

分类号_____

学校代码 10487

学号 M201871875

密级_____

华中科技大学

硕士学位论文

WiFi 系统中投屏与游戏业务传输质量
保证研究

学位申请人：郭志斌

学 科 专 业：信息与通信工程

指 导 教 师：吴伟民 副教授

答 辩 日 期：2021 年 5 月 日

**A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree for the Master of Engineering**

**Research on Transmission Quality Assurance of Screen
Casting and Game Services in WiFi System**

Candidate : Guo Zhibin

Major : Information and Communication Engineering

Supervisor : Assoc.Prof. Wu Weimin

Huazhong University of Science & Technology

Wuhan 430074, P.R.China

May, 2021

独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期：2021 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本论文属于 ☐ 保密， ☐ 在 _____ 年解密后适用本授权书。
☐ 不保密 ☒。

（请在以上方框内打“√”）

学位论文作者签名：

日期：2021 年 月 日

指导教师签名：

日期：2021 年 月 日

摘 要

投屏业务与游戏业务是 WiFi 场景下非常重要的业务，投屏技术实现了多设备之间的画面共享和数据互通，游戏业务则丰富了人们的业余生活，而 WiFi 技术的快速发展，使人们能够更加便捷地使用投屏技术和享受游戏乐趣。

但是，无论是投屏还是游戏，都对业务报文的传输时延、丢包率和吞吐率提出了极高的要求。在现有的 WiFi 场景下，当存在多业务并发、多设备共存时，WiFi 链路上的信道竞争十分激烈，使得投屏和游戏业务的报文传输出现时延、丢包等问题，给人们带来了糟糕的使用体验。

针对投屏业务在 WiFi 链路上的高质量传输需求，本文基于投屏技术原有的传输机制和报文特性，在无线链路底层设计了全新的投屏报文传输方案，包括使用反馈重传方案来保证投屏报文的传输可靠性，并使用调度传输方案以实现投屏业务的低时延快速传输，以及基于空口质量感知进行投屏报文的视频帧编码，来提高投屏业务的传输吞吐率，还将蓝牙信道与 WiFi 信道进行异构融合，实现投屏业务的全双工传输。

与此同时，针对游戏业务在 WiFi 链路上的高质量传输需求，在硬件上支持双频双发功能的移动终端设备上，本文充分利用了双频多队列的传输资源，设计了高效的报文封装机制，用于实现报文的调度发送和反馈重传功能，从而保证游戏业务的传输可靠性；紧接着基于封装机制和反馈信息设计了双频下各个硬件队列的传输质量测量机制，并基于测量得到的链路质量信息，进一步设计了双频调度发送机制和动态流量调度机制，共同保证游戏业务在复杂 WiFi 环境下的低时延传输。

针对投屏业务和游戏业务传输质量保证所提出的设计方案，本文均在 Linux 平台和 Windows 平台进行了架构设计和代码实现，并对相关实现进行了性能验证。验证结果证明，经过本文提出的质量保证机制对业务报文进行处理后，投屏业务和游戏业务的传输时延波动得到明显改善，报文丢包率得到降低，业务在 WiFi 链路上的传输质量成功得到保证。

关键词： 802.11 协议 无线投屏 传输质量保证 反馈重传 流量调度

Abstract

Screen projection business and game business are very important businesses in the WiFi scenario. Screen projection technology enables screen sharing and data exchange between multiple devices. Game business enriches people's leisure life. The rapid development of WiFi technology makes people Can use the projection technology more conveniently and enjoy the fun of the game.

However, whether it is a screencast or a game, extremely high requirements are put forward on the transmission delay, packet loss rate, and throughput rate of service messages. In the existing WiFi scenario, when there are multiple services concurrently and multiple devices coexist, the channel competition on the WiFi link is very fierce, causing problems such as delay and packet loss in the transmission of messages for screen projection and game services. Brings a bad user experience.

In view of the high-quality transmission requirements of the projection service on the WiFi link, this paper designs a new projection message transmission scheme at the bottom of the wireless link based on the original transmission mechanism and message characteristics of the projection technology, including the use of feedback Transmission scheme to ensure the transmission reliability of the projection message, and use the scheduling transmission scheme to achieve low-latency and rapid transmission of the projection service; and based on the air interface quality perception to encode the video frame of the projection message to improve the projection service High transmission throughput rate; and the heterogeneous integration of Bluetooth channel and WiFi channel to achieve full-duplex transmission of screen projection services.

At the same time, in response to the high-quality transmission requirements of game services on WiFi links, this article makes full use of the transmission resources of dual-frequency and multi-queue on mobile terminal devices that support dual-frequency and dual-transmit functions on the hardware, and designs an efficient report. The message encapsulation mechanism is used to realize the function of message scheduling and feedback retransmission, thereby ensuring the transmission reliability of game services; then based on the encapsulation mechanism and feedback information, the transmission

quality measurement mechanism of each hardware queue under dual frequency is designed, and Based on the measured link quality information, a dual-frequency scheduling transmission mechanism and a dynamic traffic scheduling mechanism are further designed to jointly ensure the low-latency transmission of game services in a complex WiFi environment.

Aiming at the design ideas proposed for the transmission quality assurance of the screencasting business and the game business, this article has carried out the architecture design and code implementation on the Linux platform and the Windows platform, and performed performance verification of the related implementations. The verification result proves that after the service message is processed by the quality assurance mechanism proposed in this article, the transmission delay fluctuation of the screen projection service and the game service is significantly improved, the packet loss rate is reduced, and the service transmission on the WiFi link Quality success is guaranteed.

Key Words: 802.11 agreement Wireless screen cast Transmission quality assurance Feedback retransmission Traffic scheduling

目 录

摘 要.....	I
Abstract.....	III
目 录.....	V
第 1 章 绪论.....	1
1.1 研究背景与意义.....	1
1.2 国内外研究概况.....	2
1.3 论文的结构与安排.....	4
第 2 章 影响投屏业务与游戏业务的无线传输机制概述	6
2.1 WiFi 下的传输机制.....	6
2.2 无线投屏技术.....	12
2.3 游戏业务传输技术.....	15
2.4 驱动开发技术.....	15
2.5 章节小结	18
第 3 章 投屏业务传输质量保证的设计	19
3.1 技术路线分析.....	19
3.2 整体架构介绍.....	21
3.3 反馈重传方案.....	22
3.4 调度传输方案.....	37
3.5 基于空口质量感知的视频帧编码.....	42
3.6 异构融合反馈重传机制.....	50
3.7 本章小结	53
第 4 章 投屏业务平台实现与测试	54

4.1	软件设计实现.....	54
4.2	反馈重传方案验证.....	59
4.3	调度传输方案验证.....	61
4.4	空口质量感知功能验证.....	62
4.5	章节小结.....	63
第 5 章	游戏业务传输质量保证的设计	65
5.1	技术路线分析.....	65
5.2	高效的报文封装机制.....	67
5.3	无线链路 KPI 测量机制	70
5.4	双频多队列下的调度发送机制.....	73
5.5	动态流量调度机制.....	78
5.6	章节小结.....	82
第 6 章	游戏业务平台的实现与测试	83
6.1	软件架构与设计实现.....	83
6.2	无线链路 KPI 测量功能验证	87
6.3	双频链路下的调度传输测试.....	88
6.4	章节小结.....	90
第 7 章	总结与展望.....	91
7.1	全文总结	91
7.2	课题展望	92
致 谢	93
参考文献	94

第1章 绪论

1.1 研究背景与意义

伴随着 WiFi 技术的不断发展，无线投屏的使用也正逐渐推广开来。无线投屏技术结合了智能手机的移动便携优势、智能电视的大屏显示优势和个人电脑的高效办公优势，给人们带来了更为丰富的智能设备使用体验，如商务投屏演示，家庭投屏影视，多屏协同办公等。但是，投屏技术对传输链路的时延、丢包率和带宽提出了极高的要求，传输时延太大会导致信源和信宿画面出现肉眼可见的迟滞，并会使得投屏画面和音频难以同步；链路丢包则会使投屏画面出现卡顿、黑屏和花屏等现象；带宽更是直接影响了投屏的画质。因此，针对无线投屏场景进行链路级的传输质量保证，对投屏使用体验的提升至关重要。

与此同时，随着智能手机、笔记本电脑等便携式移动终端设备的广泛普及和使用，承载在 WiFi 链路上的游戏业务场景也已随处可见。新一代的智能手机和笔记本电脑等设备所使用的无线网卡在硬件上逐渐都已支持双频双发功能，但是目前依旧缺少一套通用成熟的方案对不同优先级的业务进行链路层的合理调度。因此，当面临多业务并发、多设备共存的复杂无线场景时，由于游戏业务的传输未能充分利用到双频链路资源，业务报文在信道上的竞争产生了更高的时延与丢包率，最终使得游戏体验变得十分糟糕。

本论文基于实验室与某知名企业的项目合作研究，目的是在自研芯片和平台上，面对复杂无线场景，基于投屏技术原有的传输机制和报文特性，实现无线投屏业务的低时延、高可靠、高吞吐率传输，以及保证游戏业务在双频 WiFi 链路上的高优先级、高质量传输。在当今复杂的国际环境下，自主制定协议和自主研发技术已成为国家的重点研究方向，对国产品牌的智能手机、笔记本电脑等自主研发设备进行技术功能开发同样至关重要。因此，通过在自研国产设备上实现 WiFi 下各业务传输质量保证的技术研究并实现设计开发，从而提高国产品牌的产品竞争力，对国家发展具有重大意义。

1.2 国内外研究概况

1.2.1 WiFi 技术的发展现状

由于移动互联网的深入普及和无线网络服务的不断丰富,近年来对无线业务流量和服务质量(Qos)的需求急剧增加^[1]。无线局域网(WLAN)以其速度快、部署灵活、成本低等优点,与蜂窝网络一起成为承载无线网络业务的主要网络^[2]。

从20世纪90年代开始,IEEE开始了IEEE 802.11无线局域网的标准化工作。802.11规定了WLAN的物理层(PHY)和介质访问控制(MAC)层,已成为目前应用最广泛的WLAN标准^[3]。

1999年发布的IEEE 802.11a^[4]采用了正交频分复用(OFDM)技术,其最大传输速率达到了54Mbps。同年,IEEE 802.11b^[5]采用直接序列扩频(DSSS),最大传输速率为11Mbps。2003年,IEEE 802.11g^[6]将OFDM引入到2.4GHz频段,最大传输速率也达到了54Mbps。2009年发布的IEEE 802.11n^[7]是一个里程碑,它引入了块确认(CB)和单用户(SU)MIMO(SU-MIMO)技术,最大传输速率可达600Mbps。与以前的版本相比,IEEE 802.11n的最大传输速率增加了10倍以上。2013年,IEEE 802.11ac^[8]引入了256正交幅度调制(256-QAM)和CB增强技术,从而达到了最大传输速率6.93Gbps。IEEE 802.11ax^[9]通过引入多用户MAC(mu-MAC)、空间复用(SR)和目标唤醒时间(TWT),显著增强了MAC层技术,提高了高密度部署场景下的接入效率和用户体验。另一方面,IEEE 802.11ax也通过引入1024-QAM、正交频分多址(OFDMA)、上行(UL)MU-MIMO、增强CB等技术,实现了最大传输速率达9.6Gbps。

1.2.2 投屏技术的发展现状

无线投屏常常又被人们称作无线同屏、屏幕共享等,其实现的功能是在无线链路上,将某一可显示设备A的屏幕画面传输到另一个可显示设备B上,使设备B的屏幕上能实时呈现设备A的屏幕内容,更进一步的功能则支持在设备B上对设备A进行远程操作。在商务投屏演示、家庭投屏影视、多屏协同办公等场景上都存在着对该技术的使用。

目前,主流的无线投屏协议主要有三种,它们分别为苹果的AirPlay Mirroring技

术^[10]，索尼、英特尔、微软等使用的 DLAN 技术^[11]和 Wi-Fi 联盟定制的 Miracast 技术^[12]。

AirPlay Mirroring 是苹果公司私有的无限互动技术。该技术基于苹果公司的 AirPlay 协议，能够将苹果公司生产的手机、平板等设备的屏幕画面通过 WiFi 链路传输到支持 AirPlay 的显示设备上显示。由于苹果公司并未将 AirPlay 协议开源，目前只有苹果公司的设备和少数得到苹果公司授权的设备支持 AirPlay 协议，限制了该技术的进一步发展。

DLAN 的全称是 Digital Living Network Alliance，即数字生活网络联盟，由索尼、英特尔、微软等公司共同推广使用。该技术支持基于 IEEE 802.3.i/u 协议的有线网络接入和基于 IEEE802.11/a/b/g/n 协议的无线网络接入，支持 IPv4 协议，支持使用 HTTP 协议进行媒体的传输，并支持 JPEG、LPCM(线性脉冲编码调制)、MPEG2 等媒体格式^[13]。DLAN 技术于 2003 年被设计使用，具有 DLAN 认证的设备还较为普遍，但是开发 DLAN 的相关组织已经在 2017 年 1 月 15 日正式宣布解散，因此未来不会再更新 DLAN 标准，该技术正逐步被 Miracast 取代。

Miracast 是目前最具发展前景的无线投屏技术，该技术伴随着 WiFi 的发展和基于 Android 操作系统的智能终端设备的普及而不断被推广开来。Miracast 的技术协议规范是 WiFi Display (WFD)，各个多媒体设备之间基于 Miracast 协议建立 WiFi 链路上的连接，并传输视频和音频数据，实现画面和声音的跨设备播放。Android 操作系统自 4.2 版本以后就原生支持 Miracast 技术，因此该技术在近几年得到了大量的普及推广^[14]。但是由于该技术仅是一个基于 Wi-Fi Direct 的软件应用，缺少底层硬件的联合支持，在延迟、帧率、分辨率等方面依旧表现得不尽人意。

除了上述几种常见的无线投屏技术，还存在许多小众的无线投屏技术仍在使用，如 GoogleCast、Samsung MultiScreen、WiDi 协议、WiMo 协议^[15]、闪联协议^[16]等等。

1.2.3 流量控制技术的发展现状

目前，针对不同优先级业务场景下的流量控制解决方案，国内外提出了许多建设性的算法研究，如漏桶算法^[17]、令牌桶算法^[18]等通用的流量整形算法。其中漏桶算法把业务传输的动作比作将水注入桶中的动作，桶底存在一个小洞，以固定速率出

水，而注水则可以使用任意速率，但是一旦桶满，多余的水将溢出，即业务被丢弃^[19]。该算法的缺点是无法满足业务的突发传输需求，因此另一个算法令牌桶便是基于此改进而来。令牌桶算法周期性地向一个桶中放入令牌，业务的传输则需要从桶中取出对应大小的令牌。放入令牌时，若桶中令牌已满，则多余的令牌被丢弃；业务传输取令牌时，若桶中令牌不足，则不发送该业务^[20]。这种算法简单有效地实现了业务的流量限制，并满足了突发业务的传输需求，因此得到了广泛的使用。

与此同时，也出现了许多应用于特殊场景的流量控制算法，比如针对 TCP 业务的 RED 算法^[21]，以及无线场景下避免 TCP 突发业务的 TCP 确认平滑算法^[22]等等。像 Killer 等各大 WiFi 设备研发厂商便是基于上述算法，通过软硬件联合开发，在自家产品中加入了流量控制调度机制。

1.3 论文的结构与安排

本文针对 WiFi 链路下的投屏业务和游戏业务传输需求，提出了全新的传输质量保证设计方案，用于实现投屏和游戏业务在 WiFi 链路上的低时延、高可靠传输，提高用户的使用体验。

第一章是绪论部分，介绍了本文的研究背景与意义，并简单介绍了国内外针对 WiFi 技术、投屏技术和流量控制的发展概况，最后阐述了论文的整体结构和安排。

第二章则介绍了当前现有的技术背景，首先对 802.11 协议下的报文发送机制进行了研究，分析了 WiFi 链路上影响投屏业务和游戏业务时延、丢包率和带宽的各个机制；接着针对无线投屏技术，阐述了该技术使用到的链路结构、传输机制和编码方式等；然后介绍了游戏业务所使用的传输层协议技术；最后介绍了在 Windows 操作系统和 Linux 操作系统下进行无线网络设计开发所会使用到的底层架构。

第三章详细阐述了针对投屏业务所提出的新的设计方案，首先进行了技术路线的分析，介绍了投屏技术的需求与现有的问题和难点，并介绍了据此提出的整体设计方案。接着详细介绍了投屏报文的反馈重传方案和调度传输方案，用于保证投屏业务的低时延、高可靠传输；然后详细讲述了基于空口质量感知的视频帧编码设计；最后还介绍了将蓝牙信道与 WiFi 信道联合使用的投屏业务异构融合传输设计方案。

通过这些设计方案，共同来保证投屏业务的无线传输质量。

第四章针对在投屏业务上提出的设计方案，在 Linux 平台上和 Windows 平台上进行了架构设计和代码实现，然后验证了复杂无线环境下投屏业务通过反馈重传方案带来的可靠性提升，和通过调度传输方案带来的传输时延改善，最后针对基于空口质量感知的视频帧编码设计在链路层进行了测量与验证。

第五章详细阐述了针对游戏业务所提出的全新设计方案。首先分析了游戏业务在 WiFi 链路上传输时面临的问题和难点，讲述了游戏业务的传输需求和目标，并针对这些难点和需求提出了整体的设计方案。接着介绍了高效的报文封装设计，该设计用于实现多优先级业务下丢包检测和调度重传；紧接着介绍了基于该封装机制所设计的双频多硬件队列下的链路 KPI 测量机制，详细阐明了该机制的具体实现思路；然后基于链路 KPI 信息，进一步设计了双频多队列下的业务报文调度发送机制，以及动态流量调度传输机制，从而保证游戏业务在多业务并发、多设备共存的复杂无线场景下的传输质量。

第六章针对保证游戏业务传输质量的设计方案，再次在 Linux 平台上和 Windows 平台上进行了架构设计和代码实现，并验证了双频多硬件队列下的链路 KPI 测量功能，然后对比验证了业务在复杂无线环境下经过双频调度发送设计所带来的传输质量改善。

第七章对全文进行了总结，以及对下一步工作的展望。

第2章 影响投屏业务与游戏业务的无线传输机制概述

2.1 WiFi 下的传输机制

投屏业务和游戏业务都对链路的传输时延、丢包率和吞吐率提出了很高的要求，为了保证这些业务在 WiFi 链路上的传输质量，本小节对 WiFi 链路上影响时延、丢包率和吞吐率的传输机制进行了深入研究，具体分析了 WiFi 系统下 CSMA/CA 机制、Qos 机制、帧聚合机制和双频双发技术，为后续的质量保证研究提供了依据。

2.1.1 CSMA/CA 机制

相比有线传输环境，无线传输环境存在一个较大的问题，那就是很难及时地检测到无线链路上发生的信号冲突。接收方收到的无线信号的强度通常远远小于发送端发出的信号，如果要像有线传输环境一样进行碰撞检测，则需要更高的硬件成本进行链路监听；而且无线链路还存在着隐藏节点问题和暴露节点问题，碰撞检测机制通常无法检测到所有的碰撞事件^[23]。

因此，为了解决无线环境下的冲突检测问题，传统 802.11 协议规定了两种机制：DCF（分布式协调功能）和 PCF（点协调功能）。其中，DCF 使用带有冲突避免的载波侦听多路访问机制，即 CSMA/CA（Carrier Sense Multiple Access with Collision Avoid）机制，该机制类似以太网的带有冲突检测的载波侦听多路访问机制，即 CSMA/CD（Carrier Sense Multiple Access with Collision Detection）机制，所有 STA（Station）竞争访问信道，目前已被广泛使用；PCF 则是通过使用 AP（Access Point）或基站对无线信号的传输进行调度，从而对各个 STA 的通信进行统一控制^[24]。

CSMA/CA 机制主要分析以下三个名词：载波侦听，多路访问，冲突避免。

载波侦听是指，当发送端要在空口发送一个报文时，首先会对空口信道进行信号监听，只有空口处于空闲状态，发送端才会在空口上发送报文数据。

多路访问是指，允许多个 STA 通过多点接入同一个 AP，每个 STA 均有占用无线链路的权利和机会。

冲突避免是指，收发端能够通过主动避免冲突的方式解决冲突问题，如发送端监听到空口信道空闲时，先等待一段时间，然后再发送报文数据，而非立即就发送报文数据；或需要发送报文数据前，先发送一个短小的请求发送帧 RTS（Ready To Send），只有在指定时间内收到接收端送来的准许发送 CTS（Clear To send）帧，才判定空口信道是空闲的，然后进行报文数据的发送^[25]。

报文基于 CSMA/CA 机制在 WiFi 空口上进行发送时，会执行以下流程：

1、发送端监听空口信道，如果当前空口信道空闲，且在一个 DIFS（分布协调功能帧间间隔）内，空口信道持续空闲，则开始发送报文数据。

2、接收端收到该报文后，等待一个 SIFS（最短帧间间隔），然后向发送端发送确认帧 ACK。

3、发送端收到确认帧 ACK 后，认定本次传输过程已经成功。

当发送端准备发送数据时，如果检测到其他设备当前正在发送数据，则只能执行 CSMA/CA 机制下的退避算法，随机得到一个退避时间，将待发送报文推迟发送。若发送端发送报文数据后，超时未接收到 ACK 帧，则判断报文发送失败，同样执行退避算法，推迟重传报文。因此，在繁忙的无线链路上，业务报文会由于竞争信道失败或传输干扰导致报文的发送出现延迟，甚至在多次发送失败后，发送动作会被放弃而导致产生丢包。

2.1.2 Qos 机制

由于传统 802.11 协议下的 DCF 机制不支持 Qos，为了对无线报文进行基于业务优先级的调度发送，IEEE 802.11e 将 DCF 机制改进成了 EDCA 机制^[26]。在 IEEE 802.11d 中定义了 8 种优先级来区分不同的用户，IEEE 802.11e 的 EDCA 机制基于这 8 种优先级，映射出了 4 种访问类型 Access Category (AC)，分别为语音服务（Voice，AC_VO）、视频服务（Video，AC_VI）、尽力传输（Best-effort，AC_BE）和背景流量（Background，AC_BK）四种，其中语音服务的优先级最高，然后依次降低，背景流量优先级最低^[27]。两种 802.11 协议之间的映射关系如表 2.1 所示（优先级的值越大，传输优先级越高）。

表 2.1 IEEE 802.11d 与 IEEE 802.11e 的优先级映射关系

IEEE 802.11d		IEEE 802.11e	
业务类型	优先级	业务类型	优先级
BK(Background)	1	AC_BK(Background)	1
--	2	AC_BK(Background)	1
BE(Best-effort)	0	AC_BE(Best-effort)	2
EE	3	AC_BE(Best-effort)	2
CL	4	AC_VI(Video)	3
VI(Video)	5	AC_VI(Video)	3
VO(Video)	6	AC_VO(Video)	4
NC	7	AC_VO(Video)	4

厂商为了遵循 802.11 协议规定，让报文能基于上述优先级进行发送，在 WiFi 驱动的 MAC 层设置了多个硬件队列（又称发送队列），每个优先级对应一个硬件队列，且每个硬件队列独立维护各自的随机退避状态机。因此，当有不同优先级的报文同时需要发送时，这些报文首先会分别进入对应优先级的硬件队列，然后不同优先级的待发送报文之间通过信道竞争，来获取访问信道的机会^[27]。具体架构如图 2.1 所示。

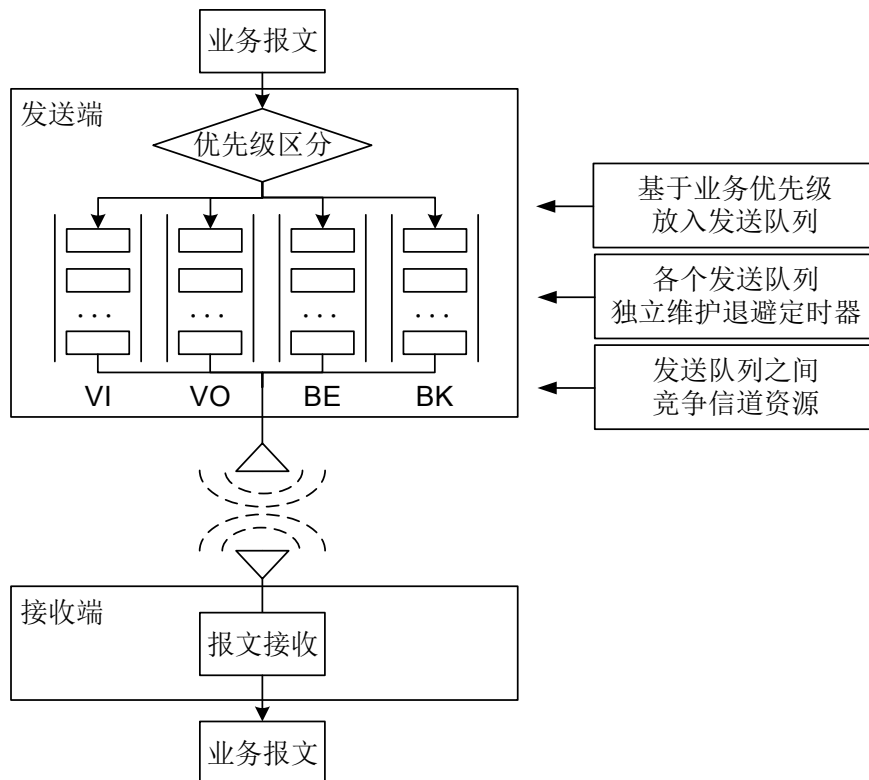


图 2.1 EDCA 机制下的发送队列

基于每个优先级, IEEE 802.11e 的 EDCA 机制规定了对应的每个硬件队列的竞争参数^[28], 主要包括以下三个参数:

1、发送或退避计时前, 进行空闲等待的 IFS 时间。

在 DCF 机制下, 报文首先需要等待 DIFS 长度的时间, 确保该期间信道始终空闲, 然后才允许进行退避计时或者进行发送。其中, 一个 DIFS 时间由 1 个 SIFS (短帧间间隔) 和 2 个 Slot (时隙) 时间组成, 其中每一个 Slot 时间中都进行了一次对信道 CCA (载波信号强度) 的检测, 因此每个 DIFS 时间中, 连续 2 次确认了信道的空闲, 才进行 backoff。

EDCA 机制对这种等待机制进行了修改和细化。在 EDCA 机制下, 报文首先需要等待 SIFS 长度的时间, 确保该期间信道始终空闲, 然后针对不同优先级可选择等待不同数量的 Slot 时间 (即检测信道 CCA 的次数不同)。如果需要等待的 Slot 时间的数量越多, 则意味着该优先级的报文每一次接入信道之前, 需要等待更长的时间, 因此优先级就更低^[29]。

2、需要进行退避等待时, 选择随机数的 CW (争用窗口) 大小。

IEEE 802.11 标准的 CSMA/CA 协议沿用了 CSMA/CD 协议中的二进制指数退避机制, 当检测到信道冲突时, 选择一个随机数进行退避等待, 该随机数即为等待的 Slot 时隙个数, 取值在 $\{0, 1, \dots, 2^k-1\}$ 之间, 其中 $k = \min\{\text{退避次数}, 10\}$, 2^k-1 则为当前的 CW (竞争窗口), CW_{\min} (最小竞争窗口) 和 CW_{\max} (最大竞争窗口) 被称作 CW 的最小值和最大值。

在 DCF 机制下, CW_{\min} 和 CW_{\max} 是固定的, 如 IEEE 802.11a 中, CW_{\min} 的值为 15, CW_{\max} 的值为 1023; 而 IEEE 802.11b 中, CW_{\min} 的值为 31, CW_{\max} 的值为 1023。当 CW 超过最大值后, 如果再发生退避, CW 便不再增大。当退避重传 16 次依旧不能成功时, 丢弃该帧并向高层报告。因此, CW_{\min} 和 CW_{\max} 决定了平均退避时间值, 这两个数值越大, 报文的平均退避时间就越长, 报文越不容易发送出去^[30]。

EDCA 机制则将 CW_{\min} 和 CW_{\max} 的值与发送队列优先级关联了起来, 针对不同的发送队列, CW_{\min} 和 CW_{\max} 的值均不相同, 从而实现了报文的 Qos 调度发送功能。

3、TXOP_{Limit} (最大传输机会) 的大小。

当发送端需要发送报文时，可以使用 RTS 帧和 CTS 帧提前协商好一段时间，这段时间别的设备不得占用空口信道资源。因此，将这段时间称为发送端协商得到的 TXOP（传输机会），而 $\text{TXOP}_{\text{Limit}}$ （最大传输机会）则表示 TXOP 的最大取值^[31]。

在 DCF 机制下，使用 RTS 帧和 CTS 帧协商得到的 $\text{TXOP}_{\text{Limit}}$ 是个固定值。

而 EDCA 机制则将业务的优先级与 $\text{TXOP}_{\text{Limit}}$ 关联起来，每个发送队列使用不同的 $\text{TXOP}_{\text{Limit}}$ ，从而实现业务发送的 Qos 调度^[32]。

最终，在 EDCA 机制下，各个发送队列的优先级对比情况如图 2.2 所示。

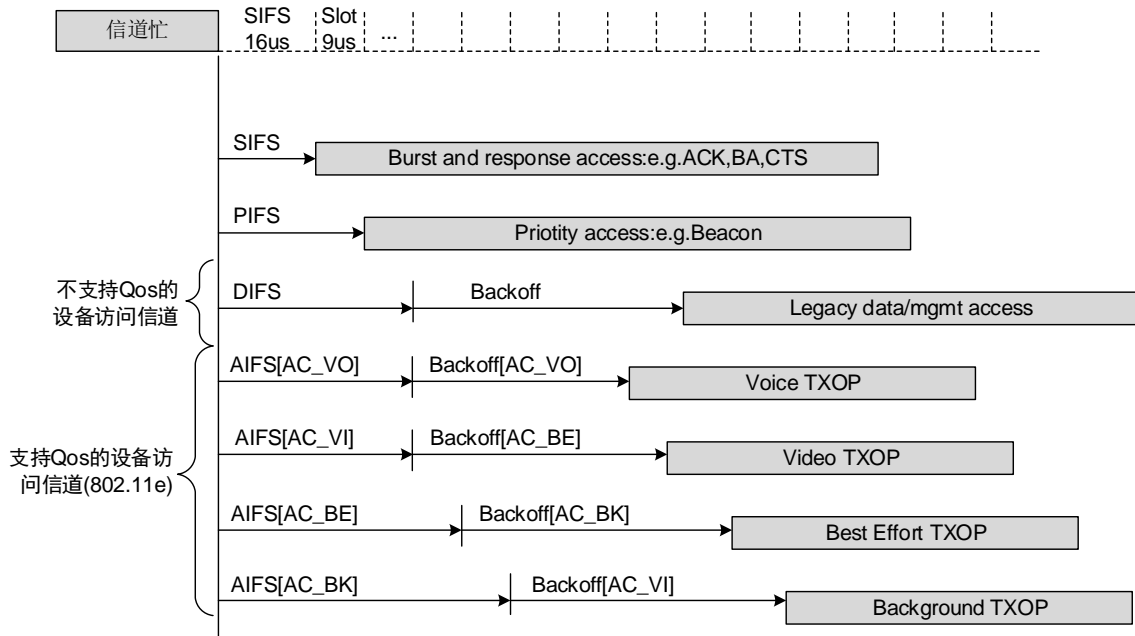


图 2.2 EDCA 机制下的发送优先级对比

因此，在 IEEE 802.11e 的 EDCA 机制下，由于不同硬件队列具有不同的空闲等待时间、退避等待时间和最大传输机会，使业务在 WiFi 链路上传输能够获得不同优先权限。在高优先级硬件队列的业务将更容易争抢到信道进行发送，而低优先级硬件队列中的业务传输则更容易因竞争失败而退避等待，从而产生更大的时延，甚至导致丢包。

2.1.3 帧聚合机制

为了提高报文在空口上的传输速率和吞吐量，充分利用信道资源，IEEE 802.11n 协议中提出了帧聚合机制^[33]。该机制通过将多个无线帧进行组合，得到一个长度更

长的无线帧，然后一次性将该帧进行传输。这种机制使得本应该进行多次竞争的传输变成了一次竞争即可的传输，大大减少了竞争的开销，减少了需要等待的空口时间，从而降低了业务的传输时延，并提高了传输吞吐率。

IEEE 802.11n 协议定义了两种帧聚合的方法。第一种帧聚合方法，称为 A-MSDU（针对 MSDU 的帧聚合），该方法将多个 MSDU（MAC 服务数据单元）聚合成单个帧进行传输。第二种称为 A-MPDU（针对 MPDU 的帧聚合），该方法将多个 MPDU（MAC 协议数据单元）进行组合，然后一次发送，最后专门由 BA（块确认）帧对各个 MPDU 的发送情况进行反馈^[34]。两种聚合方式可以同时使用，聚合格式如图 2.3 所示。

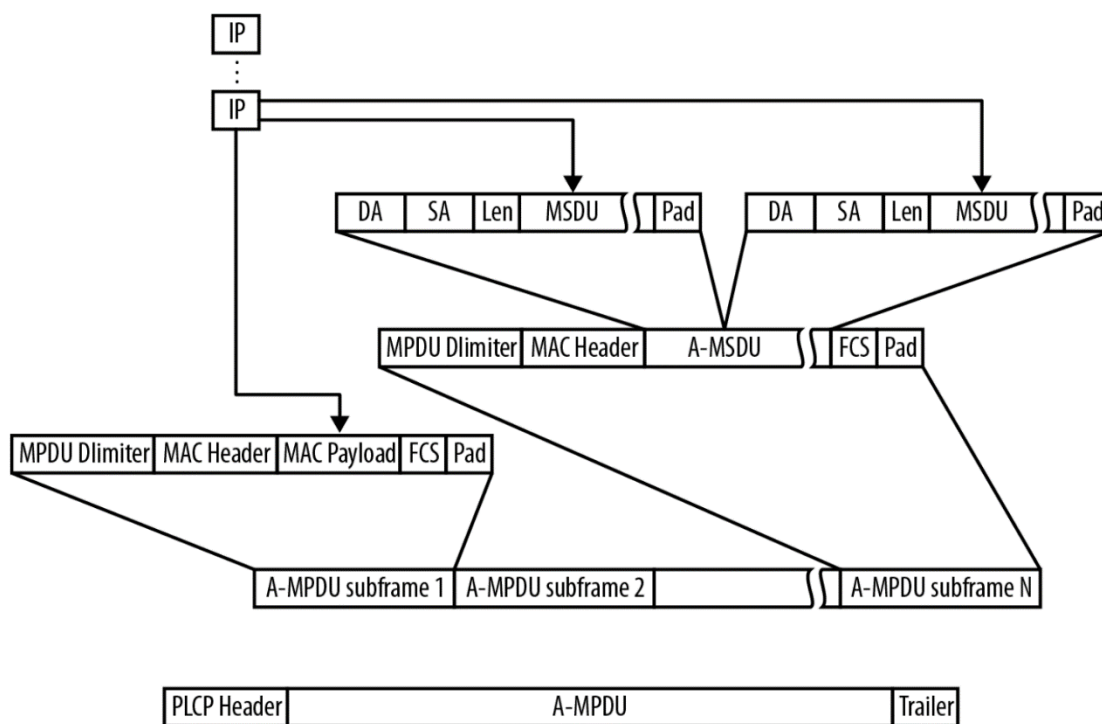


图 2.3 对 A-MSDU 报文的 A-MPDU 聚合

在 IEEE 802.11ax 协议中，对帧聚合技术进行了进一步的使用^[35]。IEEE 802.11n 协议下的帧聚合机制中，只有相同的 IEEE 802.11e 的 Qos 接入类别的 MPDU，才可以聚合成同一个 A-MPDU，这就对聚合动作提出了更高的前置要求，无法将聚合机制充分利用起来。而在 IEEE 802.11ax 协议中，在 A-MPDU 聚合机制中新增了 Multi-TID（多流量标识符）^[36]，当进行 A-MPDU 聚合时，可以通过 Multi-TID 识别

不同报文的 QoS 接入类别，因此不同 QoS 接入类别的报文也可以进行 A-MPDU 聚合，从而更有效地减少了空口的竞争开销，提高了数据在空口上的整体传输效率。

2.1.4 双频双发技术

为了提高 WiFi 的吞吐量，WiFi 芯片开发厂商提出了 DBDC（双频双发）技术，通过在芯片中集成 2 套完整的信号收发处理通路（2 个独立的基带处理和射频前端），实现了 WiFi 芯片同时在 2.4GHz 和 5GHz 频段上进行数据收发能力，从而提升了业务传输的带宽。

通常情况下，路由器使用的 WiFi 芯片均支持 DBDC 技术，因为路由器需要支持分别在 2.4GHz 和 5GHz 频段上产生独立的无线网络，从而在 2.4GHz 和 5GHz 频段下能够同时连接多个终端，并保证连接的终端能够在不同频段下同时且独立地工作。而常见的终端设备通常仅支持 SBSC（单频单发）或 DBSC（双频单发），即射频前端同一时间仅能选择一个频道进行工作。有的设备即使支持了 DBDC 的硬件，在实际使用场景依旧通过时分技术进行数据传输，无法真正实现双频下的全双工通信。但是随着技术的不断发展和人们对更高 WiFi 性能的追求，支持完整 DBDC 技术的终端设备也正在被逐渐研发和推广中。本文的研究便是基于支持完整 DBDC 技术的国产并发芯片上的开发平台，通过全新的传输质量保证设计，深度挖掘 DBDC 技术所能实现的双频传输功能，在提升带宽的同时保证业务低时延和高可靠传输。

2.2 无线投屏技术

本小节对投屏业务在 WiFi 下的传输机制进行深入研究，分别介绍了 WiFi 下投屏使用的 P2P 技术、传输层协议和投屏时的编码机制，为后续的投屏业务传输质量保证设计提供了理论支持。

2.2.1 WiFi 下的 P2P 技术

在常见的 WiFi 场景下，各个 STA 通过和 AP 建立连接，并使用 AP 进行中转实现 STA 与 STA 之间、STA 与互联网之间的数据交换。但是有时候，需要在 STA 和 STA 之间进行独立的数据通信，不经过 AP 中转，且不影响原有 WiFi 链路的正常使

用。基于这种需求，便诞生了 WiFi P2P（点对点传输）技术^[37]。

WiFi P2P 技术旨在实现多个 WiFi 设备在不经过 AP 中转的情况下直连并通信，其实现的功能类似于蓝牙技术，但是相较蓝牙具备更快的搜索和连接速度，更远的传输距离以及更大的传输带宽^[38]。通过 WiFi P2P 技术连接的多个设备之间，可以进行文件传输、音视频传输、投屏业务数据传输等基于点对点连接的数据传输功能。

要实现基于 WiFi P2P 技术的连接，需要一台设备作为 GO（Group Owner），通过控制 P2P 组，使支持 P2P 的设备能够连接进来；其余设备作为 GC（Group Client），来连接作为 GO 的设备^[39]。

本文所述的无线投屏技术便是基于 MAC 层的 WiFi P2P 技术，实现投屏业务数据在 WiFi 链路上的传输。

2.2.2 传输层协议的选择

目前主流的网络传输协议主要有 TCP 和 UDP 两种。在传输投屏业务流的场景下，这两种传输协议互有利弊。

TCP 协议作为被广泛使用的传输协议，其较高的传输可靠性和基于拥塞窗口的带宽自适应调节机制能够适应绝大部分业务场景的使用^[40]。然而在投屏业务场景下，当 WiFi 链路传输时延增大或出现丢包时，其拥塞控制机制会对传输带宽进行调整，从而影响投屏业务质量；且 TCP 丢包场景下，通过超时等待实现的重传机制，也会增大投屏业务的传输时延。

UDP 协议是语音、视频流场景下常用的传输协议，该协议简单且高效，具有最快的数据传输速度^[41]。但是报文确认重传机制的缺失使得 UDP 的数据传输相当不可靠，这使得对丢包敏感的投屏业务场景无法直接使用该协议。目前市面上常见的 Miracast 技术便是基于 UDP 协议进行投屏数据传输的，其实际传输效果也因此不尽人意。RUDP（可靠 UDP）协议相对标准 UDP 协议增加了确认重传机制，但是较长的重传时间使得实际投屏中依旧存在频繁的画面卡顿现象。

本文所述的无线投屏技术基于 UDP 协议传输，通过在 MAC 层进行针对投屏业务的技术改进，以保证投屏业务的在底层链路传输的可靠性与稳定性，并与 UDP 协议的高速传输特性共同配合，保证投屏业务的传输质量。

2.2.3 投屏帧编码方式

视频流数据如果不进行压缩和编码而直接传输,会对传输带宽提出极为巨大的要求。例如,对于 24bit 像素位深、1080P 分辨率、60fps 帧率的视频流,若直接传输每一帧的视频画面,则需要的链路传输带宽为

$$\begin{aligned} BW &= 24(\text{bit/pixel}) \times (1920 \times 1080)(\text{pixel/frame}) \times 60(\text{frame/second}) \\ &= 2,985,984,000(\text{bit/second}) \\ &= 2.78\text{Gbps} \end{aligned}$$

而在 IEEE 802.11ac 协议下的 WiFi 传输,最大带宽也仅支持 1Gbps,即该协议下甚至无法实现标准 1080P 视频流的正常播放。由此可见,对传输视频帧的压缩和编码对减小投屏业务的传输压力至关重要。

目前主流的视频压缩标准是 H.264 标准,该标准下视频帧被区分为 I 帧、P 帧以及可选的 B 帧,不同帧类别使用不同的编码方式^[42]。

I 帧的全称为帧内编码图像帧(Intra-coded picture)。由于 I 帧中包含了一帧画面的全部数据,解码时只需独立的该帧就可以得到对应画面内容,因此 I 帧通常被作为关键帧,给其他帧当作参考使用。I 帧使用了基于离散余弦变换 DCT(Discrete Cosine Transform)的压缩技术^[43],能在达到 7:1 的压缩率的情况下无明显的画面损失。

P 帧的全称为前向预测编码图像帧(Predictive-coded Picture)。P 帧包含的内容是当前帧与上一个关键帧或 P 帧的差别,该差别由运动矢量和预测误差两部分组成,解码时需要先从参考帧中基于运动矢量获取预测值,再将预测值和预测误差求和,最终得到实际的帧内容^[44]。由于 P 帧仅记录了帧间的差别信息,因此压缩率通常可以达到 20:1,从而节省了大量的空间,但是解码时需要依赖上一个帧的内容作为参考。

B 帧的全称为双向预测编码图像帧(Bidirectionally predicted picture)。B 帧包含的内容是当前帧与前后两个参考帧之间的差别,压缩率可高达 50:1,极大地节省了空间^[45]。但是 B 帧的解码需要前后两个参考帧都解码成功后才能进行,因此需要缓冲更多的视频帧才能对其解码,这就使得传输时延相对更高;且 B 帧的编解码的复杂度更高,也会占用更多的 CPU 资源。考虑到投屏对实时性的要求,投屏业务场景

下往往不使用 B 帧进行编码。

本文所述的无线投屏技术基于 H.264 视频压缩标准实现，且仅使用 I 帧和 P 帧对投屏业务的视频帧进行编码。

2.3 游戏业务传输技术

游戏业务同样是对时延和丢包率非常敏感的业务，较大的时延会导致玩家在游戏过程中感到操作不跟手，并出现明显的预测误差，而丢包则会使游戏画面出现卡顿，直接影响游戏体验。

在游戏业务场景下，使用 UDP 协议进行数据传输能带来更小的传输时延，但是不可靠的传输机制需要开发者进行额外的丢包检测和重传机制；而 TCP 协议在带来可靠数据传输的同时，会由于其拥塞机制导致业务传输出现极大的延迟时间。通常情况下，使用 TCP 协议进行数据传输的游戏更注重让每个玩家都能看到同样的场景，但是某个玩家的卡顿会造成所有玩家的共同卡顿，即牺牲了整体的流畅度；而使用 UDP 协议的游戏则更注重大部分玩家的使用体验，网络糟糕的玩家需要独自承担卡顿和延迟的后果。

除了棋牌类等对时延要求不高的游戏，目前大部分类型的游戏都是用的是 UDP 协议进行游戏数据的传输，如 CSGO、绝地求生等 FPS 类型游戏和 DOTA2、王者荣耀等 MOBA 类型游戏，并在游戏上层的服务器中进行可靠性的维护。当然这不意味着游戏就不使用 TCP 协议了，通常游戏业务在非游戏过程中的数据交互，如刚进游戏时的信息确认、连接过程中的心跳包等，由于对时延要求不高，会使用 TCP 协议来保证数据传输的可靠性。

2.4 驱动开发技术

目前承载投屏业务和游戏业务的主流操作系统是安卓智能手机使用的 Linux 平台和笔记本电脑使用的 Windows 平台，因此若要对业务进行传输质量保证的设计开发，就需要对相关平台下的报文传输管理机制进行研究。

2.4.1 Windows 下的 WFP 架构

为了实现 Windows 下的网络过滤需求，微软设计了一套 WFP（Windows 过滤平台）框架^[46]。WFP 框架是一组系统服务和 API（应用程序接口）的集合，开发者可以通过这个框架，对经过网络协议栈的报文进行截获、修改并重新发送。

WFP 框架主要由五个部分组成，分别为过滤引擎（Filter Engine）、基础过滤引擎（Base Filtering Engine）、垫片模块（Shims）、呼出接口（Callouts）和应用程序接口（Application Programming Interface）^[47]，微软提供的各个模块在 Windows 操作系统下的具体层次框架图如图 2.4 所示。

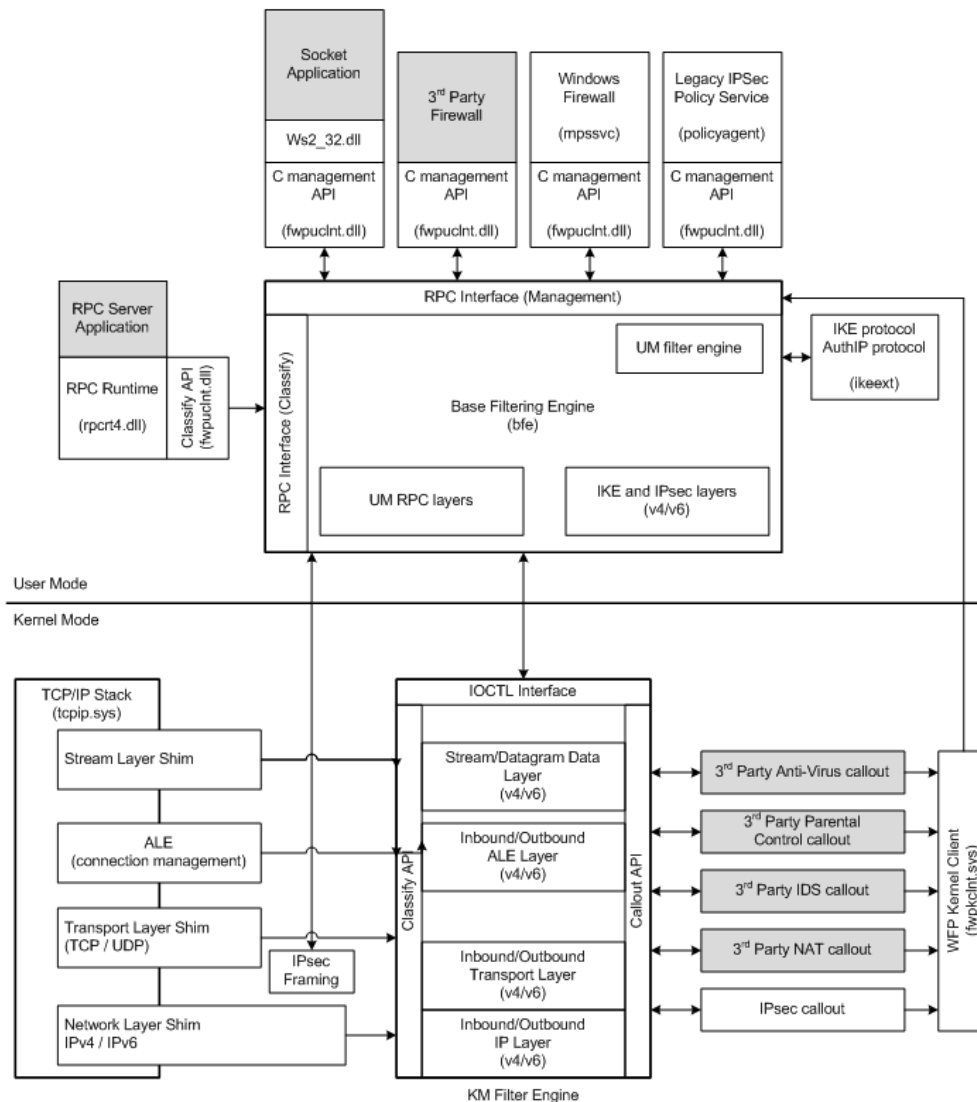


图 2.4 WFP 设计架构

在该架构下，开发者可以按需选择从网络协议栈中截获报文的位置，如传输层、网络层或数据链路层等。被截获的报文通过垫片模块送入过滤引擎模块中，最终调用呼出接口对报文进行相应的处理。因此，开发者在初始化好截获方案后，只需通过对呼出接口进行开发完善，就能实现网络报文的修改设计需求。

2.4.2 Linux 下的 WiFi 驱动架构

自 Linux 2.6.22 版本以后，802.11 协议栈便正式加入了内核，现如今内核中支持多种网卡驱动模块的设计架构^[48]。有些旧型号的驱动在内核中直接和用户态交互，使用的是 `wext`（无线拓展）接口；也有部分网卡直接在硬件中实现 MLME（MAC 层管理实体）的功能，由 `cfg80211` 层进行网卡设备和网络接口之间关联关系的管理，这类网卡驱动被称为 FullMac 驱动；而目前大部分网卡，MLME 功能都是在软件实现，通过 `mac 80211` 层实现 MLME 的设计框架，开发者只需在 SoftMac 驱动中实现所需的具体功能。有的厂商会将功能在网卡的软硬件之间进行分配，此时的驱动就成为 HalfMac^[49]。具体的层次结构如图 2.5 所示。

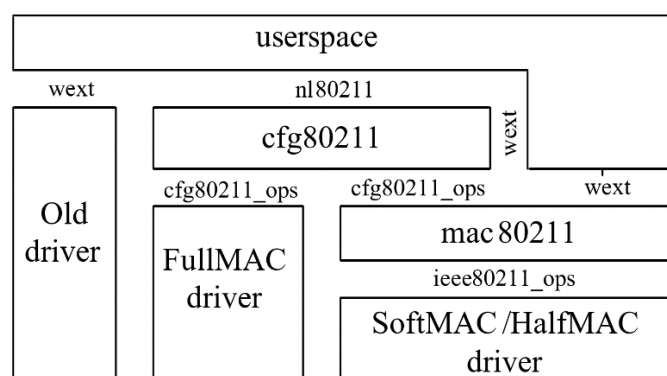


图 2.5 Linux 802.11 协议栈架构

本文所使用的 Linux 驱动开发平台基于 SoftMac 驱动实现 802.11 功能。驱动中设计了 UMAC 和 LMAC 两个部分，以分别处理 802.11 协议栈的不同功能。其中 UMAC 层主要实现 802.3 至 802.11 的报文转换、链路管理等功能，LMAC 层主要实现无线报文的发送调度、硬件交互等功能。通过两个层次模块的协同工作，共同实现 802.11 协议下的无线报文收发流程。

2.5 章节小结

第二章首先针对投屏和游戏业务在传输过程中对时延、丢包率和带宽的需求，介绍了 IEEE 802.11 协议下相关的报文传输机制，具体阐述了目前 WiFi 技术下，报文发送所使用的 CSMA/CA 机制、Qos 机制、帧聚合机制和双频双发技术；然后针对无线投屏技术，具体阐述了该技术在 MAC 层所使用的 WiFi P2P 技术、在传输层所使用的传输协议及在应用层对投屏报文所使用的压缩编码机制；接着分析了游戏业务所使用的传输层协议技术；最后描述了 Windows 下和 Linux 下进行网络调度开发所需的基本知识背景。针对投屏和游戏业务的传输质量保证需求，通过对相关技术的研究，为后续提出的全新设计方案提供了技术的支持。

第3章 投屏业务传输质量保证的设计

投屏业务是一个对丢包和时延敏感，且对带宽要求极高的业务。针对投屏业务在 WiFi 链路上传输的高质量需求，本章分析了现有的传输问题与难点，针对这些难点和需求，提出了许多新的设计方案，并对这些设计方案进行了具体的描述和分析，从而保证投屏业务的无线传输质量。

3.1 技术路线分析

3.1.1 投屏业务的需求与目标

无线投屏传输技术对 WiFi 链路的丢包率、时延及带宽提出了极高的要求。链路丢包会直接导致投屏画面出现花屏与黑屏，高时延则会使得投屏画面出现明显延迟、触控操作变得迟缓且声音视频无法同步，而过低的带宽则更是直接影响了画质。因此，本文的设计目标，是保证投屏业务能够在 WiFi 链路下得到低时延、高可靠性和大带宽的传输，从而给用户带来高质量的投屏体验。

3.1.2 现有传输问题与技术难点

在多设备共存的复杂 WiFi 环境下，投屏业务传输设备需要和其他设备共同竞争空口信道资源，当投屏报文在信道竞争中失败时，会触发退避算法导致发送动作被延迟；若连续多次出现信道竞争失败的场景，投屏报文的传输时延则不断增大，严重情况下甚至会导致底层放弃此次发送动作，从而使投屏业务产生丢包，导致画面花屏、黑屏和卡顿现象的发生。

与此同时，投屏业务的传输也会因为 WiFi 协议本身的限制导致拥塞。虽然 WiFi6 协议已经发布，但是目前市面上大部分移动终端设备还是仅支持 WiFi5 协议，如本文提出的设计方案所使用的验证平台使用的便是 WiFi5 协议的无线网卡芯片。在该协议下，已经开始使用 A-MPDU 聚合机制对业务报文聚合发送，但是协议规定 BA（块确认）帧最多仅支持对 64 个报文进行确认，这就意味着一旦链路出现丢包，发送端仅能从丢包报文开始的后续 64 个帧中选择报文进行聚合发送，以保证可以得到

确认。当发送端对 64 个报文进行聚合发送时，若聚合帧的起始几个位置出现丢包，而后面的报文正常传输，那么由于 BA 帧的限制，后续的新传报文依旧会被阻塞而无法发送，只有等待丢帧被单独重传成功后才能继续发送后续报文，这就最终造成了整个硬件队列的拥塞，从而使得投屏业务产生额外的传输时延。

投屏业务在传输过程中，投屏帧除了有 I 帧和 P 帧的区别，每一个视频帧还基于帧号、子图号及层号被分割成多份，每一份报文又为了满足 MTU 限制而被进一步分割成更小的报文。在这之中，具有不同帧类型、帧号、子图号及层号信息的报文，对传输的可靠性要求均有所不同，而目前的底层发送机制缺少对相关报文的 Qos 发送调度，而仅是统一使用 AC_VI 硬件队列进行发送，这使得实际投屏业务的发送效率不尽人意。

在传统的投屏机制下，投屏报文的编码端按照预设好的编码规则对视频数据进行编码和发送，但是底层传输链路的质量则是处于动态变化的，固定不变的编码规则往往难以适应不断变化的传输链路，从而导致无线传输链路要么过于空闲，要么负载压力过大，空口信道资源无法得到高效的使用，最终使得投屏报文传输质量存在波动，直接影响了投屏业务的使用体验。

3.1.3 投屏质量保证的整体设计思想

为了实现上述的需求和目标，针对投屏业务在无线传输中的各个问题和难点，本文提出了以下设计思想：

(1) 为了保证投屏业务在复杂 WiFi 环境下的传输可靠性，本文设计了投屏报文的反馈重传方案，该方案针对投屏报文在 WiFi 链路上的丢包现象，使用了基于 LLC over IP 封装的丢包检测与反馈重传机制，并使用高效的反馈帧构造设计进一步降低链路开销，同时保证投屏报文的可靠传输，接着基于 M/G/1 排队模型对投屏报文的平均传输时间进行了分析；

(2) 为了保证投屏业务的传输时延，本文还设计了投屏报文的调度传输方案，该方案针对投屏报文发送过程中面临的多设备竞争和 BA 帧长度限制导致的硬件队列阻塞问题，设计使用了多硬件队列调度机制和主动聚合机制进行报文传输，从而充分利用宝贵的硬件队列传输资源，降低报文的传输时延；考虑到不同投屏报文的

优先级不同，还设计了基于视频帧优先级的调度发送机制，从而进一步提高投屏业务在无线信道上的传输效率，保证投屏报文在 WiFi 链路下的传输质量。

(3) 本文提出了基于空口质量感知的视频帧编码设计。该设计思路通过在链路层对反馈重传方案下的反馈帧进行解析，实时地获取每一个视频帧编码单元的传输状态，并测量当前的空口链路质量，然后将视频帧编码单元的传输状态和链路质量信息及时上报至投屏业务的应用层编码端，触使投屏编码端进行容量压缩和编码调整，从而实现投屏业务应用层的链路自适应编码功能，提高投屏业务的传输效率与吞吐率，改善投屏业务的传输质量。

(4) 本文还提出了异构融合反馈重传的设计方案，通过将蓝牙技术与 WiFi 技术联合使用于投屏业务，实现投屏业务的全双工高效传输。该设计方案使用蓝牙的跳频传输技术进行反馈帧的传输，不仅提高了反馈帧的传输及时性与可靠性，还能避免反馈帧对 WiFi 信道的抢占，增加投屏业务的空口数据吞吐量，从而进一步提高投屏业务的传输质量。

3.2 整体架构介绍

在介绍投屏业务的传输质量保证研究之前，首先需要了解一下投屏报文的默认传输流程。在投屏业务的运行过程中，信源端的视频画面被源源不断地传输至信宿端进行显示。首先由投屏业务的应用层进行投屏画面信息的采集，并对画面数据进行编码和分片，以满足链路的 MTU 限制；然后应用层将每一个投屏报文送至 TCP/IP 协议栈，再由 TCP/IP 协议栈转发至 802.11 网卡驱动程序，由驱动程序进行 802.3 报文至 802.11 报文的转换，最后 802.11 网卡驱动程序通过 WiFi 链路将报文发送至接收端。接收端的执行流程则与发送端正好相反，802.11 网卡驱动程序从空口接收到投屏报文，将其转换至 802.3 报文后送入 TCP/IP 协议栈，然后提交至投屏业务的应用层进行视频帧的重组、解码及显示。

为了保证投屏业务的传输质量，本方案对发送端和接收端的默认报文传输流程进行了修改，分别在 TCP/IP 协议栈和 802.11 驱动层之间插入了投屏业务处理层进行开发设计，后续的各个设计方案都在投屏业务处理层中进行实现。方案设计的层次结

构如图 3.1 所示。

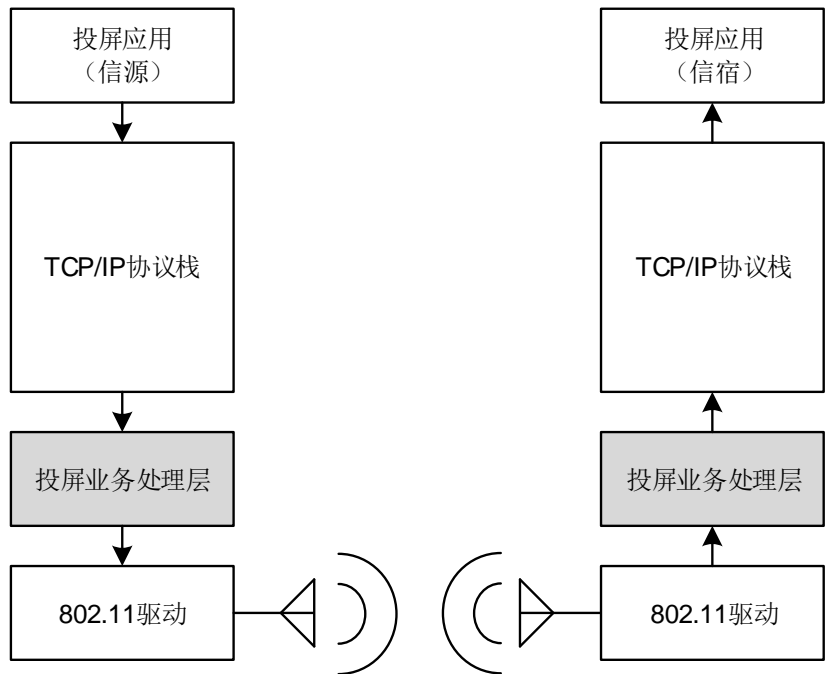


图 3.1 方案设计的层次结构

3.3 反馈重传方案

3.3.1 丢包检测与反馈重传

3.3.1.1 设计需求分析

投屏报文在空口传输过程中，会因为各种原因产生丢包，如多次信道竞争失败、空口信号干扰等。在默认的投屏流程下，丢包信息的检测需要在投屏业务应用层实现，这使得丢包场景发生时，接收端需要从投屏业务应用层构造丢包信息，然后将该信息一路送回发送端的投屏业务应用层，发送端的投屏业务应用层再将待重传报文重新传输至接收端的应用层。在这过程中，报文在发送与接收过程中的流量调度会引入传输时延，丢包信息的默认空口传输优先级过低亦不利于该报文的信道竞争，从而增大了重传时延，收发端应用程序的处理调度开销更是使得重传时延进一步变大。由于投屏业务对实时性具有极高的要求，这使得丢包场景对投屏质量产生的影响极大。

为了及时检测到投屏业务的丢包现象,本方案参考 TCP 滑动窗口机制与选择重传协议^[50],在 TCP/IP 协议栈和 802.11 驱动层之间设计了针对投屏业务的丢包检测与反馈重传机制。在该机制下,投屏报文在发送端会被缓存和重新封装,并基于发送窗口进行发送;接收端则通过封装信息判断传输过程是否出现丢包,并基于接收窗口向发送端进行接收情况的反馈。

3.3.1.2 丢包检测与重传设计

为了实现投屏业务的按序传输与丢包判断,本机制针对投屏报文设计了 LLC over IP 格式的封装。通过这种封装格式,发送端的投屏业务处理层为每一个按序到来的投屏报文分别分配了一个 LLC ID 信息,该 LLC ID 随着报文到来的顺序而连续递增,抵达最大值的时候循环归零。

在接收端,当封装后的报文被截获和处理时,投屏业务接收线程通过读取每个报文的 LLC ID 信息,判断报文的到来是否连续。当投屏报文的 LLC ID 数值连续时,意味着传输过程没有丢包,此时接收线程可以直接将投屏报文解封装并送入 TCP/IP 协议栈,交由上层投屏应用进行解码显示;而一旦检测到 LLC ID 数值不连续,该报文及其后续报文都不会被上交,而是会被缓存进接收端的缓存数组中。接收端通过构造反馈帧,将接收情况告诉发送端,通知发送端进行对应丢包的重传操作。

在发送端,每一个被封装的投屏报文都会先被按照 LLC ID 缓存进缓存数组,然后再发送。当发送端从反馈帧中解析出报文出现丢包时,即可从缓存数组中取出对应报文进行重传。每个被缓存的投屏报文都维护着一个确认信息,一旦报文超时未被反馈帧确认接收,同样会触发报文的超时重传。

基于 LLC over IP 格式封装的反馈重传机制具体流程如图 3.2 所示。

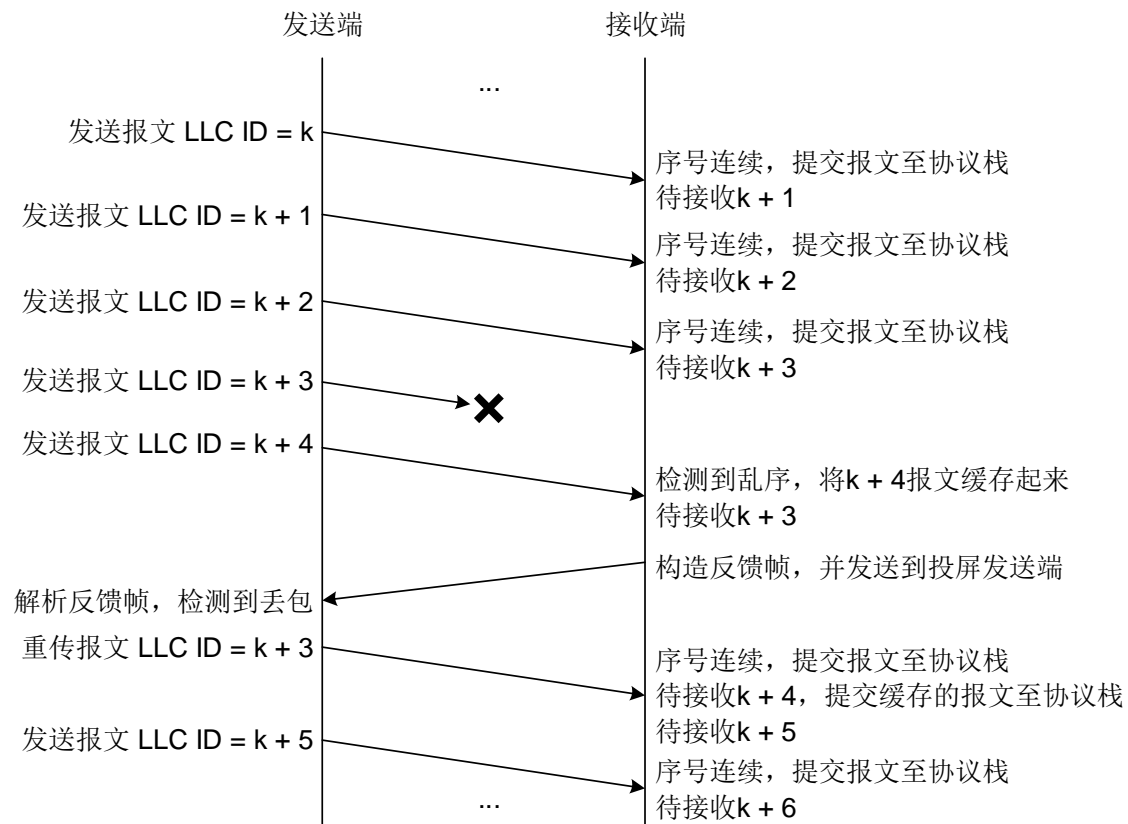


图 3.2 基于 LLC ID 连续性判断传输丢包

为了保证接收端的报文接收信息能及时、准确地抵达发送端，接收端同时使用包驱动和定时器驱动两种方式触发反馈帧的发送。对于包驱动反馈，接收端一旦检测到丢包事件的发生，则立即触发反馈帧的发送；对于定时器驱动，接收端使用高精度定时器，周期性发送反馈帧进行冗余反馈。

3.3.1.3 滑动窗口设计

当发送端源源不断地将投屏报文封装并发送时，由于接收端并不知道发送端的报文确认情况，接收端将无法确定发送端送来的报文是新传的后续报文还是反馈帧丢失导致的确认超时重传报文。具体举例来说，假设发送端已经将 LLC ID 序号为 0 到 2047 的投屏报文发送至对端，而接收端也收到了对应的报文，并也将接收信息放入反馈帧中发送至发送端。若此时接收端又收到一个 LLC ID 序号为 0 的报文，接收端面临两种情况：

1. 该 LLC ID 序号为 0 的报文是上一个序号为 2047 的报文的后续报文，接收端将其解封装后，按序提交至 TCP/IP 协议栈即可。
2. 由于之前发送给发送端的反馈帧全部阻塞或丢失，发送端因为确认超时，将报文进行重传，因此该 LLC ID 序号为 0 的报文是发送端的重传报文。此时接收端需要丢弃该报文，并继续发送反馈帧至发送端。

为了解决这种歧义的现象，本机制参考 TCP 协议下的滑动窗口机制，基于 LLC ID 信息，分别在投屏业务的发送端和接收端维护一个窗口，窗口主要维护两个指针：窗口下沿和窗口上沿。

针对发送端，窗口下沿指向最新待确认报文，窗口上沿则指向最新待发送报文；设置发送窗口固定为一半的 LLC ID 序号最大值，只有发送窗口内的报文可以发送至空口，LLC ID 序号在窗口外的报文则不被允许发送。发送窗口的具体结构如图 3.3 所示。

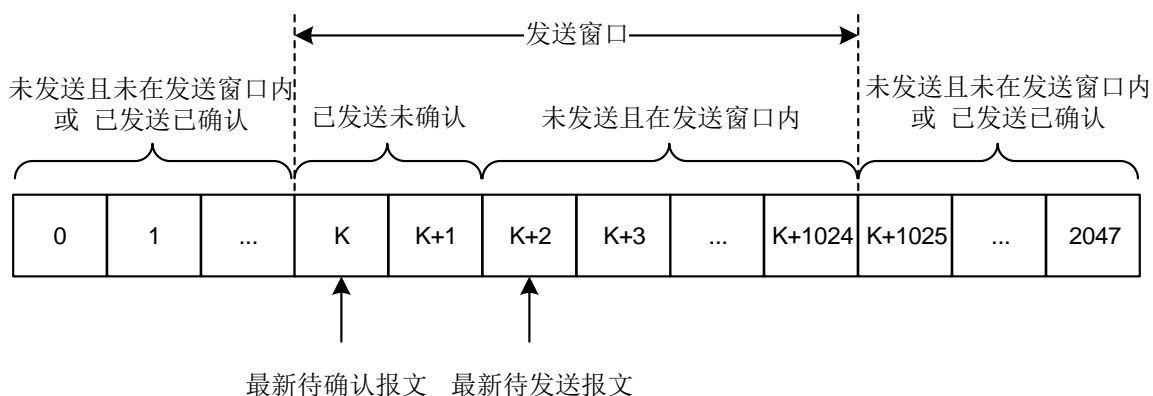


图 3.3 发送窗口结构示意图

针对接收端，窗口下沿指向最新被解封装并上报 TCP/IP 协议栈的报文，窗口上沿则指向最新已接收到的报文。若接收端收到报文的 LLC ID 序号始终连续，则窗口下沿始终与窗口上沿重合；一旦出现 LLC ID 序号不连续的现象，则窗口上沿将在窗口下沿的右侧。设置接收窗口固定为一半的 LLC ID 序号最大值，在接收窗口内，从第一个未接收到的报文到最新已被接收的报文的信息将参与接收情况的反馈，而若报文的 LLC ID 序号在窗口外，则不被允许参与反馈。接收窗口的具体结构如图 3.4 所示。

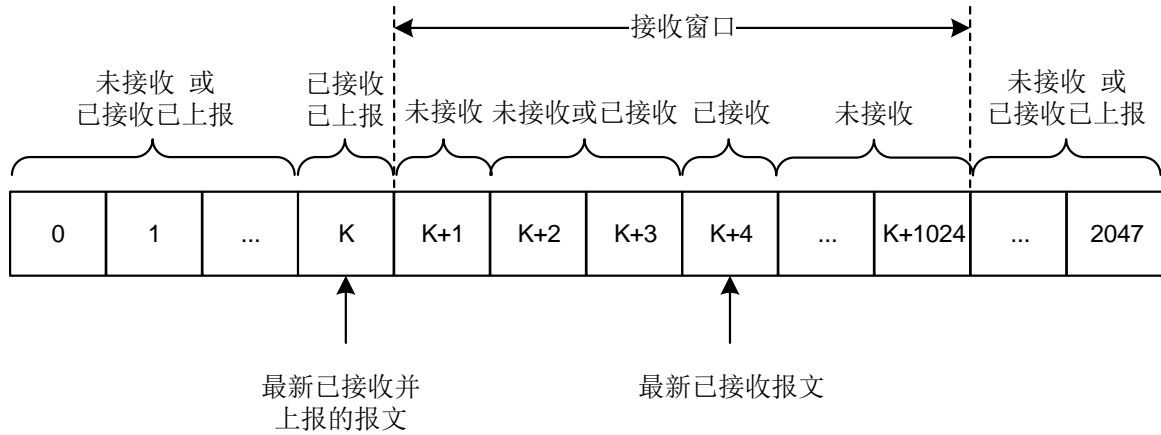


图 3.4 接收窗口结构示意图

3.3.2 高效反馈帧设计

3.3.2.1 设计需求分析

反馈帧的构造与发送是反馈重传方案中非常关键的部分。发送端的投屏业务处理层需要通过反馈帧获取每一个投屏报文的接收确认情况，从而进行丢包重传、超时重传以及后续的报文调度等工作，因此要求反馈帧快速及时地送至发送端；但是若对每一个投屏报文都发送一个反馈帧，又会带来极大的链路开销，反而影响投屏业务的正常传输。为了同时兼顾反馈重传的及时性和链路开销问题，本方案同时使用包驱动和定时器驱动两种方式进行接收信息的反馈，并设计了全新的反馈帧构造方案，通过将位图确认反馈机制和偏移确认反馈机制联合使用，极大地降低反馈帧的大小，从而将接收端的报文接收情况进行高效的反馈。

3.3.2.2 反馈机制设计

为了节省反馈帧的链路传输开销，本小节提出了一种全新的高效反馈帧构造设计。在该设计下，接收端使用位图确认反馈机制和偏移确认反馈机制进行接收端 LLC ID 信息的反馈。反馈信息从接收端的滑动窗口中获取，将当前待接收报文的 LLC ID 序号记录为起始序号，记录范围为起始序号到最新收到的报文的 LLC ID 序号。若接收端报文始终按序接收，则反馈信息中将仅包含起始序号。

偏移确认反馈机制具体还包括肯定偏移确认反馈和否定偏移确认反馈，最终根据报文的实际丢包情况决定选择哪一种确认反馈格式。反馈报文的负载由一个反馈头

部和多个反馈数据组成，反馈头部记录了起始序号、反馈报文数量、各个反馈数据的类型等信息。为了降低反馈数据给链路带来的开销，本机制限制每个反馈报文最大携带 4 个反馈数据，每个反馈数据最大 4 个字节。

(1) 位图确认反馈

当使用位图确认反馈时，从起始序号或上一个反馈数据的最后一个序号开始，每个比特代表一个报文的接收情况。在位图中，值为 0 的比特代表该报文未被接收，值为 1 的比特则代表报文已被成功接收。由于每个反馈数据最大 4 个字节，因此一个位图最大能反馈 32 个报文的接收情况。位图确认的格式如图 3.5 所示。

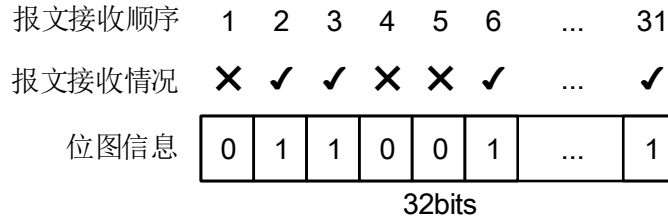


图 3.5 位图确认格式

位图反馈的方法简单且直观，但是当丢包数量偏少、偏多且连续出现时，偏移确认相对能带来更高的确认效率。

(2) 否定偏移确认反馈

当使用否定偏移确认反馈时，从起始序号或上一个反馈数据的最后一个序号开始，每个数值代表上一个丢包报文的 LLC ID 序号与下一个丢包报文的 LLC ID 序号之间的差值，具体格式如图 3.6 所示。

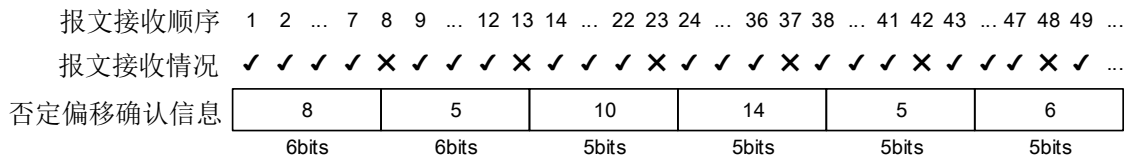


图 3.6 否定偏移确认格式

为了充分利用 4 个字节的反馈数据，将反馈数据拆成 2 个 6 比特的数值和 4 个 5 比特的数值，当数值二进制全为 1 时，特殊表示为该数值范围内的报文全部都没有丢包。在这种机制下，一个否定偏移确认理想情况下最大能反馈的报文数量为：

$$FeedbackNum_{Max} = 2 \times 2^6 + 4 \times 2^5 = 288$$

当接收端的滑动窗口中仅有极少量的报文丢失时,使用否定偏移确认反馈将能带来更高的反馈效率。

(3) 肯定偏移确认反馈

当使用否定偏移确认反馈时,从起始序号或上一个反馈数据的最后一个序号开始,每个数值代表上一个接收成功报文的 LLC ID 序号与下一个接收成功报文的 LLC ID 序号之间的差值,具体格式如图 3.7 所示。

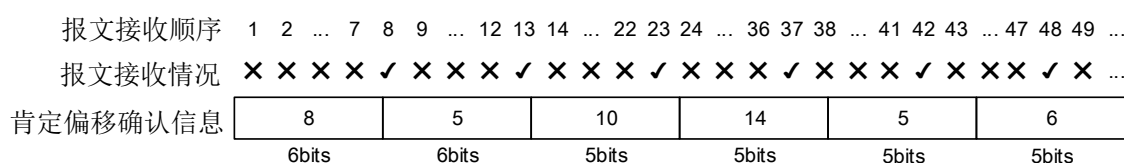


图 3.7 肯定偏移确认格式

同样地,将 4 字节的反馈数据拆成 2 个 6 比特的数值和 4 个 5 比特的数值,当数值二进制全为 1 时,特殊表示为该数值范围内的报文全部丢包。在这种机制下,一个肯定偏移确认理想情况下最大能反馈的报文数量同样为 288 个。

当接收端的滑动窗口中仅有极少量的报文接收成功时,使用肯定偏移确认反馈将能带来更高的反馈效率。

3.3.3 数据格式定义

3.3.3.1 LLC over IP 报文封装格式

LLC over IP 报文的封装主要包括 LLC 头的封装和 VLAN TAG 头的封装,其中 LLC 头中记录了报文的 LLC ID 信息和投屏帧的类型信息,而 VLAN TAG 头的封装用于在发送过程中进行硬件队列的选择,以及在接收过程中对封装后投屏报文进行类型标识。具体封装格式如图 3.8 所示。

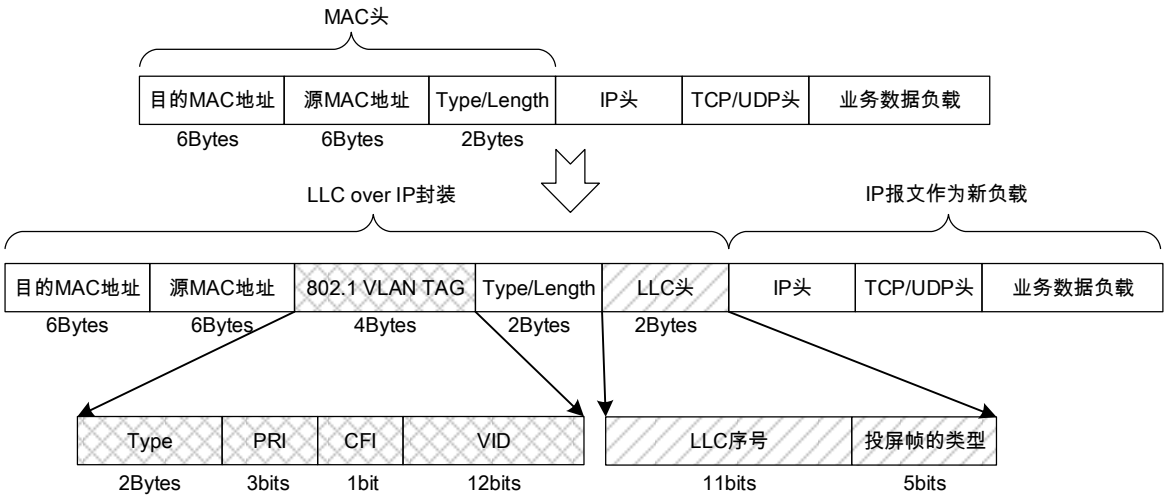


图 3.8 LLC over IP 报文封装格式

(1) LLC 头封装

在 LLC over IP 报文封装格式中，LLC 头中的各个字段信息及其含义如表 3.1 所示。

表 3.1 LLC 头封装格式

字段名称	字段长度	字段含义
LLC 序号	11bits	所有投屏报文共用一组 LLC 序号，根据报文到达发送端的顺序编号
投屏帧的类型	5bits	00001：触控帧 10000-10111：视频帧，I 帧，LayerID0-7 11000-11111：视频帧，P 帧，LayerID0-7 00010-01111：其它帧，预留拓展

(2) VLAN TAG 头封装

在 LLC over IP 报文封装格式中，VLAN TAG 头中的各个字段信息及其含义如表 3.2 所示。

表 3.2 VLAN TAG 头封装格式

字段名称	字段长度	字段含义
Type	2Bytes	此处设置为 0x8100，表示 802.1Q Tag 帧，该类型下驱动会根据优先级字段选择硬件队列。
PRI	3bits	表示帧的优先级，取值范围为 0~7，值越大优先级越高，驱动根据这 3bit 的优先级字段选择硬件队列。此处根据业务优先级设置 0、1、5、

		6 四个值对应 BE、BK、VI、VO 硬件队列。
CFI	1bit	经典格式指示符，设置为 0
VID	12bits	可配置的 VLAN ID 取值范围为 1~4094，0 和 4095 是协议保留。此处设置为 4094 来标识是 LLC over IP 封装后的帧。

3.3.3.2 反馈帧封装格式

在反馈帧封装格式中，主要包含 Feedback_head 和 Feedback_data 两个字段，其中 Feedback_head 字段记录了反馈帧的最新待接收报文的 LLC 序号（FSN）、反馈报文的数量、各个反馈数据的类型等各种信息，而 Feedback_data 字段则是变长字段，基于 Feedback_head 字段中的反馈数量设置字段实际长度，并基于 Feedback_head 字段中的反馈类型记录了接收端的报文实际接收情况。具体反馈帧的封装格式如图 3.9 所示。

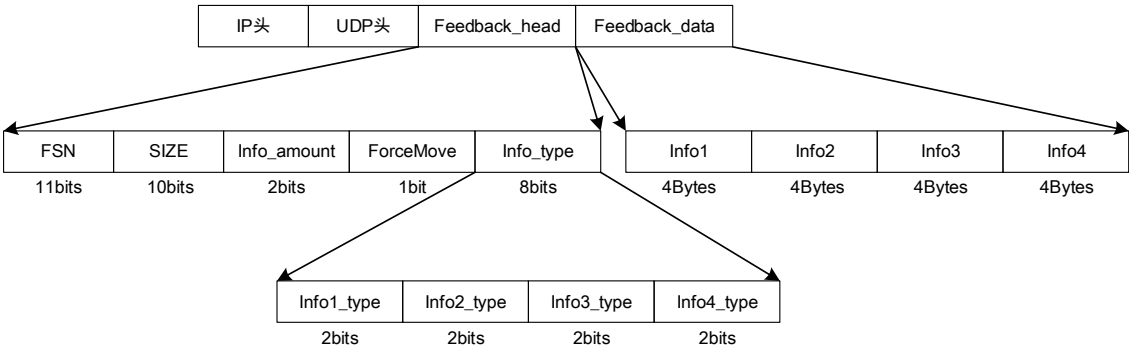


图 3.9 反馈帧封装格式

(1) Feedback_head 字段

反馈帧中 Feedback_head 字段的各个含义如表 3.3 所示。

表 3.3 Feedback_head 字段

字段名称	字段长度	字段含义
FSN	11bits	确认帧的起始 LLC ID 序号值，即接收端当前待接收的报文序号。
Size	10bits	确认 LLC ID 连续的投屏帧接收状态个数。
ForceMove	1bit	强制偏移指示。当该字段值为 1 时，指示接收端放弃了 LLC 帧，使得待接收的 LLC ID 强制偏移。
Info_amount	2bits	Response_info 中 info 单元的个数，最多 4 个。
Info_type	8bits	4 个 Info_type (n=1、2、3、4)，每个两位，指示对应的 info 单元的类型，仅对应 info 存在时有效。

(2) Feedback_data 字段

Feedback_data 字段有两种格式：偏移确认格式和位图确认格式，其中偏移确认格式又分成肯定确认偏移和否定确认偏移两种。因此，Feedback_data 字段一共有三种类型：位图，肯定确认偏移和否定确认偏移，三种类型在 Feedback_head 字段中对应的 Info_type 值如表 3.4 所示。位图、肯定确认偏移和否定确认偏移三种字段的实际封装格式如图 3.5、3.6 和 3.7 所示。

表 3.4 Info_type 字段取值

Type 值	Info 类型
0	BITMAP（位图）
1	RIGHTOFFSET（肯定确认偏移）
2	ERROROFFSET（否定确认偏移）

3.3.3.3 缓存数组结构

(1) 发送端缓存数组

发送端的缓存数组负责对已发送的投屏报文进行缓存，并记录了报文的发送时间、LLC ID 序号、报文类型等各种信息。具体格式如图 3.10 所示。

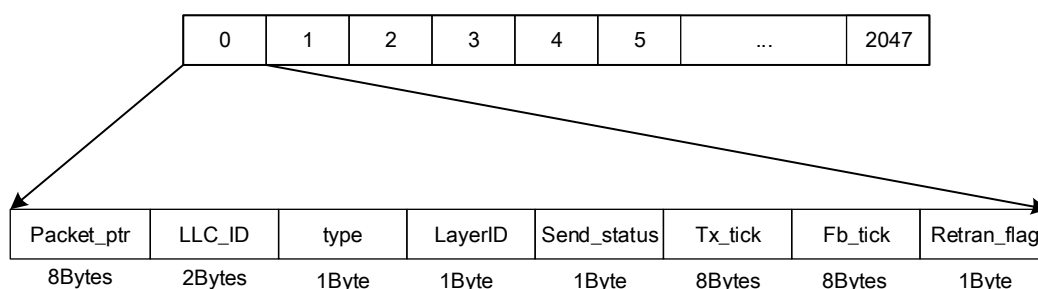


图 3.10 发送端缓存数组格式

发送端缓存数组的各个字段及其含义如表 3.5 所示。

表 3.5 发送端缓存数组字段

字段名称	字段长度	字段含义
Packet_ptr	8Bytes	投屏报文的数据指针。
LLC_ID	2Bytes	投屏报文的 LLC ID 序号。

type	1Byte	投屏报文的类型。 0: 触控帧 1: 视频帧 I 帧 2: 视频帧 P 帧
LayerID	1Byte	视频帧的层号。
Send_status	1Byte	投屏报文的发送状态。 0: NOT_SEND, 报文未发送 1: SEND_AND_RESPONSE, 报文已发送, 且得到接收确认 2: SEND_NO_RESPONSE, 报文已发送, 还未得到接收确认 3: NEED_RETRAN, 报文需要进行重传发送
Tx_tick	8Bytes	报文发送时间, 用于判定超时重传
Fb_tick	8Bytes	报文重传时间, 用于避免冗余重传
Retran_flag	1Byte	标记报文是否被重传

(2) 接收端缓存数组

当接收端检测到投屏业务出现丢包后, 接收端的缓存数组负责对后续的投屏报文进行缓存, 并记录了报文的接收时间、LLC ID 序号、报文类型等各种信息。具体格式如图 3.11 所示。

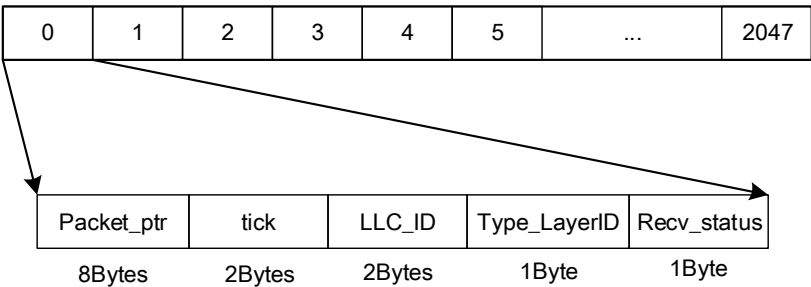


图 3.11 接收端缓存数组格式

接收端缓存数组的各个字段及其含义如表 3.6 所示。

表 3.6 接收端缓存数组字段

字段名称	字段长度	字段含义
Packet_ptr	8Bytes	投屏报文的数据指针。
tick	2Bytes	记录报文在换成数组中滞留的时间, 用于判断是否等待超时。
LLC_ID	2Bytes	投屏报文的 LLC ID 序号。

Type_LayerID	1Byte	基于 LLC 头中的类型字段，使用 5bit。 00001：触控帧 10000-10111：视频帧，I 帧，LayerID0-7 11000-11111：视频帧，P 帧，LayerID0-7 00010-01111：其它帧，预留拓展
Recv_status	1Byte	投屏报文的接收状态。 0：NOT_RECEIVED，报文未收到 1：RECEIVED_AND_FORWARD，报文已收到，已提交上层 2：RECEIVED_AND_CACHED，报文已收到，且被缓存

3.3.3.4 滑动窗口结构

(1) 发送窗口

发送窗口由窗口下沿和窗口上沿组成，用于表示发送缓存数组中可以发送的报文，只有在窗口内的报文，才可以用于发送。发送窗口的各个字段及其含义如表 3.7 所示。

表 3.7 发送端的发送窗口字段

字段名称	字段含义	字段含义
Tx_lowedge	窗口下沿	指向缓存数组某一元素的指针，初始值即为第一个经封装的新传帧在缓存数组中的位置。在收到反馈帧时，根据接收端报文的接收情况实时更新窗口下沿。
Tx_upedge	窗口上沿	指向缓存数组某一元素的指针，初始位置和窗口下沿重合。在收到新传帧和反馈帧时，均会实时更新窗口上沿。
Tx_winsize	窗口长度	根据窗口下沿和窗口上沿计算窗口大小，其值为 $(Tx_upedge - Tx_lowedge + 2048) \% 2048$ 。
MAX_LENGTH	窗口最大长度	设置 MAX_LENGTH 为发送缓存数组大小的一半，即为 1024。

(2) 接收窗口

接收窗口由窗口下沿和窗口上沿组成，遍历窗口内的报文构造反馈帧。接收窗口的各个字段及其含义如表 3.8 所示。

表 3.8 接收端的接收窗口字段

字段名称	字段含义	字段含义
Rx_lowedge	窗口下沿	指向缓存数组某一元素的指针，初始值即为第一个解封装的报文在缓存数组中的位置。在报文提交上层的过程中，实时更新窗口下沿。
Rx_upedge	窗口上沿	指向缓存数组某一元素的指针，初始位置和窗口下沿重合。在缓存报文时，会实时更新窗口上沿。
Rx_winsize	窗口长度	根据窗口下沿和窗口上沿计算窗口大小，其值为 $(\text{Rx_upedge} - \text{Rx_lowedge} + 2048) \% 2048$ 。
MAX_LENGTH	窗口最大长度	设置 MAX_LENGTH 为接收缓存数组大小的一半，即为 1024。

3.3.4 传输时延分析

在本小节设计的反馈重传方案下，投屏业务报文在 WiFi 链路上的传输可以得到非常高的可靠性保证，从而降低了链路丢包导致传输层或应用层进行报文重传而产生的时延开销。考虑到投屏业务对传输时延十分敏感，为了对反馈重传方案下的传输时延进行分析，本小节使用 M/G/1 排队模型对该传输过程进行描述。

假设投屏业务在传输过程中，投屏数据报文在链路层的平均到达率为 λ ，且 λ 服从泊松分布，其分布函数为：

$$F(t) = \begin{cases} 1 - e^{-\lambda t}, & t \geq 0 \\ 0, & t < 0 \end{cases} \quad (3-1)$$

与此同时，投屏报文的系统传输过程模型基于等效服务时延思想来建立，设第 i 个数据的等效服务时延为 X_i ($i = 1, 2, \dots$)， X_i 是相互独立同分布的随机变量，且和数据报文的到达时间间隔相互独立，则等效服务时延的均值 \bar{X} 和二阶矩 \bar{X}^2 分别为：

$$\bar{X} = E\{X\} \quad (3-2)$$

$$\bar{X}^2 = E\{X^2\} \quad (3-3)$$

假设当第 i 个报文到达系统时，第 j 个报文正在处于排队模型的服务中，它的剩余服务时间是 R_i ，此时等待队列中还有 N_i 个报文等待处理，队列状态如图 3.12 所示。

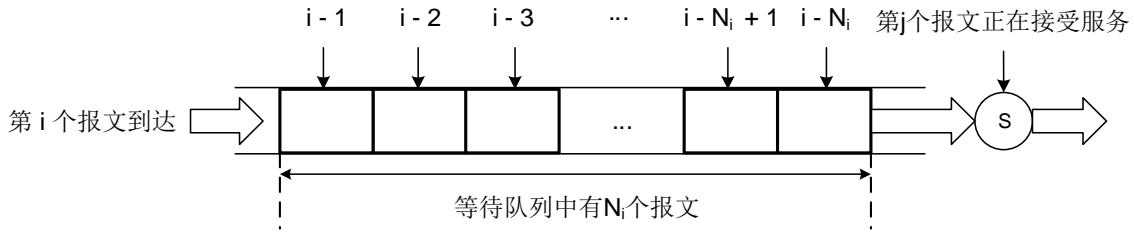


图 3.12 第 i 个报文到达时的系统状态

由图 3.9 可知，第 i 个报文的等待时延为：

$$W_{qi} = R_i + N_i \text{ 个报文的服务时延} = R_i + \sum_{k=i-N_i}^{i-1} X_k \quad (3-4)$$

为了计算系统稳定状态下的平均等待时延，先对式(3-4)求平均，得到：

$$\overline{W_{qi}} = E\{R_i\} + E\left\{\sum_{k=i-N_i}^{i-1} X_k\right\} = E\{R_i\} + \bar{X} \cdot E\{N_i\} \quad (3-5)$$

式(3-5)中， X_k 、 N_i 都是随机变量， \bar{X} 是 X_k 的均值， $E\{N_i\}$ 是 N_i 的均值，代表等待队列中的平均报文数量。

接着令 $i \rightarrow \infty$ ，且设 $W_q = \lim_{i \rightarrow \infty} \overline{W_{qi}}$ ， $R = \lim_{i \rightarrow \infty} E\{R_i\}$ ， $N_Q = \lim_{i \rightarrow \infty} E\{N_i\}$ ，整理式(3-5)并代入(3.2)可得：

$$W_q = R + \bar{X} \cdot N_Q \quad (3-6)$$

与此同时，由利特尔法则（Little's theorem）可得，在稳态的情况下，系统中的平均顾客人数 N 等于顾客在系统中的平均逗留时间 W ，乘以顾客的到达率 λ ，即：

$$N = \lambda \cdot W \quad (3-7)$$

将式(3-7)应用于本模型中，代入式(3-6)，可得：

$$W_q = R + \bar{X} \cdot \lambda \cdot W_q \quad (3-8)$$

整理式(3-8)得：

$$W_q = \frac{R}{1 - \lambda \cdot \bar{X}} \quad (3-9)$$

由式(3-9)可知，M/G/1 排队模型下的报文平均等待时延 W_q 和报文的平均到达率 λ 、平均服务时间 \bar{X} 和平均剩余服务时间 R 均相关。假设系统存在稳态解，同时也具有各态历经性，那么剩余服务时间就可以通过图 3.13 来表示。

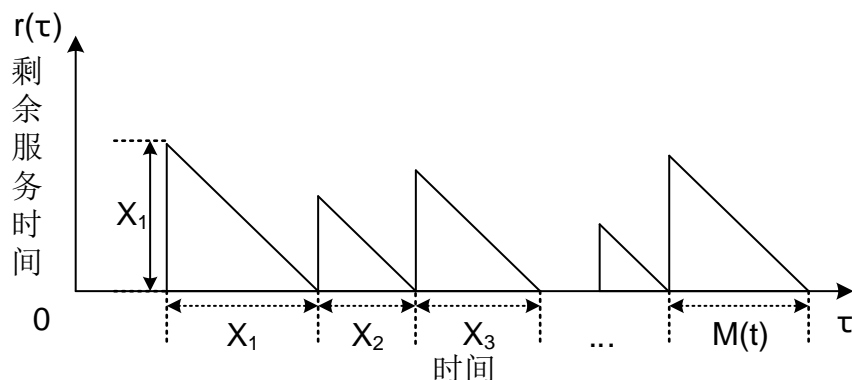


图 3.13 剩余服务时间变化曲线

为了方便分析，取 t 为 $r(\tau) = 0$ 的时刻进行计算，设 $M(t)$ 是 $[0, t]$ 这段时间上到达的报文个数，则平均剩余服务时间的值为：

$$R_t = \frac{1}{t} \cdot \int_0^t r(\tau) d\tau = \frac{1}{t} \cdot \sum_{i=1}^{M(t)} \frac{1}{2} X_i^2 \quad (3-10)$$

对式(3-10)进行修改调整，可得：

$$R_t = \frac{1}{2} \cdot \frac{M(t)}{t} \cdot \frac{\sum_{i=1}^{M(t)} X_i^2}{M(t)} \quad (3-11)$$

式中， $\frac{M(t)}{t}$ 即为报文的平均到达率 λ ， $\frac{\sum_{i=1}^{M(t)} X_i^2}{M(t)}$ 即为 X_i 的原点二阶矩，令 $t \rightarrow \infty$ ，就可以得到平均剩余服务时间为：

$$R_t = \frac{1}{2} \cdot \lambda \cdot \overline{X^2} \quad (3-12)$$

代入式(3-9)，得到 P-K 公式：

$$W_q = \frac{\lambda \cdot \overline{X^2}}{2(1 - \lambda \cdot \bar{X})} \quad (3-13)$$

于是，报文的平均传输时延为：

$$W = \bar{X} + W_q = \bar{X} + \frac{\lambda \cdot \overline{X^2}}{2(1 - \lambda \cdot \bar{X})} \quad (3-14)$$

从式(3-14)中可以看出，投屏报文的平均传输时延，与报文的平均到达率和报文在 WiFi 链路上的服务时间有关。其中，报文的平均到达率和投屏业务的传输带宽直接相关，即投屏业务的带宽越大，投屏链路上的平均时延越大；而报文在 WiFi 链路

上的服务时间则代表了从发送当前投屏报文到发送下一个投屏报文的时间间隔，即报文在硬件队列中的阻塞时间。因此，投屏报文在硬件队列中阻塞的时间越长，投屏业务的平均传输时延也会越大。

为了降低投屏业务的平均传输时延，保证投屏业务的高质量传输，就要求投屏报文在硬件队列中的拥塞时间尽可能短，并能够实现投屏业务带宽的调整，提高投屏数据在 WiFi 链路上的传输效率。为此，本文针对投屏业务，设计了报文的传输调度机制、基于空口质量感知的视频帧编码机制和异构融合反馈机制，共同改善投屏业务的带宽和队列阻塞时间，保证投屏业务的传输质量。

3.4 调度传输方案

3.4.1 多硬件队列调度

在反馈重传方案下，WiFi 链路上传输失败的投屏业务报文会被重传发送。但是，当链路上已经发生报文传输阻塞或丢包的情况时，往往意味着此时的硬件队列已经经历了多次退避动作，并存在着较长的退避等待时间。为了避免投屏报文被单个硬件队列阻塞而无法发送，本机制设计了多硬件队列调度发送机制，通过联合调度使用 AC_VI 队列和 AC_VO 队列两个硬件队列发送投屏业务报文，保证投屏业务能够以低时延快速传输。

在投屏业务的正常传输场景下，该机制选择使用默认的 AC_VI 队列发送业务报文，并通过反馈确认机制检测链路传输情况。发送端在缓存数组中记录每个报文的发送时间，并使用高精度定时器检测报文的确认等待时间。一旦检测到报文超时未收到反馈确认，或从反馈帧中检测到传输链路出现丢包，则立即触发多硬件队列下的调度重传。

当发送端出现投屏报文丢包重传或超时重传时，则意味着硬件队列中存在投屏报文未能正常传输，出现了报文阻塞现象，此时的硬件队列已经存在相对较长的退避等待时间。而投屏业务对传输时延非常敏感，若仍使用原有硬件队列进行报文重传，可能导致队列进一步阻塞，使得投屏业务在队列中的阻塞时间不断增大。因此，在投屏报文的重传场景下，本机制通过将待重传的报文送入 AC_VO 队列进行发送，以

保证投屏业务报文发送的及时性，重传调度流程如图 3.14 所示。

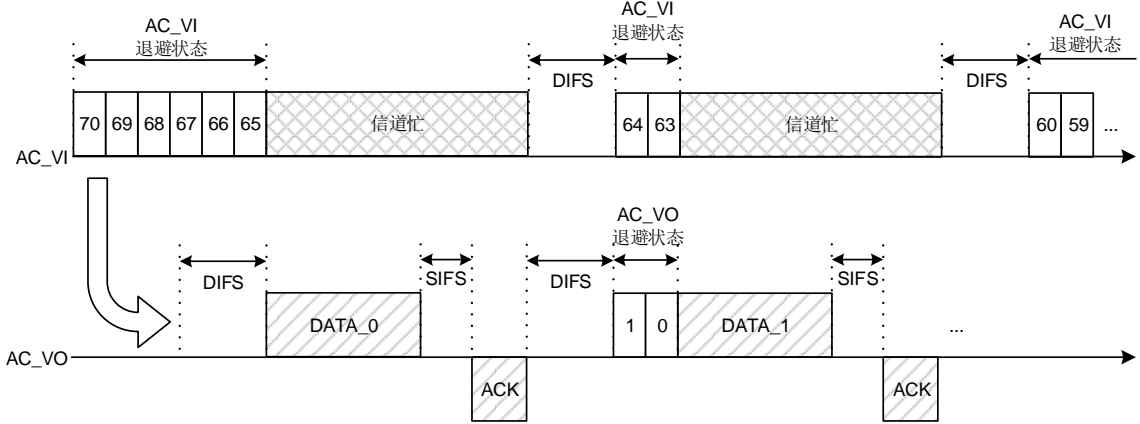


图 3.14 AC_VO 队列下的报文重传流程

为了给予 AC_VI 队列充足的时间进行退避等待，不仅重传的报文选择 AC_VO 队列进行发送，后续 16 个新传的投屏报文也都使用 AC_VO 队列发送，从而避免后续投屏报文在 AC_VI 队列继续阻塞，影响投屏业务的传输。等到 AC_VI 队列不再阻塞后，恢复使用 AC_VI 队列进行投屏报文的传输。对应流程如图 3.15 所示。

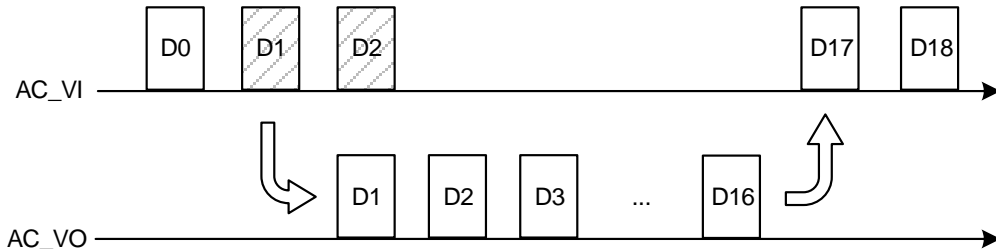


图 3.15 AC_VO 队列下的报文新传流程

通过两个硬件队列的联合调度使用，可以降低丢包或链路繁忙场景下，投屏报文在硬件队列中的阻塞时间，从而让投屏报文能够快速及时地送至对端，降低了报文在 WiFi 链路上的传输时延，保证了投屏业务的高质量传输。

3.4.2 主动聚合机制

自 IEEE 802.11n 协议开始，无线网卡驱动便开始支持将硬件队列中的报文使用 A-MPDU 进行聚合发送。但是，由于 WiFi5 协议下的 BA 帧最大仅支持对 64 个报文进行确认，当聚合帧的起始位置出现丢包时，后续的投屏报文便会因此无法继续送入硬件队列进行聚合发送，使得报文发送过程发送出现阻塞。为了解决 WiFi5 协议

下 BA 帧最大仅支持对 64 个报文进行确认而导致的硬件队列中投屏报文阻塞的问题，本方案还提出了投屏业务帧的主动聚合发送机制。该机制的设计思路如图 3.16 所示。

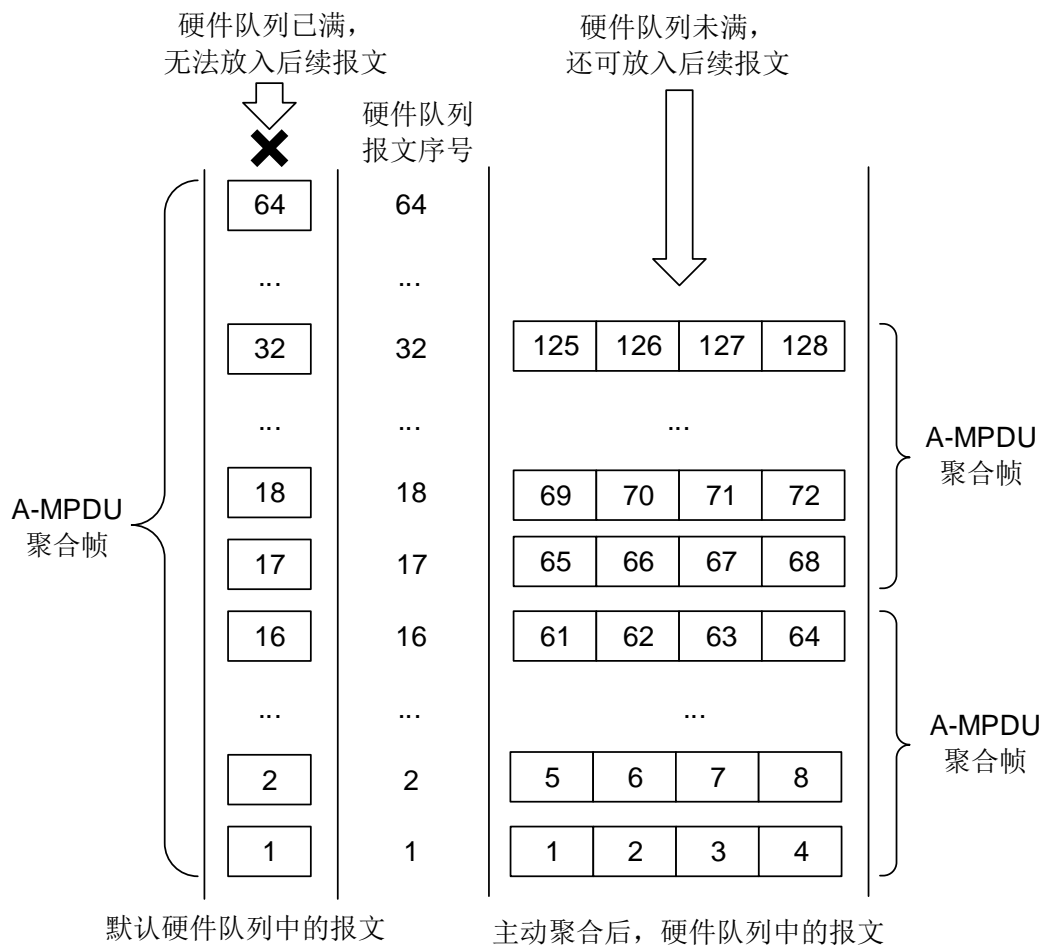


图 3.16 投屏业务帧的主动聚合发送机制

在该机制下，发送端的报文发送线程首先会对多个投屏报文进行主动聚合，然后再将聚合后的报文送入硬件队列，此时硬件队列中的每个报文包含了多个投屏报文数据。因此，硬件队列中将可以承载数量更多的投屏报文。当硬件队列以同样的报文长度进行 A-MPDU 聚合时，每个 A-MPDU 聚合帧仅使用了硬件队列中的部分报文，而不是将所有的 64 个报文一起聚合。当 BA 帧通知发送端进行丢帧重传时，硬件队列中待重传的报文即可与其他新传报文共同聚合发送，保证投屏报文能够及时地发送出去。因此，该设计能够避免投屏报文在硬件队列中出现不必要的阻塞问题，进一步降低了报文在 WiFi 链路上传输的时延。

3.4.3 优先级调度机制

投屏业务的视频帧具有不同的优先级，优先级高的报文的延时或丢失会造成更严重的画面缺失，而优先级较低的报文造成的影响则相对不那么大。因此，本节设计了基于视频帧优先级的调度发送机制，通过对视频帧进行解析，获取报文的优先级信息，然后在底层对报文进行基于优先级的调度发送，从而充分利用信道资源，改善投屏业务的传输质量。

3.4.3.1 投屏报文优先级分析

本文所述的无线投屏技术基于 H.264 视频压缩标准实现，使用 I 帧和 P 帧对投屏业务的视频帧进行编码。与此同时，为了满足链路 MTU 限制，投屏业务还将每一个视频帧继续基于帧号、子图号和层号进行细化分割，最终在底层链路传输的视频报文负载中则分别携带有各自的帧号、子图号、层号及分片序号等信息。

针对传入的投屏报文，发送函数通过解析报文负载，读取报文的各个信息，具体读取信息内容如表 3.9 所示。

表 3.9 投屏报文负载中的相关信息

字段名称	字段长度	字段含义
Type	1Byte	报文类型，触控帧、视频帧、其它帧（音频、文件等）。
FrameType	2bits	视频帧类型，I 帧、P 帧。
naluParam	1Byte	视频帧的帧号
SPN	4bits	视频帧的子图号
LayerID	3bits	视频帧的层号
EncodeUnit_ID	2Bytes	编码单元 ID，由帧号、子图号和层号构成。
SN	11bits	每一个编码单元下的各个分片的包序号
E_flag	1bit	最后一个分片的标志位

在投屏业务数据中，不同视频帧具有不同的优先级，其中 I 帧在视频帧中具有最高的优先级，I 帧的丢失意味着整个 GOP（画面组）都无法解析，此时的投屏画面会出现短暂的黑屏画面；当某一个 P 帧丢失时，P 帧及后面的所有帧信息都会解析出错，相对应的则是画面的花屏与马赛克现象。而基于视频帧划分的各个报文分片之中，又基于层号具有不同的优先级，层号越大，传输优先级越高。与此同时，投屏业务

数据中还包括触控帧及其他类型报文，其中触控帧在所有类型的帧中具有最高的优先级，其他类型的帧则作为最低优先级进行处理。

3.4.3.2 基于优先级的调度设计

为了实现基于投屏业务优先级的链路传输调度，本机制针对不同优先级的业务报文，在发送端分别设置了对应的超时重传时间，并在接收端分别设置了对应的最大等待时间。

发送端的投屏报文发送线程中，对每一个投屏报文均进行了缓存，并在缓存数组中记录了每个报文的发送时间，接收端则通过反馈帧更新发送端缓存数组中的报文确认情况。发送端使用高精度定时器周期性地检查每一个报文的确认情况，一旦发现某个报文未在超时重传时间内得到确认，则判定该报文发送出现阻塞或丢包，从而触发投屏报文的重新传输。对于超时重传时间更短的报文，发送端会更为“急迫”地想得到接收端的接收确认，因此在相同时间内会进行更多次的重传；而对于超时重传时间更长的报文，发送端则不那么“着急”得到确认，重传次数相对更少。因此，本机制针对不同优先级的报文，通过设置不同的超时重传时间，来实现投屏报文的优先级调度传输。

在投屏业务的接收端中，接收线程针对每一个收到的投屏报文，基于报文中封装的 LLC ID 信息进行丢包检查。接收线程会维护一个待接收的 LLC ID 信息，当链路上未出现丢包时，待接收的 LLC ID 信息即为接收端最新收到的投屏报文的下一个报文对应的 LLC ID 信息；而当链路出现丢包时，接收端收到的投屏报文 LLC ID 信息将与待接收的 LLC ID 信息不一致，此时该报文会被放入接收端缓存数组进行缓存，并基于前后报文信息判断丢失报文的类型，设置最大等待时间。如果在这段最大等待时间内，接收端未能收到预期收到的投屏报文，接收线程则会放弃等待该预期报文，然后跳过该报文，继续按 LLC ID 的顺序将缓存报文上报，并等待后续报文。对于最大等待时间更长的报文，接收端将更有“耐心”等待该报文，即该报文有更多的机会重传至接收端；而最大等待时间更短的报文，接收端则更容易放弃该报文。因此，本机制针对不同优先级的报文，通过设置不同的最大等待时间，进一步实现了投屏报文的优先级调度传输。

针对投屏报文的各个报文类型，本机制设置的超时重传时间和最大等待时间如表 3.10 所示。

表 3.10 不同报文优先级的调度策略

报文类型			超时重传时间	最大等待时间
视频帧	I 帧	LayerID = 0~3	2ms	8ms
视频帧	I 帧	LayerID = 4~7	3ms	8ms
视频帧	P 帧	LayerID = 0~3	4ms	7ms
视频帧	P 帧	LayerID = 4~7	5ms	7ms
其它类型帧			5ms	6ms

通过对投屏报文进行基于优先级的调度传输，可以降低低优先级投屏报文在链路上出现拥塞或丢包时对高优先级报文产生的时延影响，并使高优先级投屏报文能够更可靠地传输，从而提高投屏业务在 WiFi 链路上的传输效率，保证投屏业务能够高质量传输。

3.5 基于空口质量感知的视频帧编码

3.5.1 设计思路分析

在现有的投屏技术下，投屏业务的应用编码层与数据链路层属于两个完全独立的部分，上层的应用程序负责对投屏的画面数据进行编码、分片和下发，而底层的无线网卡驱动负责保障数据报文在空口的传输。然而，随着人们对投屏业务的需求不断增加，厂商对于投屏业务的传输质量要求也在不断提高，而投屏业务应用层固定的编码方式难以匹配质量不断变化的空口链路，使得投屏业务在无线链路上传输存在瓶颈。

投屏业务使用 I 帧和 P 帧对视频画面进行编码传输，其中 I 帧使用帧内预测编码，帧内包含了一帧画面的全部数据，通常作为参考帧给 P 帧使用；P 帧使用的则是帧间编码预测，需要从参考帧中进行运动补偿预测，才能得到实际的画面数据。因此，I 帧报文相对需要使用更大体积的报文进行传输，但是具有更强的抗干扰能力，在复杂环境下使用更多的 I 帧进行传输，能得到更稳定可靠的投屏质量；而 P 帧的传输在深度依赖 I 帧的同时，能带来更小的链路开销，在链路通畅稳定的情况下，使用更

多的 P 帧进行投屏业务的传输，能够极大提高投屏业务的传输吞吐率，实现更高清晰度的投屏显示效果。

为了让投屏业务应用层能基于空口质量动态灵活地进行编码，本节设计了底层的链路质量测量方案与编码单元传输监测方案，通过将底层的每个视频帧传输情况及时汇报上层，使上层应用实时获取视频帧的链路传输情况，进行 I 帧和 P 帧编码比例的调整，提高信道利用率；并及时发现投屏帧的传输出错信息，从而发送新的 I 帧来重建 GOP（画面组），避免画面出现黑屏、花屏与卡顿。与此同时，通过测量底层的链路质量信息并上报应用层，一同触发投屏应用进行编码方式与传输策略的调整，从而充分利用链路带宽，提高投屏报文的空口传输效率。

3.5.2 编码单元传输监测

在投屏业务的应用层，会将每一个待发送的视频帧按照帧号、子图号和层号进行分割，业务收发端的应用层以该分割后的单位为基本单位，称作编码单元，因此，每一组帧号、子图号和层号对应一个编码单元。而为了满足链路的 MTU 限制要求，应用层又将每一个编码单元继续分割成更小的投屏报文，底层的发送链路处理的最小单元就是这种投屏报文。

当投屏业务应用层检测到底层的编码单元传输持续通畅时，就可以进行编码机制的调整，通过增加 P 帧的使用，提高视频传输的压缩率，从而使链路能够传输更高清晰度的投屏画面，提高投屏业务的传输质量。当投屏业务应用层检测到底层某一个编码单元发送失败时，投屏业务应用层可以通过建立新的 GOP（画面组），重新发送新的 I 帧，从而在接收端建立整幅画面信息，避免黑屏和花屏画面的出现。这就要求投屏业务应用层能够快速、及时地发现底层视频报文的发送情况，从而实现编码机制的快速切换。因此，本机制在底层设计了对每个编码单元传输情况的实时检测上报功能。

3.5.2.1 编码单元传输信息记录

由于链路的 MTU 限制，上层应用侧对每一个编码单元都进行了分片操作，因此发送端需要记录每个编码单元所使用的报文。考虑到编码单元的每个报文到达底层

发送端的随机性，发送端使用链表保存每个编码单元和报文的对应关系。具体使用的存储结构如图 3.17 所示。

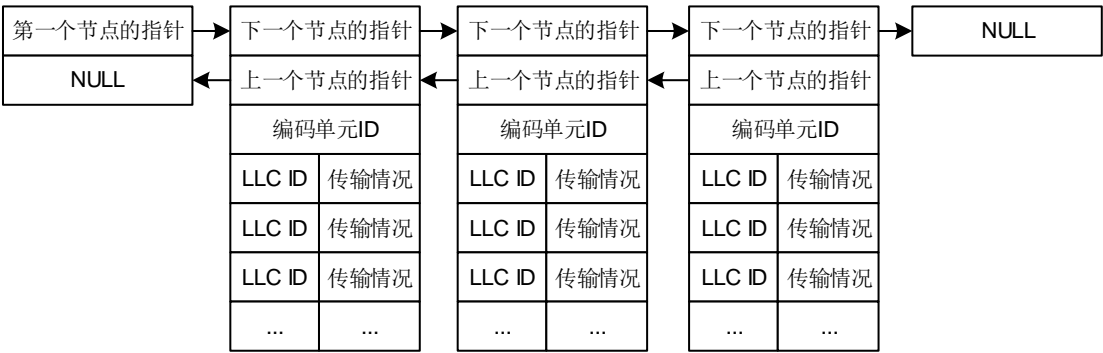


图 3.17 编码单元与报文对应关系的存储结构

该链表的维护工作在底层的投屏报文发送线程中实现。当发送线程截获到新的投屏报文时，经过报文信息的解析，发送线程会遍历链表寻找同一编码单元的节点。若存在对应节点，则在节点内记录报文的 LLC ID 信息，并设置传输情况为未得到确认；若未找到对应编码单元的节点，则创建新的节点，然后再记录相关 LLC ID 信息。

3.5.2.2 获取编码单元的传输情况

当发送端收到新的反馈帧时，经过反馈帧的解析，发送线程可以获取接收端中哪些报文得到了接收确认。对于每个被确认的报文，发送线程会通过编码单元信息找到链表中的对应节点，然后更新对应 LLC ID 信息下的报文传输情况。具体流程如图 3.18 所示。

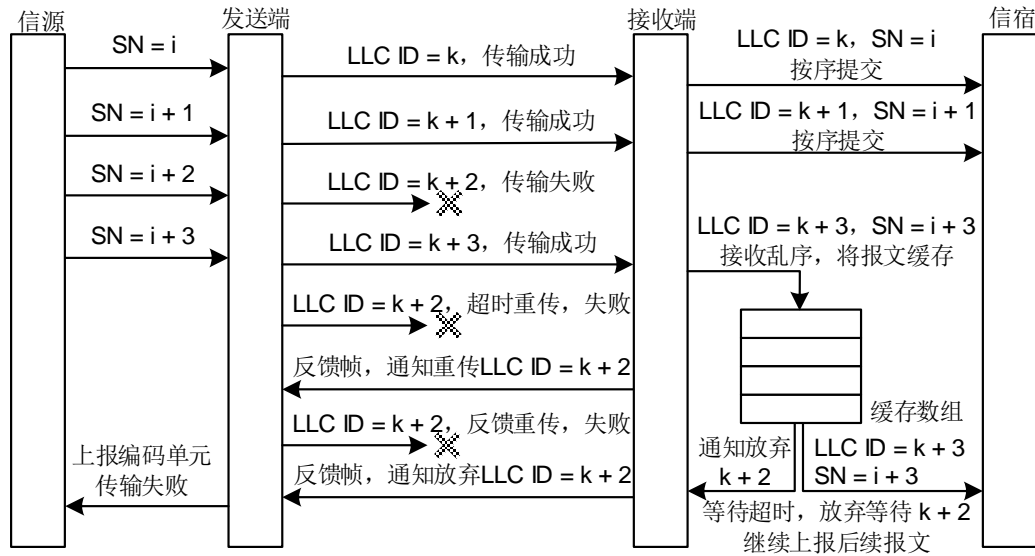


图 3.18 编码单元的传输情况监测流程

当一个节点的所有报文都被接收端成功确认，则标志该编码单元传输成功，编码单元的传输信息会被记录下来，节点也会被释放。传输信息会由定时器周期性上报，触发上层应用灵活编码；当一个节点的任意一个报文被判定发送失败，则判定该编码单元发送失败。同样地，编码单元的传输信息会被记录下来，节点也会被释放，但是此时会立即触发编码单元传输信息的上报工作，从而确保上层投屏应用及时发现视频帧的传输失败问题并进行补救，避免黑屏、花屏现象影响用户投屏体验。

3.5.3 链路质量测量

为了让投屏业务应用层能基于底层链路质量进行调整，需要进行底层链路质量信息的测量。本机制通过发送端和接收端联合进行投屏业务的空口质量测量，从而测量报文的往返时延、平均往返时延、报文的拥塞时延及硬件队列拥塞长度。

3.5.3.1 测量报文往返时延

投屏报文的往返时延指发送端发送报文的时间与接收到包含肯定确认的反馈帧的时间之差，该时延直观地反应了当前时刻下空口链路的传输质量。较小的时延意味着链路传输通畅，而更大的时延则代表了此刻链路正存在拥塞、干扰等问题导致报文无法顺利传输。由于接收端的反馈帧在高精度定时器驱动下，以 1ms 为周期进

行反馈，因此实际链路时延测量精度即为 1ms。

为了实现往返时延的测量，发送端在缓存数组中，针对每一个缓存报文均设置了报文的发送时间信息。当报文被送入硬件队列进行发送时，对应的发送时间 T_{tx} 会被记录至缓存数组中。相关流程如图 3.19 所示。

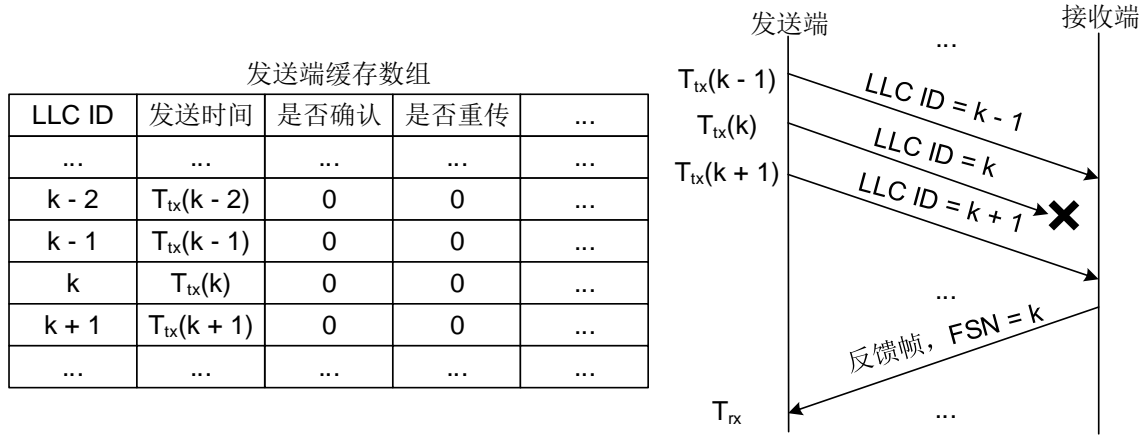


图 3.19 缓存数组的发送时间更新流程

接收端周期性地向发送端发送反馈帧，反馈帧中包含了各个报文的接收确认情况。发送端每当接收到反馈帧时，记录下反馈帧的接收时间 T_{rx} ，然后对反馈帧进行解析，得到接收端最新接收报文的 LLC 序号。接着，发送端读取缓存数组，获取该 LLC 序号对应报文的发送时间 T_{tx} ，计算得到往返时延为 $T_{rx} - T_{tx}$ 。具体流程如图 3.20 所示。



图 3.20 往返时延的计算流程

由于接收端使用冗余传输机制进行反馈帧的传输, 发送端针对同一报文会得到多次确认, 因此只在报文被首次确认时计算时延信息。与此同时, 当发送端对报文进行重传时, 不更新报文发送时间, 因此往返时延计算的是报文从首次发送到最终成功传输至对端并得到确认的整体使用时间。

3.5.3.2 测量队列拥塞信息

投屏业务报文在传输过程中, 网卡驱动会将其送入底层软件队列进行缓存, 然后再送入硬件队列中进行发送。在这过程中, 一旦硬件队列中的报文未能及时得到底层 ACK 的确认, 报文就无法从硬件队列中得到释放, 这就使得一旦硬件队列被填满, 后续待发送报文就无法进入硬件队列中进行正常发送, 投屏业务报文将会被阻塞。为了能及时检测底层队列的阻塞信息, 本机制设计了测量底层队列拥塞时延和拥塞长度的测量方法。

拥塞时延记录了当前时刻的传输流程下报文会在底层队列中停留的最大时间, 通过这个时间, 可以直观地反映出当前链路传输的质量情况。本机制下测量链路的拥塞时延的方法十分简单, 由于在丢包检测与重传机制下, 发送线程已经记录了每个

报文的发送时间，当需要当前时刻的拥塞时延时，当前时刻下最早的一个没有被反馈帧确认的报文在底层停留的时间就是目前底层的拥塞时延，因此只需要获取当前时间，并计算与最早未被确认的报文的发送时间的差值即可。

拥塞长度则记录了当前时刻下在底层队列还有多少个报文未能成功发送，该长度信息同样反应了当前情况下的链路质量。由于本机制在发送端使用了滑动窗口机制，因此同样可以很容易地得到底层队列的拥塞长度信息。滑动窗口的下沿记录了当前已发送报文中最早没有被反馈帧确认的报文，而上沿则记录了最新的已发送且未被确认的报文，在这之间的所有报文就可以被认定是当前底层被阻塞的报文。因此只需通过计算当前时刻下滑动窗口两个边沿的距离，即可得到底层队列的拥塞长度信息。

3.5.4 信息上报格式

底层会将每个编码单元的传输情况存储下来，并构造信息上报帧送至投屏业务应用层。编码单元信息上报由定时器触发和事件触发两种方式驱动，其中定时器周期性地将传输成功的编码单元信息进行上报，以实现上层应用对投屏传输情况的实时监测；而事件触发则由编码单元传输失败这个事件触发信息上报，从而保证上层应用能快速、及时地处理传输出错事件。具体上报报文的格式如图 3.21 所示。

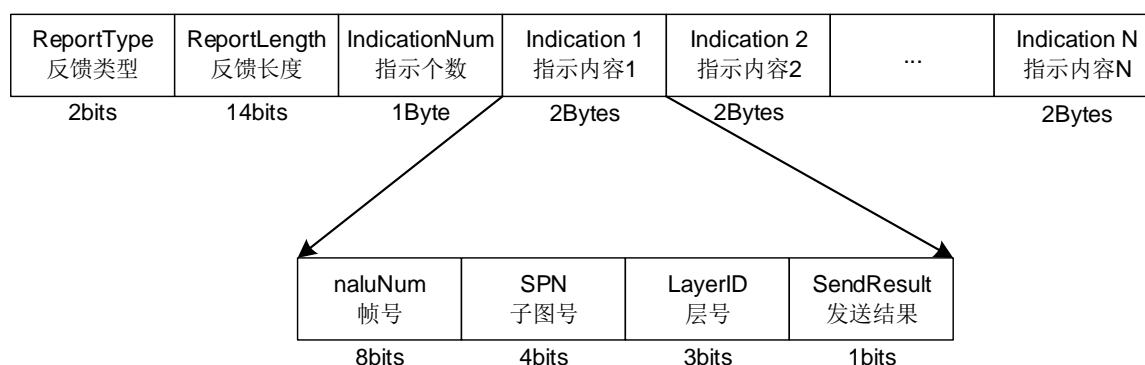


图 3.21 编码单元传输信息上报格式

链路质量信息上报则仅由定时器触发，上报间隔设置为一秒一次，从而触发上层应用对链路进行编码策略的调整和信道切换等功能。具体上报报文的格式如图 3.22 所示。

ReportType 反馈类型	ReportLength 反馈长度	Transmission_Delay 往返时延	Congestion_Delay 拥塞时延	Congestion_Length 拥塞长度	Packet_Loss_Rate 报文丢包率
2bits	14bits	2Bytes	2Bytes	2Bytes	2Bytes

图 3.22 链路质量信息的上报格式

3.5.5 性能分析

当使用了基于空口质量感知的视频帧编码设计后，投屏业务的上层应用即可根据底层的链路质量，实时动态更新当前的视频帧编码方案。在 24bit 像素位深、60fps 帧率的投屏场景下，不同分辨率、不同 I 帧与 P 帧的编码比例下，投屏业务的传输带宽如表 3.11 所示。

表 3.11 不同分辨率和编码机制下，投屏业务的传输带宽对比

分辨率 编码比例	720P (1280×720)	1080P (1920×1080)	2K (2560×1440)	4K (4096×2160)
1 : 1	102.97 Mbps	274.60 Mbps	488.17 Mbps	1171.61 Mbps
1 : 5	69.92 Mbps	186.45 Mbps	331.47 Mbps	795.54 Mbps
1 : 10	62.41 Mbps	166.42 Mbps	295.86 Mbps	710.06 Mbps
1 : 20	58.12 Mbps	154.97 Mbps	275.51 Mbps	661.22 Mbps
1 : 50	55.34 Mbps	147.57 Mbps	262.34 Mbps	629.62 Mbps

为了更加直观地对比不同投屏分辨率下，不同 I 帧与 P 帧的编码比例与投屏传输所需带宽的关系，将上述数据使用图 3.23 进行体现。

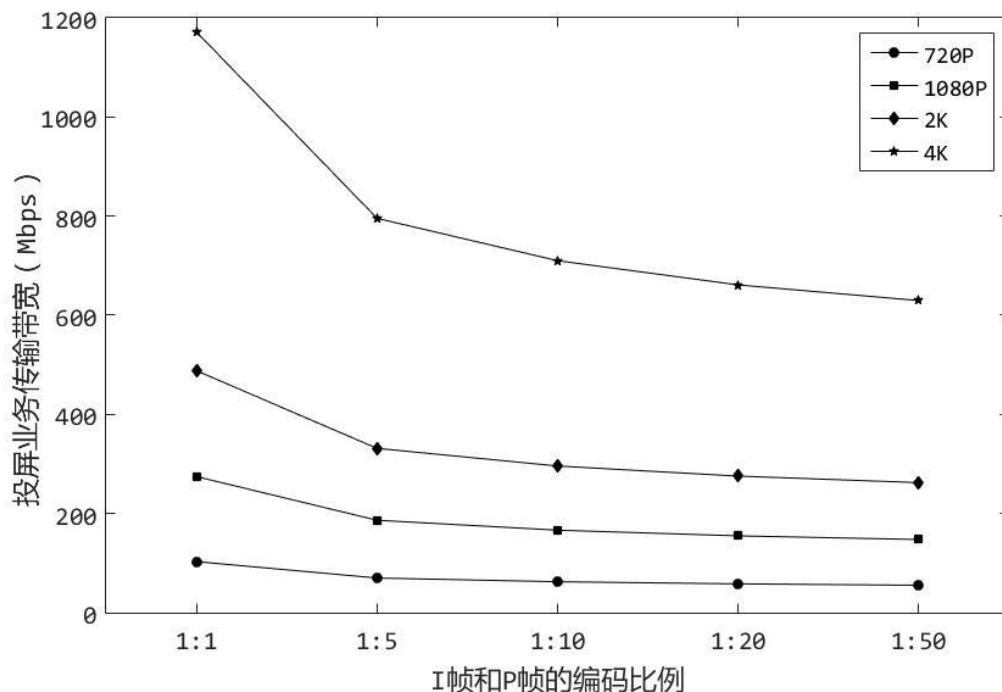


图 3.23 不同分辨率和编码机制下，投屏业务的传输带宽对比

由上述分析可得，当使用了基于空口质量感知的视频帧编码机制后，上层投屏应用可以更为灵活地调整视频帧的编码机制，在链路质量优秀的环境下提高 P 帧的使用数量，从而提高投屏业务的传输压缩率，降低投屏业务的传输带宽，由式(3-14)可知，这能够带来投屏传输时延的改善。

与此同时，在链路质量糟糕的环境下，本机制还能通过在编码过程中降低 P 帧的使用数量，使用更多 I 帧进行画面传输，从而在牺牲些许带宽的前提下保证投屏业务的稳定、可靠传输，避免投屏业务在复杂无线环境下因传输丢包而导致的卡顿与黑屏问题。

3.6 异构融合反馈重传机制

3.6.1 设计思路分析

在当前的丢包检测和重传机制下，投屏业务的传输时延计算、拥塞检测、重传调度等各个机制，都需要发送端能及时通过反馈帧获取接收端的报文接收信息。但是，单一的 WiFi 链路传输使得反馈帧的传输存在不足。由于反馈帧承载于 WiFi 链路进

行传输，即使对反馈帧使用了最高优先级的发送策略，反馈帧依旧需要参与空口的信道竞争，当链路繁忙时不得不进行退避等待，而潜在的高优先级业务数据使得反馈帧的发送阻塞现象进一步加剧，最终会使得反馈帧无法及时地到达发送端，如图 3.24 所示。

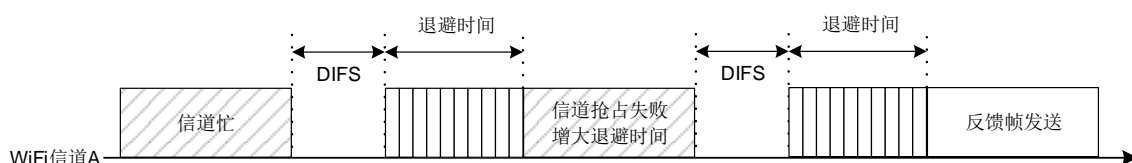


图 3.24 反馈帧在繁忙链路上传输

反馈帧的阻塞会导致发送端无法及时获取投屏报文的传输情况，从而无法及时进行调度传输和重传工作，最终影响了投屏业务的传输质量。

与此同时，蓝牙由于使用跳频技术进行数据传输，能够避开干扰信道继续通信。在跳频技术下，链路的传输带宽被划分成多个互不重叠的频率点，发送端与接收端通过伪随机码选取频率点作为数据传输的载波频率进行数据传输，从而使得各个报文的传输在不同载波频率之间随机地跳动改变。由于传输频率点的选择是随机的，信道中的干扰信号无法以同样的规律进行干扰，因此蓝牙技术下的数据传输具有很强的抗干扰能力。

针对反馈帧在单一 WiFi 链路上传输存在的问题，同时考虑到当前的常见便携设备上均同时支持 WiFi 技术和蓝牙技术，为了充分利用各种传输链路资源，实现投屏业务在不同架构上的全双工传输，本文提出了基于蓝牙技术与 WiFi 技术的异构融合反馈重传机制。

3.6.2 异构融合反馈设计

由于蓝牙使用的是跳频技术，因此蓝牙信道下的反馈帧可以与 WiFi 链路下的投屏业务同时传输，从而实现投屏业务的全双工通信，如图 3.25 所示。

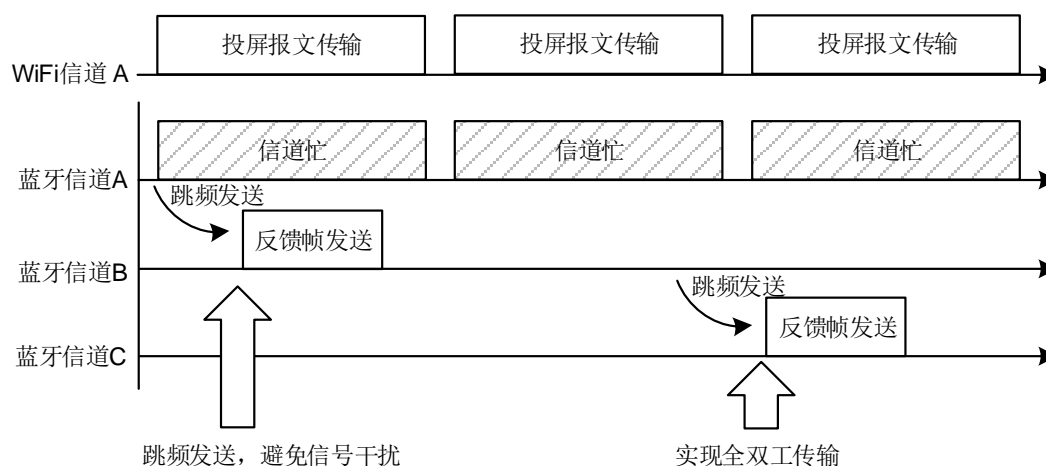


图 3.25 蓝牙的跳频传输能力

通过将蓝牙技术与 WiFi 技术联合应用于投屏业务的传输，反馈帧可以避免参与 WiFi 信道的竞争，因此不仅降低了反馈的时延，还降低了投屏业务在 WiFi 链路上的传输时延，提高了投屏业务的传输吞吐率，如图 3.26 所示。

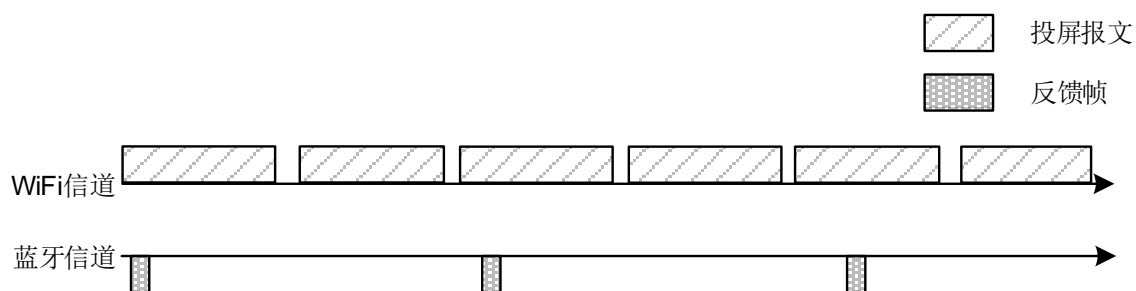


图 3.26 WiFi 与蓝牙下的全双工传输

与此同时，考虑到蓝牙信道上的反馈帧也存在潜在的丢包问题，当投屏业务接收端收到相同 LLC ID 序号的冗余投屏业务报文时，判定发送端出现了错误的超时重传，即反馈帧的传输出现错误，此时则选择在 WiFi 链路和蓝牙链路上同时进行反馈帧的传输，从而保证反馈帧的传输成功率，如图 3.27 所示。

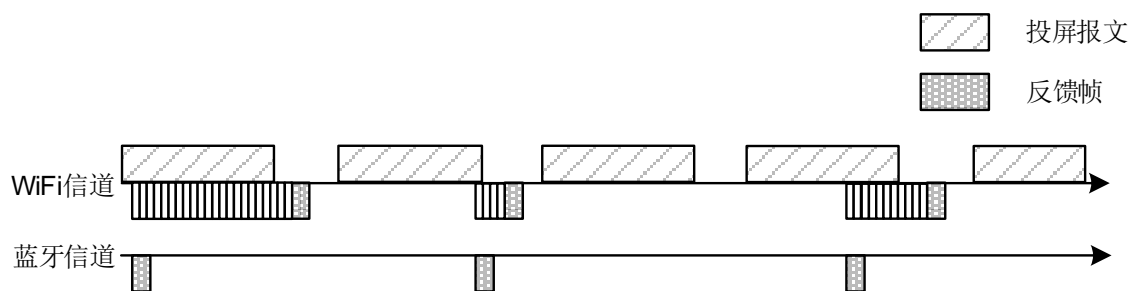


图 3.27 WiFi 与蓝牙下的冗余反馈传输

通过在蓝牙链路和 WiFi 链路上联合进行反馈帧的传输,使得反馈信息能够更可靠、更快速地送至发送端,发送端因此能更及时地发现链路丢包情况,触发报文重传,并更新链路质量信息,进行传输调度和信息上报等操作,从而进一步增强投屏业务的传输质量。

3.7 本章小结

本章的主要内容是投屏业务在 WiFi 链路上的传输质量保证方案的构建和设计。首先分析了投屏业务现有的传输问题和技术难点,以及投屏技术的传输需求和目标,并提出了多条用于保证投屏传输质量的设计方案。为了降低投屏业务的无线链路丢包率,设计了投屏报文的反馈重传方案,该方案基于 LLC ID 的报文封装格式设计实现了投屏报文的丢包检测和反馈重传功能,高效的反馈帧构造机制则是极大地优化了反馈帧在链路上的传输开销,接着还基于 M/G/1 排队模型对投屏报文在反馈重传方案下的平均传输时间进行了理论分析;为了降低投屏业务的传输时延,设计了针对投屏报文的调度传输方案,通过使用多硬件队列的调度发送与报文主动聚合机制克服了硬件队列上的报文阻塞问题,使得投屏业务的传输更为通畅和及时;基于报文优先级的调度发送机制更是能够进一步提高投屏报文在 WiFi 链路上的传输效率。在基于空口质量感知的视频帧编码机制下,编码单元传输情况的上报不仅使得投屏业务在丢包场景下能够及时恢复图像的正确传输,还能触发投屏业务应用层灵活地调整编码机制,从而让投屏业务在高质量链路上能传输更高分辨率的画面,提高了投屏传输吞吐率,且在链路恶劣的情况下保证投屏画面的稳定传输,提高了投屏传输可靠性;最后介绍了异构融合反馈重传机制,通过联合使用蓝牙链路和 WiFi 链路,改进了投屏报文的反馈重传机制,实现了投屏业务的全双工传输,使投屏业务的传输质量有了更进一步的提升。

第4章 投屏业务平台实现与测试

上一章详细地描述了针对保证投屏业务传输质量所提出的各个思想与技术方案，本章在此基础上，在开发平台上对上述技术进行了设计实现与测试，并对比分析了上述技术的传输保证效果和性能水平。

4.1 软件设计实现

4.1.1 开发平台与架构

为了对上一章节所提出的各个方案与技术方案进行性能验证，通过与某知名企业进行项目合作研究，在 Linux 平台和 Windows 平台均进行了发送端和接收端的代码实现，从而同时满足 Linux 平台至 Windows 平台的投屏业务的传输质量保证需求（如手机投屏至笔记本的多屏协同功能等），和 Windows 平台至 Linux 平台的投屏业务的传输质量保证需求（如笔记本投屏至安卓平板的多屏协同功能等）。

在基于 Linux 操作系统的终端平台上，使用的是企业自主研发的无线网卡设备，因此通过对该无线网卡设备的驱动代码进行深度开发，实现了多硬件队列重传方案下的各个技术设计与基于空口质量感知的视频帧编码设计，并基于实际投屏场景进行了投屏业务的传输测试与性能分析。

与此同时，同样使用企业自主研发生产的 Windows 操作系统笔记本作为投屏场景的另一类终端平台，通过在 Windows 操作系统的 WFP 框架下进行网络报文的截获与处理，从而同样地实现了多硬件队列重传方案下的各个技术设计与基于空口质量感知的视频帧编码设计，并联合 Linux 平台进行了投屏业务的传输与测试。

无论是 Linux 平台还是 Windows 平台，都是用同一套框图来实现本文提出的设计，区别主要在和操作系统交互的接口函数和实现的投屏业务处理层所在的层次，如 Linux 平台下的投屏业务处理层在无线驱动程序的 UMAC 层截获报文实现，而 Windows 平台则使用 WFP 框架对 TCP/IP 协议栈的报文进行过滤来实现。

整体架构框图如图 4.1 所示。

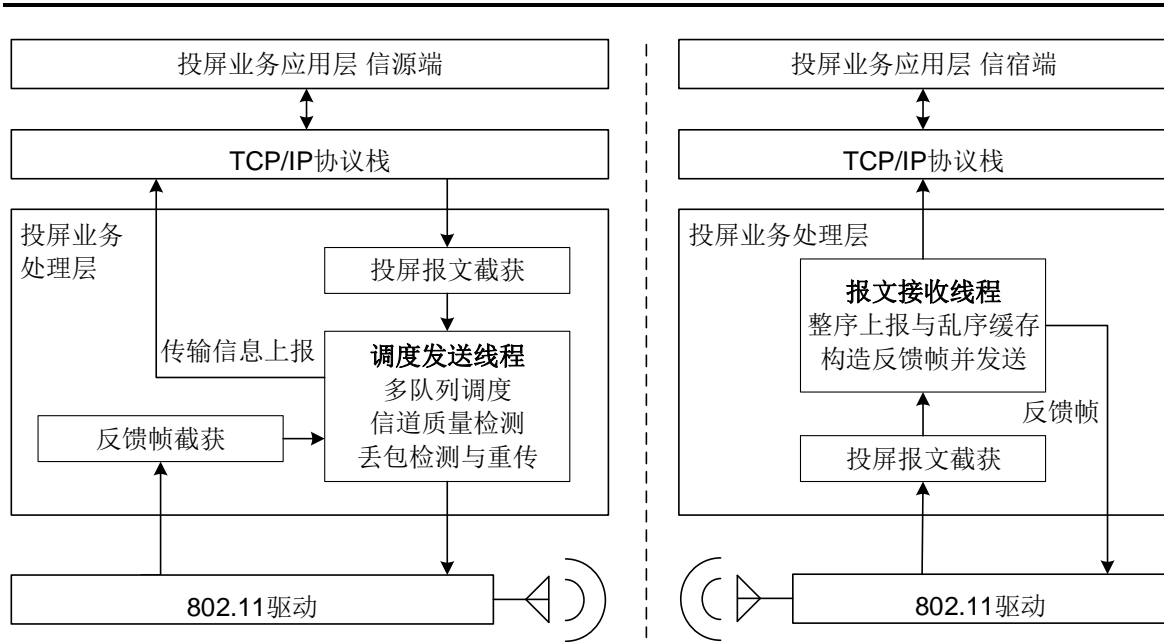


图 4.1 投屏业务传输质量保证流程框图

为了避免冗余，本章仅介绍了在 Linux 平台作为发送端、Windows 平台作为接收端的投屏场景下，收发端的投屏业务处理层的设计实现。在 Windows 平台作为发送端、Linux 平台作为接收端的投屏场景下，除了与操作系统的交互（使用的函数接口等）有所不同，实际的设计实现与前者基本相同。

4.1.2 发送端设计实现

在 Linux 平台下，发送端的投屏业务处理层运行在无线网卡驱动的 UMAC 层中，因此相关的资源分配、线程创建等操作伴随着驱动程序被加载而开始。Linux 操作系统将每一个网络设备都抽象成 `struct net_device` 结构体，并使用 `struct net_device_ops` 结构体作为与上层 TCP/IP 协议栈的交互接口，该结构体中的函数指针 `ndo_start_xmit` 负责将协议栈的报文送至无线网卡驱动进行发送。因此，通过对函数指针 `ndo_start_xmit` 对应的发送函数 `wal_bridge_vap_xmit()` 进行修改，即可实现待发送投屏报文的过滤和截获。

驱动程序使用特定的函数接口来上报上报接收到的报文，该函数接口最终使用 Linux 操作系统下的 `netif_rx_ni()` 接口把报文送至上层 TCP/IP 协议栈。因此，通过对函数接口中的代码进行对应修改，即可实现待接收投屏报文的过滤和截获。

为了实现多硬件队列重传方案，还创建了投屏业务发送线程，专门负责投屏报文

的封装、缓存、发送等操作，使用环形队列实现进程间无竞争的报文传输，并通过运行一个高精度定时器，使用完成量周期性唤醒发送线程进行超时重传、链路信息上报等功能。Linux 平台下投屏业务发送端的具体设计实现框图如图 4.2 所示。

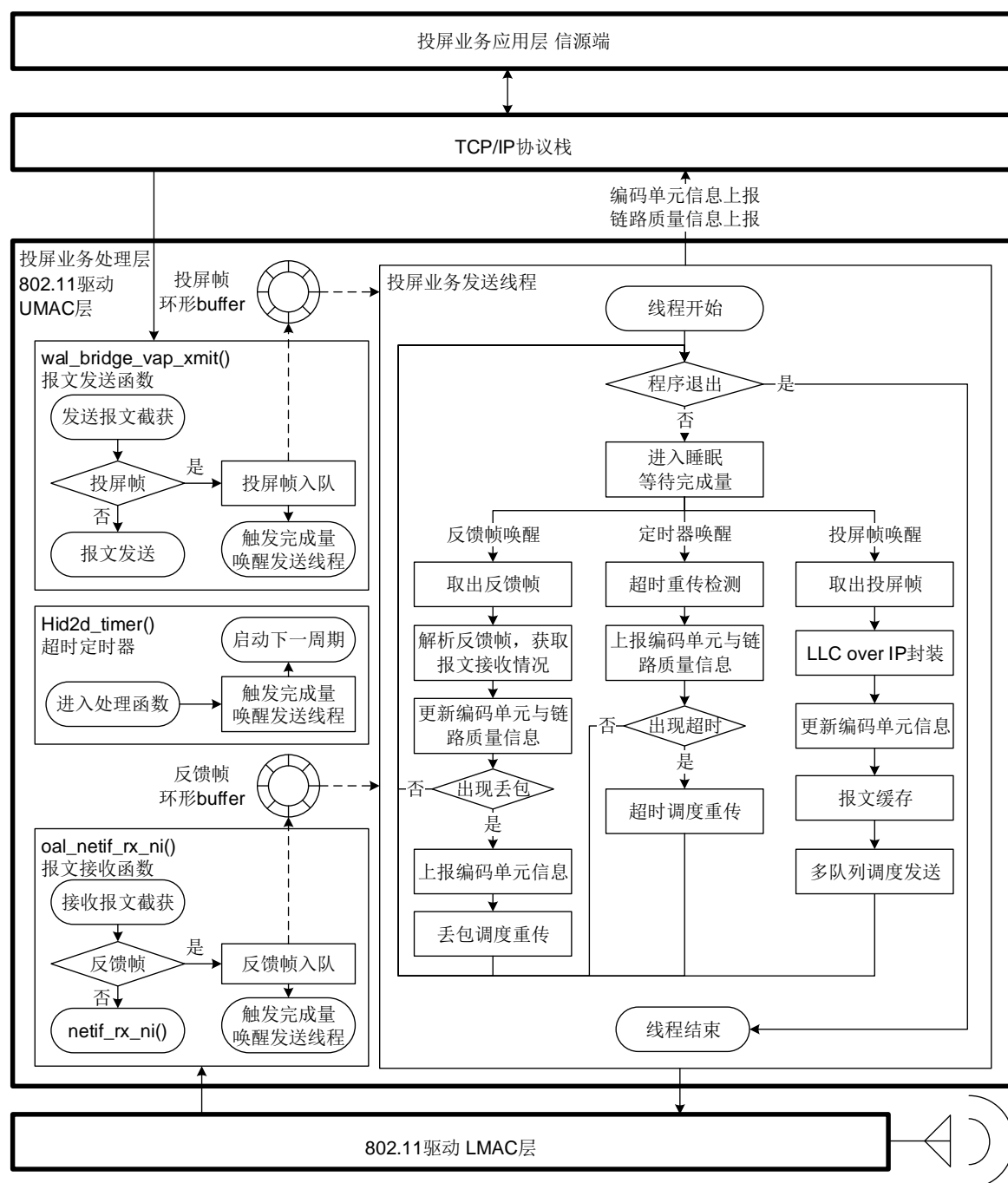


图 4.2 投屏业务质量保证设计实现 发送端框图

当投屏报文在报文发送函数中被截获时，报文会被送入指定环形队列，并通过触

发完成量来唤醒发送线程处理投屏报文；当反馈帧在报文接收函数中被截获时，反馈帧也会被送入对应环形队列，然后唤醒发送线程进行反馈帧的处理；与此同时，定时器也会周期性唤醒发送线程执行确认超时检查等任务。

针对投屏帧的唤醒操作，发送线程会逐个将报文取出，进行 LLC over IP 封装，并基于 LLC ID 更新对应的编码单元传输信息链表，用于信源信道的联合编码；然后报文会被缓存进缓存数组，并基于多硬件队列调度机制和主动聚合机制进行调度发送。报文的发送时间会被记录进缓存数组，用于确认超时的判断和链路质量信息的测量。

针对反馈帧的唤醒操作，发送线程会逐个将反馈帧取出，通过解析反馈帧获取接收端的丢包信息和等待超时信息，然后使用这些信息更新编码单元传输信息链表，并从缓存数组中取出丢包的报文，基于多硬件队列调度机制进行调度重传。若发现接收端因为等待超时而放弃对应报文的接收，则立即将编码单元传输信息送至投屏业务应用层，通知上层迅速发送新的参考帧，以保证投屏业务的传输不会因此中断卡顿。

针对定时器的唤醒操作，发送线程会计算当前时间与发送窗口中报文发送时间的差值，判断是否存在报文超时未收到反馈帧的接收确认；针对确认超时的报文，发送线程会基于多硬件队列调度机制进行调度重传。同时，发送线程还会周期性地当前的链路质量信息和编码单元传输情况上报至投屏业务应用层，确保上层能及时发现链路质量的变化，灵活地调整编码机制。

4.1.3 接收端设计实现

在 Windows 平台下，接收端的投屏业务处理层运行在 WFP 框架下的过滤驱动程序中。过滤驱动在初始化过程中，配置报文过滤的位置为 TCP/IP 协议栈中的 MAC 层，并设置对应的处理函数，从而对接收到的投屏报文进行截获。与此同时，过滤驱动程序的初始化流程还创建了接收线程、定时器等模块，实现投屏报文的解封装、按序上报、发送反馈帧等功能。Windows 平台下投屏业务接收端的具体设计实现框图如图 4.3 所示。

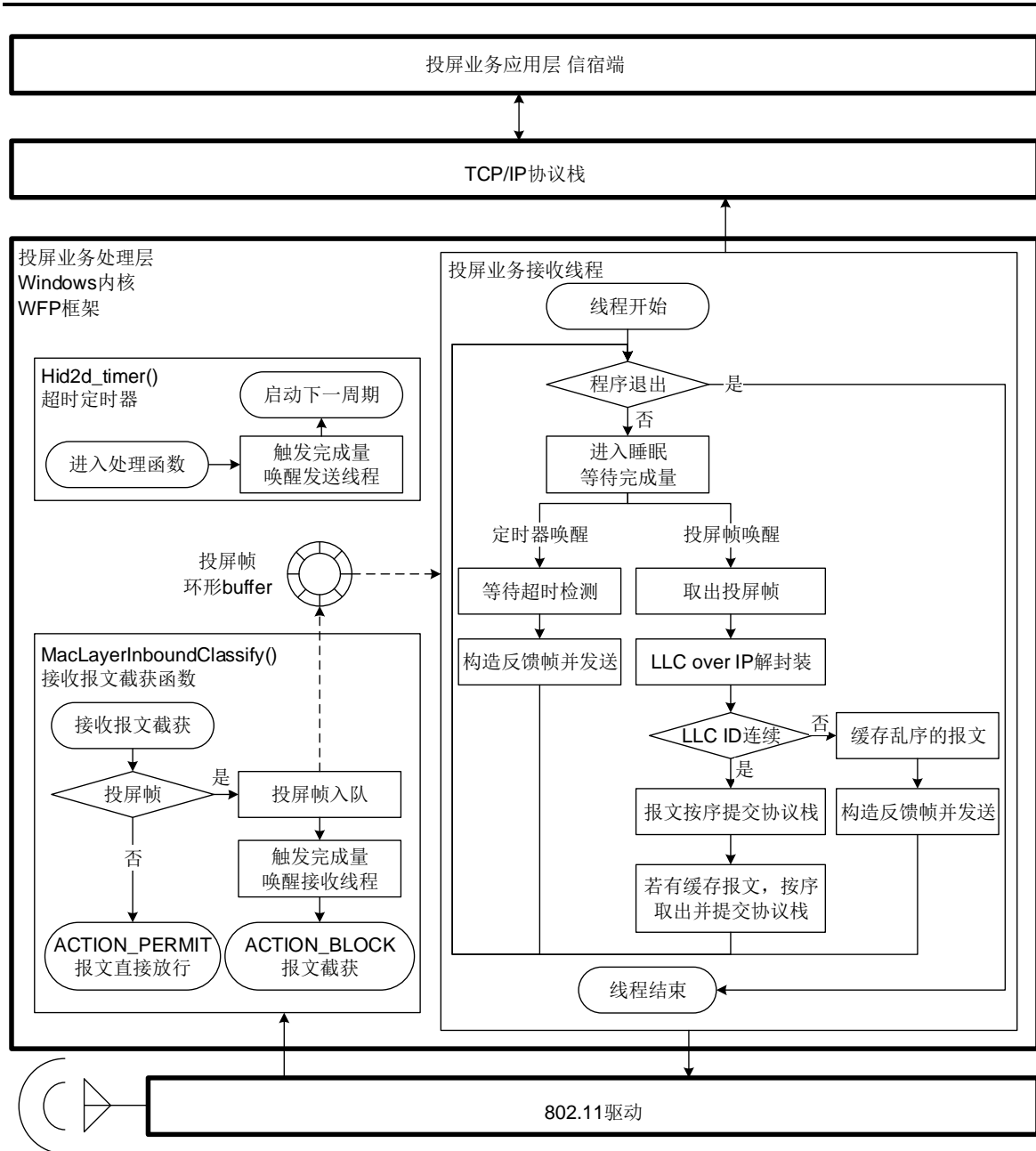


图 4.3 投屏业务质量保证设计实现 接收端框图

当接收到的投屏报文在过滤驱动的处理函数中被捕获时，报文会被送入投屏帧对应的环形队列，并通过触发完成量来唤醒接收线程处理投屏报文；与此同时，定时器也会周期性唤醒接收线程，来判断是否存在报文出现等待超时，并构造反馈帧送至发送端。

针对投屏帧的唤醒操作，接收线程会将报文从环形队列取出，然后解封装报文，获取 LLC ID 信息、报文优先级信息和投屏帧的 IP 报文。接收端本地记录待接收投

屏帧的 LLC ID 序号，当收到投屏帧的 LLC ID 与待接收序号一致时，认为报文传输正常，于是将投屏帧 IP 报文正常送至协议栈；若 LLC ID 序号不一致，则判定为出现丢包，将该报文缓存至缓存数组，并记录下缓存的时间。检测到乱序时，接收端会立即构造并发送一个反馈帧，然后设置冷却时间，避免冗余传输。

针对定时器的唤醒操作，接收线程会基于当前时间和缓存数组中的报文缓存时间，判断报文的等待重传是否出现超时，当等待超时则放弃对该帧的等待，继续处理后续报文，并记录下放弃等待的报文信息，由反馈帧通知发送端。除此以外，接收线程还会读取缓存数组中的报文接收情况，基于高效反馈机制构造反馈帧，并在高优先级队列上发送至业务发送端，通知发送端及时重传。

4.2 反馈重传方案验证

4.2.1 测试流程介绍

为了验证本文设计的投屏质量保证机制能否正确地进行链路拥塞和丢包检测并进行反馈重传，本章搭建了投屏业务传输的实际环境。测试使用的设备是 Linux 平台下的手机终端和 Windows 平台下的笔记本终端，并使用厂商自主研发的投屏程序进行投屏。与此同时，为了模拟复杂的无线链路环境进行测试，本测试还额外使用的多个手机终端设备和笔记本终端设备，在测试的同时在 WiFi 链路上传输大带宽视频流数据，来构造多设备抢占信道资源的测试环境。

4.2.2 测试结果分析

当投屏业务在复杂无线环境下运行时，使用反馈重传方案来保证投屏质量的手机终端设备会实时地记录底层的链路质量测量信息和接收到的反馈信息，并通过 adb 工具进行打印输出，测试过程中得到的输出信息如图 4.4 所示。

[illegible]

图 4.4 链路质量信息和反馈信息的打印

在使用和不使用反馈重传方案的设备上均进行十次独立的复杂 WiFi 环境下投屏测试，并分别记录投屏过程中的视频帧丢包率，测试结果如图 4.5 所示。

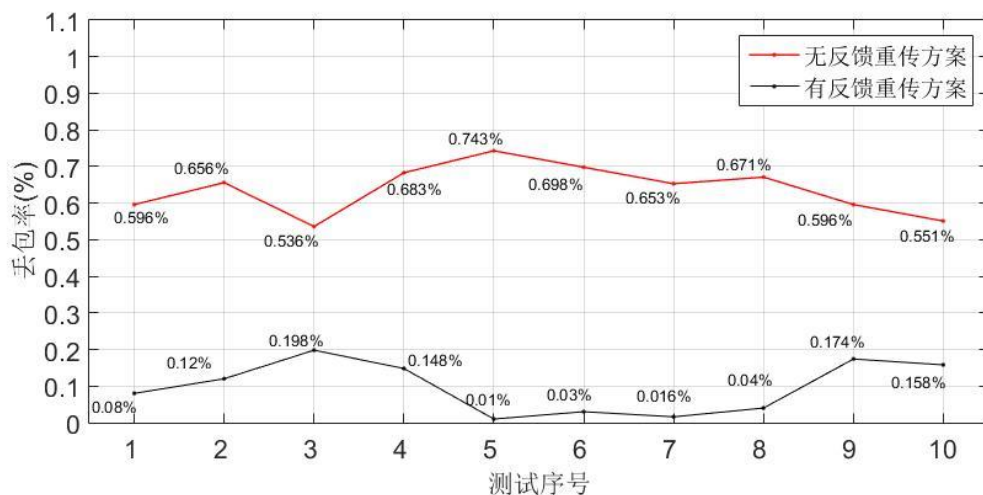


图 4.5 投屏业务传输过程的丢包率信息

质量成功得到了保证。

4.3 调度传输方案验证

4.3.1 测试流程介绍

验证了反馈重传方案的正常工作后,本文进一步验证了投屏业务的调度传输方案对传输时延的改善效果。测试过程中,视频帧的端到端时延从投屏业务的应用层中获取,并且同样使用额外设备在 WiFi 链路上传输大带宽数据来模拟多设备信道抢占环境,通过对比使用与不使用调度传输方案的两个场景下的测试数据,验证本文的调度传输方案对投屏业务的性能改善。

4.3.2 测试结果分析

在测试过程中,通过对空口报文进行抓包,可以发现投屏业务的报文实际调度传输情况,抓包结果如图 4.6 所示。

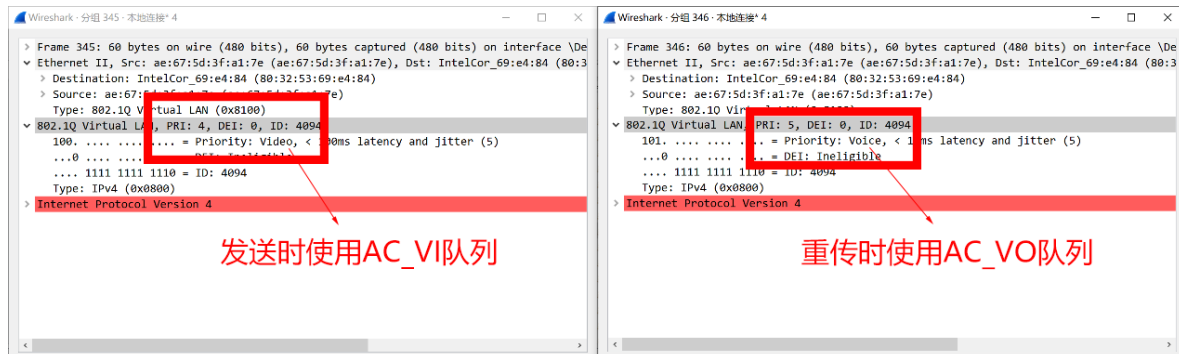


图 4.6 投屏业务报文的空口抓包信息

从空口抓包信息中可以看出,当链路出现拥塞或丢包时,成功地触发了业务报文的重传,并且重传报文及后续新传报文均在高优先级队列进行传输,避免了拥塞导致的新传报文无法传输的问题,成功地实现了投屏业务的多硬件队列联合调度传输功能。在不使用本文设计的调度传输方案和使用调度传输方案的场景下,通过从投屏业务应用层获取视频帧的传输时延信息并绘图,得到投屏业务传输过程中的时延信息如图 4.7 所示。

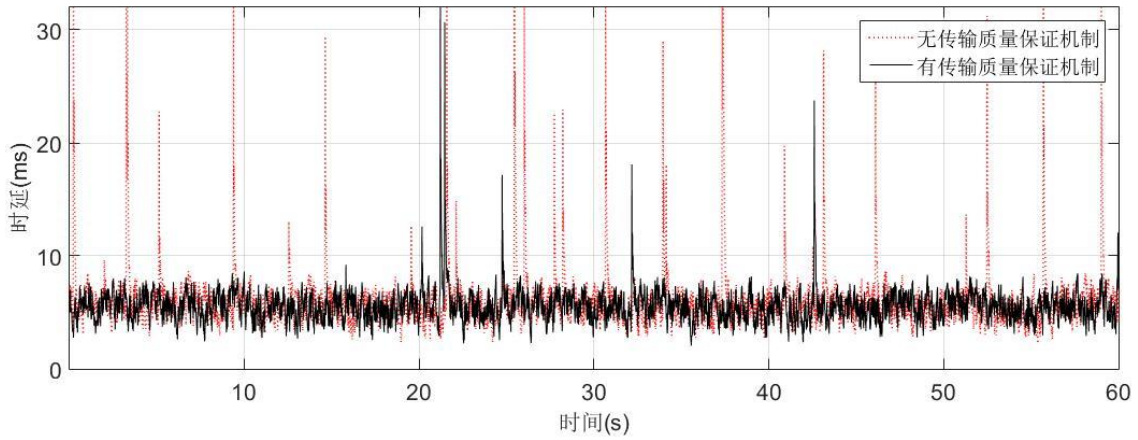


图 4.7 投屏业务传输过程的时延信息

从测试结果中可以得出，在链路通畅的情况下，两个场景的实际的传输时延相差不大，都能满足正常的投屏传输需求；但是在复杂的无线环境下，不使用反馈重传方案和调度传输方案的设备在传输投屏业务报文时，会频繁地出现拥塞和丢包现象，导致投屏传输时延波动明显，造成画面卡顿花屏；经过本文设计的反馈重传方案和调度传输方案进行传输质量保障之后，投屏业务的时延抖动得到明显改善，传输稳定性显著提高，证明了本文提出的设计可以保证 WiFi 系统中的投屏业务传输质量。

4.4 空口质量感知功能验证

4.4.1 测试流程介绍

在本文提出的基于空口质量感知的视频帧编码机制中，投屏业务应用层基于底层的视频帧传输情况进行编码机制的动态调整，实现更高的信道利用和更可靠的画面传输。因此，本设计的关键之处在于底层能否及时通过反馈帧，获取视频帧编码单元的链路传输情况，并及时地上报投屏业务应用层，从而触发编码机制的调整。

由于投屏业务应用层的程序由合作厂商负责实现，目前合作厂商还没有把动态编码机制的相关功能开发好，因此测试只进行链路级的验证。在测试过程中，通过运行完整的投屏业务流程，判断底层是否能够对编码单元传输情况进行实时监测并及时上报，从而验证底层设计是否可以有效地支持应用程序进行动态编码。

4.4.2 测试结果分析

在 1080P 分辨率下的投屏业务传输过程中,底层基于反馈帧监测每个编码单元对应报文的接收确认结果,实时上报编码单元的传输情况,并测量底层从开始发送编码单元首个报文到上报整个编码单元传输情况的时延,一分钟内测试得到的上报时延分布图如图 4.8 所示。

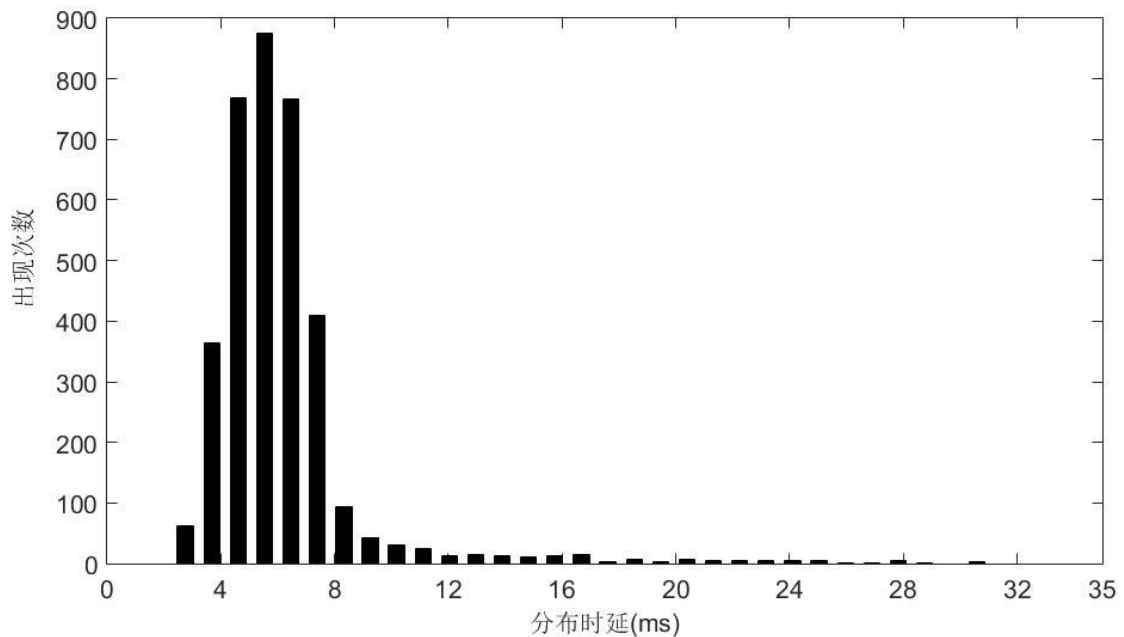


图 4.8 投屏业务视频帧的上报时延分布统计

本文设计的编码单元传输情况检测机制成功地对每一个编码单元的传输情况进行了确认和上报。从分布图中可以看出,在链路通畅的情况下,大部分编码单元都可以在 16ms 以内确认传输情况并进行上报;少量情况下,由于编码单元对应的报文数量过多,或由于链路出现拥塞导致报文的确认被延迟,编码单元的传输情况需要更长时间才能得到确认。因此,测试证明了底层链路上的设计能够有效地支持上层应用进行编码策略的灵活调整,实现基于空口质量感知的视频帧编码,进行投屏业务吞吐率的提升。

4.5 章节小结

针对保证投屏业务在 WiFi 链路上传输质量的设计方案,本章同时在 Linux 平台

和 Windows 平台上进行了代码实现。接着通过实际投屏场景的运行，在使用和不使用本文设计的反馈重传方案和调度传输方案的前提下，分别进行了投屏场景的运行测试，通过测试结果可以看出，本文设计的反馈重传方案有效地降低了投屏业务在复杂无线环境下的丢包率，调度传输方案则显著改善了投屏业务在复杂无线环境下的传输时延波动现象。最后验证了本文的空口质量感知设计能够有效地监测视频报文的传输情况，并及时地将编码单元传输信息上报至投屏业务应用层，证明了该设计在底层链路上可以很好地支持上层应用程序进行动态灵活的视频帧编码。

第5章 游戏业务传输质量保证的设计

随着便携式移动终端设备的普及使用与上层软件生态的日益丰富，承载在 WiFi 链路上的游戏业务需求也在日益增加，这就对底层无线传输链路针对游戏业务的传输调度提出了更高的要求。针对游戏业务在 WiFi 链路上传输的高质量需求，本章分析了当前存在的问题与难点，并基于支持双频双发的 WiFi 芯片平台，进一步提出了保证游戏业务的传输质量的设计方案。

5.1 技术路线分析

5.1.1 现有难点与问题分析

随着笔记本、智能手机等便携式终端设备的不断普及，运行在这些终端设备上的游戏业务生态也已经十分丰富。游戏业务对链路的时延和丢包率十分敏感，而 WiFi 链路是终端设备和互联网之间的重要数据传输通道，因此用户对使用 WiFi 链路承载的游戏业务的传输质量要求也越来越高。

由于在当前的 WiFi 机制下，当终端设备中的游戏业务与多业务并发传输时，底层缺少针对各类报文优先级足够细致的区分，终端设备无法正确对游戏业务的报文优先级进行正确判定处理，使得实际报文传输机会被低优先级业务抢占，空口业务的传输出现拥塞，游戏业务的传输因此被阻塞。

在多设备共存的复杂 WiFi 场景下，游戏报文未被送至足够高优先级的队列中，因此报文在发送时常常会由于优先级过低而在竞争信道过程中失败，导致报文在底层硬件队列中不断退避和等待，从而出现延时和卡顿，影响了游戏业务的使用体验。

当前市面上已有少量网卡设备支持 DBDC 双频发送，但是由于缺少底层驱动与网卡的软硬件配合机制，实际的空口无线报文发送依旧使用时分发送等伪双频机制，无法充分利用 DBDC 的双频发送工作效率。因此，针对支持 DBDC 功能的无线网卡设备，设计一套完善的双频调度发送机制，对游戏业务的无线传输质量的保证与提高有极大作用。

5.1.2 游戏业务的需求与目标

无线传输链路的时延与丢包现象直接影响了上层游戏业务带给用户的使用体验，因此，本设计的目标是充分利用 WiFi 两个频段下的传输通道，在多业务并发的场景下，确保游戏业务能够优先地在空口链路上进行传输，并在多设备共存的复杂无线环境下，确保游戏业务的传输可靠性，降低链路传输时延，从而高效率地利用多条信道资源，保证游戏业务能够在 WiFi 链路上得到可靠、低时延的高质量传输。

5.1.3 游戏业务质量保证的整体设计思想

针对上述游戏业务的传输难点和问题，为了实现游戏业务传输的需求与目标，本文提出了以下设计思想：

(1) 为了实现各个优先级业务的调度传输，保证游戏业务的高优先级可靠传输，本文设计了高效的报文封装机制。该机制在使用极少量封装开销的同时，实现了对多种类型以太网帧的调度发送，并能够支持丢包检测与报文重传功能，从而使终端设备在多设备竞争的复杂无线环境下，保证游戏业务在空口上能够得到高质量的传输。

(2) 本文基于报文封装机制，进一步设计了无线链路 KPI（关键性能指标）测量机制，通过联合使用业务报文封装信息和反馈帧数据进行计算，实时地探测当前双频下每个硬件队列的业务传输时延、队列拥塞时延、队列拥塞长度等链路 KPI 信息，为后续的业务动态调度机制提供数据支撑。

(3) 基于无线链路的 KPI 测量信息，本文设计了双频多队列调度发送机制，将游戏业务与其它各个业务的报文同时承载在 2.4GHz 和 5GHz 频段上，进行双频与多队列下的灵活调度，并针对丢包和拥塞场景，对业务报文进行了跨队列和跨频段的重传，从而充分利用无线网卡设备的 DBDC 功能，降低游戏业务在无线链路上传输时延和丢包率，保证游戏业务的传输质量。

(4) 针对终端设备多业务并发的场景，本文设计了基于链路 KPI 的动态流量调度机制。在该机制下，本文将传输时延、队列拥塞情况等链路质量信息与令牌桶、RED 算法、TCP 平滑降速等技术进行结合使用，从而使各个优先级下业务的传输调

度方案能随着当前链路质量进行动态地调整，避免低优先级业务抢占游戏业务带宽而导致空口拥塞，在保证游戏业务具有高质量的传输效果的同时，确保各个业务在空口链路上能够高效传输。与此同时，该机制还能够兼容其他普通终端设备的正常使用，具有终端透明的特点。

5.2 高效的报文封装机制

5.2.1 设计需求分析

在多任务并发和多设备共存的无线传输场景下，游戏业务报文的无线传输会受到其他业务及其他设备的干扰与抢占，为了保证游戏业务在无线信道竞争中取得优势，底层在发送业务报文时需要支持传输优先级的配置与硬件队列的选择；复杂无线环境下游戏业务报文的传输也可能出现阻塞和丢包，因此底层同样需要一套机制保障游戏业务报文的可靠传输。因此，本章基于投屏业务下的反馈重传方案，设计了一种高效的报文封装机制，从而在最大程度降低链路传输开销的同时，支持对多个优先级下的业务报文同时进行独立的调度传输与丢包检测重传功能。

5.2.2 报文封装结构设计

为了满足不同优先级的业务报文之间能够独立地进行调度传输，本文针对四个硬件队列对应的报文优先级，设计了基于子通道号和子通道序号的丢帧检测和重传机制，并设计了高效的封装结构进行报文封装。具体的报文封装格式如图 5.1 所示。

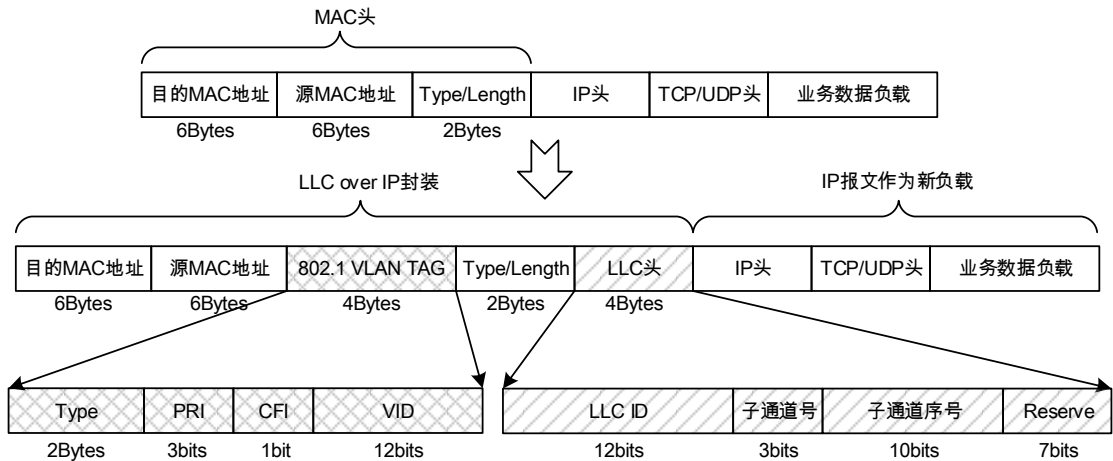


图 5.1 高效的报文封装格式

该封装机制包括 LLC 头和 VLAN TAG 头两个部分。其中 LLC 头负责实现各个业务优先级下的丢包检测与报文重传功能，VLAN TAG 头则负责实现硬件队列的选择和封装后报文的类型识别功能。

5.2.2.1 LLC 头封装

在 LLC 头中主要设计了三个字段：LLC ID、子通道号和子通道序号。

LLC ID 字段负责实现报文的缓存和高效反馈功能。所有待发送的业务报文都使用同一组 LLC ID，按照报文到达底层的顺序进行编号，并基于 LLC ID 将报文缓存至发送端的缓存数组中用于重传；接收端的反馈帧同样基于 LLC ID 进行报文接收情况的反馈，通过将多个优先级业务的接收信息放入同一个反馈帧进行反馈，降低了反馈帧的链路开销，提高了信道资源的使用效率。

子通道号则对应业务的优先级信息，每个子通道号与实际使用的硬件队列相对应。当前设计为该字段分配了 3 个比特，以支持今后的拓展使用。

子通道序号则对应特定业务优先级下的报文传输顺序。在每个子通道号下，独立维护各自的子通道序号信息，不同子通道号下的子通道序号之间则相互独立。

通过在用户态对各个业务的五元组信息进行监测，并将五元组与实际优先级的映射关系下发至底层，底层可以基于五元组信息对报文的子通道号和子通道序号信息进行封装。业务发送端和接收端通过子通道序号信息，来判断对应优先级下业务报文的丢包和阻塞情况，并获取对应业务优先级下各个硬件队列的传输质量信息。

5.2.2.2 VLAN TAG 头封装

VLAN TAG 头主要由 Type 字段、PRI 字段、CFI 字段和 VID 字段组成。在本机制下，通过将 Type 字段设置为 0x8100，使底层网卡驱动将封装报文识别成 802.1Q Tag 帧，以实现硬件队列的选择和封装报文的识别功能。对于 802.1Q Tag 帧，通过配置 PRI 字段选择发送报文所使用的硬件队列。CFI 作为经典格式指示符，固定设置为 0。VID 字段则标识了封装后的业务报文的类型，业务接收端基于该字段判断报文是不是需要进行解封装的业务报文。

5.2.3 报文封装效率对比

针对业务报文在数据链路层的隧道封装，目前常见的封装技术有 PPTP 点对点隧道协议、L2TP 隧道协议、CAPWAP 隧道协议等。

PPTP 点对点隧道协议是对点到点协议（PPP）的拓展，为 PPTP 客户机与 PPTP 服务器之间提供了加密通信服务^[51]。在该协议下，业务报文载荷经过多层封装，添加了 PPP 报头、GRE 报头、IP 报头及新的二层封装，具体封装格式如图 5.2 所示。

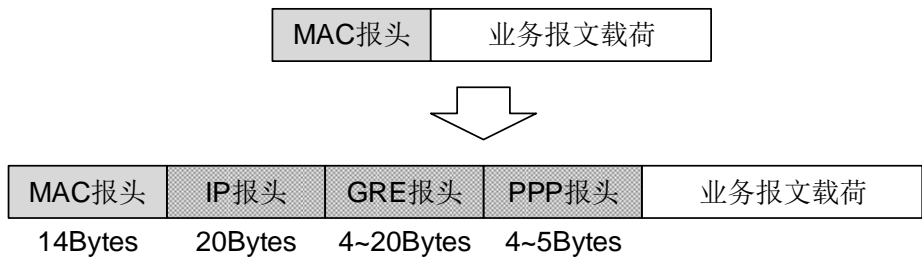


图 5.2 PPTP 协议下的隧道封装

L2TP 隧道协议综合了 L2F 隧道协议和 PPTP 隧道协议制定而成，功能上与 PPTP 隧道协议类似，但是相较 PPTP 仅支持对 IP 报文的封装，L2TP 隧道协议只要求面向数据包的点对点连接，并支持多隧道封装、包头压缩、隧道验证等功能^[52]，其具体封装格式如图 5.3 所示。

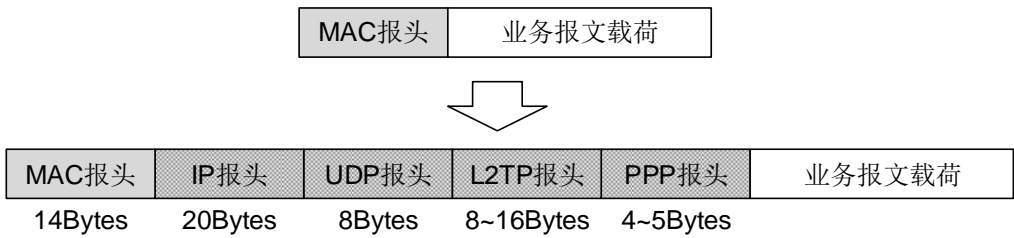


图 5.3 L2TP 协议下的隧道封装

CAPWAP 协议是一个通用的隧道协议，该协议通过将 802.11 协议的无线报文隧道封装成 802.3 协议的报文，实现了无线终端接入点 AP 和无线网络控制器 AC 之间的数据传输，从而让 AC 能够对其所关联的 AP 进行集中的管理和控制^[53]。该协议下，无 DTLS 加密的数据报文封装格式如图 5.4 所示。

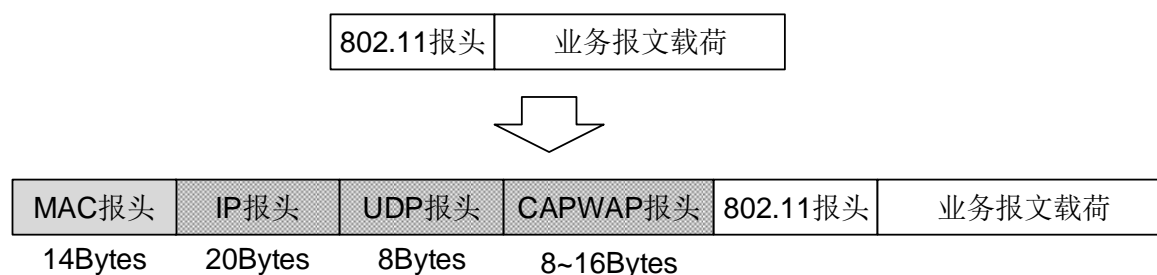


图 5.4 CAPWAP 协议下的隧道封装

为了对比本文设计的报文封装机制与目前主流使用的封装机制的封装效率，设业务报文负载均为 1400 字节，主流封装机制选择开销最小的封装策略，报文封装效率的计算公式为：

$$\eta = \frac{Length_{\text{业务报文负载}}}{Length_{\text{封装开销}} + Length_{\text{业务报文负载}}} \times 100\% \quad (5-1)$$

于是，各个封装机制下的报文封装效率如表 5.1 所示。

表 5.1 不同报文封装机制的效率对比

封装协议	业务报文负载	封装开销	封装效率 η
PPTP 隧道	1400 Bytes	28 Bytes	98.039%
L2TP 隧道	1400 Bytes	40 Bytes	97.222%
CAPWAP 隧道	1400 Bytes	50 Bytes	96.552%
高效报文封装机制	1400 Bytes	8 Bytes	99.432%

根据各个封装机制下的效率计算结果可以发现，本文设计的高效报文封装机制相对目前主流的报文封装技术，具有更小的报文封装开销，因此能够带来更高的封装效率和链路传输性能。

5.3 无线链路 KPI 测量机制

5.3.1 设计需求分析

游戏业务等各种业务报文在发送过程中，可能由于并发任务信道抢占、多设备竞争干扰等因素，造成业务报文在底层硬件队列出现阻塞，并影响后续报文的正常发送，因此合理的频段与队列调度发送机制需要依据底层传输质量信息配合完成。本文基于高效的报文封装机制和反馈帧的接收信息反馈功能，设计了无线链路下的 KPI

测量机制，通过检测双频多硬件队列下各个硬件队列的报文往返时延、报文拥塞时延和报文拥塞长度信息，为后续的双频多通道调度及流量控制机制提供底层链路质量数据支持。

5.3.2 多硬件队列下的往返时延测量

业务报文的往返时延指业务发送端的报文发送时间到发送端收到反馈帧对应报文确认的时间之差，该时延直观地反应了当前时刻下对应硬件队列的传输质量，本机制下两个频段下的各个硬件队列分别能测得不同的时延信息。

为了实现各个硬件队列下往返时延的测量，发送端在缓存数组中不仅记录了报文的发送时间信息，还记录了每个报文的发送频段和使用的硬件队列（即子通道号）。相关流程如图 5.5 所示。

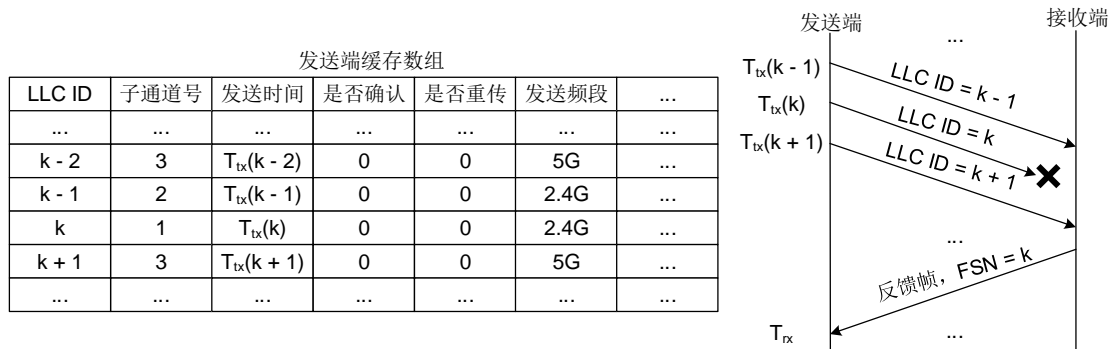


图 5.5 各个硬件队列下缓存数组的信息更新

由于发送端的缓存数组中已经记录了各个报文 LLC ID 与其子通道号、子通道序号之间的对应关系，因此接收端的反馈帧只需包含 LLC ID 信息，即可高效地同时反馈多个硬件队列下的报文接收确认情况。发送端收到反馈帧后，记录下反馈帧的接收时间，然后对反馈帧进行解析，即可得到各个频段、各个硬件队列下接收端最新接收报文的 LLC 序号。接着，发送端读取缓存数组，通过计算反馈帧与对应硬件队列下报文的发送时间的差值，即可得到各个硬件队列的报文往返时延，计算流程如图 5.6 所示。

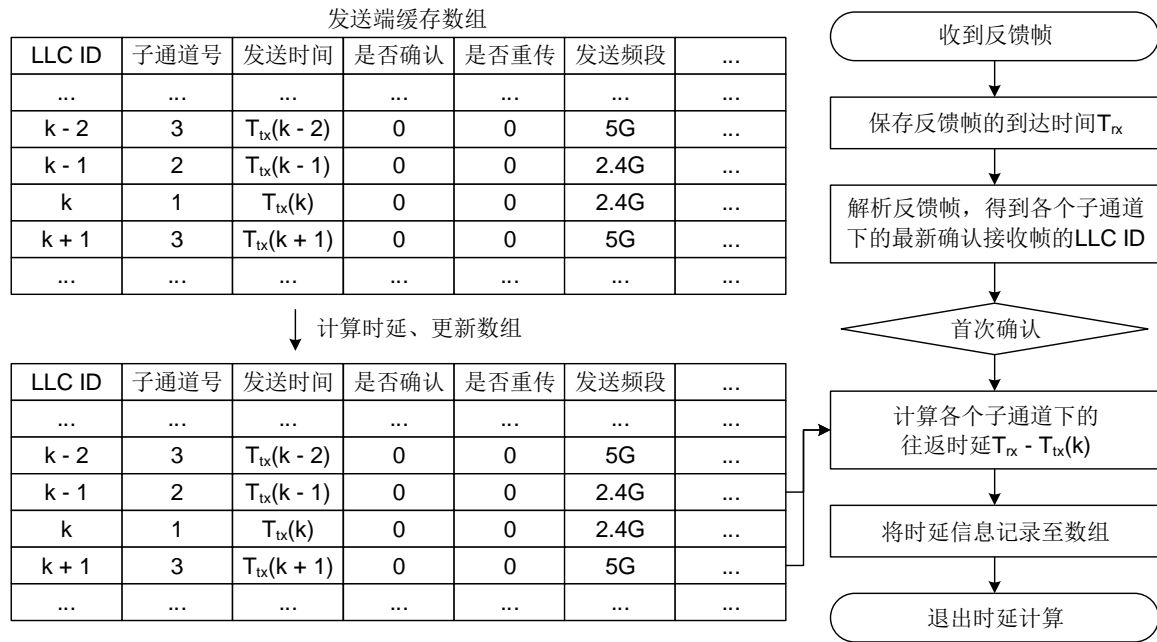


图 5.6 各个硬件队列下往返时延的计算流程

同样地，在冗余确认机制下，当同一个报文得到多次接收确认时，只在报文被首次确认时计算时间差，作为对应硬件队列下的时延信息。通过缓存数组中报文的封装信息与反馈帧信息的配合工作，共同地实现了多个硬件队列下传输时延的测量。

5.3.3 多硬件队列下的拥塞信息测量

类似于投屏业务下的拥塞信息测量，游戏业务下的拥塞信息测量也包括拥塞时延信息和拥塞长度信息组成，但是游戏业务的拥塞信息测量同时包括了两个频段下，以及每个频段下的四个硬件队列下的测量。

各个硬件队列的拥塞时延信息指示了当前时刻下报文若选择该硬件队列进行发送，该报文会在硬件队列中停留的最大时间。测量各个硬件队列对应的拥塞时延需要将滑动窗口的报文信息与缓存数组的信息进行联合使用，每当收到反馈帧并更新缓存数组后，即遍历发送端的滑动窗口获取所有已发送且未确认的报文，基于缓存数组得知每个报文所使用的频段和硬件队列，并针对每个频段和硬件队列，分别判断出对应的最早未被反馈确认的报文的 LLC ID 序号，然后通过查找缓存数组中对应报文的发送时间，计算与当前时间之差，即可得到对应频段对应硬件队列下的拥塞时延。具体流程示例如图 5.7 所示。



图 5.7 各个硬件队列下拥塞时延的计算流程

各个硬件队列的拥塞长度信息指示了当前时刻下该硬件队列中还有多少个报文还未传输成功，测量机制类似时延的测量，同样需要获取每个频段、每个硬件队列下最早未被反馈确认的报文的 LLC ID 序号，然后从该序号开始遍历缓存数组，计算出对应硬件队列下所有未被反馈确认的报文的个数，从而得到对应频段对应硬件队列下的拥塞长度信息。

5.4 双频多队列下的调度发送机制

5.4.1 设计需求分析

在 WiFi 传输链路上，游戏业务的流畅稳定运行对无线传输链路的时延、丢包率等指标提出了很高的要求，但是在当前 WiFi 机制下，针对对传输优先级具备较高要求的游戏业务报文，底层链路缺少合适的机制将其分配到更恰当的硬件队列，从而导致游戏业务报文在信道竞争中处于劣势，造成报文的阻塞与延迟。与此同时，针对当前支持 DBDC 的最新网卡设备，现有的无线网卡驱动缺少合适的调度发送机制，这使得硬件设备在实际使用过程中依旧使用时分技术进行报文的双频发送，因此链路带宽无法得到充分的利用，浪费了宝贵的硬件资源与信道资源。

针对上述原因，本文设计了一种全新的报文调度发送机制，用于业务报文在双频

多队列下进行灵活的调度发送。本机制基于双频下各个硬件队列的 KPI 测量结果，对不同类型的以太网帧、不同优先级的业务报文、反馈帧和重传帧等使用不同的调度发送策略，从而让游戏业务及其他各种优先级的业务能够在 2.4GHz 和 5GHz 两个频段，以及对应频段的各个硬件队列中得到高质量、高效率的报文传输。

5.4.2 双频多队列调度设计

考虑到不同类型的报文具有不同的传输优先级，因此在本文设计的双频多队列调度机制下，实际的调度发送机制根据待发送的报文类型而定，目前主要针对普通的待发送业务报文、携带接收信息的反馈帧和需要重传的业务报文进行不同的调度发送。

5.4.2.1 业务报文的发送

针对普通的待发送业务报文，硬件队列的选择主要依据待发送报文的业务类型，不同业务类型对应的优先级由用户态程序进行配置，底层则基于用户态程序下发的业务五元组信息与优先级信息映射关系表进行硬件队列的选择。发送频段的选择需要结合考虑两个频段上对应硬件队列的链路 KPI 信息。

由于链路 KPI 信息的测量基于反馈帧实现，而反馈帧的反馈间隔可能会低于报文的发送间隔，因此若是仅对链路质量进行择优发送，报文的发送过程中则会出现传输抖动现象，即在一个反馈帧的反馈周期内，所有报文全都选择同一个较优的频段进行发送，造成该频段传输质量变差；而新的反馈帧到来并更新链路质量信息后，所有报文又全部选择另一个频段进行发送，最终导致两个频段下的业务轮流出现突发的业务传输，降低了信道资源的利用率与报文的传输可靠性。

为了避免业务报文在传输过程中出现抖动，本设计了一种调度机制，通过使用双频下各个队列的链路质量信息，计算各个频段下的报文发送比例，按比例对报文进行调度发送。

设当前某个优先级的硬件队列在 2.4GHz 下的传输带宽为 $BW_{2.4G}$ ，报文往返时延为 $RTT_{2.4G}$ ，拥塞时延为 $CT_{2.4G}$ ，拥塞长度为 $CL_{2.4G}$ ；在 5GHz 下的传输带宽为 BW_{5G} ，报文往返时延为 RTT_{5G} ，拥塞时延为 CT_{5G} ，拥塞长度为 CL_{5G} ；鉴于不同频段的带宽

不同，链路负载的承载能力也不同，因此引入比例因子 τ ：

$$\tau_{2.4G} = \frac{BW_{2.4G}}{BW_{2.4G} + BW_{5G}} \quad (5-2)$$

针对不同链路质量信息，分别设置不同的权重因子为 μ_{RTT} 、 μ_{CT} 和 μ_{CL} 。由于目前只使用这三种质量信息作为频段选择依据，因此三个权重因子满足以下关系：

$$\mu_{RTT} + \mu_{CT} + \mu_{CL} = 1 \quad (5-3)$$

基于这几个参数，业务报文在 2.4GHz 频段和 5GHz 频段下的发送比例分别设置为：

$$P_{2.4G} = \tau_{2.4G} \cdot \left(\frac{\mu_{RTT} \cdot RTT_{5G}}{RTT_{2.4G} + RTT_{5G}} + \frac{\mu_{CT} \cdot CT_{5G}}{CT_{2.4G} + CT_{5G}} + \frac{\mu_{CL} \cdot CL_{2.4G}}{CL_{2.4G} + CL_{5G}} \right) \quad (5-4)$$

当业务报文到来时，为了让报文在两个频段上基于比例进行调度发送，针对该报文取一个随机数 $R \in [0, 1]$ ，然后判断随机数 R 所在的比例范围，若 $R \in [0, P_{2.4G}]$ ，则将该报文放入 2.4GHz 频段对应的硬件队列发送；若 $R \in (P_{2.4G}, 1]$ ，则选择 5GHz 频段的硬件队列发送，从而保证业务报文在两个频段上平稳有序地传输。

5.4.2.2 反馈帧与重传报文的发送

反馈帧由业务的接收方构造和发送，负载中包含了接收端的业务报文接收情况，业务发送端基于反馈帧进行丢包的检测和重传，以及对链路质量进行探测。因此，反馈帧能否及时、可靠地送至发送端，决定了业务报文的丢包与阻塞能否及时地重传至接收端，以及发送端能否及时更新链路当前的传输质量信息。因此，反馈帧具有极高的传输调度优先级。

重传报文则是由发送端负责发送，具体是在发送端解析反馈帧后，判断出丢包进行重传，或者发送端检测到报文超时未收到确认，因此进行超时重传动作。重传报文能否及时、可靠地送至接收端，直接决定了当前传输的业务能否实现低时延、高可靠的传输。因此，重传报文也具有极高的传输调度优先级。

针对反馈帧与重传报文，本机制通过选择最高优先级的硬件队列 AC_VO 进行报文发送，并均在两个频段上同时发送报文，从而实现报文的冗余传输，确保高优先级的报文能快速且可靠地送至对端。对应流程如图 5.8 所示

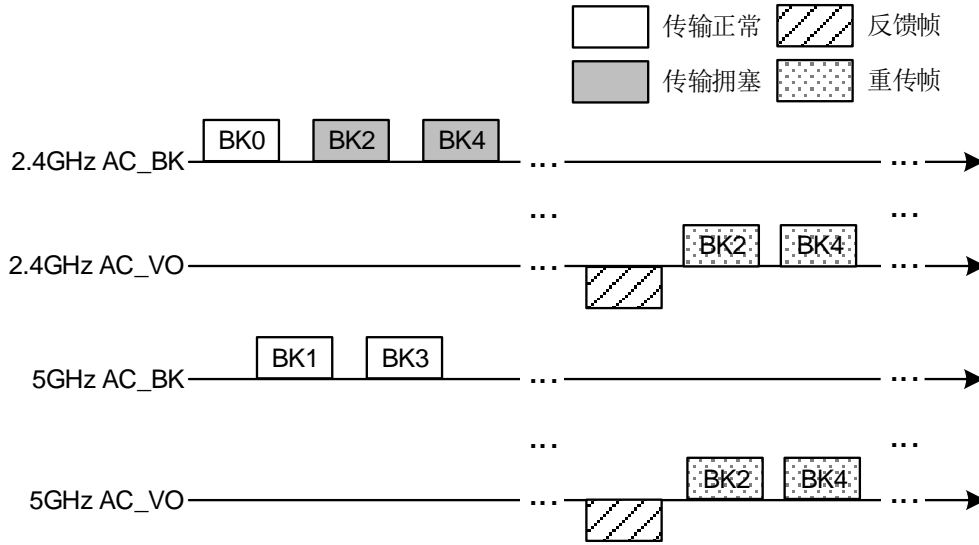


图 5.8 跨队列冗余反馈和重传

5.4.3 性能仿真测试

为了对比分析调度算法的性能，针对仅在 2.4GHz 频段上传输、仅在 5GHz 频段上传输和通过反馈重传与双频调度机制同时在 2.4GHz 频段和 5GHz 频段上传输三个场景，分别使用 matlab 进行传输时延的仿真。

在仿真使用的双频 WiFi 信道下，设置 2.4GHz 频段下的链路平均处理时延为 5 毫秒，平均丢包率为 1%，5GHz 频段下的链路平均处理时延为 1 毫秒，平均丢包率为 0.1%。根据式(3-14)可知，报文在链路上的平均传输时延和报文抵达率、链路平均处理时延相关，因此可以根据单位时间内对应频段上的报文数量和链路平均处理时延，实时地计算出报文在对应频段上的平均传输时延。报文的实际传输时延则以平均传输时延作为高斯分布的均值进行随机取值，丢包情况则基于平均丢包率进行随机判决。对于双频调度机制下的报文发送，通过反馈帧实时地测量两个频段下的 KPI 信息，并基于测量信息进行双频调度发送。在仿真测试中，设计每秒传输 1000 个报文，测试时间为 60 秒，并将每秒内所有报文的传输时延进行平均计算，得到的传输时延仿真结果如图 5.9 所示。

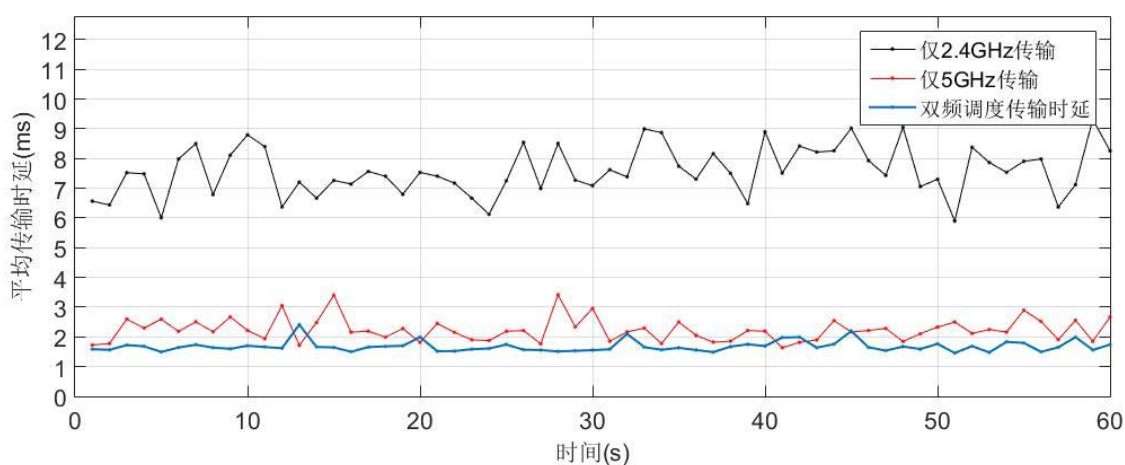


图 5.9 双频调度机制平均传输时延仿真测试

从时延的仿真测试结果中可以看出，经过双频下的调度传输，报文在传输过程中由于可以使用链路时延更小的频段进行发送，因此相较仅使用 2.4GHz 或 5GHz 频段的场景，报文的平均传输时延更小且更稳定。

与此同时，连续进行十次仿真测试，并分别统计每次测试的报文丢包率，得到的丢包率仿真结果如图 5.10 所示。

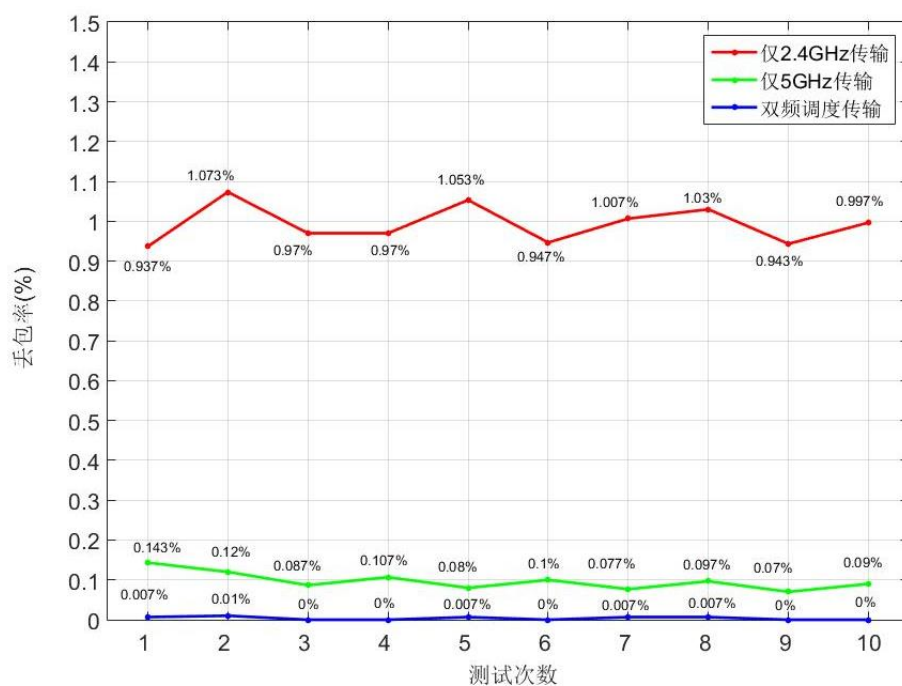


图 5.10 双频调度机制丢包率仿真测试

从丢包率的仿真测试结果中可以看出，在 WiFi 链路存在丢包的情况下，使用双

频调度进行反馈重传的报文传输流程，相比不使用该机制的流程，报文的丢包率显著降低。因此，通过在双频下进行确认反馈和冗余重传，能够极大地降低报文的丢包率，从而实现业务在 WiFi 链路上的高可靠传输。

5.5 动态流量调度机制

5.5.1 设计需求分析

虽然经过双频多队列下的调度机制，游戏业务得以使用质量更好的频段及更高优先级的硬件队列进行报文发送，但是在多业务并发场景下，当低优先级业务进行突发或长期的大带宽数据传输时，低优先级业务将长时间占据宝贵的空口信道资源，导致高优先级业务的发送出现迟滞与阻塞，从而造成空口的拥塞现象。为了保证游戏业务在空口上的传输质量，本文设计了动态流量调度机制，针对不同优先级的业务报文使用对应令牌桶策略进行流量控制，并在 AP 侧设计了基于链路 KPI 信息的 TCP 业务平滑降速机制，从而确保游戏业务在多用户场景下得到足够的发送机会。

5.5.2 令牌桶设计

令牌桶是一种用于网络流量整形和带宽限制的算法，该算法支持对数据流量的传输带宽进行限制，又支持一定的突发传输，且算法实现简单，因此本文选择使用该算法来保证游戏业务在多任务并发下的传输质量。

在令牌桶算法中，设置有一个容量为 S 字节的桶，并向桶中以每秒 n 字节的速率放置令牌，当桶中的令牌数量大于 S ，则丢弃多余的令牌；当有 m 字节的报文需要发送时，若桶中的令牌数量大于等于 m 字节，则取出 m 字节令牌，并将报文发送；若桶中的令牌数量小于 m 字节，则将待发送报文丢弃或缓存。在这个算法下，业务的传输带宽被限制在每秒 n 字节，同时又支持最大 S 字节的突发业务传输，算法的示意图如图 5.11 所示。

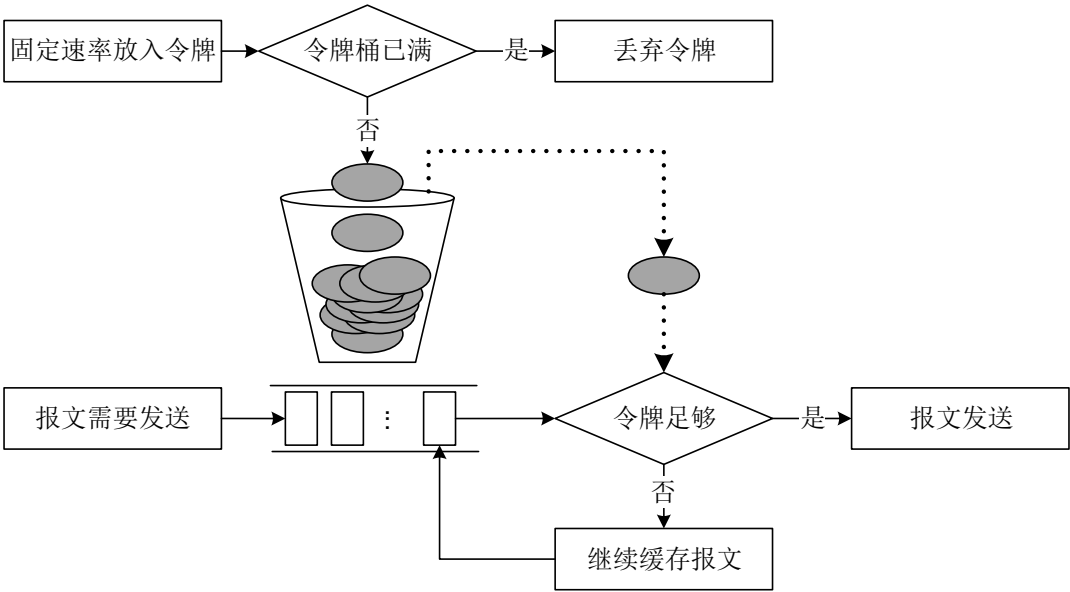


图 5.11 令牌桶算法

在本文设计的动态流量调度机制下，针对四种业务优先级和底层的四个硬件队列，进行了不同的令牌桶调度策略。其中针对游戏业务和最高优先级业务的传输，不使用令牌桶进行限速，而是选择直接送至双频多通道调度机制进行调度发送，从而保证这类报文的传输优先级；针对其他三种业务优先级，分别使用三个独立的令牌桶进行调度，从而避免较低优先级的业务抢占高优先级业务，造成空口拥塞。令牌桶的使用如图 5.12 所示。

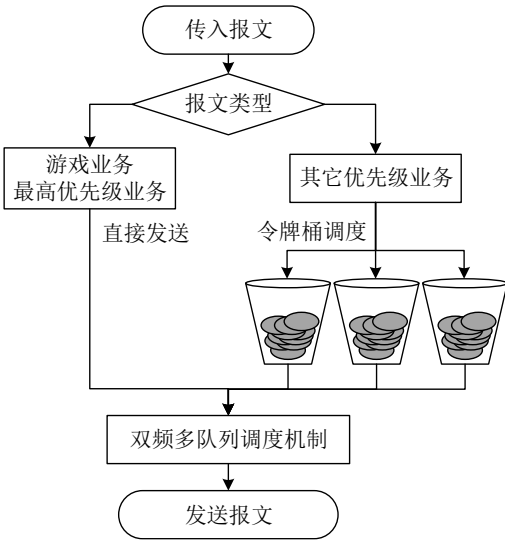


图 5.12 多优先级下的令牌桶调度

5.5.3 TCP 平滑降速设计

TCP 平滑降速算法的作用是在不发生丢包的前提下，使 TCP 业务的传输带宽平滑下降，从而使低优先级业务在保证正常稳定传输的前提下实现带宽限制和网络整形功能。TCP 业务的发送过程由滑动窗口控制，待发送报文只有在滑动窗口内才可以发送；而滑动窗口的大小受限于接收窗口和拥塞窗口，其中接收窗口由 TCP 业务的接收端告知，拥塞窗口则由发送端基于拥塞算法得到。拥塞算法基于 TCP 业务的 ACK 帧来判断业务的 RTT 时延信息和丢包情况，从而实时调整拥塞窗口大小，最终影响滑动窗口大小与流量传输带宽。据此，本机制通过对 TCP 业务的 ACK 帧进行延迟传输，来实现 TCP 业务带宽的平滑调整。

本机制在 AP 侧同时针对上行和下行的低优先级业务进行 TCP 平滑降速，不仅能够确保游戏业务在多业务场景下得到优先调度传输，还能使支持和不支持双频调度的终端设备均能正常连接 AP 并进行流量控制，具有终端透明的特点，最终确保游戏业务在多业务和多设备共存场景下均能够优先持有空口信道资源，保证游戏业务在 WiFi 信道上的传输质量。

为了实现低优先级 TCP 业务的平滑降速，本机制设计使用了环形数组对 TCP 业务的 ACK 帧进行延迟操作。环形数组的每个元素都是链表头节点，链表中保存了被延迟的帧；同时维护了一个定时器和一个出队指针，以定时器周期为最小精度进行报文延迟，对于需要延迟 k 个定时器周期的报文，以出队指针为起点，选择其后续第 k 个元素，将报文缓存进对应链表中。定时器周期性地将出队指针向后移动，并获取指针指向的数组元素，将链表中的帧发送出去。延迟流程如图 5.13 所示。

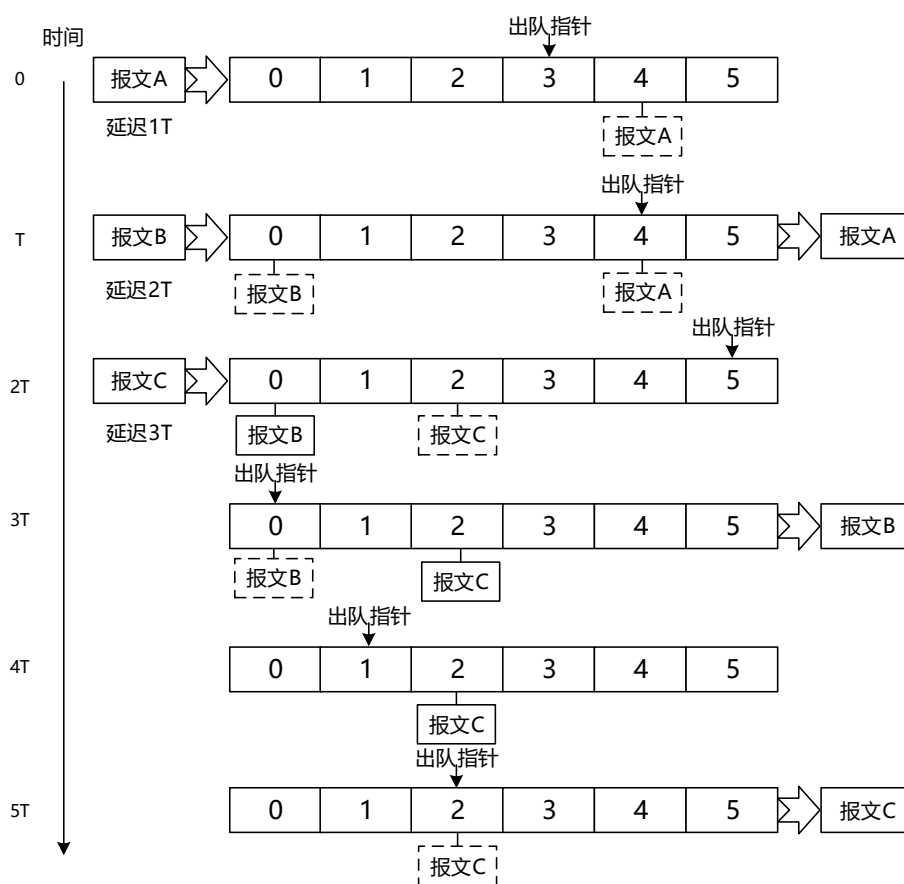


图 5.13 TCP 平滑降速下的延迟机制

在本机制下,通过无线链路的 KPI 信息实时监测游戏业务等最高优先级的业务的链路发送质量。在网络流量正常的情况下,不对业务进行平滑降速,各个优先级的报文均直接发送;当发现最高优先级的业务的往返时延、拥塞时延或拥塞长度超过阈值时,判定链路出现拥塞,此时启用环形队列进行平滑降速,环形队列的执行流程设置如下:

- (1) 针对三种较低优先级的业务设置三个入队指针 In_VI 、 In_BE 和 In_BK , TCP 业务的 ACK 帧选择对应优先级的指针进行缓存和延迟。
- (2) 初始时, In_BK 与出队指针的距离为 3 个单位;若网络持续拥塞,则每次定时器超时,以步长为 3 个单位增大 In_BK 与出队指针的距离。当 In_BK 与出队指针的距离达到 30 个单位时,停止增加距离,并开始使用次低优先级指针 In_BE 。

- (3) 初始时, In_BE 与出队指针的距离为 2 个单位; 若网络持续拥塞, 则每次定时器超时, 以步长为 2 个单位增大 In_BE 与出队指针的距离。当 In_BE 与出队指针的距离达到 30 个单位时, 停止增加距离, 并开始使用次高优先级指针 In_VI 。
- (4) 初始时, In_VI 与出队指针的距离为 1 个单位; 若网络持续拥塞, 则每次定时器超时, 以步长为 1 个单位增大 In_VI 与出队指针的距离。当 In_VI 与出队指针的距离达到 30 个单位时, 停止增加距离。
- (5) 当网络不再拥塞时, 出队指针照常移动, 并将对应报文出队发送; 入队指针 In_VI 、 In_BE 和 In_BK 的位置则固定下来不再移动。当环形队列中的所有 ACK 帧均被发送出去时, 恢复初始状态。

5.6 章节小结

本章的主要内容是游戏业务在 WiFi 链路上传输质量保证方案的构建和设计。首先分析了游戏业务在 WiFi 链路上传输时存在的问题, 提出了传输需求目标, 然后针对难点和目标, 提出了多条保证游戏传输质量的设计思想路线。为了游戏业务的可靠和优先传输, 本章设计了高效的报文封装机制, 在非常小的封装开销基础上实现了业务报文对硬件队列的选择发送功能和反馈重传功能; 然后基于封装和反馈流程设计了无线链路的 KPI 测量机制, 从而实时检测各个频段各个硬件队列下的报文传输 KPI 信息; 紧接着基于链路 KPI 信息设计了双频多队列下的调度发送机制, 保证游戏等各个业务在双频下的稳定、可靠、低时延传输, 并对调度性能进行了仿真测试; 最后针对游戏业务与多业务并存的场景, 设计了动态流量调度机制, 在终端透明的前提下避免了游戏业务的传输被低优先级业务干扰, 从而进一步保障了游戏业务在 WiFi 链路上的高质量传输。

第6章 游戏业务平台的实现与测试

上一章详细地描述了针对保证游戏业务传输质量所提出的各个思想与技术方案，本章在此基础上，分别在 Linux 平台和 Windows 平台上对上述技术方案进行了代码实现，并测试验证了部分功能模块的工作效果。

6.1 软件架构与设计实现

6.1.1 整体架构介绍

现有的主流游戏业务不仅在基于安卓系统的手机终端设备运行，在基于 Windows 操作系统的笔记本设备上也十分常见，为了让本文提出的设计方案能够覆盖足够多的使用场景，本文在基于 Linux 平台的路由器设备和 Windows 平台的笔记本设备上均进行了代码实现，设计实现的整体框图如图 6.1 所示。

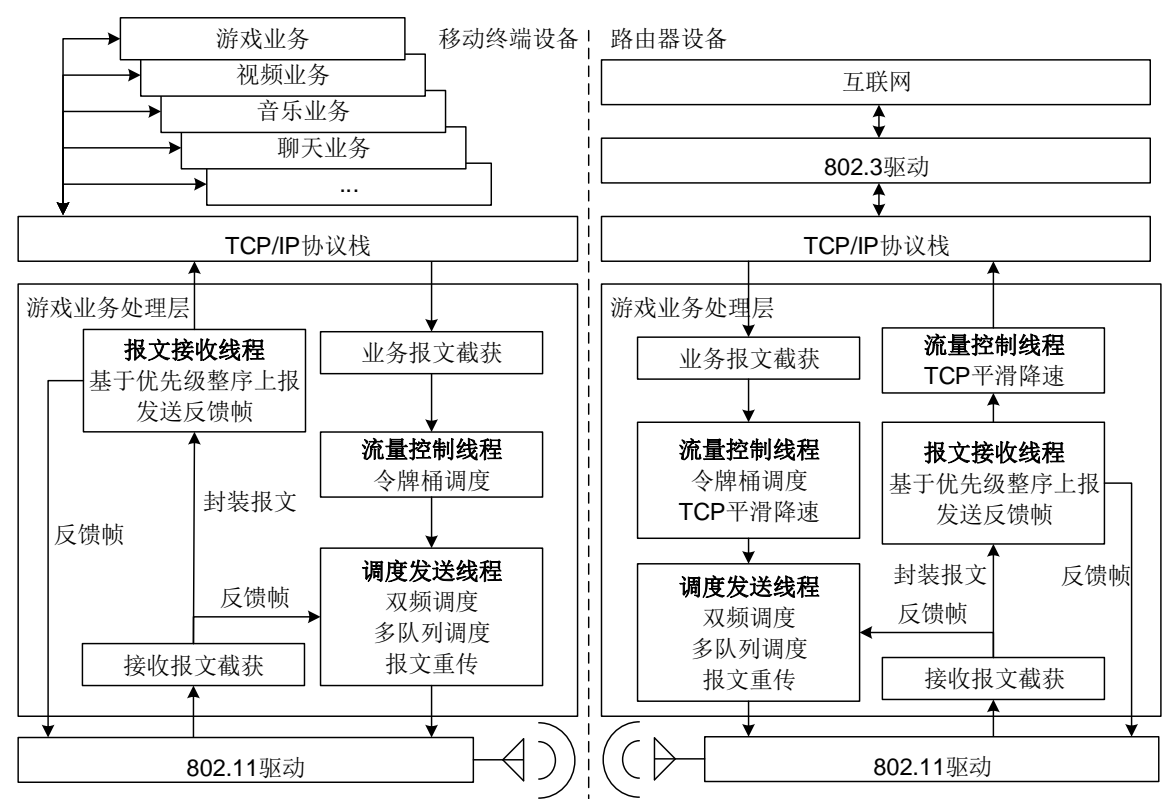


图 6.1 整体架构框图

本设计在设备的 TCP/IP 协议栈与无线网卡驱动之间插入了游戏业务处理层。在整体层次架构中, 包含了报文收发端的截获模块、流量控制线程、业务调度发送线程与业务报文接收线程, 其中路由器设备相较于移动终端设备, 具有更完善的流量控制模块, 从而满足针对多种移动终端设备的流量调度需求。

6.1.2 代码实现具体框图

在 Windows 平台下, 游戏业务处理层运行在 WFP 框架下的过滤驱动程序中。通过配置报文过滤的位置为 TCP/IP 协议栈中的 MAC 层, 并设置对应的处理函数, 从而对上行和下行的所有报文进行截获, 并创建了流量控制线程、游戏业务发送线程、游戏业务接收线程和定时器模块, 实现了上行和下行业务的共同调度传输。Windows 平台下的具体设计实现框图如图 6.2 所示。

在 Linux 平台下, 游戏业务处理层运行在无线网卡驱动的 UMAC 层中, 通过对驱动程序中的业务发送接口函数 `wal_bridge_vap_xmit()` 和业务接收接口函数 `oal_netif_rx_ni()` 进行修改, 截获上行和下行的所有报文, 并同样创建了流量控制线程、游戏业务发送线程、游戏业务接收线程和定时器模块。其中当 Linux 平台作为 AP 设备时, 同时设置了上行和下行的流量控制线程, 从而实现对支持和不支持双频传输的终端设备共同进行流量调度。Linux 平台下的具体设计实现框图如图 6.3 所示。

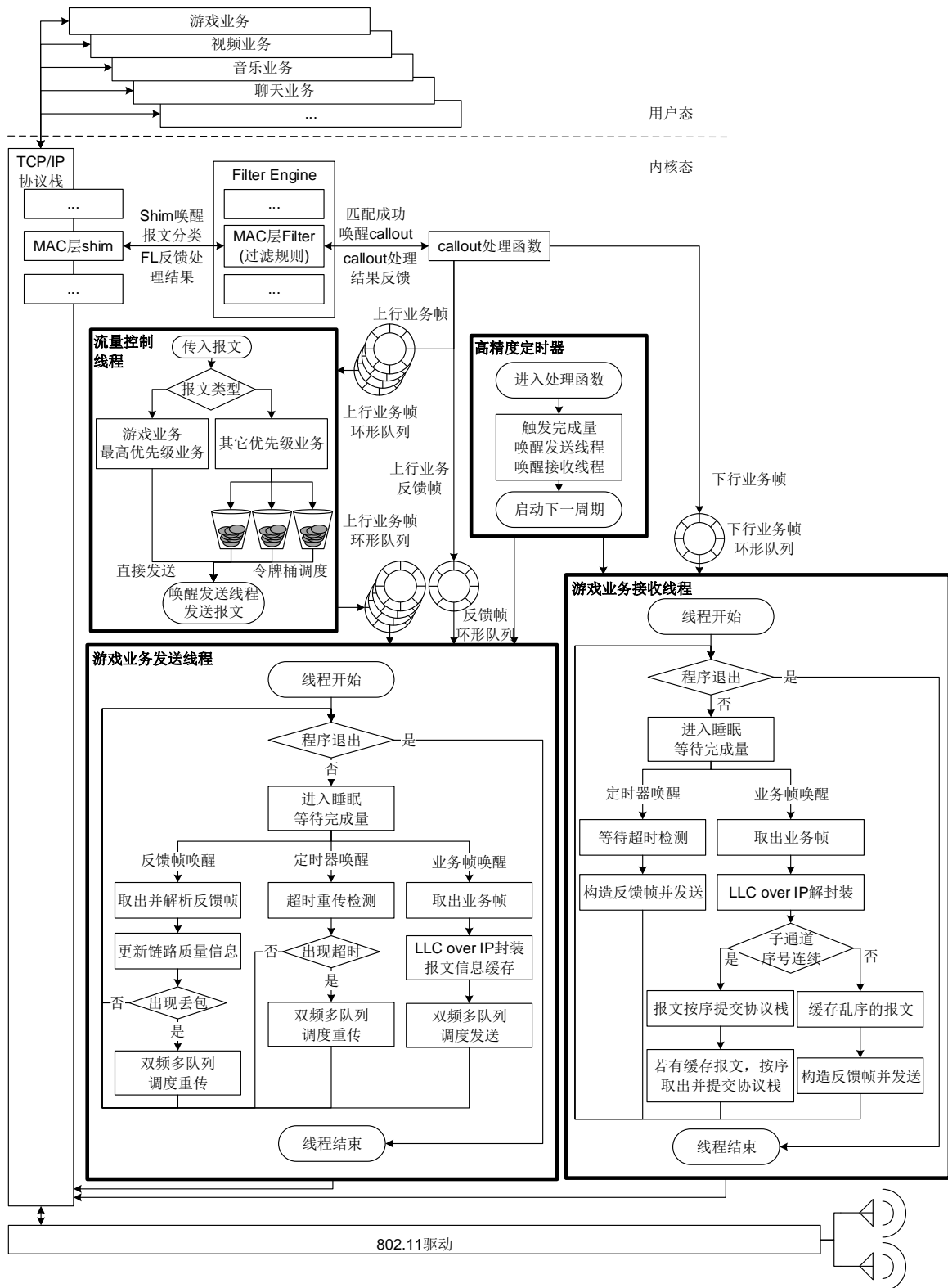


图 6.2 Windows 平台下的设计实现

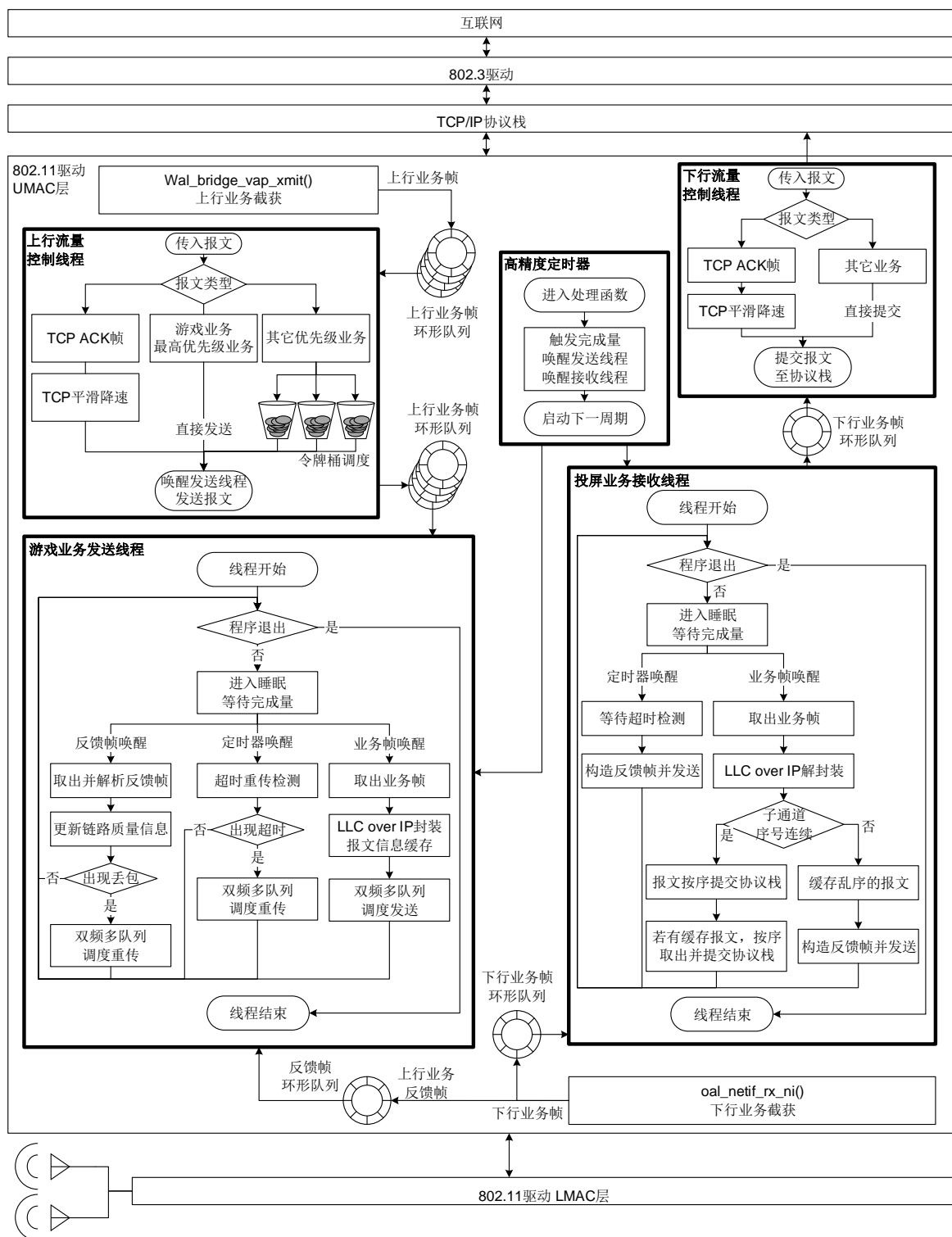


图 6.3 Linux 平台下的设计实现

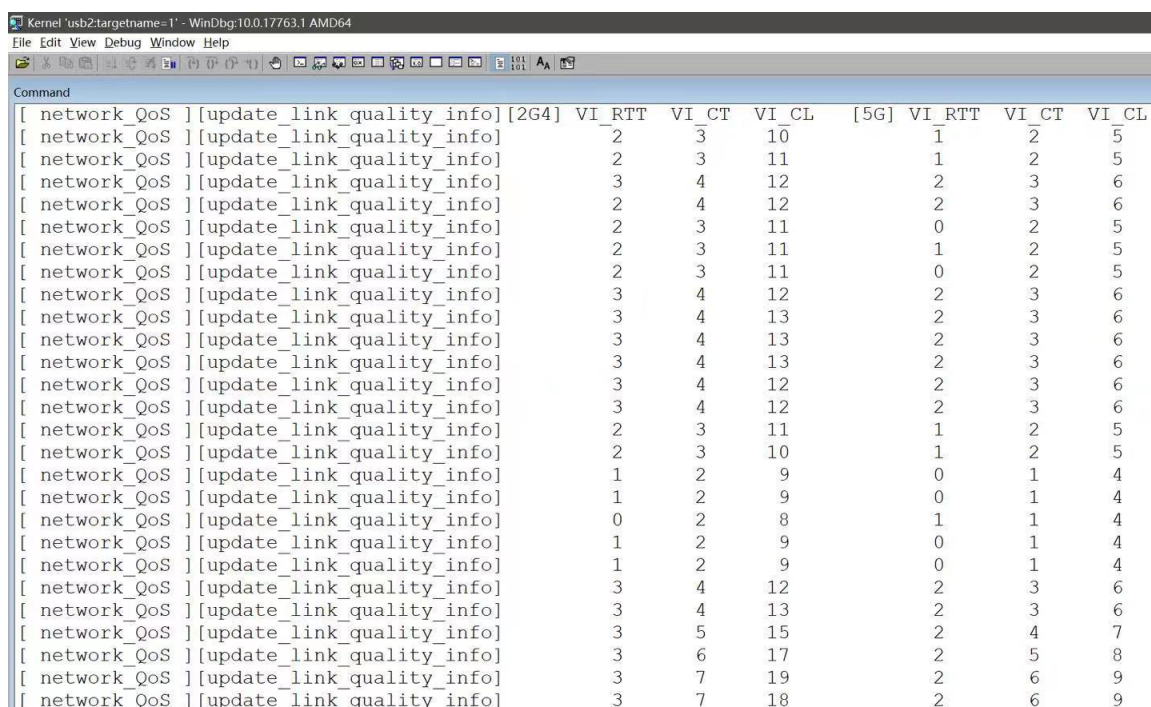
6.2 无线链路 KPI 测量功能验证

6.2.1 测试流程介绍

为了验证无线链路的 KPI 测量机制是否有效,本文在 Linux 平台的路由器设备和 Windows 平台的笔记本设备之间进行了数据流传输测试。测试使用 iperf 工具进行 UDP 打流,笔记本作为发送端,路由器作为接收端,设备上均进行了游戏业务传输质量保证设计的代码实现,从而模拟移动终端设备传输上行游戏业务数据的场景。

6.2.2 测试结果分析

在 UDP 流传输过程中,发送端会基于报文封装信息和反馈帧对各个硬件队列的报文往返时延、拥塞时延和拥塞长度进行计算,测试过程中驱动程序将计算结果通过 Windbg 调试工具实时打印至主机中。此处选择了 AC_VI 硬件队列在 2.4GHz 和 5GHz 下的 KPI 信息进行打印显示,实际打印结果如图 6.4 所示。



	[2G4]			[5G]		
	VI	RTT	CT	VI	RTT	CT
[network_QoS][update_link_quality_info]	2	3	10	1	2	5
[network_QoS][update_link_quality_info]	2	3	11	1	2	5
[network_QoS][update_link_quality_info]	3	4	12	2	3	6
[network_QoS][update_link_quality_info]	2	4	12	2	3	6
[network_QoS][update_link_quality_info]	2	3	11	0	2	5
[network_QoS][update_link_quality_info]	2	3	11	1	2	5
[network_QoS][update_link_quality_info]	2	3	11	0	2	5
[network_QoS][update_link_quality_info]	3	4	12	2	3	6
[network_QoS][update_link_quality_info]	3	4	13	2	3	6
[network_QoS][update_link_quality_info]	3	4	13	2	3	6
[network_QoS][update_link_quality_info]	3	4	13	2	3	6
[network_QoS][update_link_quality_info]	3	4	12	2	3	6
[network_QoS][update_link_quality_info]	3	4	12	2	3	6
[network_QoS][update_link_quality_info]	2	3	11	1	2	5
[network_QoS][update_link_quality_info]	2	3	10	1	2	5
[network_QoS][update_link_quality_info]	1	2	9	0	1	4
[network_QoS][update_link_quality_info]	1	2	9	0	1	4
[network_QoS][update_link_quality_info]	0	2	8	1	1	4
[network_QoS][update_link_quality_info]	1	2	9	0	1	4
[network_QoS][update_link_quality_info]	1	2	9	0	1	4
[network_QoS][update_link_quality_info]	3	4	12	2	3	6
[network_QoS][update_link_quality_info]	3	4	13	2	3	6
[network_QoS][update_link_quality_info]	3	5	15	2	4	7
[network_QoS][update_link_quality_info]	3	6	17	2	5	8
[network_QoS][update_link_quality_info]	3	7	19	2	6	9
[network_QoS][update_link_quality_info]	3	7	18	2	6	9

图 6.4 无线链路 KPI 信息测量结果

由打印结果可以得知,本文设计的无线链路 KPI 测量机制成功地对底层的传输链路进行了测量,得到了双频多队列下各个硬件队列的报文往返时延、拥塞时延和拥

塞长度的数据，这证明了报文的封装和反馈重传机制能够正确地运行，并为后续的双频多队列下业务传输调度和动态流量调度提供了可靠的数据支持。

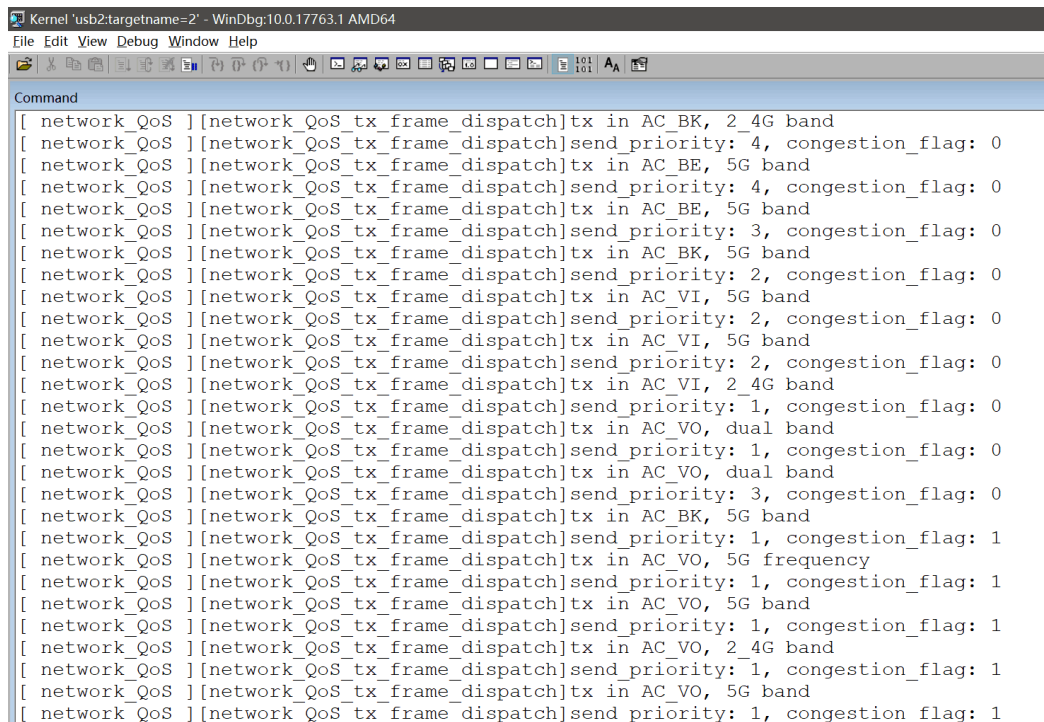
6.3 双频链路下的调度传输测试

6.3.1 测试流程介绍

本文基于无线链路上的 KPI 信息，对业务的传输进行了双频上的调度。为了测试游戏业务在双频调度下的性能，使用 Linux 平台的路由器设备和 Windows 平台的笔记本设备进行双频连接，并使用 iperf 工具模拟游戏数据流进行传输。与此同时，为了模拟复杂无线空口环境，使用额外的设备在 WiFi 链路上进行大带宽流量传输，参与空口信道的竞争。测试通过对比使用双频调度机制和不使用双频调度机制下，UDP 流的传输时延，来验证双频链路下调度传输的功能和性能。

6.3.2 测试结果分析

Windows 平台下的双频调度发送机制在运行过程中，会实时地记录业务的调度发送流程，并通过 Windbg 工具将日志信息打印出来，调度打印信息如图 6.5 所示。



```

[ network_QoS ][network_QoS_tx_frame_dispatch]tx in AC_BK, 2_4G band
[ network_QoS ][network_QoS_tx_frame_dispatch]send_priority: 4, congestion_flag: 0
[ network_QoS ][network_QoS_tx_frame_dispatch]tx in AC_BE, 5G band
[ network_QoS ][network_QoS_tx_frame_dispatch]send_priority: 4, congestion_flag: 0
[ network_QoS ][network_QoS_tx_frame_dispatch]tx in AC_BE, 5G band
[ network_QoS ][network_QoS_tx_frame_dispatch]send_priority: 3, congestion_flag: 0
[ network_QoS ][network_QoS_tx_frame_dispatch]tx in AC_BK, 5G band
[ network_QoS ][network_QoS_tx_frame_dispatch]send_priority: 2, congestion_flag: 0
[ network_QoS ][network_QoS_tx_frame_dispatch]tx in AC_VI, 5G band
[ network_QoS ][network_QoS_tx_frame_dispatch]send_priority: 2, congestion_flag: 0
[ network_QoS ][network_QoS_tx_frame_dispatch]tx in AC_VI, 5G band
[ network_QoS ][network_QoS_tx_frame_dispatch]send_priority: 2, congestion_flag: 0
[ network_QoS ][network_QoS_tx_frame_dispatch]tx in AC_VI, 2_4G band
[ network_QoS ][network_QoS_tx_frame_dispatch]send_priority: 1, congestion_flag: 0
[ network_QoS ][network_QoS_tx_frame_dispatch]tx in AC_VO, dual band
[ network_QoS ][network_QoS_tx_frame_dispatch]send_priority: 1, congestion_flag: 0
[ network_QoS ][network_QoS_tx_frame_dispatch]tx in AC_VO, dual band
[ network_QoS ][network_QoS_tx_frame_dispatch]send_priority: 3, congestion_flag: 0
[ network_QoS ][network_QoS_tx_frame_dispatch]tx in AC_BK, 5G band
[ network_QoS ][network_QoS_tx_frame_dispatch]send_priority: 1, congestion_flag: 1
[ network_QoS ][network_QoS_tx_frame_dispatch]tx in AC_VO, 5G frequency
[ network_QoS ][network_QoS_tx_frame_dispatch]send_priority: 1, congestion_flag: 1
[ network_QoS ][network_QoS_tx_frame_dispatch]tx in AC_VO, 5G band
[ network_QoS ][network_QoS_tx_frame_dispatch]send_priority: 1, congestion_flag: 1
[ network_QoS ][network_QoS_tx_frame_dispatch]tx in AC_VO, 2_4G band
[ network_QoS ][network_QoS_tx_frame_dispatch]send_priority: 1, congestion_flag: 1
[ network_QoS ][network_QoS_tx_frame_dispatch]tx in AC_VO, 5G band
[ network_QoS ][network_QoS_tx_frame_dispatch]send_priority: 1, congestion_flag: 1

```

图 6.5 双频调度传输过程中的调度信息

与此同时，通过空口抓包，也可以查看业务报文在链路上的调度传输情况，如图 6.6 所示。

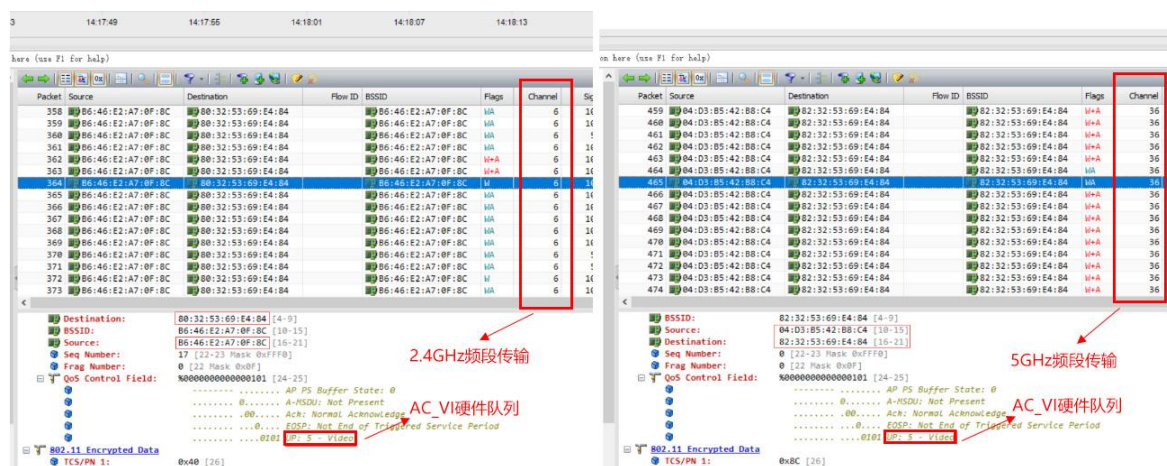


图 6.6 双频调度传输过程中空口抓包信息

从运行结果中可以看出，通过基于无线链路 KPI 信息的双频调度机制对业务的发送调度，iperf 工具发出的 UDP 流成功地在 2.4GHz 和 5GHz 两个频段上进行了调度传输，从而提高了业务的传输带宽。并且在链路出现 congestion 的场景下，调度机制成功地进行了业务的 congestion 和丢包检测，并实现了双频冗余重传功能，保障了业务的传输可靠性。

与此同时，测试还在多设备共存竞争信道的复杂干扰环境下，分别运行了不使用和使用双频调度机制的业务在 5GHz 频段下的传输场景，并将两个场景下的传输时延信息进行了记录和对比，将时延数据导出并绘成图表，测试结果如图 6.7 所示。

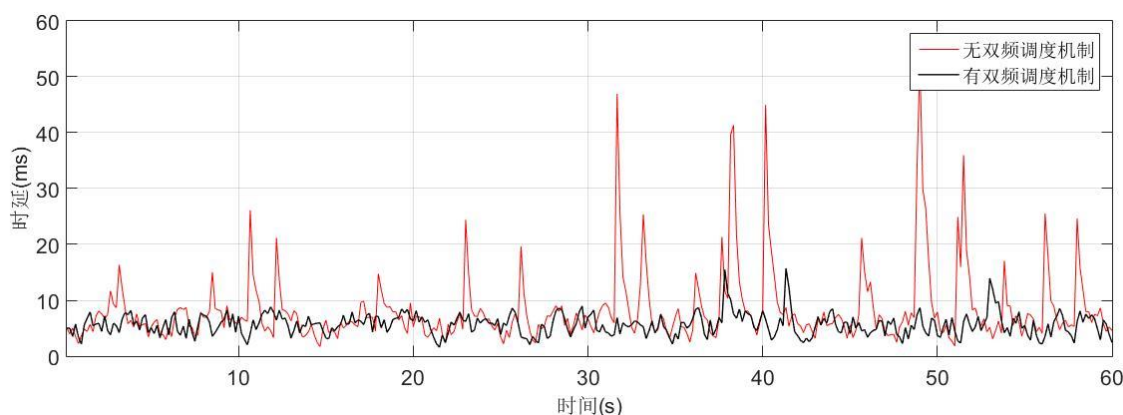


图 6.7 复杂干扰环境下的业务传输时延对比

从测试结果中可以看出，在基于高效报文封装的双频调度传输过程中，业务在复

杂无线环境下的传输具有更低和更稳定的传输时延。高效的报文封装机制为业务的传输带来了更高的可靠性，双频的联合调度使用则确保了业务能选择更优的频段进行发送，因此最终带来了业务在 WiFi 链路上的更高质量传输。

6.4 章节小结

本章针对游戏业务在 WiFi 链路上的传输质量保证的设计方案，在 Linux 平台的路由器设备上和 Windows 平台的笔记本设备上均进行了代码实现。接着测试了双频及多个硬件队列下 KPI 信息测量机制，成功地实现了多硬件队列质量信息的实时监测功能。然后验证了基于 KPI 信息的双频调度发送机制，并对比测试了双频调度发送机制所带来的传输性能提升，测试结果成功地证明本文所提出的设计方案能够给游戏业务在 WiFi 链路上的传输带来传输质量的保证。

第7章 总结与展望

7.1 全文总结

随着 WiFi 技术的不断发展, 流畅稳定和功能齐全的投屏技术正逐渐成为重要的用户需求, 而游戏业务则永远追求着更低的时延和更高的可靠性, 这对业务在 WiFi 链路上的传输提出了更高的要求。为了保证投屏业务和游戏业务在 WiFi 链路上能够高质量传输, 本文分别提出了多个设计方案, 并通过代码实现验证实际性能。

本文的主要工作包括:

(1) 针对 WiFi 链路上投屏业务的传输, 提出了一套完善的设计思路来保证传输质量, 包括使用反馈重传方案来降低投屏业务在无线链路上的丢包率, 使用调度传输方案来降低投屏业务的传输时延。

(2) 提出了基于空口质量感知进行视频帧编码的设计思路, 在 WiFi 链路层针对投屏报文设计了编码单元传输情况检测机制, 并实时探测链路质量信息, 从而触发投屏编码层进行编码调整, 及时对出错的视频帧进行恢复, 并灵活地匹配链路质量, 实现高效传输。

(3) 提出了蓝牙信道与 WiFi 信道联合进行投屏业务传输的异构融合设计方案, 用于实现投屏业务的全双工传输, 提高投屏业务数据在 WiFi 链路上的传输效率。

(4) 在 Linux 平台和 Windows 平台分别上对投屏业务的质量保证设计方案进行了代码实现和部分功能的测试。

(5) 针对游戏业务在 WiFi 链路上的传输, 设计了高效的报文封装方案以保证业务传输的及时性与可靠性, 并基于封装信息和反馈信息设计了无线链路 KPI 测量机制, 实时监测各个硬件队列的发送质量。

(6) 基于链路的 KPI 信息, 设计了业务在双频多通道下的灵活调度机制和动态流量调度机制, 保证了游戏业务在多业务并发、多设备共存的场景下的空口传输质量。

(7) 在 Linux 平台和 Windows 平台上分别对游戏业务的质量保证设计方案进行

了代码实现和部分功能的测试。

7.2 课题展望

本文针对投屏业务和游戏业务在 WiFi 链路上传输质量保证的研究做了大量的工作，并获得了一定的效果，但由于实际业务场景复杂，研究与设计实现的时间有限，个人能力有所不足，很多设计方案的测试验证和性能分析还有待进一步完成。

（1）基于空口质量感知的视频帧编码机制有待投屏业务的应用层程序的进一步开发设计，从而进行完整功能的验证和测试。

（2）投屏业务在蓝牙信道与 WiFi 信道上全双工传输的异构融合设计方案还有待功能实现和性能验证。

（3）在多业务并发、多设备并发的复杂 WiFi 场景下，动态流量调度机制为游戏业务带来的空口传输质量提升水平有待进一步测试。

致 谢

不知不觉，研究生生涯已经快要结束。在吴老师的实验室团队中，我接触到了许许多多的科研项目，全面地学习了网络通信的理论知识，收获了丰富的项目开发经验，我的毕设开题也源于这些项目。如今我的毕业论文即将完成，学习生涯也即将结束，此时此刻，我怀着诚挚的情感，向一路上对我帮助和支持的人致以我最衷心的感谢。

首先，我要感谢我的导师吴伟民老师。吴老师在我的整个研究生生涯给予了极大的帮助，在他孜孜不倦的教导和帮助下，我得以逐渐将学习到的理论知识不断应用于实践，得以接触到更加广阔的科研世界。我还要感谢刘应状老师，他在我的学习和生活中提出了许多宝贵的意见，使我能够不断地进步，顺利完成论文工作。

然后，我要感谢实验室团队的刘寺杰、周谋成、梁芷馨、郁帅鑫、郭康、刘一雄、黎育才、刘振宇、王禹磊同学以及师弟师妹们，他们不仅在学习和生活上陪伴和鼓励着我，还在我的毕设项目上提供了大量的支持，在大家的共同努力下推进着项目的设计与开发。

我还要感谢我的父母和家人，你们是我坚强的后盾，也是驱动我不断向前的不竭动力。感谢徐晓俐在论文期间给予我的陪伴，让我顺利地度过最后的挑战。

最后，衷心感谢所有在生活中帮助和支持着我的人！

参考文献

- [1] 丛静. 优化 WIFI 网络的问题研究[A]. 中国武汉决策信息研究开发中心、决策与信息杂志社、清华大学经济管理学院.“如何建立科学决策机制理论研讨会——决策论坛”论文集(下)[C].中国武汉决策信息研究开发中心、决策与信息杂志社、清华大学经济管理学院:《科技与企业》编辑部,2015:1.
- [2] Ericsson, “ Ericsson mobility report: On the pulse of the networked society,” Ericsson, Stockholm, Sweden, Tech. Rep. EAB-16:018498, 2016.
- [3] 代静. 下一代 Wi-Fi MAC 层关键技术研究[D].华中科技大学,2019.
- [4] IEEE 802.11a, IEEE Standard for Local and Metropolitan Area Networks SpecificRequirements-Part 11:Wireless Lan Medium Access Control(MAC) and Physical Layer(PHY)Specification:High-speed Physical Layer in the 5 GHZ Band[S],Dec.1999.
- [5] IEEE 802.11b, IEEE Standard for Local and Metropolitan Area Networks SpecificRequirements-Part 11:Wireless Lan Medium Access Control(MAC) and Physical Layer(PHY)Specification:High speed Physical Layer Extention in the 2.4 GHz Band[S],Sep.1999.
- [6] IEEE 802.11g, PartII:Wireless Lan Medium Access Control(MAC) and Physical Layer(PHY)Specification,Amendment 4:Further Higher Data Rate Extention in the 2.4GHzband[S],Jun.2003
- [7] IEEE 802.11n, Part 11: Wireless LAN medium access control (MAC) and physical layer(PHY) specifications: Enhancements for Higher Throughput[S]. 2009.
- [8] IEEE 802.11ac/D2.0, Draft Standard for Information Technology-Telecommunications andInformation Exchange Between Systems:Local and Metropolitan Area Networks SpecificRequirements,Part 11:Wireless LAN Medium Access Control(MAC) and PhysicalLayer(PHY) specification,Amendment 5:Enhancement for VeryHigh Throughput forOperation in Bands below 6 GHz[S].2011.

- [9] IEEE Draft Standard for Information Technology Telecommunications and InformationExchange Between Systems Local and Metropolitan Area Networks Specific RequirementsPart 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY)Specifications Amendment Enhancements for High Efficiency WLAN, in IEEE P802.11ax/D6.0[S], November 2019, pp.1-780.
- [10] DearMob; 5KPlayer Updated in Time for iPhone 12 AirPlay Mirroring[J]. Computer Technology Journal,2020.
- [11] 高宏.多屏互动的原理和设置步骤[J].广播电视信息,2019(09):49-53.
- [12] 贺鹏. 基于 Miracast 的多媒体共享系统设计与实现[D].西南交通大学,2016.
- [13] 王廉,付东海,谷飞.基于数字生活网络联盟的跨局域网共享媒体方案的设计[J].计算机应用,2014,34(S1):31-34.
- [14] 谭鹏飞. 安卓无线同屏技术研究及系统实现[D].中国科学院大学(中国科学院人工智能学院),2020.
- [15] 陈琛.中国移动 WiMo 开启多屏娱乐新纪元[J].通信世界,2010(47):19.
- [16] 李丕杉. 基于闪联协议的多媒体共享系统研究与实现[D].广东工业大学,2014.
- [17] 谭小兰,吴德建,罗敬.一种自适应的漏桶算法[J].湖南工程学院学报(自然科学版),2006(02):42-44.
- [18] 仲思惠. 基于令牌桶算法的流量控制服务的设计与实现[D].大连理工大学,2016.
- [19] 卢选民,赵健,史浩山.基于漏桶算法的自适应多媒体同步控制机制[J].陕西师范大学学报(自然科学版),2003(01):55-61.
- [20] M. Zuberek Wlodek,Strzeciwiłk Dariusz. Modeling Traffic Shaping and Traffic Policing in Packet-Switched Networks[J]. Journal of Computer Sciences and Applications,2018,6(2).
- [21] 李敏.一种新的自适应 RED 算法[J].中国科技信息,2018(08):80-81.
- [22] 铁玲,诸鸿文,蒋铃鸽,周天翔.确认平滑:一种改善无线链路上的 TCP 性能的实现[J].小型微型计算机系统,2003(06):1002-1003.
- [23] 罗浪. 无线局域网 CSMA/CA 协议的研究与实现[D].成都理工大学,2016.
- [24] 赵青芝. 无线局域网 802.11e MAC 层 EDCA 机制研究[D].北京交通大学,2009.

- [25] Baher MAWLAWI, Jean-Baptiste DORÉ, Nikolai LEBEDEV, Jean-Marie GORCE. Analysis of Frequency Channel Division Strategy for CSMA/CA with RTS/CTS Mechanism[J]. International Journal on Smart Sensing and Intelligent Systems, 2020, 7(5).
- [26] 邓皓文, 江虹, 李维. 支持 QoS 的无线局域网自适应 MAC 协议算法[J]. 传感器与微系统, 2021, 40(02): 146-149.
- [27] Wei Feng. Performance analysis of IEEE802.11e EDCA wireless networks under finite load[J]. Wireless Networks, 2020(prepublish).
- [28] 林尚娟. 无线局域网中高效 MAC 层接入机制研究[D]. 北京邮电大学, 2019.
- [29] 邓皓文. 基于 EDCA 机制的无线局域网 QoS 性能研究[D]. 西南科技大学, 2020.
- [30] 张化明. WIFI 传输与接入技术的发展研究[J]. 通讯世界, 2018, 25(12): 38-39.
- [31] Mehmet Ali Erturk, Luca Vollero, Muhammed Ali Aydin. Optimal joint load balancing and EDCA configuration of IEEE 802.11 wireless hotspots[J]. International Journal of Communication Systems, 2018, 31(2).
- [32] Ahmed Ismail Abu-Khadrah, Zahriladha Zakaria, Mohdazlishah Othman, Mohd Shahril Izuan Mohd Zin. Enhance the Performance of EDCA Protocol by Adapting Contention Window[J]. Wireless Personal Communications, 2017, 96(2).
- [33] 张方毅. 一种基于 802.11n 无线局域网标准的 A-MPDU 软件实现算法[D]. 哈尔滨工程大学, 2009.
- [34] 毕敬腾, 李鸥, 杨延平, 林洪文. 一种改进的 WLAN 中基于 MPDU 最小帧间间距的帧聚合机制[J]. 计算机应用研究, 2017, 34(07): 2128-2133.
- [35] 秦梦. 802.11ax 样机 AP 软 MAC 的设计与实现[D]. 华中科技大学, 2017.
- [36] 张驰. 802.11ax 样机 STA 侧软 MAC 设计与实现[D]. 华中科技大学, 2017.
- [37] 王伟. 基于 P2P 的 VPN 网关设计和实现[D]. 电子科技大学, 2015.
- [38] Lai C F, Huang Y M, Chao H C. DLNA-based multimedia sharing system for OSGI framework with extension to P2P network[J]. Systems Journal, IEEE, 2010, 4(2): 262-270.
- [39] Wi-Fi Alliance. Marketing Requirements Document for Testing and

Certification of Wi-Fi Display Devices [M]. 2008.

- [40] 刘晓天. 基于 TCP 协议的网络应用设计与开发[J]. 山东农业工程学院学报, 2017, 34(08): 157-158.
- [41] 朱新星. 网络传输过程中 UDP 协议实现[J]. 数字技术与应用, 2016(05): 23.
- [42] 易琰. 基于 H.264 标准的视频传输系统的设计[D]. 广东工业大学, 2018.
- [43] 张明明. 基于 H.264 帧内预测的静态图像压缩[D]. 天津工业大学, 2016.
- [44] 曹蕊. 基于 H.264 压缩编码标准的视频加密技术[D]. 南京航空航天大学, 2012.
- [45] 孟浩. 基于 H.264 编码标准的视频加密技术研究[D]. 中北大学, 2020.
- [46] 李若枫. 基于 Windows 驱动过滤的网络入侵检测技术研究[D]. 中国矿业大学, 2019.
- [47] 徐峰. 基于 WINDOWS X64 系统环境内安全监控软件的设计与实现[D]. 北京邮电大学, 2019.
- [48] 沈华. 基于 Linux 的 802.11 协议与驱动研究[J]. 微型电脑应用, 2013, 29(04): 32-34+38.
- [49] Quan Le-Trung. An open-source framework for integration of wireless ad-hoc drivers into Linux kernel[P]. Information and Communication Technology, 2016.
- [50] 骆德腾. 基于长短滑动窗口的自适应增强 TCP 网络编码技术研究[D]. 西南交通大学, 2019.
- [51] 倪洁, 徐志伟, 李鸿志. PPTP VPN 与 L2TP/IPSec VPN 的实现与安全测试[J]. 电子技术与软件工程, 2019(12): 192.
- [52] 陈洪恩. 基于 L2TP 的 VPN 设计与实现[D]. 湖北工业大学, 2018.
- [53] 张明龙. 基于 IPv6 的 CAPWAP 协议系统设计与实现[D]. 北京邮电大学, 2014.