数据序列格式。

交互事件	数据格式	数据说明
用户验证	serviceCode + phoneNum	服务代码 + 手机号
验证结果	serviceCode + phoneNumResult	服务代码 + 验证结果
短信验证	serviceCode + phoneNum + verifyCode	服务代码 + 手机号 + 验证码
验证结果	serviceCode + verifyCodeResult	服务代码 + 验证结果
注册提交	serviceCode + phoneNum + passwordHash	服务代码 + 手机号 + 密码哈希
注册结果	serviceCode + registerResult	服务代码 + 注册结果

图 5-6 用户注册交互事件数据格式说明

（3）通信接口

业务通信模块的通信接口包括 SMSSDK 客户端通信接口和 SSL 通信接口，前者由 Mob 短信验证平台提供，后者由 Android SSLSocket API 的支持。用户注册流程比较复杂，同时涉及到这两个通信接口。其余功能均只使用 SSL 通信接口与管理服务端交互。

5.3 管理服务端设计与实现

管理服务端处于该原型系统的中心，负责移动终端的业务通信管理和代理服务端的资源通信管理，汇聚了移动用户的个人信息和代理系统的资源信息，因此通信安全是管理服务器首要考虑的问题。其次，由于管理服务器需要与移动终端直接通信处理注册、登录、退出等基本用户业务，所以数据存取效率和数据通信效率也是必须要考虑的因素。基于安全和效率的二者考虑，将管理服务端的模块和接口设计如图 5-7 所示。

管理服务端有三个接口和三个模块组成，包括 SMSSDK 服务端通信接口、SSL 通信接口、数据存取接口、数据预处理模块、业务通信管理模块和代理资

源管理模块。SMSSDK 服务端通信接口是 Mob 短信验证平台提供的短信验证码服务的服务端接口，用于管理服务端向 Mob 服务器验证客户端短信验证码的有效性。SSL 通信接口基于 Linux C Socket 和 OpenSSL 加密库实现，保障业务通信管理和代理资源管理的通信安全。数据存取接口基于 Linux C 和 MySQL 实现，对数据库的基本增删改查操作进一步封装，使得函数接口更佳简单易用。数据预处理模块在接收 SSL 通信接口回传的数据序列后，解析数据序列的服务类型，并将数据分流至相应的业务通信管理模块或代理资源管理模块。业务通信管理模块对 WSAS 移动终端用户的基本信息管理，包括用户注册、登录、退出等基本操作和 VPN 的预连接。代理资源管理模块对 WSAS 的后端代理服务器集群的系统资源、网络资源和系统日志集中管理，并根据当前代理服务器的负载状态提供移动终端最优的代理服务器线路。

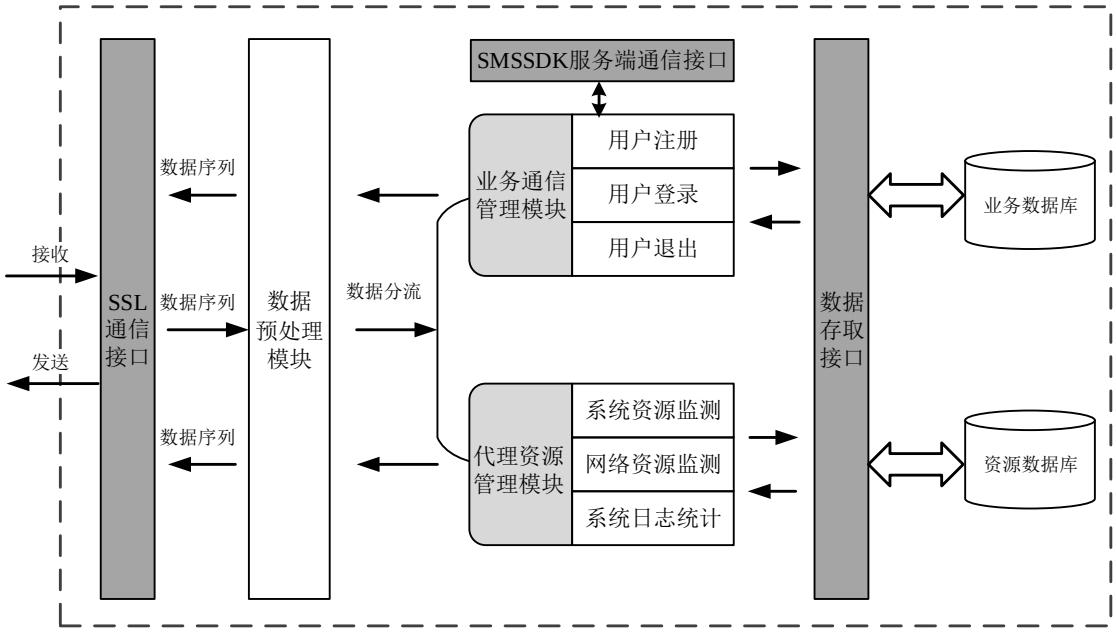


图 5-7 管理服务端整体结构

5.4 代理服务端设计与实现

图 5-8 描绘了代理服务端的模块工作流程，按照通信接口不同可以将其功能实现分为两部分：与移动终端的 VPN 通信和与管理服务器的管理网络通信。

在与移动终端的 VPN 通信方面，该端 VPN 通信接口处理与移动终端往来的流量，包括 CS-MVPN 握手协议流量和传输协议流量。对于握手协议流量，该端初始化 SSL 连接，身份认证模块与客户端进行双向身份认证，并建立标准 SSL 安全通道。随后，该端密钥交换模块与客户端在 SSL 安全通道内进行

安全 Diffie-Hellman 密钥交换，并且通信双方生成最终对称密钥。之后，该端 DHCP 模块为移动终端 MVPN 客户端分配虚拟 IP 地址。密钥交换模块和 DHCP 模块在标准 SSL 通道的保护下，为客户端生成了 VPN 隧道建立的必要参数，包括对称密钥、认证参数和虚拟 IP 地址。这些参数同样会被该端加载到 VPN 隧道管理模块，作为客户端 VPN 连接准入凭证。至此，SSL 安全通道完成所有握手阶段的交互工作。对于传输协议流量，直接由该端快速传输模块对来往流量快速加解密后转发至目标地址。

在与管理服务器的管理网络通信方面，该端 SSL 通信接口向管理服务端周期性上传 VPN 服务端的实时状态和运行日志。该端系统管理模块对系统资源和网络资源实时监测，系统资源包括 CPU 占用率和内存占用率，网络资源包括网络占用带宽、网络传输时延和丢包率。除此之外，该模块还实时记录 VPN 服务端的运行日志。上述该端的所有参数被周期性统计，并使用 SSL 通信接口发往管理服务端。

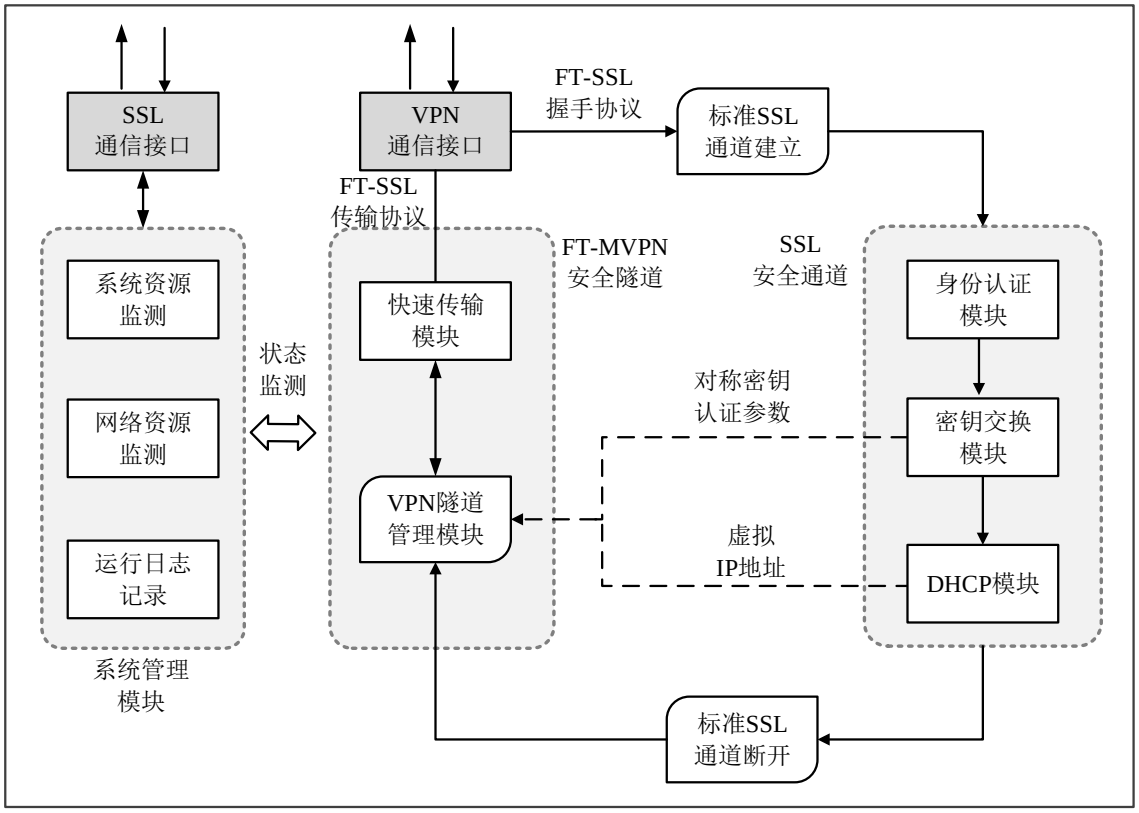


图 5-8 代理服务端模块工作示意图

5.5 测试与结果分析

本节内容主要对 WSAS 客户端进行系统测试，首先介绍整体测试环境，

随后对其分别进行功能测试、性能测试和系统测试，最后给出测试评价。

5.5.1 测试环境

实验环境的网络拓扑如图 5-1，所涉及硬件设备主要包括一台 Android 移动终端，一台 Linux 管理服务器和一台 Linux 代理服务器，具体配置参数信息如表 5-1 所示。

表 5-1 MVPN 稳定性实验的硬件设备配置表

设备类型	配置参数	配置描述
移动终端	设备型号	MI 2S
	CPU 型号	Snapdragon APQ8064 Pro 1.7GHz
	RAM 容量	2G
	操作系统	Android 4.1.1
管理服务器	CPU 型号	Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz
	内存容量	1G
	操作系统	CentOS Linux release 7.0.1406
代理服务器	CPU 型号	Intel® Xeon® CPU E5-2630 0 @ 2.30GHz
	内存容量	1G
	操作系统	CentOS Linux release 7.0.1406

其中，移动终端安装了 WSAS 客户端应用，WSAS 管理服务端和代理服务端程序分别部署在 Linux 管理服务器和 Linux 代理服务器上。

5.5.2 功能测试

功能测试对 WSAS 客户端的主要功能进行验证，包括用户注册功能、用户登录功能、VPN 连接功能和 Wi-Fi 控制功能。前三个功能的测试在图 5-9 所示的系统 UI 进行，Wi-Fi 控制功能的测试则在系统后台进行。

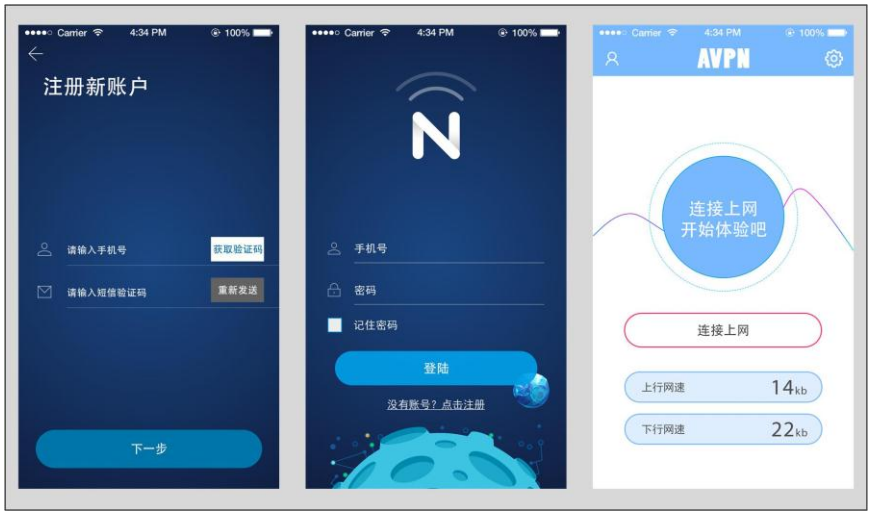


图 5-9 WSAS 客户端的主要功能 UI

（1）用户注册功能

用户注册需要经历手机号验证、验证码验证、密码长度验证三步过程，手机号验证需要向管理服务端发出请求，检测当前手机号是否注册；验证码验证也需要在管理服务端验证该验证码是否与第三方短信平台发放的验证码一致；密码长度验证则在移动终端自身完成。测试用例和测试结果如表 5-2 所示，其中合法的手机号长度、验证码长度和密码长度分别为11位、4位和6至20位。

表 5-2 WSAS 用户注册功能测试用例和测试结果表

测试子功能	用例描述	测试数据	预期结果	实际结果
手机号验证	长度不合法	1876631	验证失败	验证失败
	字符不合法	1876631379*	验证失败	验证失败
	手机号合法	18766313792	验证成功	验证成功
验证码验证	验证码不符	3792	验证失败	验证失败
	验证码相符	3797	验证成功	验证成功
密码长度验证	长度不符	123ab	注册失败	注册失败
	长度符合	1234abcd	注册成功	注册成功

用户注册功能的所有测试结果与预期均保持一致，因此该功能工作正常。

（2）用户登录功能

用户登录功能需要经历手机号验证和密码验证两步过程，两步均需要向管理服务端发出请求，前者验证当前手机号是否注册，后者验证用户登录密码是否正确。测试用例和测试结果如表 5-3 所示。

表 5-3 WSAS 用户登录功能测试用例和测试结果表

测试子功能	用例描述	测试数据	预期结果	实际结果
手机号验证	未注册的手机号	18766313791	验证失败	验证失败
	已注册的手机号	18766313792	验证成功	验证成功
密码验证	密码不符	1234abce	登录失败	登录失败
	密码相符	1234abcd	登录成功	登录成功

用户登录功能的所有测试结果与预期均保持一致，因此该功能工作正常。

（3）VPN 连接功能

用户在 WSAS 客户端 UI 点击“连接上网”按钮，即可使用 MVPN 正常上网。移动终端所有其他应用的数据流量均通过 MVPN 安全隧道进行发送和接收，因此 VPN 连接功能正常。

（4）Wi-Fi 控制功能

Wi-Fi 控制功能测试在程序后台进行，测试接入 Wi-Fi 信号的参数提取、白名单比对和 MVPN 通信组件的开关。测试结果表明：接入非白名单 Wi-Fi 热点时，WSAS 系统后台能自动开启 MVPN 通信组件，保护客户端的数据流量。

5.5.3 性能测试

客户端的性能测试包括传输时延测试和网速测试两方面，分别测试运行 WSAS 客户端应用和未运行 WSAS 客户端时的传输时延情况和网速情况。

传输时延的统计借助于 Android 底层 Linux 系统的 ping 工具。首先使用 adb shell 命令连接移动终端，随后在运行 WSAS 客户端应用和未运行 WSAS 客户端应用两种情况下分别使用 ping 命令统计 10 次通信双方的传输时延，统计结果如图 5-10 所示。测试结果表明，除了第三次运行 WSAS 应用时延稍大外，两种通信方式下时延基本相同。通过计算平均传输时延，运行 WSAS 应用的平均传输时延为 26.48ms, 未运行 WSAS 应用的平均传输时延为 25.03ms, 二者仅相差约 1ms，因此使用 WSAS 客户端不会对传输时延造成明显影响。

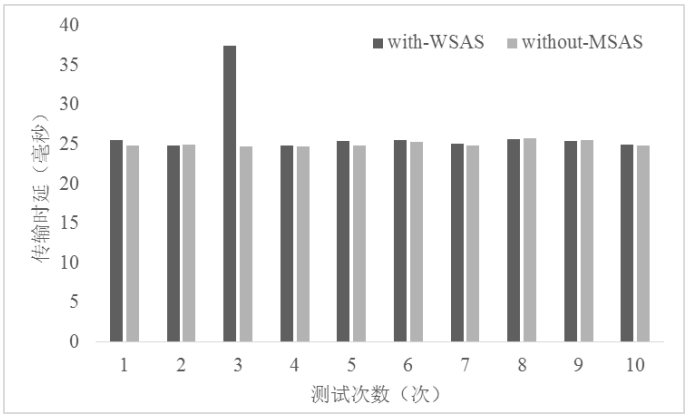


图 5-10 传输时延对比结果

网速情况的统计则通过编写 Android 移动终端的实时网速监控程序实现，网速监控程序每隔 10s 对当前网速采样一次。通过播放相同视频保证相同背景流量，先后测试运行 WSAS 客户端应用和未运行 WSAS 客户端应用两种情况下的移动终端的网速情况，统计结果如图 5-11 所示。

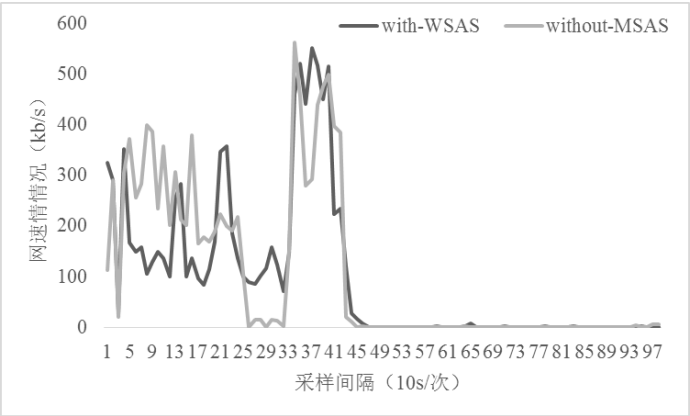


图 5-11 网速情况对比结果

从视频的加载时间来看，未运行 WSAS 客户端应用约在 450 秒加载完整个视频，运行 WSAS 客户端应用约在 455 秒加载完整个视频；从网速的峰值来看，未运行 WSAS 客户端应用网速峰值约为 560kb/s, 运行 WSAS 客户端应用网速峰值约为 550kb/s。因此，使用 WSAS 客户端应用对移动终端的网速影响不大。

5.5.4 系统测试

客户端的系统测试重点考察运行 WSAS 客户端应用和未运行 WSAS 客户端应用的两种不同情况下，移动终端操作系统的 CPU 占用率、内存占用率和整体耗电情况。两次测试期间，除了运行 WSAS 客户端应用的差异之外，需保证两次测试 Android 操作系统的其他后台进程一致。两次测试均采用后台静默运行的方式，直至客户端因电量损耗完毕而停止统计，得到系统的 CPU 占用率、内存占用率和耗电情况的变化曲线，如图 5-12 至 5.14 所示。

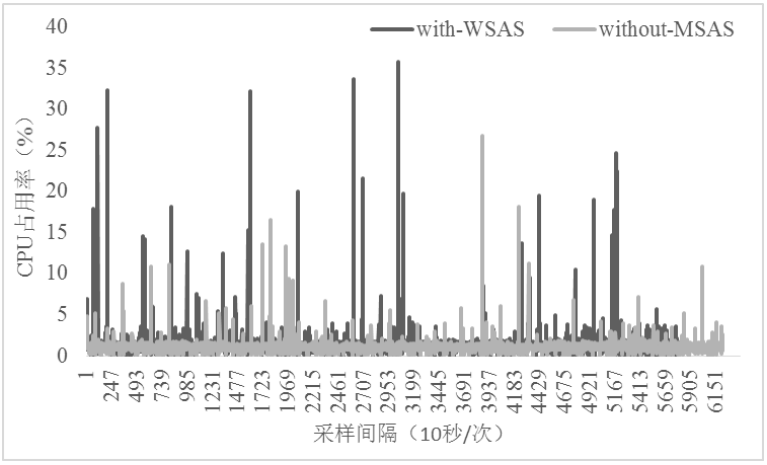


图 5-12 CPU 占用率对比结果

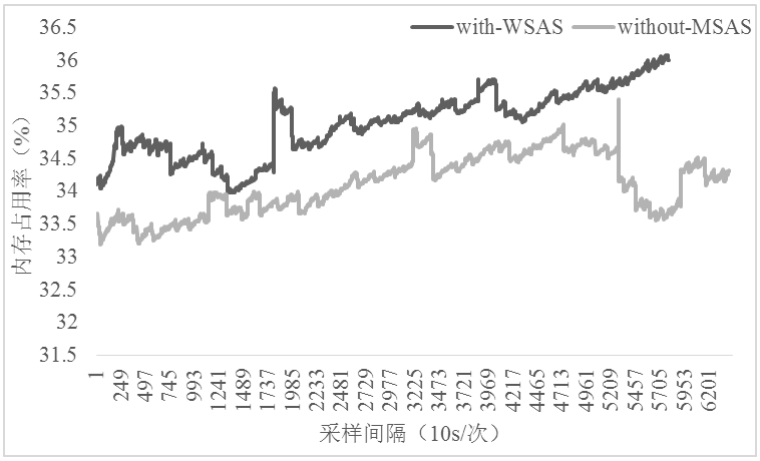


图 5-13 内存占用率对比结果

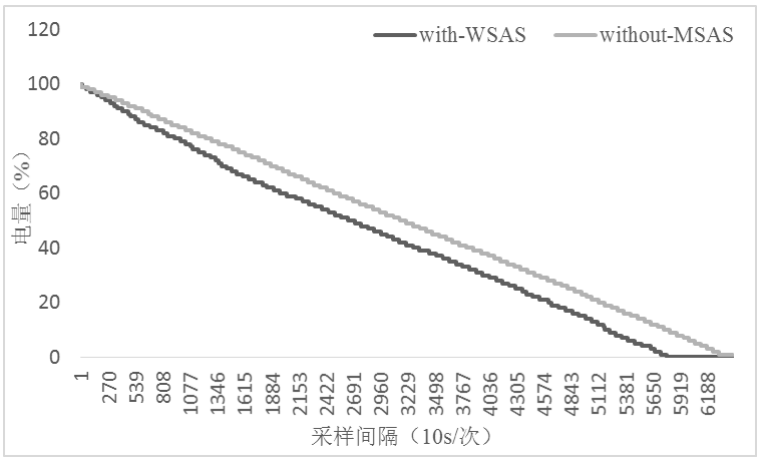


图 5-14 电量消耗对比结果

从图 5-12 的 CPU 占用的程度来看，运行 WSAS 客户端应用的平均 CPU 占用率为 1.26%，未运行 WSAS 客户端应用的平均 CPU 占用率为 1.15%，二者 CPU 占用率相差无几。从图 5-13 的内存占用的程度来看，运行 WSAS 客户端应用的平均内存占用率为 35.04%, 未运行 WSAS 客户端应用的平均内存占用率为 34.10%，二者内存占用率相差不大。从图 5-14 的电量消耗的程度来看，运行 WSAS 客户端应用的待机时间约 57920 秒，未运行 WSAS 客户端应用的待机时间约 64410 秒，前者比后者的待机时间缩短了 6490 秒，增加了移动终端约 10.08%的电量消耗，因此 WSAS 客户端的电量消耗有待进一步优化。

5.5.5 测试评价

WSAS 原型系统在实际应用环境下进行了功能测试、性能测试和系统测试。功能测试结果表明，WSAS 客户端可以提供移动终端稳定快速的 MVPN 安全隧道通信；性能测试结果表明，该 MVPN 安全隧道在传输时延和网速情况方面的性能表现与 Wi-Fi 直接接入相差无几；系统测试结果表明，该原型系统占用系统资源较低，但电量消耗问题比较明显，相比于未使用 WSAS 客户端应用的移动终端增加了 10.08%的电量消耗，因此该系统的电量消耗问题有待优化改进。

5.6 本章小结

本章介绍了课题原型系统 WSAS 的整体架构和系统主要参与实体的模块设计，并着重对 WSAS 客户端进行功能测试、性能测试和系统测试，测试结果表明 WSAS 功能满足需求，稳定性和性能达到指标，但电量消耗问题有待进一步优化。

结　论

Wi-Fi 安全接入的需求是针对无线网络攻击泛滥和 Wi-Fi 接入需求广泛的背景提出的，因此选取一种 Wi-Fi 安全接入技术对于移动用户网络流量和个人隐私的保护至关重要。本课题选择 MVPN 技术展开研究，改进了该技术在传输速度和通信连接稳定性方面的不足，并提出了基于 MVPN 的移动终端 Wi-Fi 安全接入系统。本文主要完成了以下工作：

（1）研究了 MVPN 的虚拟隧道技术和 Android 平台虚拟隧道的创建方式和工作方式，实现了简单隧道 MVPN。其功能测试结果表明：该虚拟隧道能够发送和接收移动终端的所有流量，并且未对移动终端的网速造成影响。

（2）研究了 MVPN 的安全通信协议，指出了现有 MVPN 通信协议的不足，给出了 MVPN 安全协议的设计目标，提出了一种基于对称密钥体制的分层通信协议—快速传输隧道协议，并将新型的 AEAD 加密算法应用于协议的数据加密，最后基于该协议实现了快速传输 MVPN。其性能测试表明：该安全隧道较 OpenVPN 相比具有更快的传输速度和更低的传输时延。基于该创新点，本课题发表了国家发明专利《一种适用于移动设备的 VPN 协议》。

（3）研究了 MVPN 通信连接的稳定性，结合理论分析和 MVPN 应用调研分析，阐明了现有 MVPN 应用稳定性不足的根本原因，提出了一种通用的 MVPN 的通信保障机制，并给出其形式化定义和数学模型。随后本文提出了 MVPN 通信保障模型和其原型实现—通信保障 MVPN。稳定性测试结果表明：通信保障模型的稳定性达到 96.67%。同时，该模型还具有一定的通用性和可扩展性。基于该研究内容的研究成果和创新点，本课题发表了国际会议论文《A Communication Supportable Generic Model for Mobile VPN on Android OS》和国家发明专利《一种移动网络复杂环境下的 VPN 通信能力保障方法》。

（4）研究了 MVPN 技术与 Wi-Fi 安全接入的有机结合方案，将上述改进的安全、快速、稳定的 MVPN 技术应用于 Wi-Fi 安全接入的应用场景，设计了 Android 移动终端的 Wi-Fi 安全接入系统，给出了系统整体架构、系统客户端核心组件设计和系统主要参与实体的模块设计。系统测试结果表明：该系统功能能够满足移动终端 Wi-Fi 安全接入的需求，整体性能满足预期目标，但电量消耗问题有待进一步优化。基于该创新点，本课题发表了国家发明专利《一种公共 WiFi 的安全接入系统》。

本文主要研究了 MVPN 的虚拟隧道技术、安全通信协议和通信保障模型，

改进了 MVPN 的传输速度和稳定性，实现了基于 MVPN 的 Wi-Fi 安全接入系统。根据本课题研究、实现和实验中遇到的问题，未来工作可以从以下两个方面继续开展：

（1）理论上，对本文提出的快速传输隧道协议进行形式化分析与验证；实际中，模拟网络攻击行为攻击该通信模型。从理论和实际两方面进一步验证该通信协议的安全性。

（2）现有的 Wi-Fi 安全接入系统的耗电量值得进一步优化，主要从客户端代码、通信模型和协议算法三个角度进行改进。

参考文献

[1] 中国互联网络信息中心. 第 37 次中国互联网发展状况统计报告[R]. (2016-01). https://cnnic.cn/gywm/xwzx/rdxw/2015/201601/W020160122639198410766.pdf.

[2] 360 天巡实验室. 2015 企业无线网络安全报告[R]. (2015-07-10). http://www.freebuf.com /articles/wireless/72084.html.

[3] Kindberg T, Bevan C, Woodgate D, et al. Authenticating ubiquitous services: A study of wireless hotspot access[C]//11th ACM International Conference on Ubiquitous Computing. ACM, 2009: 115-124.

[4] 陈伟,顾杨,李晨阳, 等. 无线钓鱼接入点攻击与检测技术研究综述[J]. 武汉大学学报(理学版), 2014, 60(1): 13-23.

[5] Nikbakhsh S, Manaf A B A, Zamani M, et al. A Novel Approach for Rogue Access Point Detection on the Client-Side[C]// 2012 26th International Conference on Advanced Information Networking and Applications Workshops. IEEE, 2012: 684-687.

[6] Lee C, Shen C, Sahin G, et al. A Novel and Scalable Communication-History-Based Knapsack Authentication Framework for IEEE 802.11 Networks[C]// 2015 IEEE Conference on Communications and Network Security. IEEE, 2015: 44-52.

[7] Sanatinia A, Narain S, Noubir G. Wireless spreading of WiFi APs infections using WPS flaws: An epidemiological and experimental study[C]// 2013 2013 IEEE Conference on Communications and Network Security. IEEE, 2013: 430-437.

[8] Adrian D, Bhargavan K, Durumeric Z, et al. Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice[C]// Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, 2015: 5-17.

[9] Bhargavan K, Leurent G. Transcript Collision Attacks: Breaking Authentication in TLS, IKE, and SSH[C]// Network and Distributed System Security Symposium. IEEE, 2016: 21-24.

[10]Liang J J, Jiang J, Duan H X, et al. When HTTPS Meets CDN: A Case of Authentication in Delegated Service[C]// 2014 IEEE Symposium on Security and Privacy. IEEE, 2014: 67-82.

[11]Berger T. Analysis of current VPN technologies[C]// International Conference on Availability, Reliability and Security. IEEE, 2006: 108-115.

[12]Naylor D, Finamore A, Leontiadis I, et al. The Cost of the "S" in HTTPS[C]// Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies. ACM, 2014: 133-140.

[13]Haase S, Seeling P. SOCKx—An application layer network switching framework using SOCKSv5 protocol extensions[C]// International Conference on Electro/Information Technology. IEEE, 2011: 1-4.

[14]Chaabane, A, Manils, P, Kaafar, M.A. Digging into Anonymous Traffic: A Deep Analysis of the Tor Anonymizing Network[C]// International Conference on Network & System Security. IEEE, 2010: 167-174.

[15]Shu M, Tan C, Wang H. Mobile VPN Scheme Based on SOCKS V5[C]// International Conference on Machine Vision and Human-Machine Interface. 2010: 792-795.

[16]Uskov A V. Information security of mobile VPN: Conceptual models and design methodology[C]// International Conference on Electro/Information Technology (EIT). IEEE, 2012: 1-6.

[17]Uskov A V. Information Security of IPsec-based Mobile VPN: Authentication and Encryption Algorithms Performance[C]// International Conference on Trust, Security and Privacy in Computing and Communications. IEEE, 2012:1042-1048.

[18]Liyanage M, Gurtov A. Secured VPN Models for LTE Backhaul Networks[C]// 2012 IEEE Vehicular Technology Conference, 2012: 1-5.

[19]Lakbabi A, Orhanou G, El Hajji S. VPN IPSEC & SSL technology Security and management point of view[C]// Next Generation Networks and Services. IEEE, 2012: 202-208.

[20]Mao H Q, Zhu L, Qin H. A Comparative Research on SSL VPN and IPSec VPN[C]// 2012 8th International Conference on Wireless Communications, Networking and Mobile Computing. IEEE, 2012: 1-4.

[21]Yu D, Chen N. The Improving of IKE with PSK for Using in Mobile Computing Environments[C]// International Conference on Information Assurance & Security. IEEE, 2009: 331-334.

[22]Kim H M, Yoo J S. An effective calibration of VOIP internet telephony performance using VPN between PAC and PNS[C]// Global Mobile Congress.

IEEE, 2009: 82-89.

[23]Zuquete A, Frade C. Fast VPN mobility across Wi-Fi hotspots[C]// International Workshop on Security and Communication Networks. IEEE, 2010: 1-7.

[24]Oya T, Kamiyama H, Miyoshi J, et al. Evaluation of the route optimization architecture for MOBIKE-based mobile communication systems[J]. Ieice Technical Report Information Networks, 2010, 109: 55-58.

[25]Liyanage M, Gurtov A. Secured VPN Models for LTE Backhaul Networks[C]// 2012 IEEE Vehicular Technology Conference. IEEE, 2012: 1-5.

[26]Yu D, Chen N, Tan C. Design and Implementation of Mobile Security Access System (MSAS) Based on SSL VPN[C]// International Workshop on Education Technology and Computer Science. IEEE, 2009: 152-155.

[27]Hong Y R, Kim D. Security Enhancement of Smart Phones for Enterprises by Applying Mobile VPN Technologies[C]// Computational Science and ITS Applications - Iccsa 2011 - International Conference. Proceedings. 2011:506-517.

[28]Liao J M, Zhang H L, Zhan S Y, et al. Analysis and design of VPN based on wireless Android intelligent home center[C]// International Computer Conference on Wavelet Active Media Technology and Information Processing. IEEE, 2013: 251-254.

[29]Kilinc C, Booth T, Andersson K, et al. WallDroid: Cloud Assisted Virtualized Application Specific Firewalls for the Android OS[C]// Proceedings of the 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications. IEEE Computer Society, 2012: 877-883.

[30]Anastasia S, Anh L, Minas G, et al. Demo: AntMonitor: A System for Mobile Traffic Monitoring and Real-Time Prevention of Privacy Leaks[C]// Proceedings of the 21st Annual International Conference on Mobile Computing and Networking. ACM, 2015: 170-172.

[31]Choi B, Choi S K, Cho K. Detection of Mobile Botnet Using VPN[C]// Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. IEEE, 2013: 142-148.

[32]Alshalan A, Pisharody S, Huang D. A Survey of Mobile VPN Technologies[J]. IEEE Communications Surveys & Tutorials, 2015, 72(5): 1-1.

[33]Hu M, Zhao Q, Kuramoto M, et al. Research and implementation of Layer Two Tunneling Protocol (L2TP) on carrier network[C]// 2011 4th IEEE International

Conference on Broadband Network and Multimedia Technology. IEEE, 2011: 80-83.

[34]Bruce S, Mudge, David W. Cryptanalysis of Microsoft's PPTP Authentication Extensions (MS-CHAPv2)[C]// Proceedings of the International Exhibition and Congress on Secure Networking - CQRE (Secure) '99. ACM, 1999: 782-782.

[35]Qu J, Li T, Dang F. Performance Evaluation and Analysis of OpenVPN on Android[C]// Fourth International Conference on Computational and Information Sciences. 2012: 1088-1091.

[36]Anupam Datta, Ante Derek, John C. Mitchell, et al. Secure Protocol Composition[J]. Electronic Notes in Theoretical Computer Science, 2003, 83: 11-23.

[37]Mihir Bellare, Chanathip Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm[M]// Advances in Cryptology — ASIACRYPT 2000. Springer Berlin Heidelberg, 2000: 469-491.

[38]Langley A. ChaCha20 and Poly1305 for IETF Protocols[S]. USA: IETF, 2015-05. https://tools.ietf.org/html/rfc7539.

[39]Tzvetkov V D. Virtual Private Networks for mobile environments. Development of protocol for mobile security and algorithms for location update[D]. Technische Universität Darmstad: Ph.D. Thesis, 2010.

[40]Goff T, Moronski J, Phatak D S, et al. Freeze-TCP: a true end-to-end TCP enhancement mechanism for mobile environments[J]. Proc IEEE Infocom Mar, 2010, 3(3): 1537-1545.

[41]Eronen P. IKEv2 Mobility and Multihoming Protocol (MOBIKE) [S]. USA: IETF, 2006-06. https://www.ietf.org/rfc/rfc4555.txt.

[42]Dutta A, Schulzrinne H. Mobility Protocols and Handover Optimization: Design, Evaluation and Application[J]. Wiley-IEEE Press, 2014.

[43]Liu Z H, Chen J C, Chen T C. Design and analysis of SIP-based mobile VPN for real-time applications[J]. IEEE Transactions on Wireless Communications, 2009, 8(11): 5650-5661.

攻读硕士学位期间发表的论文及其他成果

（一）发表的学术论文

[1] Fu C L, He Q G, Wang B L, et al. A Communication Supportable Generic Model for Mobile VPN on Android OS[C]// The Twenty-First IEEE Symposium on Computers and Communications. IEEE, 2016. (Accepted)

（二）申请及已获得的专利

[1] 傅春乐，何清刚，孙云霄，王佰玲，刘扬，张昭. 一种公共 WiFi 的安全接入系统：中国，201510874872.5 [P]. 2015-12-04.

[2] 傅春乐，何清刚，孙云霄，王佰玲，刘扬，张昭. 一种适用于移动设备的 VPN 协议：中国， 201510874885.2 [P]. 2015-12-04.

[3] 傅春乐，何清刚，王佰玲，刘扬，陈彬，张昭. 一种移动网络复杂环境下的 VPN 通信能力保障方法：中国，201610202955.4[P]. 2016-04-05.

哈尔滨工业大学学位论文原创性声明和使用权限

学位论文原创性声明

本人郑重声明：此处所提交的学位论文《Android 移动终端 Wi-Fi 安全接入关键技术的研究与实现》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：  日期：2016 年 6 月 19 日

学位论文使用权限

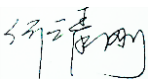
学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

（1）学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；（2）学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；（3）研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。

本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：  日期：2016 年 6 月 19 日

导师签名：  日期：2016 年 6 月 19 日

致　谢

又是一年毕业季，却是不同离别人。本科毕业的那一幕似乎还在眼前，转眼之间，研究生离别的那一刻也不再遥远。两年间紧张忙碌的学习生活给予了我许多宝贵的财富，见证了我的收获与成长。饮水思源，一切财富的根源则是那些与我一路同行的长者、伴侣、知己，在此向你们表示最诚挚的感谢。

首先，要感谢我的导师何清刚副教授。自本科阶段认识何老师以来，已经过去六年的时光。从最初的课余交流到如今的亲自指导，何老师至始至终以一位长者的姿态，关心我的学习生活，宽容我的缺点不足，指导我的学术科研，引领我的前进方向。一日为师，终身为父，我永远不会忘记您的谆谆教诲。

特别感谢王佰玲教授。研究生期间，王老师一直关心我的科研成果和工程进度，总能在我迷茫无措时为我指点迷津，在我束手无策时为我提供思路，在我毫无动力时为我调整状态。王老师一直以朋友的姿态，与我一起钻研工作难题，探讨科研方案，交流学术心得，激励着我在科研道路上勇攀高峰。

特别感谢孙云霄学长。孙学长作为项目带头人，在工程实践上经验丰富、思维活跃、技术非凡，总能在我遇到技术瓶颈时指导我解决问题；在日常工作中关心组员、建设团队，为实验室的发展尽心尽力。

特别感谢网络技术研究所团队的所有老师和同学。感谢刘扬老师、辛国栋老师在项目工程上的指导和帮助，感谢韩希先老师、宋佳老师、黄俊恒老师在学术科研上经验和教训的传授，感谢王孝朋老师、戚晓红老师在实验室日常生活上的关心和呵护，感谢李超学长、吕芳学姐、王锐同学、顾建伟同学、穆瑞超同学、吴昊同学在学习工作中对我的支持和帮助。

特别感谢计算机学院领导老师对研究生的关心和照顾，感谢黄蕊副书记、苑新玲老师在党建工作和教学工作中的辛勤付出。

特别感谢我的家人对我的理解和支持，特别是我的父母和爷爷奶奶，对我无时不刻的挂念和关心，鼓励着我在学业上不断勇攀高峰，我唯有不断努力才能报答你们对我无私的爱。特别感谢张昭，已经与我一同走过 3 年时光，陪伴我在学习工作中相互激励，在生活中相互照顾，互相成就最好的彼此。即使未来不可预知，但未来会来，有你有我，最终成为彼此的灵魂伴侣。

最后，感谢论文评阅专家对本文提出宝贵的意见。