

分类号_____

学校代码 10487

学号_____

密级_____

华中科技大学

硕士学位论文

(学术型☐ 专业型☐)

基于 Wi-Fi 的智能家居系统的高可靠 数据传输研究

学位申请人：郁帅鑫

学 科 专 业：信息与通信工程

指 导 教 师：吴伟民 教授

答 辩 日 期：2022 年 5 月 XX 日

**A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Master Degree in Engineering/Science （工学/理学硕士）
/the Professional Master Degree (专业学位)**

**英文题目， Times New Roman， 小二号，
实词的首字母大写**

**Candidate : xxx（中文习惯，姓在前且姓全部大写）
Major : ***
Supervisor : Prof. xxx**

**Huazhong University of Science and 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： 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。

保 密 ☐，在 _____ 年解密后适用本授权书。

本论文属于 不保密 ☐。

（请在以上方框内打“√”）

学位论文作者签名：

指导教师签名：

日期： 年 月 日

日期： 年 月 日

摘要（需更新）

近年来，随着网络技术的发展，智能家居（Smart Home）技术逐渐进入家家户户。与传统 Wi-Fi 家庭业务场景不同，智能家居业务是典型的密集用户接入场景，Wi-Fi 链路上的信道竞争十分激烈，对 Wi-Fi 的 MAC 层无线信道接入带来了极大的压力，使得智能家居终端出现信道接入时冲突概率极高，系统吞吐量低下等问题，给用户带来了很糟糕的体验，阻碍了智能家居产业的进一步发展。

本文主要针对单个接入点，终端设备数量较大，网络环境复杂的智能家居业务场景。首先分析了 Wi-Fi 网络在终端密集场景下出现信道接入困难，系统吞吐量低下的原因，指出现有信道接入机制所存在的不足之处以及隐藏终端问题对系统性能的影响，在此基础上，针对两个因素分别提出了对应改进优化策略：

1、基于 802.11 协议，对传统 DCF 机制进行改进。根据终端的预测流量大小将其均衡分配为多个分组，只有同组内的终端同时竞争信道资源，限制了直接参与信道竞争的终端的规模，使得信道竞争有序化，降低了终端传输数据时的碰撞概率。基于 Bianchi 模型提出改进 Markov 模型，用其对 DCF 改进机制进行了理论性能分析，结果证实了该方案对降低冲突概率，提升系统吞吐量的有效性。在此基础上提出了两种 Linux 平台上的具体软件实现方案；

2、在上述“分而治之”的思想之上，进一步提出了基于地理分区的隐藏终端优化方案。将接入点的覆盖区域划分为若干扇形，每个扇形区域内的终端分配为一组，不同组内的成员禁止同时竞争信道资源，打破了隐藏终端问题发生的条件。NS-3 仿真分析结果表明，该优化方案在减少隐藏终端问题上有明显的优势，并且适用于密集用户场景。

关键词： 802.11 协议；智能家居；信道接入；Markov 模型；隐藏终端

Abstract (需更新)

In recent years, with the development of network technology, Smart Home technology has gradually entered every household. Different from the traditional Wi-Fi home service scenario, the smart home service is a typical dense user access scenario. The channel competition on the Wi-Fi link is very fierce, which brings great influence to the wireless channel access of the MAC layer of Wi-Fi. The pressure of the smart home terminal makes the channel access probability extremely high, and the system throughput is low, which brings a bad experience to users and hinders the further development of the smart home industry.

This article is mainly aimed at smart home business scenarios with a single access point, a large number of terminal devices, and a complex network environment. Firstly, the reasons for the difficulty of channel access and low system throughput in Wi-Fi networks in the terminal-intensive scenario are analyzed, and the shortcomings of the existing channel access mechanism and the impact of hidden terminal problems on system performance are pointed out. On this basis, corresponding improvement and optimization strategies are proposed for two factors:

1. Based on the 802.11 protocol, the traditional DCF mechanism is improved. According to the predicted traffic size of the terminal, it is evenly distributed into multiple groups. Only the terminals in the same group compete for channel resources at the same time, which limits the scale of the terminals directly participating in the channel competition, makes the channel competition orderly, and reduces the time when the terminal transmits data. collision probability. Based on the Bianchi model, an improved Markov model is proposed, and the theoretical performance of the DCF improvement mechanism is analyzed with it. The results confirm the effectiveness of this scheme in reducing the collision probability and improving the system throughput. On this basis, two specific software implementation schemes on Linux platform are proposed;

2. On the basis of the above "divide and conquer" idea, a hidden terminal optimization scheme based on geographic division is further proposed. The coverage area of the access point is divided into several sectors, and the terminals in each sector are assigned to a group.

Members in different groups are prohibited from competing for channel resources at the same time, which breaks the conditions for the terminal hiding problem. The NS-3 simulation analysis results show that the optimization scheme has obvious advantages in reducing the hidden terminal problem and is suitable for dense user scenarios.

Key words: 802.11, Smart Home, Channel access, Markov model, Hidden terminals

目 录

摘 要（需更新）	I
Abstract（需更新）	II
目 录.....	IV
1 绪论.....	1
1.1 研究背景与意义	1
1.2 国内外研究概况	3
1.3 论文的研究内容	9
1.4 论文组织与结构（需更新，逻辑图？）	11
2 研究背景技术概术	13
2.1 Wi-Fi 下的传输机制.....	13
2.2 排队论相关概念	18
2.3 802.11 DCF 性能分析模型	20
2.4 Linux 网卡驱动开发技术	25
2.5 NS-3 仿真平台介绍.....	26
2.6 本章小结（需更新）	30
3 基于改进 DCF 机制的智能家居业务传输优化研究.....	32
3.1 引言	32
3.2 现有无线信道接入机制分析	32
3.3 Wi-Fi 智能家居场景系统性能.....	38
3.4 流量均衡的受控并行随机接入机制系统模型.....	40
3.5 改进 Markov 模型	44
3.6 受控并行随机接入机制性能理论分析	49
3.7 受控并行随机接入机制的仿真与分析（需更新）	57

华中科技大学硕士学位论文

3.8	本章小结（需更新）	58
4	受控并行随机接入机制的设计与实现	60
4.1	引言	60
4.2	集中式受控并行随机接入机制的方案设计	60
4.3	集中式受控并行随机接入机制的软件实现	73
4.4	分布式受控并行随机接入机制的方案设计	76
4.5	分布式受控并行随机接入机制的软件实现	79
4.6	两种实现方案的实验结果与分析（需更新）	81
4.7	本章小结（需更新）	86
5	基于地理分区的智能家居隐藏终端优化研究	87
5.1	引言	87
5.2	隐藏终端问题概述	87
5.3	现有隐藏终端问题研究分析	88
5.4	基于地理分区的隐藏终端优化方案设计	90
5.5	基于地理分区隐藏终端优化方案的仿真与分析（需更新）	92
5.6	本章小结（需更新）	94
6	总结与展望	96
6.1	本文主要内容及结论（需更新）	96
6.2	本文的主要创新点（未完成）	97
6.3	课题展望（需更新）	97
	致 谢	99
	参考文献（未完成）	103

1 绪论

1.1 研究背景与意义

新技术的创新、扩展和传播在很大程度上推动了人类的进步。在现代社会中，新技术在提高社会福利和定义人类与环境互动的新方式等方面，一直发挥着核心推动作用。在上个世纪，人类达到了前所未有的舒适和幸福水平。当然，这种进步得益于技术在人类生活的方方面面的大规模引入。

然而，资源开采是不可持续的，如果继续以目前的水平消耗能源，自上世纪以来习惯的繁荣时期可能会暂停甚至结束。此外，社会正面临着严重挑战，已经影响了经济和社会发展模式^[1]。因此，人类必须想办法合理利用现有资源，同时，对现有习惯的纠正也是不可避免的。另外，随着医疗水平和生活条件的提升，人类的平均寿命明显提高^[2]。在中长期范围内，出生率下降，社会老龄化愈发严重，这将给医疗保健系统带来巨大负担^[3,4]，各地政府未来可能将无法负担这些费用。因此，老年医疗保健已成为重大的健康和伦理问题，也是政府政策的重要组成部分^[5]。

一方面，不可持续的能源消费模式，另一方面，老龄化加剧。克服这两大挑战需要政府和人民共同面对，一个新兴想法是将一部分责任转移给普通公民，使大家能够在资源的合理化使用中发挥积极作用，贡献出自己的一份力量。

智能家居，作为一种新的范式可以让人们有意识地管理家庭能源资源，并改善他们的行为以减少能源消耗^[6,7]。半导体技术的快速发展和小型化使计算和电子设备激增；微控制器单元 (MCU) 处理能力的指数增长；在非常小的传感器中使用复杂处理技术测量和存储数据来集成先进的信号处理能力；无线技术的快速发展和进步，尤其是短距离和低功率应用。上述四个因素不断推动智能家居成为克服上述两大挑战的一个答案。

智能家居概念基于服务和功能之间的交互。这个想法源于多个领域的融合：娱乐、安全、能源管理和医疗保健。智能家居范式可以满足这些需求，与传统家居产品相比，智能家居通过数据交互及远程控制等功能形态将家居产品从被动静止型发展

华中科技大学硕士学位论文

至主动智能型，智能家居可以根据家庭居民的要求进行预测和响应，以提升家居环境的安全性、智能性、舒适性、节能性、便捷性，基于智能设备的精准控制可以矫正人们的生活习惯，为各种能源（比如水、电费）费用节约资金。[4]

伴随着云计算、边缘计算、人工智能技术、通信技术等技术的不不断突破，智能家居行业进入快速发展阶段 [2-3]。根据中国报告大厅发布的《2016-2021 年中国智能家居行业发展分析及投资潜力研究报告》测算，我国智能家居潜在市场规模约为 5.8 万亿元，2018 年市场总规模达到了 225 万亿元。预计我国智能家居市场未来 3~5 年的整体增速约为 13% [1.3]。

近年来政府制定了一系列政策文件支持智能家居行业的发展，详细见表 1-1。2017 年 12 月，工信部印发的《促进新一代人工智能产业发展三年行动计划(2018-2020)》提出，要优先推动智能家居等产品的突破。2020 年 8 月，国家标准化管理委员会正式将智能家居标准体系纳入《国家新一代人工智能标准体系建设指南》，提出在 2023 年全面完成智能家居标准体系的构建工作 [1.4]。

表 1-1 全国智能家居相关支持性政策文件

发布时间	发布单位	发布文件名
2016 年 12 月	国务院办公厅	《国务院关于印发“十三五”国家战略性新兴产业发展规划的通知》
2017 年 07 月	国务院办公厅	《国务院关于印发新一代人工智能发展规划的通知》
2018 年 07 月	国务院工业和信息化部 发展改革委	《扩大和升级信息消费三年行动计划（2018-2020 年）》
2019 年 09 月	国务院办公厅	《国务院办公厅关于加快发展流通促进商业消费的意见》
2020 年 08 月	国家标准化管理委员会	《国家新一代人工智能标准体系建设指南》

从目前智能家居具体产品来看，随着技术多元应用场景渗透以及中国国民消费结构的转型升级，智能家居单品品类得以不断创新与扩张，可划分为智能安防、智能照明、智能中控、智能家电、智能门禁等。它们之间的交互和远程控制离不开互联网，而家庭局域网作为接入互联网的最后一公里，其性能好坏直接影响人们的上网体验。随着智能家居等可移动设备种类和数量的剧增，传统的需要预先铺设的有线网络已

经不能满足人们生活的需求，无线局域网（WLAN）以其部署灵活、成本低等优点逐渐成为家庭局域网的优选。通过 Wi-Fi 接入互联网的智能家居设备也越来越多，而在智能家居设备中添加 Wi-Fi 接入互联网的功能也促进了智能家居的发展[x]。

随着智能家居的发展，普通家庭可能超过 30 个甚至 50 个智能设备，未来甚至会更多，而一般只有一个无线路由器，属于典型的单 AP，终端设备的高密场景。在高密场景中，现有 Wi-Fi 标准的性能表现不容乐观。在家庭 Wi-Fi 网络中，大量智能家居设备的引入，不仅使得原本手机、平板以及笔记本等传统移动设备上网体验变得很糟糕，而且智能家居设备本身的时延很大，网络带宽受限，甚至出现掉线问题，使用体验也很差。严重影响了智能家居的进一步发展。因此，对承载大量智能家居业务的传统家庭 Wi-Fi 的数据传输性能的优化，减少无线设备之间的干扰，提升智能家居用户体验，对于智能家居的进一步发展与普及有着充分的必要性。

1.2 国内外研究概况

1.2.1 智能家居的发展现状

在计算机网络、自动化控制、物联网及人工智能等技术快速发展的大背景下，社会智能化已成为本世纪的发展趋势。在过去的几年里，智能家居技术也随之飞速发展起来。上世纪 80 年代初，智能家居概念在美国诞生，称之为 Smart Home。智能家居有几个同义词，例如，智能屋、家庭自动化等。

智能家居主要经历了四个阶段的发展：第一阶段只是实现了灯光、窗帘及少量的安防控制等，各设备是使用同轴线及两芯线连接而成；第二阶段引入了总线和 IP 技术组网，能够实现可视对讲及安防等业务；第三阶段是集中式的智能控制系统，通过独立的中控机，来完成安防、计量等方面的功能；第四阶段则基于智能家居技术，根据用户的实际需求定制个性化的功能，旨在使得用户可以随时随地获取家中的信息，并且可以远程控制。

早在几十年前，家庭自动化的想法就已经存在。然而，当时因为硬件和安装成本高昂，其发展进程缓慢。物联网 (IoT) 的迅速发展促进了智能家居技术与各种电子设备的融合。伴随电器行业在家用电器上引入了 Wi-Fi 技术，使得设备可以通过 Wi-

Fi 互联，从而使普通消费者能以可控的成本，使用自动化智能设备，为智能家居的发展增添动力。

智能家居系统可以提供的两个关键功能：简单控制；智能自动化[4]。这两个功能旨在为居民提供舒适、方便的电器控制以及房间或建筑物的安全。它还促进节能生活，以支持可持续和绿色环境。智能家居与智能手机的无缝集成，它允许消费者使用他们的智能手机从任何地方控制和管理电器。Zion Market Research [7] 报告称，2016 年全球智能家居市场规模为 241.0 亿美元，预计到 2022 年将达到 534.5 亿美元。智能家居市场增长是描述该领域整体经济和产业进步的有用指标。

随着云、无线网络及各智能终端的出现与发展，除了支持 Wi-Fi 的家居设备外，制造商还为智能电器提供附加功能。最近几年，这些设备已与基于云的服务集成，以提供增值服务、运营和管理[2]，逐渐萌生了“云+端”的智能家居系统模式，即采用互联网中的基于云计算技术的专用功能的服务平台，其模式构成如图 1-1 所示。

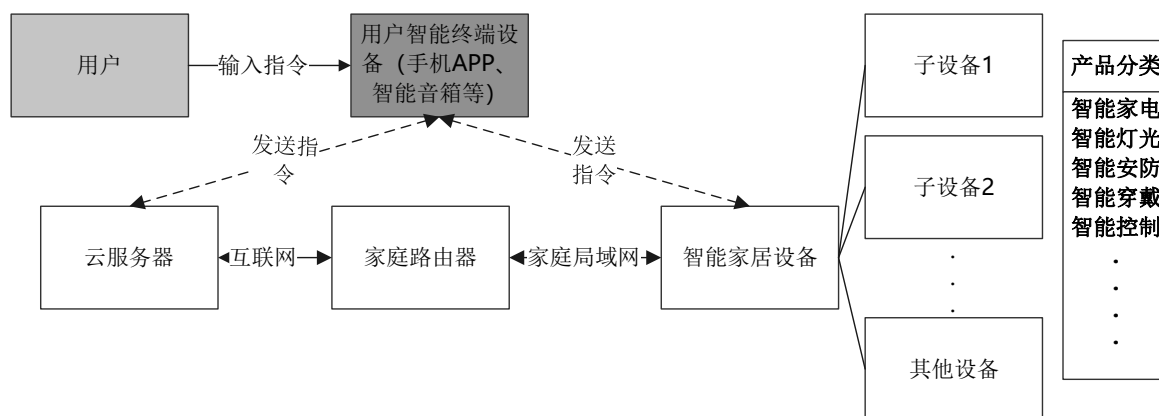


图 1-1 智能家居实现模型及分类

云服务器主要任务是实现大规模数据的存储、管理及移动终端(智能手机或 Pad 等)的可靠介入及并发访问。家居终端内集成了自动化采集和控制系统，还具备与云服务器交互的功能。用户在智能手机等移动终端上安装了对应的移动应用程序后，就可以随时随地实现对智能家居的远程监测、控制和管理。比如，LG 的 Cloud Center 向用户提供了一个应用程序，用户通过这个应用程序，可以不打开冰箱门，直接检查和查看冰箱的内容。这种方式避免了打开冰箱门时，其中冷空气的流失，从而不需要消耗更多的能量来保持低温。因此，可以节省能量。另外，基于谷歌云平台的飞利浦

照明的照明系统。用户可以云平台将家庭照明风格更改为各种模式,以获得比普通灯更舒适的效果。还可与 Nest 安全系统联动,安装在屋外的监控摄像头检测人体运动时,可打开家中的灯,就好像居民已经听到或知道了外面有人一样。

智能家居的核心要素是设备之间的互联互通。得益于 Wi-Fi 的普及程度,通过 Wi-Fi 把家庭内所有的智能设备连接起来确实为智能家居的发展与普及提了速。但是随着智能设备数量的攀升,通过 Wi-Fi 接入互联网的设备数量急剧上升,而设备之间相互干扰,带来视频卡顿、网络带宽受限,甚至出现 Wi-Fi 断连等问题。大量智能家居设备的引入,不仅使得原本手机、平板、笔记本等常用设备上网体验变差,而且智能家居设备本身的用户体验也很差。严重影响了智能家居的进一步发展。解决 Wi-Fi 在智能家居场景中应用的不足之处是一个亟待解决的问题。

1.2.2 Wi-Fi 标准的发展

无线局域网(WLAN)以其部署灵活、速度快、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]。随着移动互联网、物联网、智能家居等急速发展,无线网络业务内容不断丰富,业务量急剧上升,对 WLAN 的带宽和服务质量(Qos)的要求越来越高。

IEEE802.11 已成为目前应用最广泛的 WLAN 标准[3]。IEEE802.11 相关标准是由 IEEE802.11 委员会主导制定的,委员会成立于 1990 年。并于 1997 年,发布了 802.11 的第一个版本——802.11-1997 [7]。1999 年 Wi-Fi 联盟正式成立。随后又先后推出了 802.11b、802.11g、802.11n、802.11ac、802.11ax。2019 年 Wi-Fi 联盟在 802.11ax 发布的同时,将其协议版本重新命名为 Wi-Fi 6,并对其以往版本进行了重命名,具体见表 1-2,用简单的数字表示 Wi-Fi 版本,使普通用户更容易理解,更好的向消费者宣传。目前,随着 Wi-Fi 6 相关技术逐渐成熟,相关设备也逐渐进入人们的生活。

表 1-2 历代 Wi-Fi 标准命名及协议对应

协议旧命名	协议新命名	英文全称(简称)	中文
802.11n	Wi-Fi 4	High Throughput(HT)	高吞吐
802.11ac	Wi-Fi 5	Very High Throughput(VHT)	甚高吞吐
802.11ax	Wi-Fi 6	High Efficiency(HE)	高效能

在标准不断演进过程中，不断引入新的技术，有正交频分复用(OFDM)技术、块确认(CB) 和单用户(SU) MIMO (SU-MIMO)技术和 256 正交幅度调制(256-QAM)等等。自 Wi-Fi 4 以来，Wi-Fi 标准核心技术特征如表 1-3 所示：

表 1-3 Wi-Fi 技术特征发展

Wi-Fi 版本	Wi-Fi 4	Wi-Fi 5	Wi-Fi 6
频段	2.4G/5G	5G	2.4G/5G/6G
带宽/MHz	40	160	160
最大速率/Gbps	1.2	3.5	9.6
子载波间隔/kHz	312	312	78.125
空间流数目	4	8	8
调制方式	64QAM OFDM	256QAM OFDM	1024QAM OFDMA
多用户	SU-MIMO	DL MU-MIMO	DL/UL MU-MIMO

在 Wi-Fi 标准更新迭代历程中，前期主要重视系统吞吐量的提升，后面开始重视效能。实际上，Wi-Fi 4 发布后，就已经进入了 WLAN 的高速无线接入的时代，在物理速率上已经超出了其他无线网络，为 WLAN 的长远发展奠定了夯实的基础。但是随着万物互联时代的到来，需要接入互联网的设备在呈指数级增长。而作为部署使用方便的 Wi-Fi 终端的数量和种类也是日益丰富多样，给 Wi-Fi 标准的更新迭代增添了无尽动力。

随着市场的不断发展，业务业务日渐丰富多样，Wi-Fi 标准在更新迭代的同时，802.11 协议组也会不断出征，希望占据一些其他领域的网络业务，比如说 IoT（即 802.11ah），VR/AR 这种近距离的图像传输（802.11ad/ay）。这些新扩展的协议往往是基于 802.11 的核心框架做改进。

802.11ah 就是面向 IoT 场景的专项协议，又名 HaLow。同样是 Best-effort。其具有五个主要特点：大连接；远距离；高可靠；低功耗；安全性好。802.11ah 工作在 Sub-1GHz 频段的 ISM 信道上。虽然 802.11ah 在商业上不是很成功，但是其协议设计中有很多内容还是比较值得借鉴和参考的。

实际上, 802.11ah 相比于其他的一些协议, 从性能上而言, 确实比较均衡, 而且质量都还是不错。不过最后没有成功的原因, 主要在于实现该技术的成本和稳定性上, 具体设计过程中, 802.11ah 更倾向于直接基于 802.11ac 协议做改进。

已经商用两年多的 802.11ax, 是一个全方位碾压 802.11ac 的金牌骑手, 不仅 Wi-Fi 的速度比 802.11ac 提升近 3 倍, 而且多设备并发连接有 4 倍容量提升, 而且支持高密度连接。在这一点上和 802.11ah 如出一辙。这两个实际上是 802.11 在 IoT 布局的两个部分的协议。Wi-Fi 6 主要面向类似智能家居这样的室内场景的 IoT, 同时追求速率和功耗, 其还是工作在 2.4GHz 和 5GHz 频段, 与当前的 802.11 协议兼容。HaLow 实际上是面向传感器网络这一类的场景的。实际上, 802.11ah 和 802.11ax 的一些技术特征上是类似的, 比如 TWT, BSS Color, 两者都支持, 而且实际上 ax 是借鉴 ah 的, 但是在细节设计上, 两者还是有比较大的差异。

虽然说 Wi-Fi 6 支持高密度连接, 但是其还处于普及过程中。它虽然向前兼容其他 Wi-Fi 版本, 但是设备数量更大的 Wi-Fi 5、Wi-Fi 4 的设备无法使用这些新特性。这使得对之前版本的 Wi-Fi 协议, 在高密度连接场景的提升性能, 提高单个终端的使用体验显得格外有意义。

1.2.3 Wi-Fi 密集用户场景研究现状

无线 WLAN 密集场景是主要有两种: 一种是多 AP 场景, 在信号覆盖范围之内有许多邻近的 WLAN 小区使用频带重叠的信道; 另一种是单 AP 场景, 同一个 AP 的信号覆盖范围之内有许多与之连接的 STA, 本文主要研究后者, 智能家居场景是典型的第二种单接入点密集场景。

高密 WLAN 场景下的覆盖优化一直是热点问题。怎样进行网络覆盖的改进和规划, 提升高密场景下的信道接入性能, 为高密、高并发的用户提供良好的使用体验, 众多学者、研究机构和企业都进行了深入和长期的研究和探索。

在密集场景中, 用户发送数据产生碰撞的概率随着节点数量的增加而明显增加, 原因之一是目前 IEEE 802.11 中使用的 DCF 退避算法存在缺陷。虽然 DCF 指数退避算法在网络中 STA 数量较少时能很好地避免冲突, 但一旦与 AP 相关联的 STA 数量超过某个阈值, 该算法的性能就会明显下降。由于这个原因, 许多研究首先关注的是

改进标准中的 DCF 指数避免算法。

当前 802.11 标准里面的指数退避算法非常简单,若某个 STA 数据发送失败,竞争窗口值 (CW) 将直接加倍[2],但是成功发送后, CW 将更新为最小,这是假设该 STA 的争用级别已下降。实际上其争用级别更可能的是缓慢变化。因此,一种很直接的优化方法就是改进竞争窗口的更新法则,使其更加合理。例如, EIED 退避算法[2],数据帧传输成功后, CW 除以一个参数 r_D ; 而如果发生冲突, CW 则乘以一个参数 r_I 。通过调整参数值,系统吞吐量可以达到较好的水平。与之类似的,还有 SD[13]、LMILD[14]、MILD[15]退避算法改进方案,它们的区别主要在于 CW 的更新快慢程度,有线性变化的,也有成倍变化的。通过合理调整 CW 的大小,可以达到减小退避等待时间,减小发送数据时发生碰撞的概率的效果。

此外,在 GDCF [16]、HBAB [17]退避算法改进方案中,通过统计某个 STA 在过往一段时间内数据帧发送成功和冲突次数的比例来对其 CW 作出相应的调整,而不仅仅参照最近一次传输的结果。显然,这样调整 CW 会更加准确,更能反映出网络的真实运行状态。但相应的算法会变得更加复杂。

VC-UORA 算法[x]借鉴了 CSMA/CA 和 OFDMA 的协议思想将大量站点(STA)随机接入时的激烈碰撞转化为可避免的虚拟竞争,从而提高频谱资源的随机接入效率,但是以少量时间资源和算法复杂度为代价。在上述方案中,将信道被划分成多个子信道,但是有的方案使用了过于复杂的资源分配方法,倾向于向 LTE 靠拢,这不符合 Wi-Fi 网络的特点,而且考虑与现有 802.11 标准的兼容性。

MPR[40]是一种上行多用户 MAC 协议,不同用户的数据帧使用 CDMA 来区分。文献[41]提出了一种分布式下行 MU-MIMO 方案,根据 AP 侧队列中数据的等待时长以及业务等级来调度下行链路多用户的数据传输。但是上述方案都有一定的缺陷,而且对于分布式的 802.11 标准的 STA 来说,实现难度较大。

综上所述,在国内外研究中,解决密集场景下 Wi-Fi 性能下降的问题的思路主要有两个方面:其一是对二进制退避算法本身做改进。使用更加合理的方式来调整 CW 的大小,来达到减少退避时间、降低抢占信道时产生冲突的概率的效果,但是在 STA 超密集场景下性能提升有限。另一种思路是引入 OFDMA 和 MU-MIMO 等技术,相

关的研究虽然很多，但都有一定的局限性，比如有的方案需要进行时频资源块的分配，实现过于复杂，不符合 WLAN 协议的特点。

1.3 论文的研究内容

本论文基于实验室与某知名通信企业的项目合作研究，目的是在自研 Wi-Fi 芯片和平台上，面对终端数量较大的智能家居场景，基于智能家居终端设备原有的传输机制和报文特性，实现各个终端设备的高可靠数据传输。

基于 Wi-Fi 的智能家居业务场景有四个重要特征：

- (1) 终端数量多。存在 30 以上甚至 50 个终端设备；
- (2) 终端类型多。存在许多不同类型的终端设备，而不同类型的终端设备通常具有特定的流量模式和数据包大小。
- (3) 网络环境复杂。存在各种射频干扰，如其他家庭的无线通信信号等，而且家中终端设备排布很分散。
- (4) 绝大部分现有智能家居终端设备不支持 Wi-Fi 6，甚至不支持 Wi-Fi 5，而且很多设备不会像换手机一样，更换很频繁。

这是典型的 Wi-Fi 密集用户接入场景。研究在单个接入点下终端设备高度密集的场景，如何让用户能够及时、公平的接入网络，设计一种公平的接入解决方案，解决用户并发接入风暴的问题，优化密集用户场景下的系统性能和传输效率一直是一个热点问题。

Wi-Fi 最初旨在为少数设备提供宽带无线互联网接入。虽然最新的 Wi-Fi 6 [2]被称为适用于物联网，但是目前支持 Wi-Fi 6 的设备的普及程度并不理想，尤其是简单设备控制类的智能终端都是 Wi-Fi 5 甚至 Wi-Fi 4 等。除此之外，具有零星物联网流量的大量 STA 会导致高碰撞概率。尽管 802.11ax 实现了 OFDMA，允许多个 STA 同时传输，但 Wi-Fi 6 的类似 ALOHA 的 OFDMA 随机接入对于大量竞争 STA 效率低下。

现有的用于优化密集场景系统性能的技术方案等，都不能完全适用于智能家居业务场景，如：

特征（1）由于设备数量大，而 Wi-Fi 通信使用的是典型的无线介质信道，介质访问接入方式是 DCF 机制，在设备数量庞大时，产生碰撞的概率会直线上升，数据帧传输成功率急剧下降。另外，在终端数量非常多时，除数据帧之外的管理帧和控制帧可能会消耗掉 5%~20% 的信道资源，极端情况下绝大多数设备都无法完成业务的传输，整个系统是处于崩溃的边缘，这种情况称之为信道接入风暴。

对于特征（2），传统 DCF 对于同质业务的设备是合理的，因为从概率上来讲，它们抢占到信道使用权的概率都有相同的，也就是说对所有设备“一致对待”。但是智能家居场景终端设备类型多，不同的终端有不同特征的业务负载，具有不同的业务数据包到达率、数据包大小和调制编码方式，这种情况下，存在小流量设备竞争信道退避次数很多，发送成功率很低，甚至无法发出数据的状况，称之为接入信道的公平性问题。

针对特征（3），在家庭局域网中网络环境复杂，设备较分散，而且可能有墙体阻挡，某些终端载波检测不到对方的存在，容易出现“隐藏终端”问题，虽然现有 RTS/CTS 机制可以解决隐藏终端问题，但是会牺牲 10% 的空口带宽。更糟糕的情况是，某些角落里的接收信号强度（RSSI）低于 -75dBm 的终端会经常出现速率不稳定甚至掉线等情况。另外还要遭受外界环境的干扰，同频或邻频干扰。这使得对智能家居场景优化的难度进一步提升。

特征（4）说明，家中即使更换了新的支持 Wi-Fi 6 的路由器，现有智能家居设备仍然无法使用 Wi-Fi 6 其中引入的新的特性。而对于设备本身来说很多设备不会更换的很频繁，另外直接对设备升级 Wi-Fi 6 有些困难，而且成本很高，还存在与主控芯片的兼容性问题，有的设备甚至无法更换 Wi-Fi 芯片，而且路由器与智能家居设备并不是同一厂商，所以要想对智能家居场景的传输网络进行优化，还需要考虑对路由器透明。

本文针对 Wi-Fi 链路下的智能家居业务特征和传输需求，提出了全新的对路由器透明的传输可靠性保证优化方案，用于实现智能家居业务在 Wi-Fi 链路上的高可靠传输，让各设备能够及时、公平的接入网络，提高用户的使用体验，为智能家居的进一步发展增添动力。

1.4 论文组织与结构（需更新，逻辑图？）

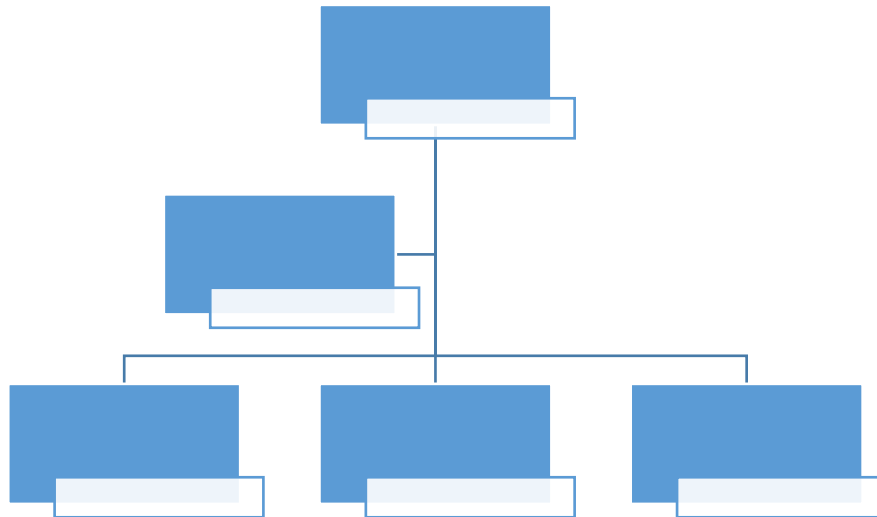


图 1.1 组织结构图

第一章是绪论部分，介绍了本文的研究背景与意义，并介绍了国内外针对智能家居、Wi-Fi 技术和密集用户场景的发展研究现状，接着对本文的研究内容进行了详细介绍，最后阐述了论文的整体组织与结构。

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 Wi-Fi 链路上影响智能家居设备业务的时延、丢包率和带宽的几个机制；然后介绍了排队系统的基本概念以及 M/M/1/K 模型，接着对单数据包流量模式和饱和流量模式下 DCF 机制性能分析模型进行了详细的阐述；接着介绍了在 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，最后对 NS-3 仿真平台进行了相关的介绍。

第三章详细阐述了针对用户密集的智能家居场景中，为了降低用户量较大时接入信道的冲突概率，提升系统吞吐量，基于改进 DCF 机制提出了新的设计方案，首先对现有的密集用户场景下降低冲突概率的信道接入机制的工作原理以及优缺点进行了详细的分析，然后通过在实际的智能家居网卡开发板的实际性能测试分析了智能家居场景现有的问题和难点。然后详细表述了所提出的受控并行随机接入机制的系统模型。接着在对新方案的冲突概率和吞吐量进行理论分析之前，基于 Bianchi 模

型提出了更加符合智能家居业务场景的改进 Markov 模型。最后通过 NS-3 仿真验证了改进 Markov 模型的准确性。

第四章，针对第三章阐述的受控并行随机接入机制的智能家居业务传输优化方案，做了两种对路由器透明的实现，对两种方案的实现细节进行了阐述，最后对两种实现方案的实际测试结果进行了分析，并与第三章理论值进行了对比。

第五章，在第三章提出的业务优化方案的思想基础上，提出了基于地理分区的智能家居场景隐藏终端优化方案，为优化隐藏终端问题提供了新的思路。

第六章对全文进行了总结，以及对下一步工作的展望。

2 研究背景技术概述

2.1 Wi-Fi 下的传输机制

从宏观角度，802.11 协议可以分为两个核心部分：链路层业务和物理层传输。

链路层业务主要制定了 WLAN 链路协商的规则，以及针对 WLAN 接入服务而设计的系列功能，例如报文重传和确认、重复报文检测、密钥协商、加密保护、漫游等等。而物理层传输则实现 WLAN 设备之间的能够完成信号的发送和接收，并致力于不断提高数据传输的物理速率。

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。而且无线链路还存在着隐藏终端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，802.11 协议中引入了两种信道接入方法：分布式协调功能（DCF，Distributed Coordination Funtion）和点协调功能（PCF，Point Coordination Funtion）。两者之间的关系如图 2-1，现在 802.11 设备大多使用 DCF，随机退避准则接入网络，PCF 则是基于轮询机制，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]，使用并不广泛。

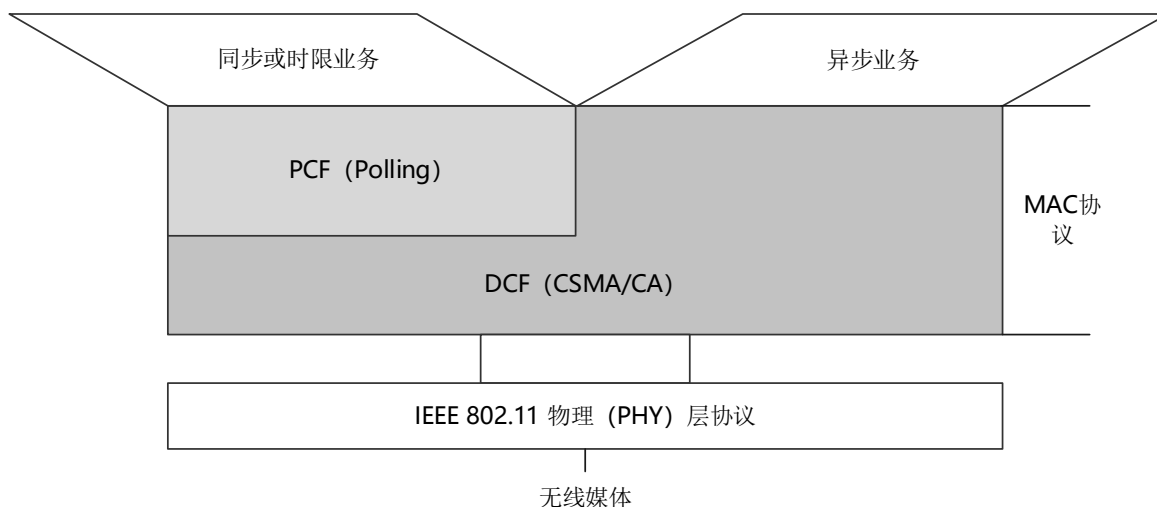


图 2-1 802.11 标准中 DCF 与 PCF 的关系

2.1.1 DCF 机制

DCF 机制主要包含 CSMA/CA 机制，二进制指数退避算法以及帧间间隔三个重要组成。DCF 一般采用载波监听多路访问/冲突避免（CSMA/CA）避免冲突接入信道。CSMA/CA 机制，在密集场景时，用户发送数据发生冲突的概率随着节点个数显著增加的一个原因就是 IEEE802.11 目前采用的 DCF 退避算法存在缺陷。在网络节点数目较少的时候，DCF 指数退避算法尚能较好的规避冲突，一旦 AP 关联的 STA 个数超过一定阈值，该退避算法的性能就会大幅下降。

2.1.1.1 帧间间隔

帧间间隔（IFS, Interframe Space），是 802.11 协议为了尽量减少碰撞引入的，协议规定所有 STA 在完成发送之后，必须在等待一段很短的时间（全程侦听）才能发送下一个帧，这段时间统称为 IFS。IFS 的长短取决于 STA 要发送的帧的类型。高优先级的帧的 IFS 较短，故可以获得优先发送权，但是低优先级的就必须等待较长的时间。

如果某个 STA 的低优先级帧还没来得及发送，其他 STA 的高优先级帧已经发出，则信道变为繁忙状态，此时低优先级的帧就只能再推迟发送了。这样做减少了发生碰撞的机会，至于各种 IFS 的具体长度，取决于使用的 PHY 层的特性。

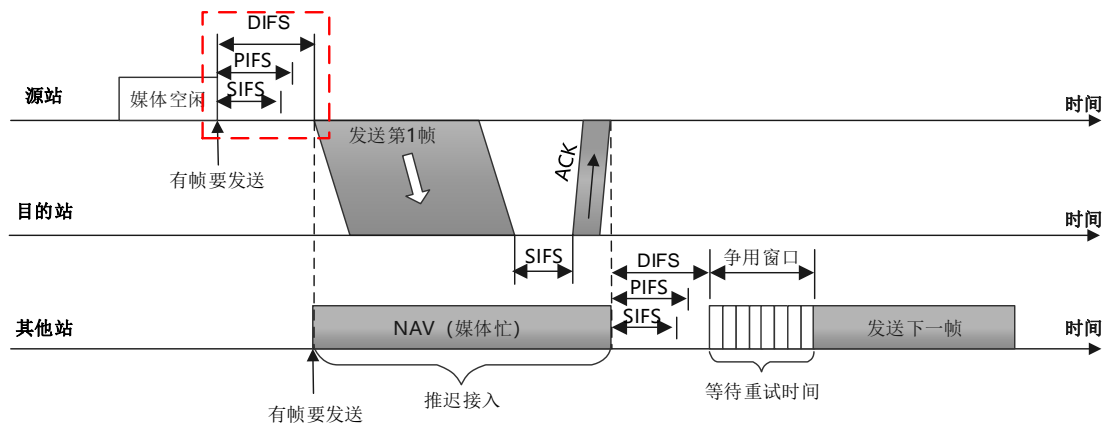


图 2-2 各种 IFS 示意图

IFS 时间越短，优先级越高，从小到大有短帧间间隔（SIFS）、集中协调功能帧间间隔（PIFS）、分布式协调功能帧间间隔（DIFS）以及错误帧间间隔（EIFS）。IFS 示意图见图 2-2。

SIFS 是最短的 IFS，用来间隔优先级最高的帧，如控制帧（RTS/CTS/ACK）等；

PIFS 只能工作在 PCF 模式的 STA 使用；

DIFS 只能工作在 DCF 模式的 STA 使用；

EIFS 在前一个帧出错的情况下，STA 不得不延迟 EIFS 而不是 DIFS 时间段，再发送下一帧。

2.1.1.2 CSMA/CA 机制

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 Wi-Fi 空口上进行发送时，会执行以下流程，如图 2-3 所示：

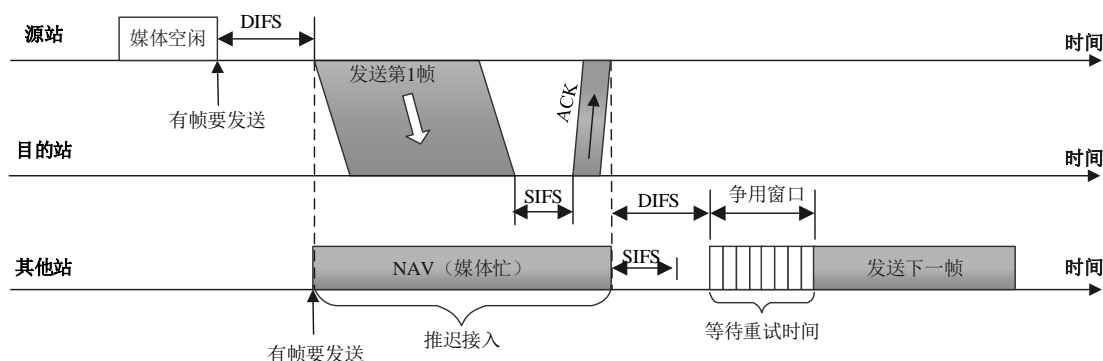


图 2-3 CSMA/CA 原理图

其中 NAV 主要是利用无线数据帧 MAC 头部中的 Duration 字段进行设置。

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或者超时未接收到 ACK 帧，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。因此，在繁忙的无线链路上，业务报文会由于竞争信道失败出现延迟，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。

2.1.1.3 二进制退避算法

前一小节介绍了 CSMA/CA 的基本工作原理，在所有终端节点执行 CSMA/CA 机制进行信道抢占的时候，某些终端节点可能会产生冲突，这些冲突节点为了降低下一次冲突的概率，会执行退避算法来随机延迟数据帧的发送。本小节来介绍二进制退避算法（BEB，Binary Exponential Back off）的原理和具体执行流程。

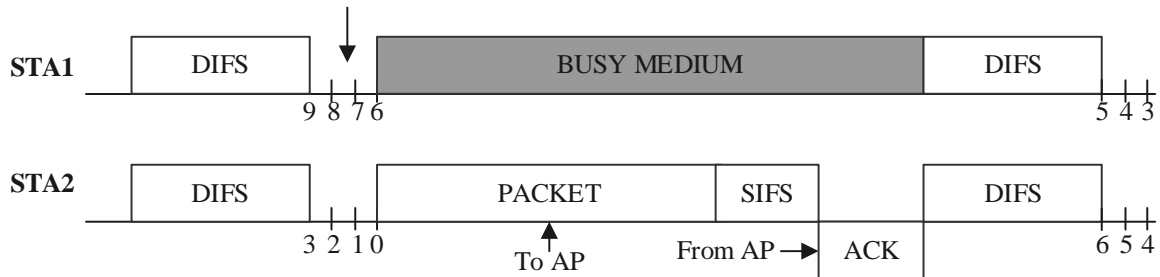


图 2-4 二进制指数退避算法工作原理

在 BEB 机制中，当 STA 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 $k = \min\{\text{退避次数}, R\}$ ，其中 R 为设定帧的最大重传次数，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。BEB 的具体执行过程举例如图 2-4。

图 2-4 中，在某一时刻 STA1 退避计数器为 9，STA2 退避计数器为 3，在时隙

内检测到信道空闲，退避计数器减 1，STA2 的计数器先归零，而且检测信道是空闲的，STA2 开始向 AP 发送数据帧，此时 STA1 检测到信道忙，冻结计数器。当 STA2 收到来自 AP 的 ACK，表示 AP 正确接收，继续后续帧的发送；如果 STA2 计数器先归零，但是检测到信道是忙的状态，则 STA2 不会立即发送数据帧，而是将退避阶数加 1，在新的竞争窗口内选择一个值作为新的退避计数器，继续检测信道，重复上述过程。

2.1.2 PCF 机制

PCF 全称是 Point Coordination Function，即点协调工作模式。实际上，在 802.11 的 MAC 层协议结构中，PCF 是基于 DCF 之上的协议，因为 PCF 是以 DCF 作为基础扩展的，所以 PCF 和 DCF 可以很好的兼容工作。

在 PCF 模式中，实际上是 DCF 和 PCF 基于一种交替工作的机制，即两者各占据一段时间，交替进行工作。这个交替周期被称为 CFP 重复周期（CFP repetition interval），CFP 周期包含 CFP 和 CP 两个部分，如图 2-5

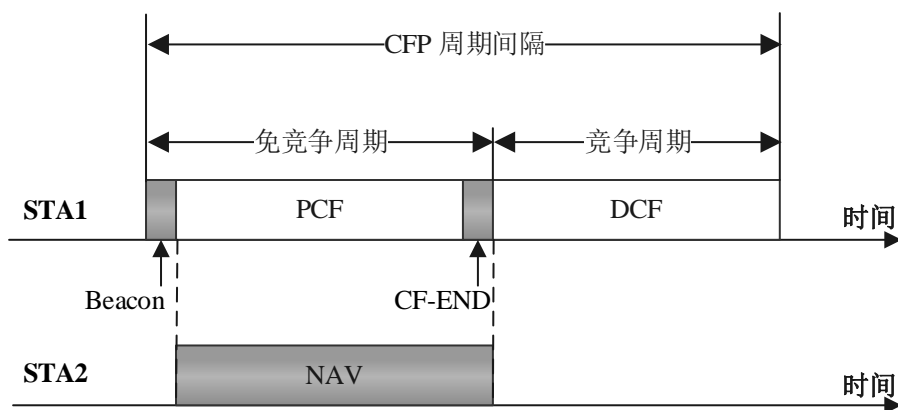


图 2-5 PCF CFP 周期

其中，无竞争时间（CFP）：Contention-Free Period，该时间是为了 PCF 工作所设定的一个时间段。CFP 时间是通过使用虚拟载波监听（NAV）的机制进行保护的，该 NAV 是由 Beacon 帧设置的，并用 CF-END 帧作为结尾；竞争时间（CP）：Contention Period，在 CP 时间段内，协议执行 DCF 机制。在某一个 CFP 重复周期准备开始的时候，AP 首先把 Beacon 帧中的 NAV 相关字段设置为 CFPMaDuration。

PCF 中的无竞争服务实际上是为了对实时业务提供 QoS 支持而引入的, PCF 具有面向连接的特点,更适于为交互式实时应用提供严格的参数化的 QoS 保证,但 PCF 不能区分业务类型和业务源,而且其实现起来效率过低且过于复杂,故在实际产品中一般不被使用。

2.2 排队论相关概念

2.2.1 排队论系统概述

“顾客”和“服务员”是排队服务系统中,两个组成重要部分,“顾客”排队系统中请求服务的人或物;“服务员”:为“顾客”服务的设备或人员。为什么会产生排队的现象呢,主要是因为顾客到达间隔时间和服务间隔时间的随机性。排队系统组成结构见图 2-6

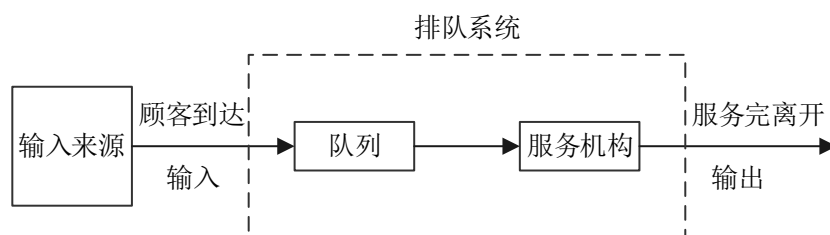


图 2-6 排队系统的组成

输入过程、排队（服务）规则和服务机制是排队系统的三个基本组成部分,其中输入过程代表顾客到达的规律,其中包含顾客的到达是独立的还是与某因素有关,是单个到达还是成批到达,是来自有限的总体还是无限的总体,顾客到达时间间隔的概率分布;排队规则代表顾客等待服务的排队准则,有先来先服务（first come,first served,FCFS）、随机服务（served in random order, SIRO）、优先服务（priority, PR）以及后到先服务（last come,first served,LCFS）等;服务机制主要包括服务机构的设置、服务台的数量、服务方式以及服务时间分布等。

排队系统主要分为三类,损失制系统（losing system）:没有顾客排队;等待制系统（waiting system）:顾客无限排队;混合制系统（mixing system）:排队长度或时间有限。

为了更方便的描述一个排队系统,提出了排队模型的符号表示为

$X/Y/Z/A/B/C$ ，其中 X 为顾客到达时间间隔分布； Y 为服务时间分布； Z 为服务台数目； A 为排队系统允许的最大顾客容量； B 为顾客总体数量； C 为排队规则。例如 $M/M/1/\infty/\infty/FCFS$ 。

排队系统的运行指标主要有六个，队长：系统中的顾客数，期望值表示为 L_s ；排队长：系统中排队等待的顾客数，期望值记为 L_q ；逗留时间：顾客在系统中的停留时间，期望值记为 W_s ；等待时间：顾客在系统中的排队的等待时间，期望值记为 W_q ；系统负荷：系统的繁忙程度，用 ρ 表示；忙期：服务机构连续工作时间的长度 T_b 。

2.2.2 泊松分布

如果离散型随机变量 X 的概率分布律为

$$P(X=k) = \frac{\lambda^k}{k!} e^{-\lambda}, k=0,1,2,\dots \quad (2-1)$$

其中 $\lambda(>0)$ 为常数，则称 X 服从参数 λ 的泊松（Poisson）分布，其期望平均值 $E(X)=\lambda$ ，方差 $D[X]=\lambda$ 。如果某一时间在特定时间间隔 $(0,t)$ 内发生的次数服从泊松分布 (λt) ，则该事件先后两次发生的时间间隔服从指数分布 (λ) 。

2.2.3 M/M/1/K 排队模型

$M/M/1/K$ 排队模型是典型的混合制排队系统，其系统中共有 K 个位置，1 个服务台， $1 \leq K$ ，当 K 个位置已被顾客全部占用时，新到的顾客就自动离开，当系统有空位置时，新到的顾客就进入系统排队等待服务，系统顾客到达的时间间隔服从参数为 $\lambda (>0)$ 的 Poisson 流，每个顾客所需的服务时间也是独立的，而且到达与服务是彼此独立的。假设系统按照先来先服务的规则服务顾客，用 p_j 表示顾客到达时看到有 j 个顾客的概率，则

$$p_j = \begin{cases} \frac{(1-\rho)\rho^j}{1-\rho^{K+1}}, \rho \neq 1 \\ \frac{1}{K+1}, \rho = 1, \end{cases} \quad 0 \leq j \leq K \quad (2-2)$$

其中, $\rho = \lambda / \mu$, 表示系统的繁忙程度。 $j=0$ 时有。

$$p_0 = \frac{1-\rho}{1-\rho^{K+1}} \quad (2-3)$$

2.3 802.11 DCF 性能分析模型

本章节介绍了两个 DCF 的分析模型, 分别针对单数据包流量模型和饱和流量模型, 其中单数据包流量模型是指在一个信标帧间隔内始终只有一个数据帧准备发送, 即缓存队列里始终只有一个数据帧待发; 而饱和流量模型是指局域网内的所有节点的流量无限大, 始终存在待发送的数据帧, 即缓存队列里始终有数据待发。

2.3.1 单数据包流量模型

在单数据包流量模型中, 终端节点在一个时隙内只有一个数据包需要传输, 并且在指定时隙到来之时, 所有的节点在准备发送数据帧时, 按 DCF 机制开始竞争, 竞争成功的节点在传输完数据后, 会再次进入睡眠状态, 此时参与信道竞争的节点数减少, 时隙内传输冲突概率也会降低, 因此传输冲突概率与时间和退避阶数相关[73]。

在 DCF 机制中, 终端节点的业务数据在传输成功之前, 在每个退避阶数都会进行一次信道资源抢占的尝试。从 0 阶开始, 若信道资源抢占失败, 则退避阶数会增加 1, 随之, 竞争窗口 (CW, Contention Window) 会增大。在每个退避阶数中, 终端设备会从竞争窗口中均匀随机选取一值作为退避计数值。如果终端设备在退避时间段内始终没有检测到其设备使用信道, 则退避计数器减 1。当退避计数器到 0 时, 进行该阶数的信道资源抢占尝试。

把第 k 阶退避的 CW 长度表示为 W_k ($k=0, 1, 2, \dots, R$), 其中 R 是一个常量, 表示最大重传次数。采用二进制指数退避时最大 CW 为 W_R , 即 $W_m = 2^{\min(k, R)} W_0$, 如果设备节点的某一个数据帧在经历了 $(R + 1)$ 次传输失败后, CW 将不会再次增大, 保持为 W_R 不变, 此时数据帧最多经历 W_{sum} 个计数 (即累计退避计数), 如式(2-4)所示

$$W_{sum} = \sum_{i=0}^R W_i \quad (2-4)$$

把设备节点的传输概率表示为 $(1 + R) \times W_{sum}$ 阶矩阵, 用符号 T 来表示。其中 $T_{m,n}$

为第 m 行第 n 列的元素，表示退避阶数为 $(m-1)$ 、网络计数为 n 个的设备节点的传输概率。

把设备节点碰撞概率也表示为 $(1+R) \times W_{sum}$ 阶矩阵，用符号 \mathbf{H} 来表示。其中 $\mathbf{H}_{m,n}$ 第 m 行第 n 列的元素，表示退避阶数为 $(m-1)$ 、网络计数为 n 个的设备节点的产生碰撞概率。因为所有节点采用相同的退避策略和参数，故所有设备节点具有相同的传输概率和碰撞概率。

当在 $(m-1)$ 轮退避中发生数据帧的碰撞后，终端设备将会从 $[0, W_m - 1]$ 之间均匀随机的选取一个整数值，作为退避计数器，故在 m 阶退避中网络计数的值均匀分布在 $[n+1, n+W_m]$ 之间，传输概率为 $1/W_m$ 。

$$\mathbf{T}_{m+1,n} = \begin{cases} 1/W_0, & m=0, n=1, \dots, W_0 \\ \sum_{j \in \beta(m,n)} \frac{1}{W_m} \mathbf{H}_{m,j}, & m=1, \dots, R, \beta(m,n) \neq \emptyset \\ 0, & \text{其他} \end{cases} \quad (2-5)$$

式中 $\beta(m,n)$ 表示节点的退避状态集合，由式(2-6)给出。

$$\beta(m,n) = \{ j: \max(n-W_m, m) \leq j \leq \min(n-1, \sum_{l=1}^{m-1} W_l) \} \quad (2-6)$$

假设终端设备每次尝试抢占信道的事件是独立的，终端设备碰撞概率表示为：

$$\mathbf{H}_{m,n} = \mathbf{T}_{m,n} (1 - (1 - \sum_{j=0}^R \mathbf{T}_{j+1,n})^{N-1}) \quad (2-7)$$

其中 N 表示时隙中的节点数。通过式(2-5)和(2-7)可得到时隙中 N 个节点（每个终端设备在一个时隙中只会发送 1 个帧）的竞争成功概率为：

$$P_N(1) = 1 - \frac{\sum_{m=0}^R \sum_{n=1}^{W_{sum}} \mathbf{H}_{m,n}}{\sum_{m=0}^R \sum_{n=1}^{W_{sum}} \mathbf{T}_{m,n}} \quad (2-8)$$

2.3.2 Bianchi 模型

2000 年 Bianchi 提出的 DCF 机制的 Markov 分析模型^[8]，引起了广泛轰动，现在描述 Bianchi 模型的建模过程以及求解过程。

在饱和流量模型中，有如下两个假设：

- (1) 共有 N 个节点；
- (2) 每个节点都有无限个数据包要发送。

用 $b(t)$ 表示局域网中任意某个 STA 退避计数器 Bcnt 的随机过程， t 和 $t+1$ 都是离散整数时间点，分别代表两个相邻的退避时隙的起始时刻。Bcnt 值在每个退避时隙开始时递减。Bcnt 的值取决于 STA 的退避阶数（或者尝试抢占信道的次数）。设 STA 最大退避阶数为 R ， R 为常量，则第 i 阶的竞争窗口大小为 $W_i = 2^{\min(i, R)} W_0$ ，将 STA 退避结束的随机过程表示为 $s(t)$ ，假设节点每次尝试传输的碰撞概率独立且为 p ，表示在该节点尝试传输时，至少有一个其他节点也在传输的概率。在节点数固定且节点流量饱和情况下， p 为常数。此时将 $\{s(t), b(t)\}$ 建模为如图 2-7 所示的离散时间马尔科夫链。在马尔科夫链的各状态的转移概率如式(2-9)所示。

$$\begin{cases} P\{i, k | i, k+1\} = 1 & k \in (0, W_i - 2) \quad i \in (0, R) \\ P\{0, k | i, 0\} = (1-p)/W_0 & k \in (0, W_0 - 1) \quad i \in (0, R) \\ P\{i, k | i-1, 0\} = p/W_i & k \in (0, W_i - 1) \quad i \in (1, R) \\ P\{R, k | R, 0\} = p/W_R & k \in (0, W_R - 1) \end{cases} \quad (2-9)$$

式(2-9)中存在简写，实际表示见式(2-10)

$$P\{i_1, k_1 | i_0, k_0\} = P\{s(t+1) = i_1, b(t+1) = k_1 | s(t) = i_0, b(t) = k_0\} \quad (2-10)$$

其中，每一行代表一个一步转移概率，第一行表示每个退避时隙开始，退避计数器减 1；第二行表示一个数据包成功传输后，退避阶数初始化为 0，退避计数器也在最小竞争窗口中随机取值；第三行表示数据包在 $(i-1)$ 阶退避中尝试传输失败，则退避阶数加 1，竞争窗口也增大为 W_i ；最后一行表明当退避阶数达到最大 R 时，此时信道抢占失败也不再增加阶数，竞争窗口保持最大竞争窗口 W_R 。

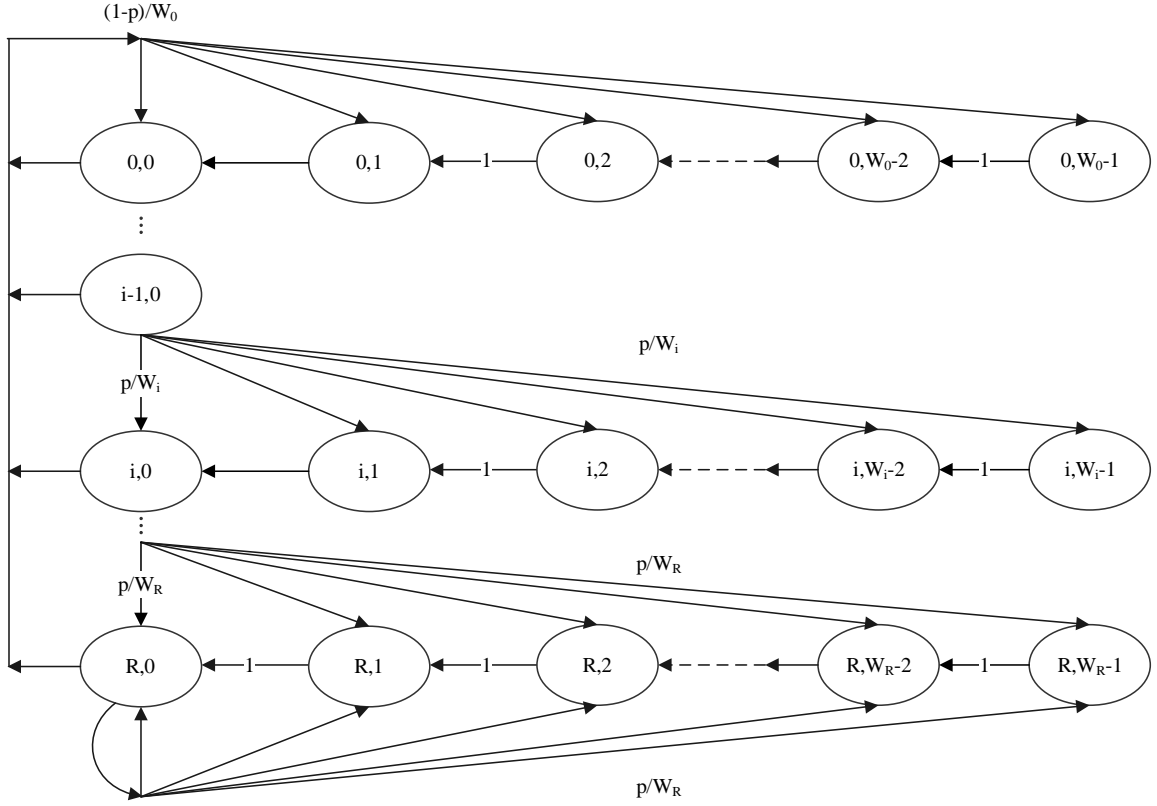


图 2-7 Bianchi 模型的马尔科夫模型状态转换图

Markov 状态转换图中，水平方向上，从右向左的每一步状态转移表示 STA 的退避计数器值减 1，而垂直方向上从上到下，每一步状态转移代表 STA 在空口上发生了一次碰撞。

$$b_{i,k} = \lim_{t \rightarrow \infty} P\{s(t) = i, b(t) = k\}, i \in (0, R), k \in (0, W_i - 1) \quad (2-11)$$

用式(2-11)表示马尔科夫链的一个平稳过程，由此可以得到一个封闭解：

$$\begin{cases} b_{i-1,0} \cdot p = b_{i,0} \rightarrow b_{i,0} = p^i b_{0,0}, 0 < i < R \\ b_{R-1,0} \cdot p = (1-p)b_{R,0} \rightarrow b_{R,0} = \frac{p^R}{1-p} b_{0,0} \end{cases} \quad (2-12)$$

$$\sum_{i=0}^R b_{i,0} = \frac{b_{0,0}}{1-p} \quad (2-13)$$

由马尔科夫链可知，对于 $k \in (0, W_i - 1)$ ，有

$$b_{i,k} = \frac{W_i - k}{W_i} \cdot \begin{cases} (1-p) \sum_{j=0}^R b_{j,0} & i=0 \\ p \cdot b_{i-1,0} & 0 < i < R \\ p \cdot (b_{R-1,0} + b_{R,0}) & i=R \end{cases} \quad (2-14)$$

将式(2-12)和式(2-13)代入式(2-14)，化简得：

$$b_{i,k} = \frac{W_i - k}{W_i} \cdot b_{i,0} \quad i \in (0, R), k \in (0, W_i - 1) \quad (2-15)$$

由式(2-12)和式(2-15)可知， $b_{i,k}$ 是 $b_{0,0}$ 和碰撞概率 p 的函数， $b_{0,0}$ 可通过归一化条件计算：

$$\begin{aligned} 1 &= \sum_{i=0}^R \sum_{k=0}^{W_i-1} b_{i,k} = \sum_{i=0}^R b_{i,0} \sum_{k=0}^{W_i-1} \frac{W_i - k}{W_i} = \sum_{i=0}^R b_{i,0} \frac{W_i + 1}{2} \\ &= \frac{b_{0,0}}{2} \left[W_0 \left(\sum_{i=0}^{R-1} (2p)^i + \frac{(2p)^R}{1-p} \right) + \frac{1}{1-p} \right] \end{aligned} \quad (2-16)$$

$p \neq 1/2$ 时，求得：

$$b_{0,0} = \frac{2(1-2p)(1-p)}{(1-2p)(W_0 + 1) + pW_0(1-(2p)^R)} \quad (2-17)$$

退避计数器递减到 0 时，节点开始尝试传输，尝试传输的概率表示为 τ ，有：

$$\tau_1 = \tau = \sum_{j=0}^R b_{j,0} = \frac{b_{0,0}}{1-p} = \frac{2(1-2p)}{(1-2p)(W_0 + 1) + pW_0(1-(2p)^R)} \quad (2-18)$$

当 $R=0$ 时，即为没有指数退避算法时的情况， τ 与 p 无关，式(2-18)会变得非常简单，仅和初始退避窗口有关。如式(2-19)

$$\tau = \frac{2}{1+W_0} \quad (2-19)$$

但是实际上， τ 是依赖于碰撞概率 p 的。式(2-18)中显示，尝试传输概率 τ 依赖于节点碰撞概率 p 。由于节点发生碰撞的条件为节点在尝试传输时，其余 $(N-1)$ 个节点中存在至少一个节点也在尝试传输，即：

$$p = 1 - (1-\tau)^{N-1} \quad (2-20)$$

式(2-18)和式(2-20)代表一个非线性系统，其中两个未知数 τ 和 p 可以用数值解

的方法解出。

由式(2-20)可知 p 的反函数为

$$\tau^*(p) = 1 - (1 - p)^{1/(N-1)} \quad (2-21)$$

τ 是一个连续单调增函数, $p \in (0,1)$, 其中从 $\tau^*(0)=0$, 上升到 $\tau^*(1)=1$, 其中式(2-18)也是一个连续函数, 对于给定值 $p=1/2$ 时, 可以简化为

$$\tau(p) = \frac{2}{1 + W_0 + pW_0 \sum_{i=0}^{R-1} (2p)^i} \quad (2-22)$$

此时 $\tau(1/2) = 2 / (1 + W_0 + RW_0 / 2)$, 也是典型的连续单调增函数。

2.4 Linux 网卡驱动开发技术

目前 Linux 平台是承载智能家居业务的主流操作系统, 要实现提高其业务的传输可靠性的设计开发, 就需要知晓 Linux 平台下的报文传输管理机制。

802.11 协议栈, 自 2.6.22 版本之后, 便正式加入了 Linux 内核, 现在 Linux 内核中支持多种设计架构的网卡驱动模块^[9]。过去, 有些老版本的驱动在内核中使用 wext (无线拓展) 接口, 可以直接和用户态交互; 还有部分网卡直接把 MLME (MAC 层管理实体) 功能在硬件中实现, 而网络接口和网卡设备之间关联关系的管理则通过 cfg80211 层完成, 这类网卡驱动统一被称为 FullMac 驱动; 但目前大部分网卡, 都是通过软件实现 MLME 功能, 使用 mac 80211 层中实现 MLME 的框架, 驱动开发者只需关注 SoftMac 驱动中具体功能的实现。有的厂商会把网卡的功能在软硬件之间进行分配, 一部分由硬件实现, 而一部分用软件实现, 这种驱动就称为 HalfMac^[10]。两种 Linux 网卡驱动的具体的层次结构如图 2-8 所示。

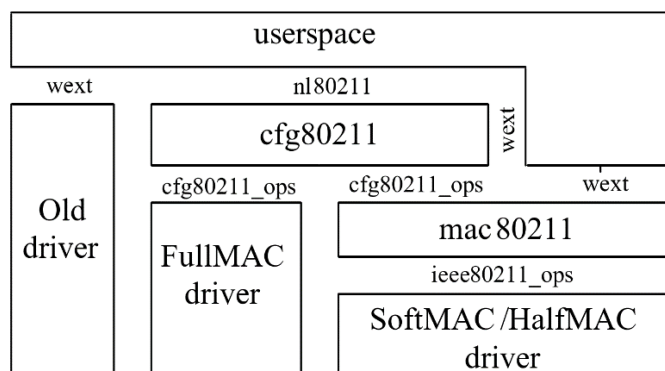


图 2-8 Linux 802.11 协议栈架构

本文所使用的 Wi-Fi 网卡的 Linux 驱动的功能是基于 SoftMac 驱动实现的。驱动中主要划分了 UMAC 和 LMAC 两个部分，分别完成 802.11 协议栈的不同功能。其中 802.3 至 802.11 的报文转换、链路管理等功能由 UMAC 层负责，LMAC 层主要负责 802.11 报文的发送调度、硬件交互等。两个层次模块协同工作共同实现 802.11 协议下的报文收发流程。

2.5 NS-3 仿真平台介绍

2.5.1 NS-3 平台简介

NS-3 (Network Simulator-3) 是一款全新的开源的离散事件驱动的网络仿真器，Linux, Windows(Cygwin), MAC OS 等主流操作系统平台上都可以使用，其各功能模块是使用 C++编写的，做网络仿真时既可以使用 C++，也可以用 Python 来编写脚本。和其他仿真平台相比，NS-3 具有更强大的可扩展性，用户可根据需求添加对应的网络功能模块，可以使用 NS-3 提供的各种仿真工具对不同的网络环境和协议策略进行仿真分析。

NS-3 并非与 NS-2 一脉相承，基本上是一个全新的仿真器，NS-3 并不包含所有 NS-2 的功能，相关接口和模块都不尽相同，两者在仿真流程上存在很大区别，NS-3 与 NS-2 部分特征对比如表 2-1:

NS-3 允许在模拟器中执行现实世界中的代码，借助 DCE (Direct Code Execution) 可以把整个 Linux 网络协议栈封装到一个 NS-3 节点中。因为 NS-3 在模块扩展上的优势，而且其与 Internet 协议设计更高的一致性，目前 NS-3 得到了越来越多的用户

的青睐，在做网络仿真使优选 NS-3。此外，由于 NS-3 对 802.11 标准 MAC 层和 PHY 层相关模型的支持，使用 NS-3 进行 Wi-Fi 网络仿真时会更简便。

表 2-1 NS-2 与 NS-3 特征比较

特征	NS-3	NS-2
发布时间	2008	1996
编程语言	C++	C++&OTcl
仿真脚本	C++&python	OTcl
仿真逻辑	分布式仿真	顺序式仿真
仿真速度	快	较慢
可视化	NS3-viz	NAM
可扩展性	强	较差
80211 模型	支持	不支持

作为一款由 C++编写的网络仿真器，NS-3 拥有一系列层次分明、功能强大的模块，常用的基本模块包括 10 类(表 2-2)。

表 2-2 NS-3 基本模块

模块	主要功能
Core	NS-3 核心的基本机制，如事件调度，日志等
Network	网络数据分组模块，一般仿真都会使用
Internet	TCP/IP 协议族实现模块，包括 TCP、UDP、IP 等
Applications	常见应用层协议实现模块，如 HTTP 等
Mobility	移动模型模块，为节点添加移动属性
Status	统计框架模块，方便对仿真数据的收集、分析和统计
Tools	作图工具
Netanim	动画演示工具
Visualizer	可视化界面工具
Wifi	802.11 协议族，包含部分 MAC 层和 PHY 层相关模型

根据不同仿真需求，可在 NS-3 原有模块上进行对应的修改，或添加一个新的模块，用于搭建需要的网络环境，完成相关仿真流程。

2.5.2 NS-3 关键概念

1. 节点

NS-3 将所有用于连接网络的计算单元抽象为一个节点(Node), 在 C++中用 Node 类来表示, Node 类用于模拟计算设备在网络中的各种表现。

可以把 Node 看成一台计算机, 基于 C++面向对象的开发思路, 用户可对 Node 类添加所需要的功能, 如各类应用程序、协议栈等, 使该节点可在相关网络中实现相应工作。

2. 应用

一般来说, 计算机软件分为两大类, 完成用户任务的应用软件和管理系统资源的系统软件。以 Linux 操作系统为例, 两类软件的区别在于所拥有的权限不同。系统软件是通过内核线程实现其功能的, 几乎所有计算机资源都可任其使用, 如内存、处理器等; 而应用软件则是基于用户线程实现的, 受到内核的控制, 只能使用部分资源, 如果想要使用更高的权限, 则需要使用系统调用、进程通信等方式。

在 NS-3 中并没有操作系统这一概念。但是, 为了模拟网络仿真中各类应用程序的网络需求, 将相关程序抽象为应用(Application), 在 C++中用 Application 类表示, Application 实际上是运行在 Node 内部的用户软件。在网络协议仿真时, 所需的用户程序可以通过对 Application 类的定制来实现, 例如 TCPClient 和 TCPServer。

3. 信道

信道即网络数据传输的通道, 在 NS-3 中基本通信子网被抽象为信道(Channel), 用 Channel 类表示。Node 之间的网络连接所使用的信道是通过 Channel 类提供的方法进行。通过对 Channel 类的面向对象编程, 可用其模拟现实世界中大部分网络传输通道, 如以太网或无线网。

NS-3 中共有三种最基本的 Channel: CsmChannel(以太网)、PointToPointChannel(点对点通信网络)和 WifiChannel(无线网络)。当用户在进行 802.11 协议仿真时, 通过设置 WifiChannel 信道模型的相关参数等, 即可实现对 802.11 MAC 层或 PHY 层特性的模拟。

4. 网络设备

网卡是计算设备连接到网络所必需使用的硬件，而且网卡若想在某一操作系统上正常运转，还需要对应平台的网卡驱动程序（net device driver）。以 Linux 为例，在 Linux 中，一块网卡和其对应的驱动程序被称为一个网络设备(net device)，该 net device 有一个特定的名称，如 wlan0 表示一个用于连接无线网络的无线网络设备。Linux 内核通过无线网络设备驱动(wireless net device driver)与相应的网卡进行数据交互，实现 802.11 协议数据收发等相关功能。

在 NS-3 中，硬件设备与软件驱动两者一起被抽象为网络设备(NetDevice)，用 NetDevice 类进行表示，NetDevice 类提供管理 Node 与 Channel 连接的方法。一个 Node 可拥有多个 NetDevice，通过多个 Channel 与另一个 Node 相互通信。用户可根据特定的需求对 NetDevice 类进行定制，用来实现不同类型的 NetDevice。

网络设备是与信道相对应的，NS-3 中共有三种基本网络设备: CsmaNetDevice(有线网卡)、PointToPointNetDevice(点对点通信设备)和 WifiNetDevice(无线网卡)。

5. 拓扑助手

在 NS-3 仿真过程中，NetDevice 与 Node 之间需要连接、NetDevice 与 Channel 之间也需要连接、分配 IP 地址等，这些等都是很常用的操作。如果每次仿真前都要进行很多机械重复的步骤，需要使用很多 NS-3 核心操作来完成，而且分步太多容易出现错误。

为此，NS-3 定义了拓扑助手(Topology Helper)类，简化了网络拓扑配置，对上述流程整合简化。用户在做网络拓扑配置时，只需选择合适的 Topology Helper，或根据自身需求定制 Helper 类，然后通过一些简单的操作，即可完成某类网络仿真环境的配置。

2.5.3 NS-3 仿真基本模型

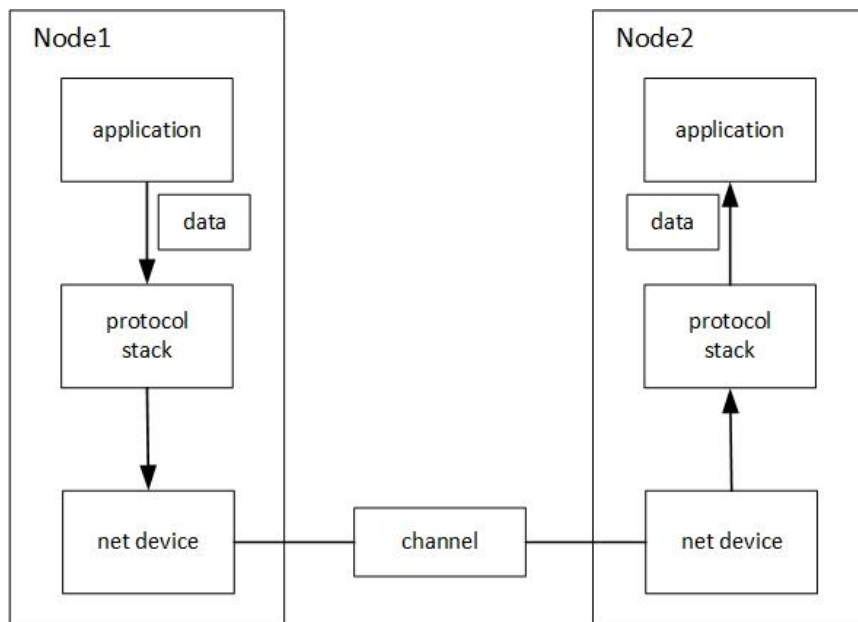


图 2-9 NS-3 仿真基本模型

NS-3 中使用 Node 模拟计算设备，用 Channel 表示传输介质，用 NetDevice 表示 Node 之间通信使用的接口。不同类型的网络结构对应不同类型的仿真模型，如 Wi-Fi 网络对应 Wi-Fi 模型。

通过 Node、Channel 和 NetDevice 等使得网络节点实现了物理连接，相当于台式机接好了网线，但实现通信，还需要通信协议的支持。Node1 中，应用层产生的数据，使用 socket 类编程，实现数据分组的向下传递，分组通过协议栈向下传递给 NetDevice，NetDevice 中包括 MAC 层、PHY 层协议，于是分组正如在现实网络中一样，由数据帧转换成二进制流，最终变成信号通过 Channel 传输到 Node2。

2.6 本章小结（需更新）

本章主要介绍了后续章节中所使用的相关技术以及工具。介绍了 802.11 协议中的 DCF 和 PCF 两种基本信道接入机制的工作原理；接着讲述了排队系统的基本组成，泊松分布、M/M/1/K 排队模型等排队论相关的理论知识；后面介绍了典型的单数据包流量模型和 Bianchi 模型两个 DCF 性能分析模型，包括建模求解等过程；接着讲述了 Linux 平台下的 802.11 标准网卡驱动的整体架构；最后介绍了 NS-3 仿真平

台的关键概念和基本仿真流程。通过本章相关技术和工具的介绍，都是本文后续章节中要使用的，在此集中介绍既有助于完成全文的理论分析和方案仿真，也方便读者理解后续内容。

3 基于改进 DCF 机制的智能家居业务传输优化研究

3.1 引言

基于 Wi-Fi 的智能家居场景，有两个重要的特点：终端数量众多，有超过 30 甚至 50 个设备；不同类型的设备的流量模式不同，即不同的终端有不同模式的业务负载。而对于家庭中的 Wi-Fi，作为典型无线通信方式，介质访问采用 CSMA/CA 机制，在终端设备产生冲突的时候，采用二进制退避算法进行延迟发送，该机制是完全分布式的，具有原理简单，易于扩展，实现简单等特点。但是 CSMA/CA 机制有着一个很明显的缺点，当终端数量达到某个门限时，设备节点在进行信道资源的抢占时，冲突概率会大大增加，导致系统吞吐量急剧下降。

因此，本章提出了一种基于改进 DCF 机制的受控并行随机接入的智能家居业务吞吐量优化方案，来降低智能家居密集用户场景下，抢占信道时产生冲突的概率，让各设备能够及时、公平的接入网络，从而提升系统吞吐量，进而提高智能家居用户的使用体验。并使用改进 Markov 模型对优化方案进行了理论性能分析，与原有 DCF 机制进行了性能对比。

3.2 现有无线信道接入机制分析

3.2.1 802.11e

DCF 和 PCF 两种信道接入机制均不支持 QoS，所以在 802.11e 中，支持 QoS 的 DCF 就变成了 EDCA（Enhanced Distributed Channel Access）模式，支持 QoS 的 PCF 就变成了 HCCA（HCF Controlled Channel Access）模式。两者最大的共同点就是都支持 TXOP。其实 HCCA 相对简单点，可以简单理解成 PCF+TXOP，而 EDCA 会复杂一点，其除了支持 TXOP 外，还需要对竞争参数有一些修改。接下来具体分析 EDCA 相对于 DCF 以及 HCCA 相对于 PCF 在信道接入机制的性能优化上做的相关改进，802.11e MAC 层协议结构如图 3-1。

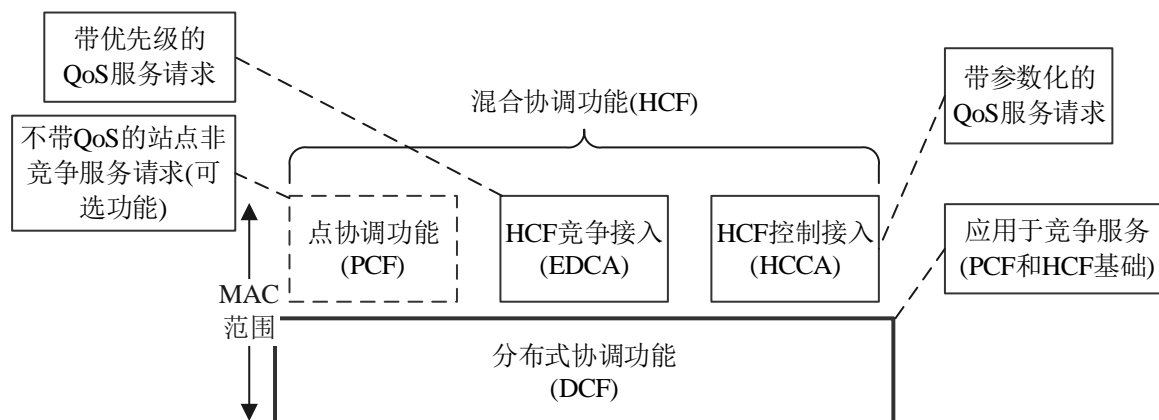


图 3-1 802.11e MAC 层协议结构示意图

3.2.1.1 DCF 与 EDCA

为了对不同流量模式，不同优先级的数据业务提供可调度发送，802.11e 在 DCF 基础上做了许多基于优先级区分机制的改进，改进机制命名为 EDCA（Enhanced Distributed Channel Access）机制。EDCA 是一种基于业务优先级调度的信道接入机制，它相对于原始的 DCF 机制，主要改进有两方面：其一，针对 DCF 机制中的几个参数的控制进行的修订；其二，引入了 TXOP 机制。

IEEE 802.11e 的 EDCA 机制基于 IEEE 802.11d 中定义的 8 种优先级，映射出了 4 种访问类型 Access Category (AC)，如表 3-1 所示。其中 AC_VO 的优先级最高，然后依次降低，AC_BK 优先级最低^[1]

表 3-1 IEEE 802.11e 的业务优先级

业务类型	优先级
AC_BK(Background)	1
AC_BE(Best-effort)	2
AC_VI(Video)	3
AC_VO(Video)	4

关于 EDCA 机制对 DCF 机制改进有以下 4 个方面：

(1) IFS (interframe space)

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 $DIFS = SIFS + 2 * Slot$ 。EDCA

机制在这种等待机制引入了 AIFS (arbitration interframe space)。在 EDCA 机制下, 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), 不同帧间间隔可由式(3-1)得出。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^[12]。

$$\begin{cases} PIFS = SIFS + 1 * Slot \\ DIFS = SIFS + 2 * Slot \\ AIFS = SIFS + n * Slot \\ EIFS = DIFS + n * Slot \end{cases} \quad (3-1)$$

其中 n 与优先级有关, n 越大, 意味着每次接入信道需要等待的时间就越多, n 与优先级的关系由式(3-2)表示

$$n = AIFSN[AC] \quad (3-2)$$

其中 AC 代表优先级, AIFSN 为 AC 到 n 的映射关系。

(2) CW (Contention Window)

DCF 机制中, CW_{min} (最小竞争窗口) 和 CW_{max} (最大竞争窗口) 是固定的, 如 802.11b 中, CW_{min} 的值为 31, CW_{max} 的值为 1023。当 CW (竞争窗口) 超过最大值后, 如果再发生退避, CW 便不再增大。因此, CW_{min} 和 CW_{max} 决定了平均退避时间值, 这两个数值越大, 平均退避时间就越长, 报文越不容易发送出去^[13]。EDCA 机制则将 CW_{min} 和 CW_{max} 的值与发送优先级关联了起来, 针对不同的优先级, 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) 重传次数限制

在 802.11 中, 默认的重传是 7 次, 如果最后一次失败才进行丢包。而在 802.11e 中, 这个也是一个可以控制的参数, 默认情况下是和 802.11 相同的, 但是实际情况下, 这个是可以控制的。

(4) TXOP 机制

当 STA 需要发送报文时, 可通过 RTS/CTS 机制提前协商好一段时间, 这段时间别的设备不得使用空口信道资源, 这段时间被称为 STA 协商得到的 TXOP (传输机会), 而 $TXOP_{Limit}$ (最大传输机会) 则表示 TXOP 的最大取值^[14]。在 DCF 机制下,

$TXOP_{Limit}$ 是个固定值。在 EDCA 机制中对 $TXOP_{Limit}$ 做了修订，将数据业务的优先级与 $TXOP_{Limit}$ 关联起来，不同优先级的业务使用不同的 $TXOP_{Limit}$ ，从而实现业务发送的 Qos 调度^[15]。

最终，在 EDCA 机制下，各个发送队列的优先级对比情况如图 3-2 所示。

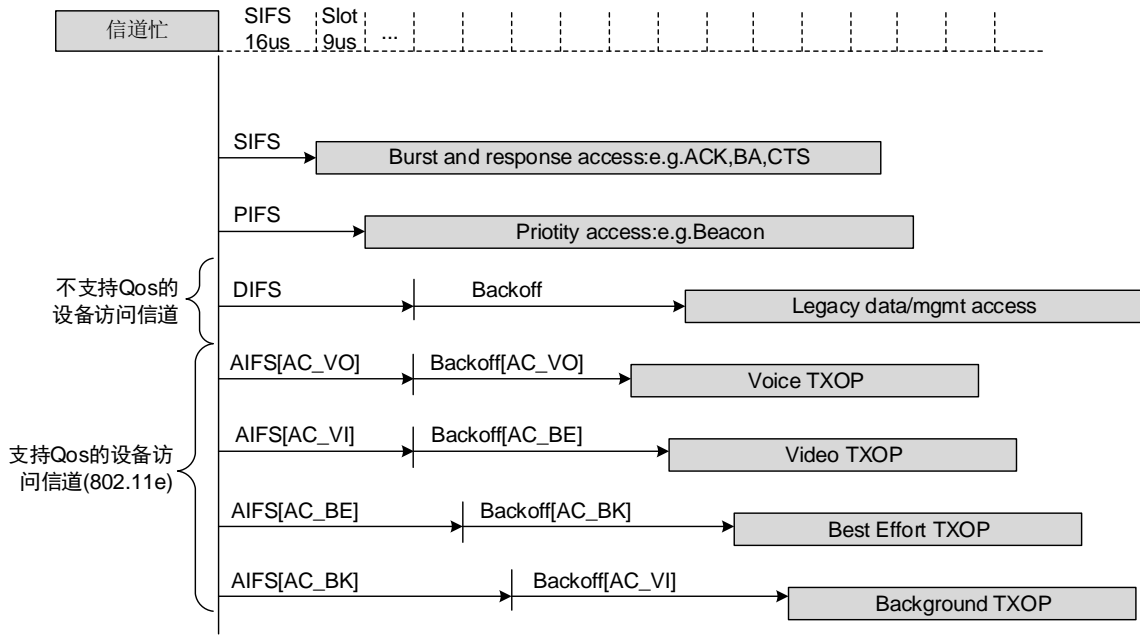


图 3-2 EDCA 机制下的发送优先级对比

因此，在 EDCA 机制下，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 Wi-Fi 链路上的传输能够获得不同优先权限。在高优先级的数据业务将更容易争抢到信道，而低优先级的数据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。

EDCA 机制，是 AP 和 STA 之间协商决定使用的，只要有一方不支持，就只能使用 DCF 机制，上述机制都无效。目前现网的智能家居设备支持 802.11e 设备数量较少，无法使用改进的新特性。

3.2.1.2 PCF 与 HCCA

2.1.2 小节讲述了 PCF 的工作原理。实际上，在 802.11 协议中，PCF 是一种可选模式，而多数无线网络产品是不支持 PCF 的。PCF 有两个明显的局限。第一，它有可能对 PCF 和 DCF 通信都造成延迟。由于免竞争时段与 DTIM 时段的频度一致，若

使用 PCF 机制的 STA 需要在当前 CFP 即将结束时发送数据帧，其可能只在下一个 CFP 中被轮询。同样，若某 STA 的数据帧需要在 DCF 时段内被发送，但在 CFP 时段内到来的帧必须等到当前 CFP 结束，并成功抢占到信道时才能被发送。这个局限对时延敏感的通信会产生严重影响。

第二，其在共享同一个信道的相邻 BSS 场景表现不佳。若某个 BSS 处于 CFP 中，则其相邻 BSS 需要等待此 CFP 结束后，才能开始自身的 CFP。此外，如果两个 CFP 的总时长超过了任何一个 BSS 的 CFP 重复间隔，那么服务承诺将无法被兑现。PCF 从未被广泛实现，迄今为止其即使有所应用，这些应用也非常有限。实践证明，DCF 机制要简单得多，而且在很多情况下更为可靠，强健性也更好。

802.11e 对 PCF 机制进行了优化，引入了 TXOP 机制，升级为 HCCA (HCF Controlled Channel Access)。HCCA 机制和 EDCA 机制一样，需要 AP 和 STA 同时支持，只要有一方不支持，就只能使用 PCF 机制。

3.2.2 802.11ah

1.2.2 小节中已经介绍了 802.11ah 的设计目标以及与 Wi-Fi 6 的关系，它是面向“大连接”低功耗场景而设计的协议。802.11ah 为了降低大规模终端设备在接入信道时产生冲突的概率，避免“接入风暴”的发生，引入了新的信道接入机制 RAW (Restricted Access Window with EDCA)。

RAW 采取分组接入的思想，限制了同时竞争信道的 STA 的数量。具体来说，使用 RAW，AP 将所有 STA 分成若干组，定义一组称为 RAW slot 的时间间隔，并将每个组分配给一个 RAW slot。通过限制同时参与信道竞争的节点个数，这种分组的方法能有效降低节点间冲突概率，提高网络承载能力。除了降低冲突概率之外，这种方法还降低了能耗，因为除了分配的时间间隔之外，STA 几乎总是可以休眠。

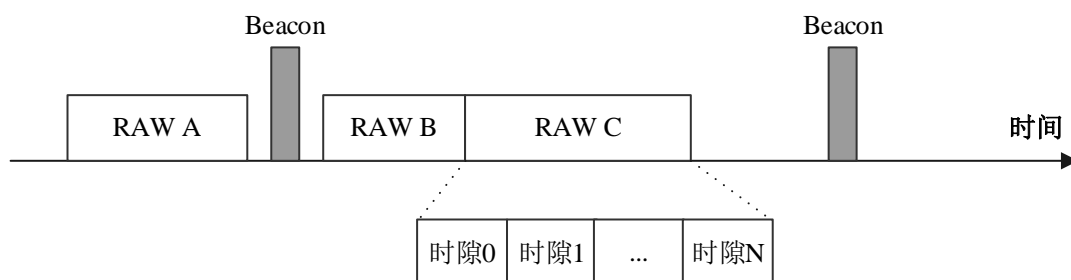


图 3-3 RAW 工作机制

标准 RAW 机制中, AP 负责把局域网中的所有节点平均分配到一定数量的 RAW 组中, 在每个 RAW 组内的节点通过自身 AID 映射到指定的时隙中。RAW 工作机制如图 3-3 所示, 这种机制和传统 802.11 协议中的 DCF 机制相比, 缓解了大量节点同时争用信道引发的冲突问题。但是, 标准 RAW 机制的分组规则在智能家居这种设备间流量特征差别很大的系统中存在极大的不合理性, RAW 机制只是按照节点的数目平均分配信道资源, 而且分配过程存在很大的随机性, 并没有把网络流量状况和节点间的业务流量差异考虑在内, 使得资源分配不合理。

举例来讲, 某些 RAW 分组流量较小, 在完成传输任务后信道处于空闲状态, 其他 RAW 组却不允许使用这个空闲状态的信道; 另外, 存在 RAW 分组的流量较大, 传输任务在所分配的 RAW 持续时间呢内无法完成。在智能家居场景中, 节点种类繁多, 不同类型的节点传输参数不同, 如 MCS 方式、发包间隔、包大小等, 导致每个节点的业务流量存在较大差异, 此时, 如果按照标准 RAW 机制分组容易造成各组间负载不平衡的情况, 极端情况下, 数据传输性能甚至不如传统 802.11 协议的 DCF 机制。

802.11ah 从协议设计上而言, 确实是对 802.11 协议框架做了一次非常不错的灵活应用。不过目前 IoT 市场的主导还是感觉是低成本, 而 802.11ah 一开始的起始点实际上是 802.11ac, 即其是在 802.11ac 基础上进行设计的, 所以相对成本还是高一些, 导致其在 IoT 的市场上, 并没有引起很大的波澜。

3.2.3 需求分析

随着家居智能化进程的推进, 家庭中通过 WLAN 接入互联网的设备数量急剧增加, 一个家庭中存在 30 个甚至 50 个终端设备, 传统的无线个人局域网在应用中显

得相形见绌。而现存的大多数智能家居设备基于成本考虑都仅支持 Wi-Fi 4 甚至 Wi-Fi 3, 抢占信道的机制大多都是传统 DCF 机制, 而且这些设备中大部分更换频率低, 而且对现有设备其网卡升级的难度以及成本都不令人满意。原始 DCF 机制随着设备数量的增加在抢占信道时的冲突率会大大提升, 会导致信道利用率, 系统吞吐量急剧下降, 严重影响智能家居的用户体验, 同时也阻碍了家居智能化进程推进。

针对大容量场景, 802.11 协议组提出了 802.11ah 协议, 尤其是其中的 RAW 机制, 和传统 DCF 机制相比, 缓解了大量终端设备同时争用信道引发的冲突问题, 在某种程度上提升了系统吞吐量, 降低了由退避过程带来的时延和能耗。但是, 标准 RAW 机制的分组规则在智能家居这种设备间流量特征差别很大的系统中存在极大的不合理性, RAW 机制只是按照终端的数目平均分配信道资源, 而且分配过程存在很大的随机性, 并没有把网络流量状况和终端间的业务流量差异考虑在内, 使得资源分配不合理。而且支持 802.11ah 协议的设备基本没有。

本文在 RAW 机制的思想基础之上, 对传统 DCF 机制进行了改进, 提出了全新的, 流量均衡的受控并行随机接入机制。缓解了大量终端设备同时争用信道引发的冲突问题, 提升了系统吞吐量, 减少了由退避过程带来的时延和能耗, 以及异构流量模式引发的信道接入的不公平问题。

3.3 Wi-Fi 智能家居场景系统性能

表 3-2 数值分析及仿真参数设置

参数	取值
数据帧速率	54Mbps
控制帧速率	24Mbps
传播时延	2 μ s
时隙时长	9 μ s
SIFS	16 μ s
DIFS	34 μ s
最小竞争窗口	15

为了验证智能家居场景下 Wi-Fi 的性能，在 NS-3 平台上进行了仿真，在单个路由器场景下，分别使用 STA 个数为 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 等几种不同规模的网络，仿真参数见表 3-2，统计了饱和和非饱和流量场景下，系统吞吐量随 STA 数量变化的情况，如图 3-4 所示，误帧率随 STA 数量变化情况，如图 3-5。

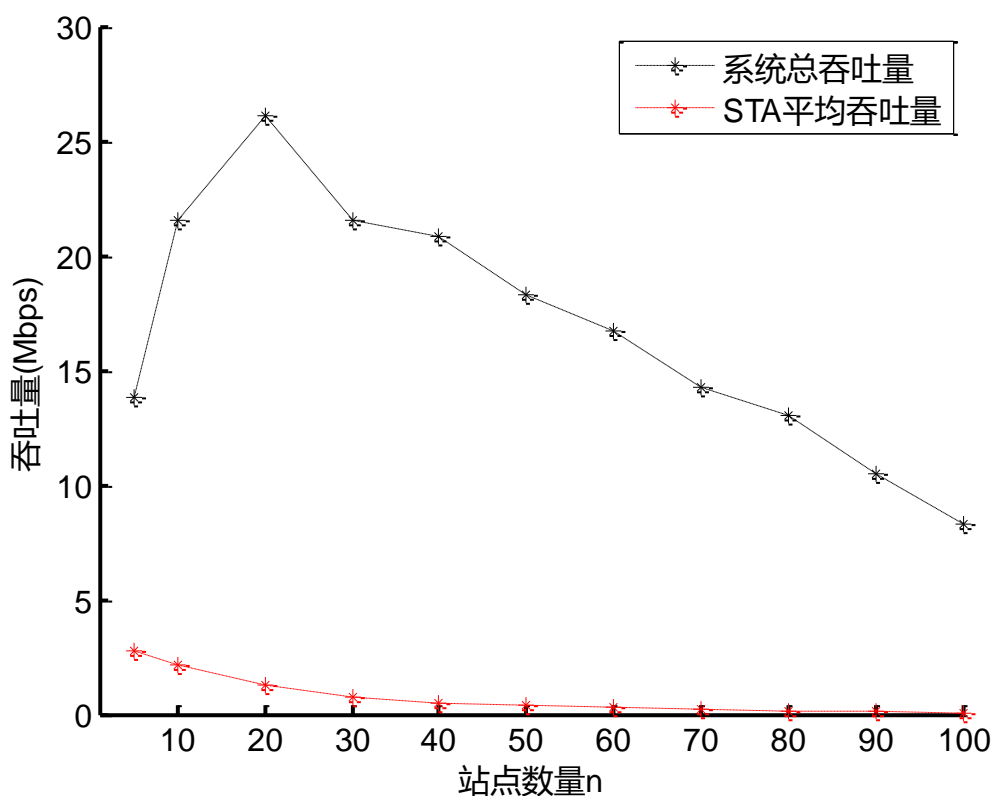


图 3-4 非饱和负载下系统平均吞吐量

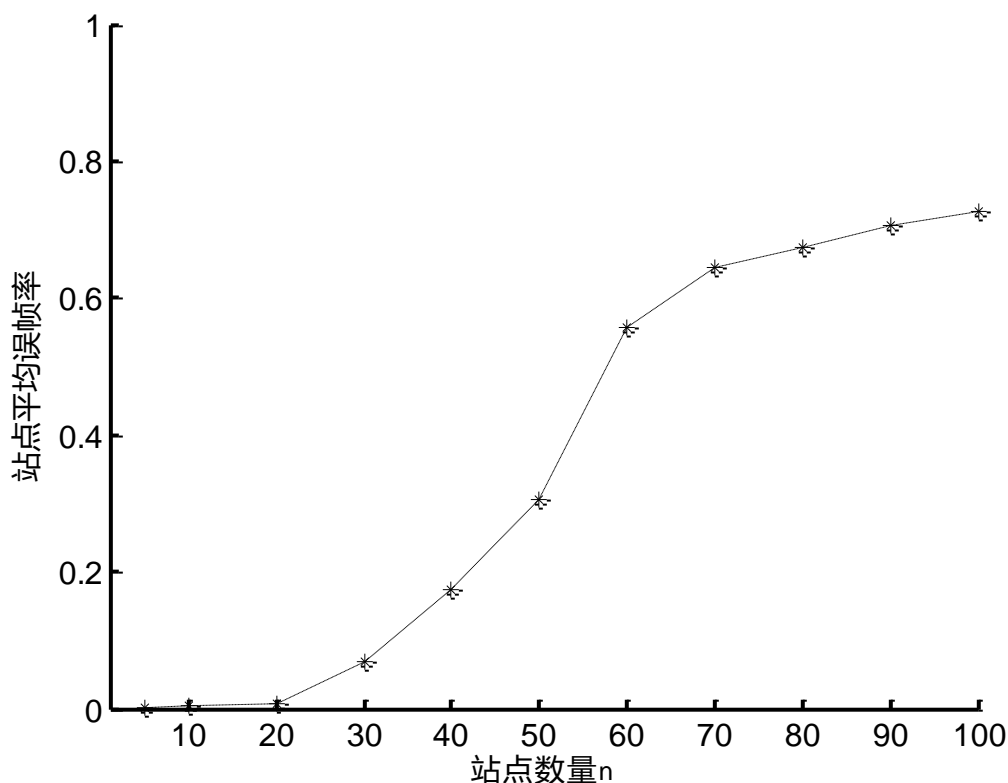


图 3-5 非饱和流量下 STA 平均误帧率

由图 3-4 和图 3-5 可知，各 STA 的产生数据帧的模型非饱和的，在 STA 数量较小时，整个家庭局域网系统处于非繁忙状态，随着 STA 数量的增大，系统吞吐量越来越大，STA 的平均误帧率只是微微变大基本不变。但是随着 STA 增多，局域网系统逐渐进入繁忙状态，逐渐变为满负载运转，后续随着 STA 数量的增大，系统吞吐量不增反减，误帧率快速增大，按照这个趋势，而 STA 平均吞吐量一直呈下降趋势，按照这个趋势 STA 的平均吞吐量将趋紧于 0。

3.4 流量均衡的受控并行随机接入机制系统模型

在建立系统模型之前，首先解释一下其中的几个关键字：

流量均衡，在本方案中认为，流量特征是影响接入的一个重要参数，每个周期内都要统计和预测下一个周期每个终端设备的流量特征，该特征是对终端设备进行分组的重要依据；

受控，传统 802.11 协议使用 DCF 机制调节多个终端同时接入信道时的冲突问题，在面对大量终端接入信道的场景，使用“分而治之”的思想，某个时间段内只允许部分终端设备抢占信道资源，所以称之为“受控”；

并行，在一个分组周期内，原本某个时间段内产生碰撞的多个结点，在本周期内都无法完成数据的传输。但是划分到不同的时间段后，可以在一个分组周期内完成所有设备的数据传输，故称之为“并行”；

随机接入，在一个周期中，划分在的某个时间段内的所有设备，使用 DCF 机制抢占信道，故称之为“随机接入”。

上文已经对流量均衡的受控并行随机接入机制的关键字的含义做了概述，接下来对该机制工作原理进行更详细的描述，首先在 3.4 节中建立了该机制的系统模型，在 3.5 节中对改进 DCF 机制建立了全新的 Markov 模型。在 3.6 节中，对改进后的 DCF 机制的冲突率和吞吐量进行了理论推导，并与原始 DCF 机制进行了对比分析。在 3.7 节中，使用 NS-3 对改进后的 DCF 机制进行了仿真，验证了改进 DCF 机制在应对智能家居场景时相对于原始 DCF 机制的性能提升，同时也证实了 3.5 节中建立的 Markov 模型的准确性。

3.4.1 系统模型

现在假设有一个基于 Wi-Fi 网络的智能家居场景，终端设备集合 S 与路由器 AP 连接， S 中共有 N 个终端设备，若干个类型。假设现在仅考虑智能家居设备的上行链路流量，即家居设备向路由器发送数据。路由器作为家庭网络的管理节点，位于智能家居网络系统的核心位置。终端设备 $s \in S$ 以泊松分布产生业务数据，经路由器汇聚转发到后端服务器。不同类型的终端设备有不同的业务负载，具有不同的业务数据包到达率 ν_s 、数据包大小 L_s 和调制编码方式 MCS，流量均衡的受控并行随机接入机制的网络拓扑图及接入规则如图 3-6 所示。

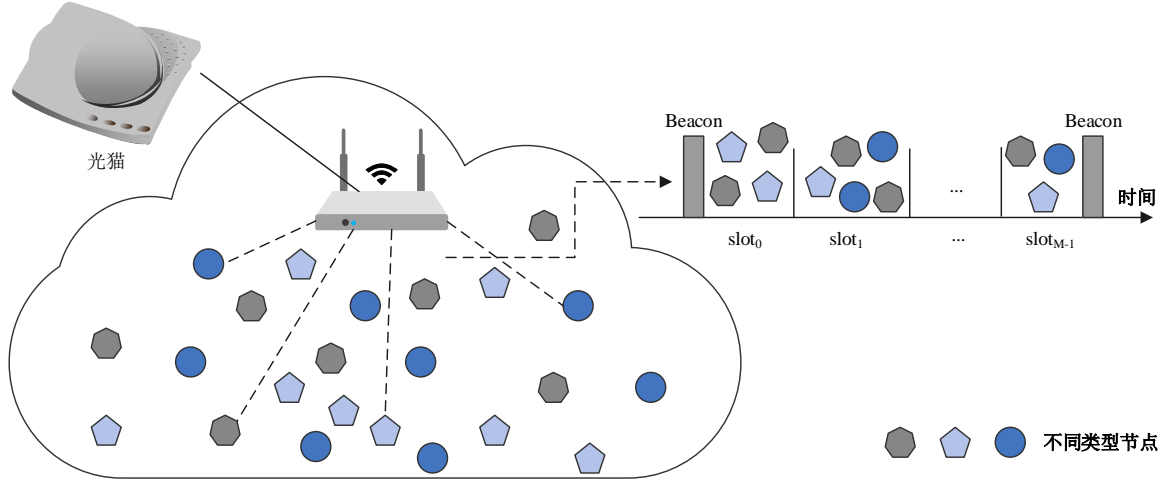


图 3-6 受控接入机制的网络拓扑图及接入规则

随着 N 的增大, 传统 DCF 机制中, N 个设备在信道接入时产生碰撞的概率会急剧上升, 极端情况下, 某个时间段内产生碰撞的设备带宽几乎为零, 不仅降低了系统总吞吐量, 而且严重影响了发生碰撞设备的业务传输的可靠性, 在设备流量模式不同的局域网内, 这种现象更为严重。本章提出的流量均衡的受控并行随机接入本质上是对 DCF 机制的改进。针对上述问题, 采用“分而治之”的思想。

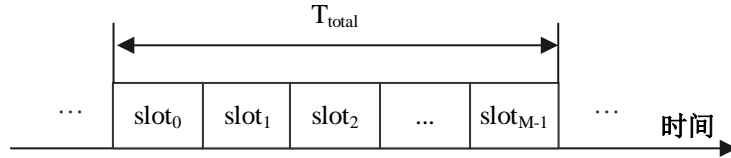


图 3-7 时隙划分

首先把一个分组周期 T_{total} , 划分成 M 个小时隙, 如图 3-7, 然后依据设备的流量大小等属性把 N 个设备分为 M 个组, 表示为 $G_M\{g_0, g_1, g_2, \dots, g_{M-1}\}$, 每个组中的设备流量之和尽可能均衡, 每个组分配一个固定时隙, 每个分组中设备数量为 n_i 个, 满足式(3-3); M 个时隙表示为 $Slot_M\{slot_0, slot_1, slot_2, \dots, slot_{M-1}\}$, 且每个设备只能被分配到一个时隙 $slot_i$ 中, 其中 M 个时隙间隔总和为一个分配周期 T_{total} , 表示为式(3-4)。

$$\sum_{i=0}^{M-1} n_i = N \quad (3-3)$$

其中 n_i 表示第 i 组终端的数量。

$$\sum_{i=0}^{M-1} slot_i = T_{total} \quad (3-4)$$

分配结果见图 3-8

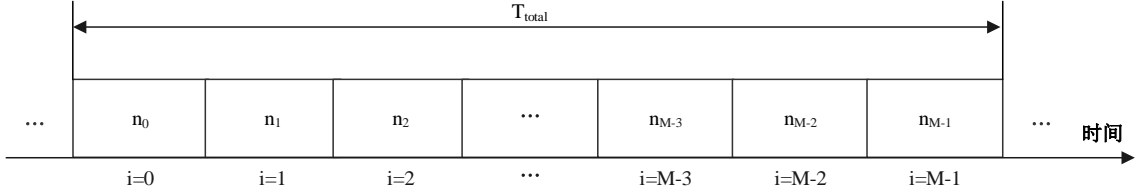


图 3-8 STA 分组效果图

分组规则数学表示见式(3-5)和式(3-6)，在每个时隙中，同一组的设备通过 DCF 机制竞争信道资源。分组由辅助节点完成，并通过私有广播帧告知已关联路由器的终端设备分组信息，设备在其所属时隙内会“苏醒”，上传缓存中的业务数据，其他时隙内设备“休眠”，将到来的数据帧缓存在队列中。

$$X_{i,j} = \{0,1\} \quad (3-5)$$

$$\sum_{j=1}^K X_{i,j} = 1, \forall i \in s \quad (3-6)$$

其中， $X_{i,j}$ 是一个二进制指标变量，它确定终端设备 i 是否加入第 j 个分组，如果终端设备 i 加入小组 j 则它为 1，否则为 0，式(3-6)限制每个设备必须加入一个时隙，并且只能加入一个时隙。

上述内容是对 DCF 机制改进的受控并行随机接入系统模型的概述。至于如何分析评判系统性能，本文将在一个分组周期 T_{total} 内的平均总吞吐量 S_{total} ，作为衡量作为所提出的受控并行随机接入机制的网络性能评估指标，表示为分组周期与吞吐率的乘积：

$$S_{total} = S \times T_{total} \quad (3-7)$$

式(3-7)中， S 表示系统吞吐率。 S 由式(3-8)给出：

$$\begin{aligned} S &= \frac{E[\text{payload information transmitted in a slot time}]}{E[\text{length of a slot time}]} \\ &= \frac{P_S P_{tr} E[P]}{(1 - P_{tr})\sigma + P_{tr} P_S T_S + P_{tr} (1 - P_S) T_c} \end{aligned} \quad (3-8)$$

其中, P_{tr} 表示在任意一个时隙中, 最少有一个站点发送数据帧的概率, 见式(3-10)。 P_s 表示传输成功的概率, 见式(3-11); T_c 表示一次成功传输所需要的时间, T_s 表示一次冲突所浪费的时间; $E[P]$ 表示平均数据帧长度; σ 表示一个时隙中信道空闲所持续的时间。当采用基本接入方式是有:

$$\begin{cases} T_s = DIFS + H + E[P] + \delta + SIFS + ACK + \delta \\ T_c = DIFS + H + E[P] + \delta + SIFS + ACK \end{cases} \quad (3-9)$$

其中 δ 表示传播时延。其中的相关参数都 802.11 协议 MAC 参数中得到。

$$P_{tr} = 1 - (1 - \tau)^n \quad (3-10)$$

$$P_s = \frac{n\tau(1 - \tau)^{n-1}}{1 - (1 - \tau)^n} = \frac{n\tau(1 - \tau)^{n-1}}{P_{tr}} \quad (3-11)$$

其中 n 表示智能家居网络中终端个数, τ 可通过式(3-15)和式(3-16)通过数值解法求得。

3.5 改进 Markov 模型

研究表明 2.3.2 中的 Bianchi 模型比以往的模型描述的更加准确, 但是比较简化, 没有考虑退避计数器冻结的情况, 没有重传次数限制, 属于饱和流量模型, 不满足智能家居业务的实际情况。为了更加符合智能家居业务特征, 本文对 Bianchi 模型做了改进。考虑了缓冲队列里没有数据帧要发送的情况, 并且考虑了退避计数器冻结的情况和重传次数限制。

为方便建模, 做以下模型假设:

- 1) 每个终端设备 STA 的数据包到达率都服从泊松分布, 包到达率为 λ , 缓存队列大小为 K ;
- 2) 局域网内的每个 STA 都可以“听”到其余的所有 STA;
- 3) 每个 STA 发送数据包的碰撞概率, 与该 STA 的发送历史无关;
- 4) 信道为理想信道, 不会在传输过程中产生错包。

用 $s(t)$ 表示在 t 时刻给定 STA 的退避阶数的随机过程, t 和 $(t+1)$ 都是离散整数时间点, 分别代表两个相邻的退避时隙的起始时刻, 最大退避阶数为 R , $b(t)$ 表示在

t 时刻给定 STA 的退避计数器 Bcnt 值的随机过程, Bcnt 值在每个退避时隙开始时递减, 当检测到信道忙时则冻结, 不再变化, 其中, 最大重传次数限制为 $(R+f)$, p 表示碰撞概率, q 表示缓存中至少有一个数据帧要发送的概率, 第 i 阶退避窗口表示为 W_i 。

I_e 表示缓存队列为空的情况, 表示非饱和状态下的等待状态。它被包含在马尔可夫链中, 以便对不同的网络负载条件进行建模。在非饱和状态下, 终端设备必须等待帧到达其传输队列。因此, 它保持在 I_e 状态的概率为 $(1-q)$ 。在饱和状态下, 一个终端设备总是有一个帧用于传输, 因此, 它立即选择一个新的退避值, 不存在 I_e 状态。

$$W_i = \begin{cases} W_0 = CW_{\min}, i = 0 \\ 2^i W_0, i \in [1, R] \\ W_R = 2^R W_0 = CW_{\max}, i \in [R+1, R+f] \end{cases} \quad (3-12)$$

此时将 $\{s(t), b(t)\}$ 建模为如图 3-9 所示的二维离散时间马尔科夫链。

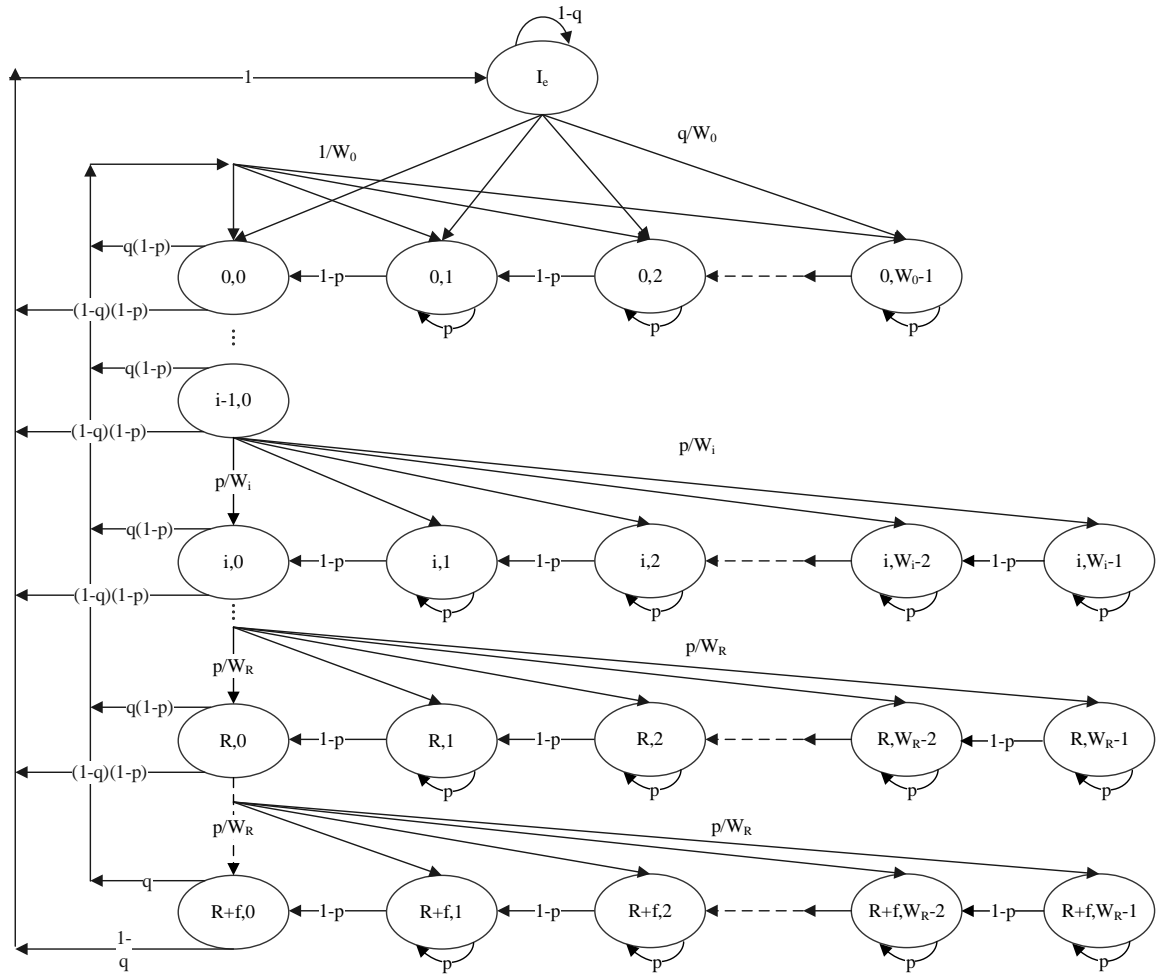


图 3-9 改进 Markov 模型的状态转换图

Markov 状态转换图中，水平方向上，从右向左的每一步状态转移表示 STA 的退避计数器值减 1，而垂直方向上从上到下，每一步状态转移代表 STA 在空口上发生了一次碰撞。

其中各状态的非零一步转移概率如式(3-13)所示

$$\left\{ \begin{array}{l} P(i, k-1 | i, k) = 1-p, i \in [0, R+f], k \in [1, W_i-1] \\ P(i, k | i-1, 0) = \frac{p}{W_i}, i \in [1, R+f], k \in [0, W_i-1] \\ P(0, k | i, 0) = \frac{q(1-p)}{W_0}, i \in [0, R+f-1], k \in [0, W_0-1] \\ P(0, k | R+f, 0) = q, k \in [0, W_0-1] \\ P(I_e | i, 0) = (1-p)(1-q), i \in [0, R+f-1] \\ P(I_e | R+f, 0) = (1-q) \\ P(0, k | I_e) = \frac{q}{W_0}, k \in [0, W_0-1] \end{array} \right. \quad (3-13)$$

式(3-13)中每个一步转移概率表示的含义如下。

其中（1）式表示，在退避时隙范围内，如果检测到信道始终是空闲的，则退避计数器减 1；（2）式表示在 $(i-1)$ 阶退避中，退避计数器减 0 之后，尝试发送数据帧时，发生了碰撞，接下来进行第 i 阶退避，其退避窗口也增大到了 W_i ，取值见式(3-12)，如果达到最大窗口，即使退避阶数增大，窗口也不再变大，维持为 CW_{max} ；（3）式表示在第 i 阶退避中，当退避计数器减为 0 时，缓存中至少存在一个数据包，而且发送成功，退避阶数归零，退避窗口减小为 W_0 ；（4）式表示在最大阶数退避中，缓存队列中至少有一个数据包，无论发送成功还是失败，退避阶数都要归零；（5）式表示在除了最大退避阶数外的任意阶退避中，在退避计数器归零时，如果缓存中没有数据包，则状态切换为 I_e ；（6）式表示在最大退避阶数，即达到了最大重传次数限制时，退避计数器归零时，如果缓存中没有一个数据包，则状态切换为 I_e ；（7）式表示处于 I_e 状态时，如果缓存队列里来了一个数据包，则进入第 0 阶退避，此时退避窗口大小为 W_0 。

按照 [章节 2.3.2](#) 的解法，可以得到改进 Markov 模型的封闭解：

$$b_{0,0} = \left[\frac{1-p}{q} + \frac{1}{2(1-p)} \left(\frac{W_0(1-(2p)^{R+1})}{1-2p} + \frac{1-p^{R+1}}{1-p} + \frac{(2^R W_0 + 1)(p^{R+1} - p^{R+f+1})}{1-p} \right) \right]^{-1} \quad (3-14)$$

则任意一个 STA 传输一个数据包的概率 τ 为

$$\tau = \sum_{i=0}^{R+f} b_{i,0} = \frac{1-p^{R+f+1}}{1-p} b_{0,0} \quad (3-15)$$

每一个时隙是一个冲突域，如果时隙中含有 n 个 STA，则有

$$p = 1 - (1 - \tau)^{n-1} \quad (3-16)$$

3.5.1 站点空闲概率

上述改进 Markov 模型中概率 q 是未知数，它与流量特性和 MAC 缓冲区排队性能有关。流量特性可以通过数据包到达缓冲区尾部的速率来反映，用 λ 来表示，假设到达过程遵循泊松过程，可以利用有限系统的排队理论。因为 STA 的发送缓存队列长度为 K ，并且每个 STA 的包到达和发送过程可以用 M/M/1/K 排队模型来描述。某个 STA 发送完一个数据帧后，其缓存队列中没有数据包要发送的概率为 $(1-q)$ 。由 M/M/1/K 性质计算 $(1-q)$ 得

$$1-q = \frac{1-\rho}{1-\rho^{K+1}} \quad (3-17)$$

其中 $\rho = \lambda D$ 为系统繁忙强度， D 为平均服务时间，实际上表示一个 STA 从一次发送成功到另一次发送成功所需要的时间。

平均服务时间 D 包括三个部分：成功传输数据所需时间 t_s ，发送失败所消耗的时间 t_c ，以及站点退避过程中消耗的时间 t_{backoff} ，即

$$D = t_s + t_c + t_{\text{backoff}} \quad (3-18)$$

式中 $t_s = T_s$ ； $t_c = i * T_c, i \in [0, R+f]$ ， i 表示重传次数； t_{backoff} 包含信道忙时退避冻结时间和空闲退避时间，如公式(3-19)。

$$t_{\text{backoff}} = \sum_{i=0}^{R+f} p^i (1-p) \cdot t_{\text{backoff}(i)} \quad (3-19)$$

其中 $t_{\text{backoff}(i)}$ 表示在第 i 阶退避中，退避所消耗的时间，可以用退避计数器减 1 的时间乘以退避次数求得，由式(3-20)表示

$$t_{\text{backoff}(i)} = t_{b0} \cdot \frac{W_i - 1}{2} \quad (3-20)$$

其中, $(W_i - 1)/2$ 表示第 i 阶退避中, 退避窗口的平均大小。 t_{b0} 表示在一次退避中, 退避计数器减 1 所需要的时间, 包括冻结时间和空闲时间, 可由式(3-21)求得。

$$t_{b0} = \sum_{j=1}^{\infty} (1-p) \cdot p^j \cdot jT_s + \sigma \quad (3-21)$$

其中, $(1-p) \cdot p^j \cdot jT_s$ 表示退避计数器减 1 之前站点冻结 j 次所需时间。给定 λ , 联立公式(3-17)-式(3-21)即可求得 q 。

3.6 受控并行随机接入机制性能理论分析

本小节, 首先通过对智能家居这个异构流量场景中, 任一终端设备 STA 在抢占信道资源传输数据帧时的竞争成功概率进行了分析。然后使用改进 Markov 模型对改进 DCF 机制的受控并行随机接入机制的冲突概率以及系统吞吐量进行了分析, 并与传统 DCF 机制进行了对比。其中每个 STA 数据包的产生和发送过程都可认为是一个 M/M/1/K 的排队模型, 令 $T_{total} = T_{beacon}$ 。本章仿真或数值分析采用 802.11 协议的参数见表 3-2, 后续无特殊说明均为此表中的参数。

3.6.1 智能家居异构流量分析

章节 1.3 中总结了智能家居业务的四个特点, 其中特点 (2) 是智能家居网络中存在许多不同类型的终端设备 STA, 而不同类型的 STA 通常具有特定的流量模式和数据包大小, 因此, 想要得到符合智能家居业务特征的确定的流量模型, 是很困难的。这种数据模型与章节 2.3 中描述的两种模型存在较大的差异, 但是想要为此异构流量场景建立一个通用的碰撞概率分析模型比较困难。虽然两个模型与实际场景并不相符, 但是具有一定参考意义。

文献[x], 应用现有模型来估计两种极端情况 (即无限流量和只有单个数据包) 的争用成功概率, 然后使用这两种情况的估计来找到精确拟合任何流量模式的争用成功概率的回归函数。实际上, 两种极端情况下的流量模型的区别在于每个终端设备的缓存队列里始终存在的数据帧的数量。实际网络场景中, 终端设备既不是单个数据包传输模式, 也不是无线数据包模式, 而是介于两者之间, 与具体场景有关。根据式

(3-11)、式(3-14)、式(3-15)和式(3-16)可以用数值解析的方法，使用 Matlab 画出竞争成功概率与 STA 数量的曲线，如图 3-10，并把异构流量情况与两个极端情况的模型进行了比较。其中所使用的相关参数如表 3-2。

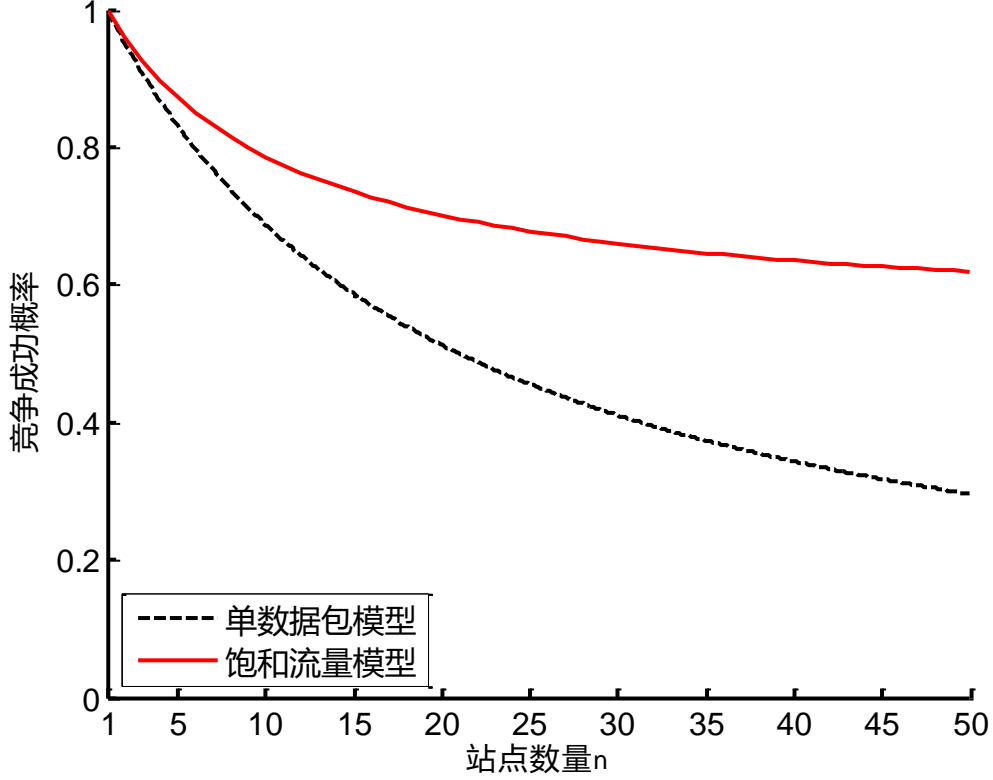


图 3-10 不同流量模型下 CSMA/CA 的竞争成功概率边界

文献[x]研究表明，实际异构流量场景的是处于两个极端模型中间，即单包数据模型是下限，而饱和流量模型是上限，具有相同对数曲线的走势。采用回归模型拟合出了竞争成功概率的近似值，其回归函数如式(3-22)的 3 个式子

$$\begin{cases} P_i^{succ} = \sum_{a \in g_i} \omega_a \times \gamma_{n_i}(N_a) \\ \omega_a = \frac{N_a \times \delta_a}{\sum_{N_j \in \mathbb{N}} N_j \times \delta_j} \\ \gamma_{n_i}(N_a) = b_1 \times \ln(N_a) + b_2 \end{cases} \quad (3-22)$$

其中， $\gamma_{n_i}(N_a)$ 表示第 i 个时隙中站点 a 的竞争成功率， n_i 表示第 i 时隙中包含站点的总数，文献中将竞争成功率拟合为了如式(3-22)中 (1) 式的对数曲线。 ω_a 表

示在一个 Beacon 帧间隔内，第 i 个时隙中包到达数为 N_a 所占权值。 N_a 表示节点 a 在 T_{beacon} 内数据包到达数， \mathbb{N} 表示各节点到达数的集合 ($N_a \in \mathbb{N}$)， δ_a 表示在 T_{beacon} 内数据包到达数为 N_a 的节点数量。(3) 式表示数据帧到达数为 N_a 的节点 a ，在总有 n_i 个站点的第 i 个时隙中传输成功的概率。(2) 式即为在第 i 个时隙中数据帧到达数为 N_a 的站点的数据帧总数占第 i 个时隙中数据帧总数的比例。(1) 式表节点 a 的传输成功率。

由图 3-10 可知对于任何给定数量的设备节点，更高的流量需求会导致更高的争用成功概率，这实际上是非常违反直觉的。根本原因是当网络拥塞时，指数退避增加了竞争窗口，因此，初始数据包比后面的数据包经历更严重的拥塞。具体来说，如果所有设备节点只发送一个数据包，则这些数据包的竞争失败概率将相当高，因为初始竞争窗口很小。然而，如果设备节点传送多个数据包，一些数据包将使用更大的竞争窗口重新传输，而一些数据包是使用初始小竞争窗口 CW_0 进行传输。

另外，无论是什么流量模式的网络状况，随着设备节点数的增多，竞争信道资源的成功率会不断减少，当设备节点数量达到一定的阈值时，成功率会非常低。相反，冲突概率会不断上升，各节点不断执行二进制退避算法，竞争窗口不断扩大，大量数据帧因为超过重传次数上限而被丢弃，信道利用率变低，系统吞吐量不断下降。

3.6.2 冲突概率分析

章节 3.6.1 说明了，对于异构流量模型来说，虽然两个极端模型相比，竞争成功概率曲线有所差异，但是走势是一模一样的。本章节以改进 Markov 模型对改进 DCF 机制的受控并行随机接入机制的冲突概率进行了分析，并且推到了改进前后的冲突概率之间的关系。

给定 R , f , W_0 , n ，根据式(3-14)、式(3-15)和式(3-16)，即可求出 p 的数值解，图 3-11 为使用 Matlab 进行数值仿真画出 p 与站点数 n 的关系图。

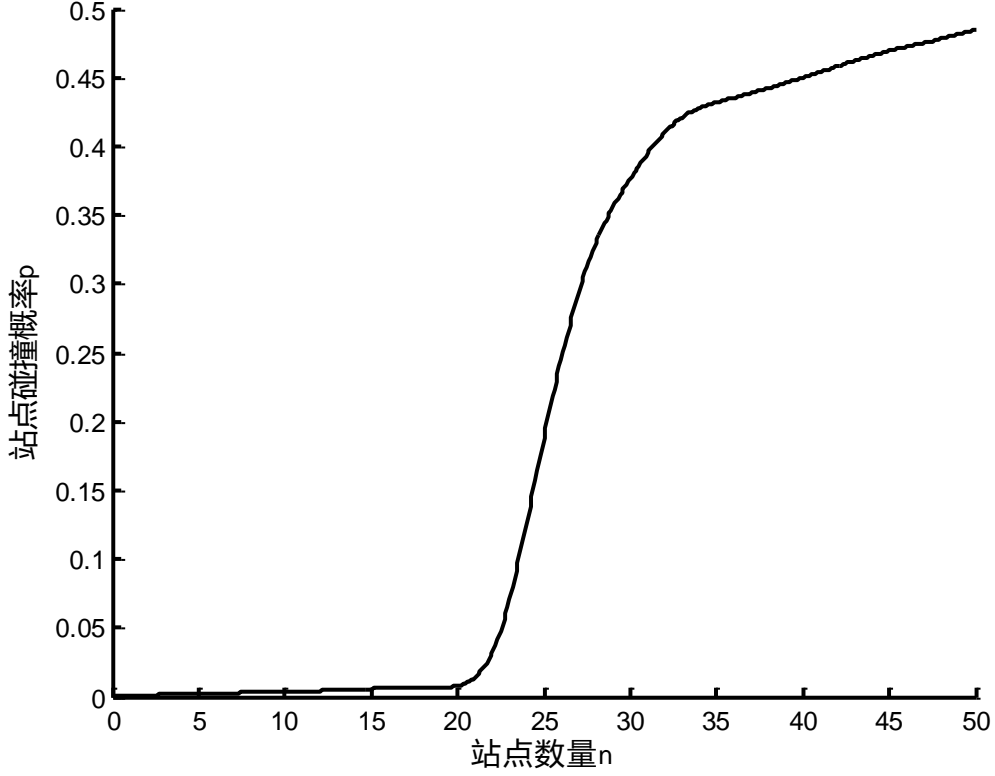


图 3-11 流量不饱和条件时不同站点数下 p 的值

由图 3-11 可知，在流量不饱和条件下，站点数较少时，STA 的在发送数据帧时发生碰撞的概率较低，虽然概率变大，但增幅比较缓慢。但是当 n 超过某个阈值时，发生碰撞的概率会快速上升。令 p 与 n 的函数关系如式(3-23)。

$$p = g(n) \quad (3-23)$$

在分析改进 DCF 机制的冲突概率之前，首先定义衡量冲突概率的指标，为一个分组周期内的平均冲突概率。在原始 DCF 机制中，由 Bianchi 模型可知， n 个 STA 在每个退避时隙内的冲突概率是一致的， n 个 STA 在一个周期内平均冲突概率等于一个退避时隙内的冲突概率，如式(3-24)

$$p_1 = 1 - (1 - \tau)^{n-1} \quad (3-24)$$

现在计算改进后 DCF 机制的一个分组周期内的平均冲突概率，首先通过图 3-8 可知分组后的情况，分组后每个组内仍然使用 DCF 机制进行信道抢占，则每个组内的冲突概率仍然满足 Markov 模型，则改进后机制，在一个周期内的平均冲突概率如

式(3-25)

$$p_2 = \sum_{i=0}^{M-1} (1 - (1 - \tau_i)^{n_i-1}) \quad (3-25)$$

其中 n_i 表示第 i 个时隙中节点的个数。每个分组中的节点数相同，即，满足式(3-26)

$$n_i = \frac{n}{M}, i \in [0, M-1] \quad (3-26)$$

即在 M 已知时， n_i 都是常数， n_i 一定时， τ 也是定值。将式(3-26)代入式(3-25)得

$$\begin{aligned} p_2 &= \frac{1}{M} \times \sum_{i=0}^{M-1} (1 - (1 - \tau)^{n_i-1}) = \frac{1}{M} \times \sum_{i=0}^{M-1} (1 - (1 - \tau)^{\frac{n}{M}-1}) \\ &= \frac{1}{M} \times M \times (1 - (1 - \tau)^{\frac{n}{M}-1}) = (1 - (1 - \tau)^{\frac{n}{M}-1}) \end{aligned} \quad (3-27)$$

式(3-23)、(3-24)和(3-27)联立可得

$$p_2 = g\left(\frac{n}{M}\right) \quad (3-28)$$

p_1 和 p_2 的对比，如图 3-12 所示，

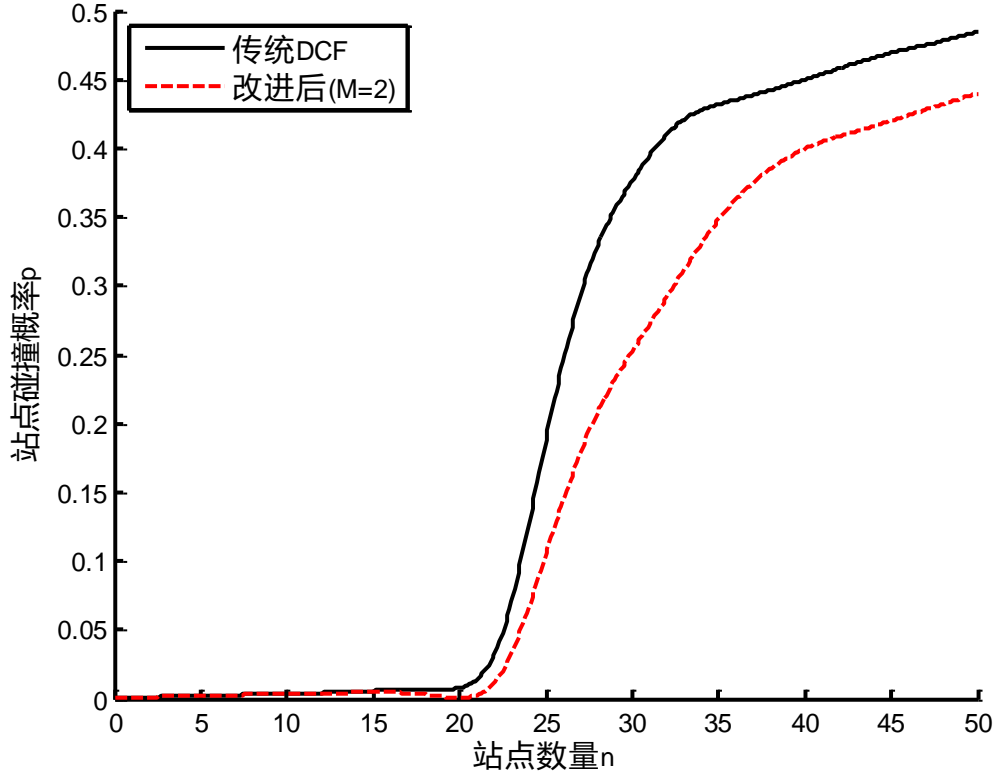


图 3-12 改进后 DCF 机制和 DCF 机制冲突概率对比

由图 3-12 可知，当总 STA 数量 n 固定，每一个时隙中的 STA 数量 n_i 一致时，一个分组周期内的平均冲突概率，改进后的 DCF 机制始终比原始 DCF 机制的小，尤其是 STA 数量较大时，改进 DCF 机制更加明显，分组机制大大降低了 STA 的平均冲突概率。

3.6.3 吞吐量分析

为了更好的进行原始 DCF 和改进后 DCF 机制吞吐量的对比，在[章节 3.4](#)中定义了新的衡量系统吞吐量的指标，一个分组周期内的平均吞吐量，见式(3-7)

当物理层参数固定时，根据式(3-8)-式(3-11)和式(3-14)-式(3-16)可以通过 Matlab 数值分析可以描绘出原始 DCF 机制非饱和负载情况下，吞吐率和 n 的关系曲线，见图 3-13。

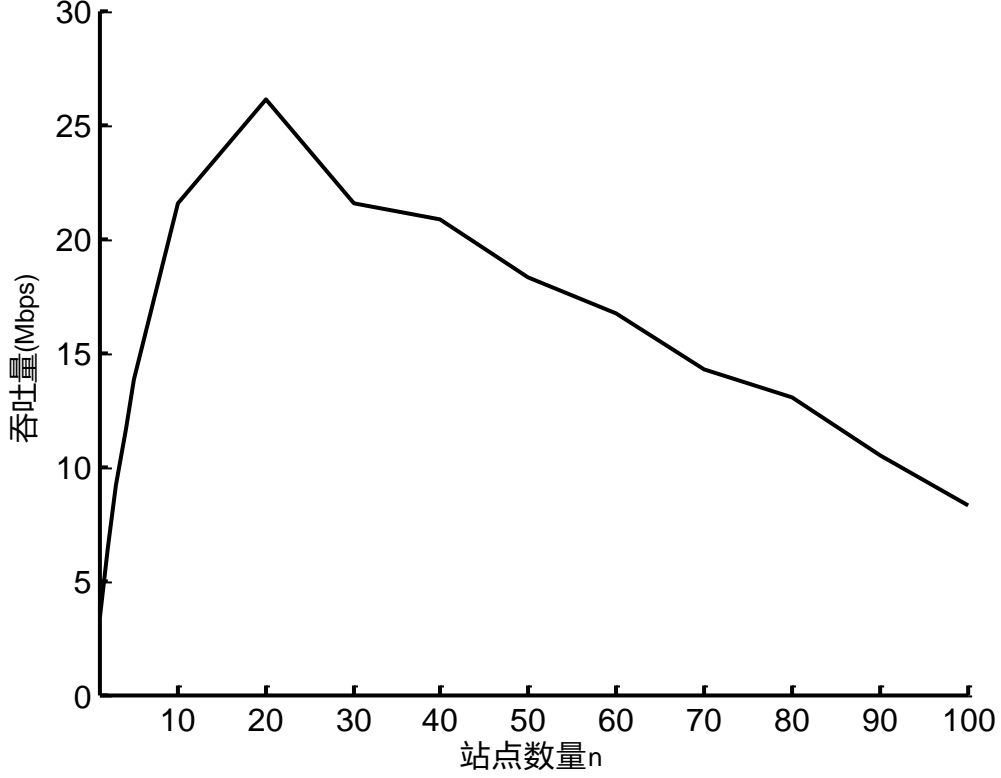


图 3-13 非饱和负载下原始 DCF 机制吞吐量和 n 的关系图

由图 3-13 可知，在 n 较小时，随着系统内竞争站点数 n 的增加，系统吞吐量逐渐变大。当 n 大于某个阈值时，STA 的吞吐量 S 逐渐降低， S 与 n 属于一对一的映射关系。并且其吞吐率只与物理层参数、初始竞争窗口 W_0 ，最大竞争窗口 W_m 有关，令 S 与 n 的函数关系如式(3-29)。

$$S = h(n) \quad (3-29)$$

由 Bianchi 模型分析可知，对于原始 DCF 机制，当 STA 数量固定时，在每一个退避时隙中，吞吐率 S 是确定的，所以，在一个分组周期 T_{total} 内的总的平均吞吐量可以表示为式(3-30)

$$S_{total} = S \times T_{total} = h(n) \times T_{total} \quad (3-30)$$

改进后的 DCF 机制，在一个分组周期内的总吞吐量 S'_{total} ，为每个小时隙的吞吐量之和，即

$$S'_{total} = \sum_{i=0}^{M-1} (h(n_i) \times T_i) \quad (3-31)$$

为了简化分析过程，假设 n_i 和 T_i 满足式(3-32)

$$\begin{cases} T_i = \frac{T_{total}}{M} \\ n_i = \frac{N}{M} \end{cases}, i \in [0, M-1] \quad (3-32)$$

式(3-31)和(3-32)联立得

$$\begin{aligned} S'_{total} &= \sum_{i=0}^{M-1} (h(n_i) \times T_i) = \sum_{i=0}^{M-1} (h(\frac{n}{M}) \times \frac{T_{total}}{M}) \\ &= M \times (h(\frac{n}{M}) \times \frac{T_{total}}{M}) = h(\frac{n}{M}) \times T_{total} \end{aligned} \quad (3-33)$$

由式(3-33)，可以知晓改进后与改进前吞吐量的关系，如图 3-14 为非饱和负载下，改进前后吞吐量的对比。

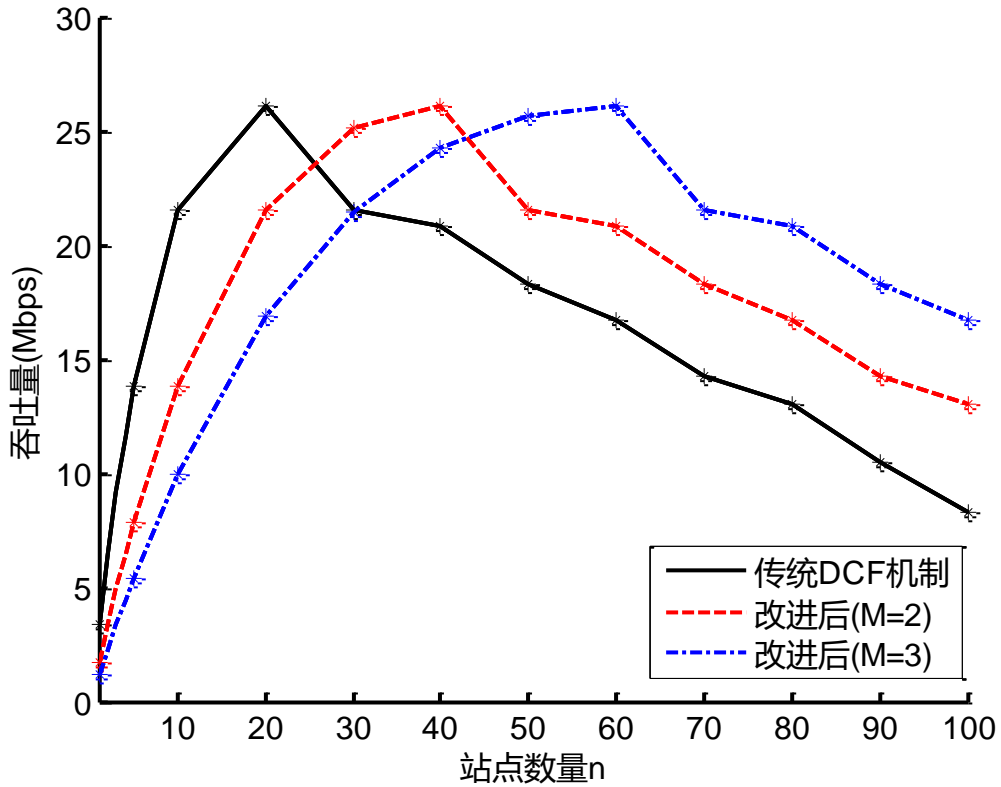


图 3-14 非饱和负载下改进前后吞吐量对比

图 3-14 中散点为理论值，线条仅为了表示趋势而连接起来。由图 3-14 可知，改进后的系统吞吐量，在站点数量比较少的时候，改进后的系统比原始 DCF 机制的吞吐量略低，这是因为此时系统处于非饱和状态，划分了时隙反而降低了信道利用率。但是随着站点数量的增加，改进后的系统吞吐量也会在达到最大值后平缓下降，但是始终比原始系统吞吐量高。因此改进后的系统达到了吞吐量提升的效果，尤其是在站点数量较大时优势越明显。另外，关于分组个数 M ，并不是分组越多越好，分组个数应该和系统繁忙程度相关，系统越繁忙， M 设置的越大，系统吞吐量提升越明显。

3.7 受控并行随机接入机制的仿真与分析（需更新）

3.7.1 仿真平台搭建

仿真平台是在国内某知名企业计算云上搭建的，Linux 内核版本为 4.14，NS-3 版本为 3.30。通过在 AP 和 STA 两端的 MAC 层添加自定义模块，实现了受控并行随机接入机制的逻辑。

其中 AP 侧，主要负责收集 STA 的属性信息，并通过属性信息对 STA 进行分组，然后把分组信息周期性的广播给当前关联的所有 STA，AP 在抢占信道时仅使用传统 DCF 机制。

STA 负责向路由器发送属性信息，同时解析来自 AP 的分组及时隙信息，屏蔽了数据帧的原始发送流程，在所属的时隙中发送数据帧，非所属时隙，有数据到来时，缓存在自定义的队列中，处于同一个分组内的 STA 使用传统 DCF 机制抢占信道资源，关闭 RTS/CTS 机制。

仿真实现中，在 MAC 上层使用 ON/OFF 模型来模拟 STA 业务不饱和业务场景。STA 在 ON 期间产生数据包，每个数据包设定为固定的长度。令两次产生数据包之间的时间间隔服从指数分布，由泊松分布与指数分布的关系可知，在 ON 期间数据包到达服从泊松分布。

使用上述修改后的模块，创建一个单接入点 AP， n 个 STA 的 802.11 协议拓扑图， n 个 STA 均匀分布在 AP 在覆盖范围内，而且所有 STA 之间无隐藏终端问题，

关闭 RTS/CTS 机制。同样的使用修改前的模块构建与之相同的拓扑图。其中 n 的值依次为 1, 5, 10, 20, 30.....90, 100, 每轮仿真统计系统平均吞吐量。

3.7.2 仿真与结果分析（需更新）

仿真参数见表 3-2。

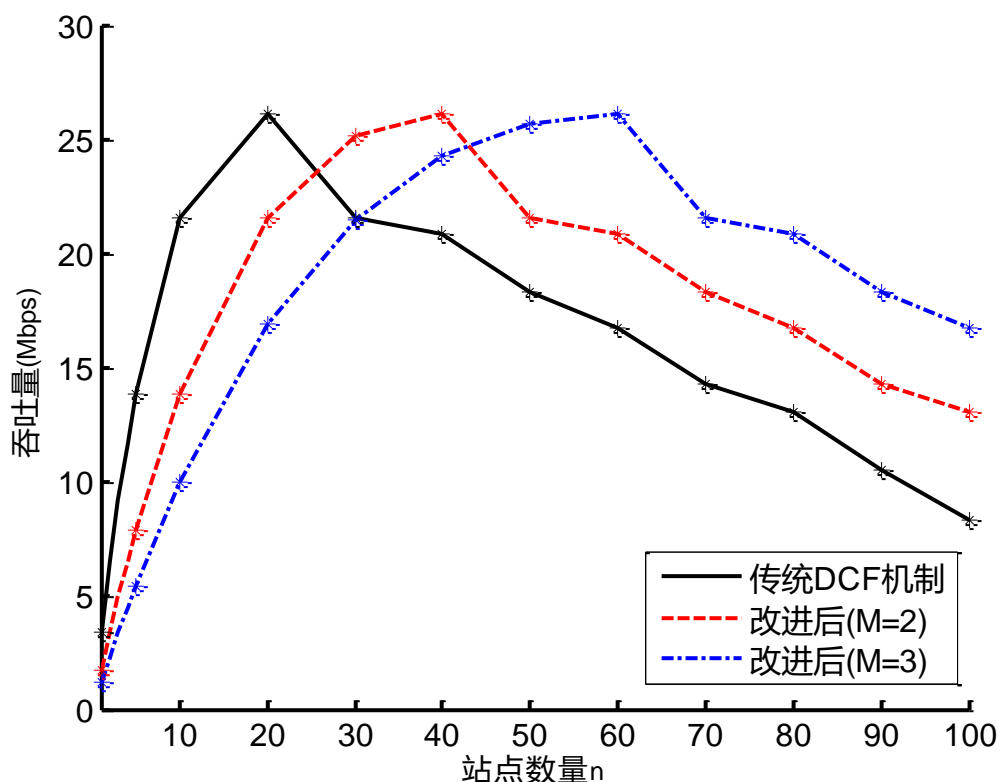


图 3-15 非饱和负载下理论与仿真对比

图 3-15 中的点为 NS-3 仿真实验结果，线条为理论分析结果，由图 3-15 可知，相同条件下，通过改进 Markov 模型推导出的理论吞吐量与实际 NS-3 仿真平台得出的数据基本一致。该结果证实了改进 Markov 模型的准确性，为后续软件实现提供了可靠的理论支撑。

3.8 本章小结（需更新）

本章主要内容是基于传统信道接入机制 DCF，针对智能家居业务传输场景，提出了受控并行随机接入的新的信道接入方案，并通过建模和 NS-3 平台仿真，证明了

本文所提出信道接入方案的可行性。首先分析了 802.11 工作组对信道接入机制中降低冲突概率的改进；然后对现有智能家居场景下的系统性能进行了仿真分析，分析影响系统性能的主要因素，现有 802.11 协议都无法完全适用于该场景；接着提出了全新的流量均衡的受控并行随机接入机制和更为准确描述智能家居业务特征的改进 Markov 模型，使用改进 Markov 模型分析了所提出的新的信道接入机制的冲突概率和吞吐量的理论值，并使用数值分析法与传统 DCF 机制进行了对比；最后使用 NS-3 对新方案进行了仿真分析，证明了理论分析的正确性，证实了改进机制在智能家居这个用户数量大，以及 STA 数量更大的场景下，降低冲突概率，提升吞吐量的优势，为后续软件实现对实际性能提升提供坚实的理论支撑。

4 受控并行随机接入机制的设计与实现

4.1 引言

第三章提出了针对传统 DCF 机制在智能家居场景下信道接入碰撞概率高，吞吐量低下的优化方案，流量均衡的受控并行随机接入机制，对方案的核心思想建立了系统模型，使用改进的 Markov 模型对改进 DCF 机制进行了理论性能分析，并且通过 NS-3 仿真平台证实了改进 Markov 对改进 DCF 机制描述的准确性，证实了受控并行随机接入机制在智能家居这一用户密集场景下对吞吐量的提升的有效性。本章在第三章优化方案基础上，考虑到现有智能家居的特点（4），详细阐述了集中式和分布式两种对路由器透明的具体实现方案，并进行了实际测试，进一步证实了流量均衡的受控并行随机接入机制在智能家居场景下降低信道接入概率，提升吞吐量的优势，并对两种实现方案的性能进行了对比，分析了两种实现方案的优缺点。

4.2 集中式受控并行随机接入机制的方案设计

4.2.1 集中式方案设计概述

章节 3.4 中已经介绍了受控并行随机接入的核心思想：分组，把家庭网络里的大量终端设备划分到不同的时隙中，同一时隙内的设备通过 DCF 机制抢占信道的使用权。集中式整体方案网络拓扑及工作原理如图 4-1 所示

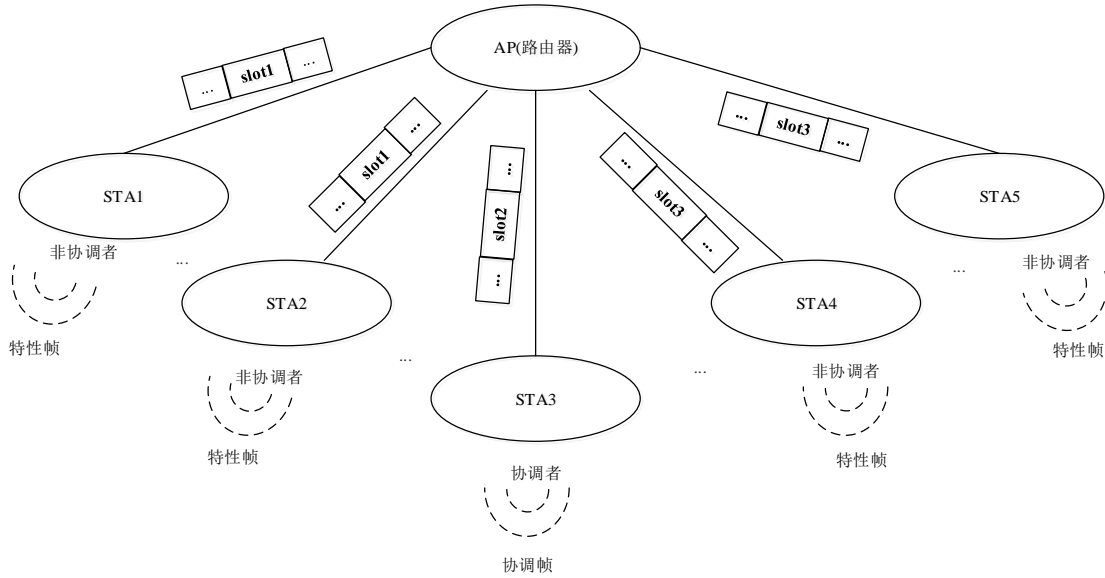


图 4-1 集中式受控并行随机接入整体方案示意图

本案对路由器透明，在此前提下，把当前局域网内的站点用两种身份区分协调者（Leader）和非协调者（Follower），协调者负责收集当前局域网内的所有站点的属性信息，并根据负载均衡分组算法对所有站点进行分组，划分到不同的时隙中，负载均衡分组算法在 4.2.6 节中讲述，协调者通过广播帧把分组信息通知给所有非协调者，非协调者按照协调者的所分配的时隙，进行数据的收发，在其他时隙中休眠，暂时缓存需要发送的数据帧。

站点的协调者和非协调者的身份可以在满足一定条件时切换，具体见 4.2.2 节的 Leader/ Follower 状态机，至于怎么选举协调者，见 4.2.3 节的分布式 Leader 选举。

在此方案中定义共定义了协调帧、特性帧、竞选帧以及竞选响应帧四种私有帧。

其中协调帧是由 Leader 状态的 STA，以 $(2 \times T_{\text{beacon}})/3$ 为周期定时构造发送的，包含当前局域网内所有 STA 的属性和分组信息，属于广播帧；另外三种帧都是单播帧，其中特性帧是非协调者向协调者告知自身属性信息的帧，主要包括终端类型，当前物理发送速率，过去三个周期内各自流量值等；竞选帧和竞选响应帧是在竞选协调者时使用的两个交互帧，发起竞选者会构造发送包括自身属性信息的竞选帧，被竞选者受到竞选帧，决定是否同意发起者成为协调者之后，会构造竞选响应帧发送给竞选发起者。

4.2.2 Leader/Follower 状态机

章节 4.2.1 中介绍了集中式受控并行随机接入的整体实现方案，方案中对站点进行了身份划分：协调者和非协调者。在满足一定条件时，STA 的身份会在两者之间尽心切换，很符合有限状态机编程的思想。

有限状态机（finite state machine）简称 FSM，表示有限个状态及在这些状态之间的转移和动作等行为的数学模型，在计算机领域有着广泛的应用。FSM 是一种逻辑单元内部的一种高效编程方法，在编程中，服务器可以根据不同状态或者消息类型进行相应的处理逻辑，使得程序逻辑清晰易懂，在完成 STA 的身份切换时的逻辑更加清晰，不容易遗漏某些切换条件以及制定状态下要执行的功能。

家庭网络中不同终端的处理器和网卡性能是不同的，协调者一般选择综合性能较强者。随着时间的推移家庭网络内可能存在新终端的加入和终端的离开，如家庭某个成员带着自己的手机和笔记本出门了，访客带着自己的手机和笔记本来拜访等。所以对于规定终端的状态切换准则很有必要。状态转换图如图 4-2 所示

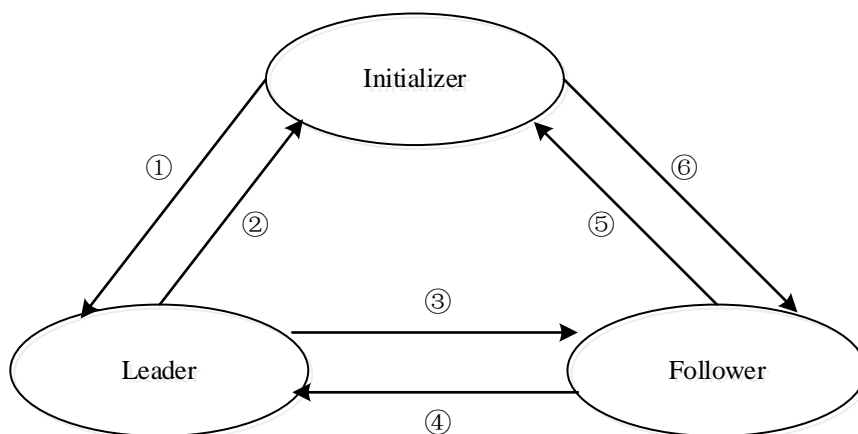


图 4-2 Leader/Follower 状态机的状态转换图

4.2.2.1 状态集

为了更好的进行状态的切换，引入了第三状态：初始状态（Initializer），Leader/Follower 状态机的状态集为 $FSMS = \{Initializer, Leader, Follower\}$ ；其中初始状态为新入网的 STA 处于所处的状态，Leader 状态表示该 STA 是当前局域网的协调者，Follower 状态即非协调者所处的状态。接下来详细阐述状态机的切换原则。

4.2.2.2 状态切换准则（格式调整）

每个状态切换通路的编号，见如图 4-2 所示的状态机的状态转换图。

其中通路 1, 为初始状态到协调者状态切换。新入网的 STA 处于 Initializer 状态。监听 1.5 倍协调帧间隔 T_1 后，若没有收到协调帧，则状态机会从 Initializer 状态切换为 Leader 状态。

通路 2, 为协调者状态到初始状态切换。暂不存在 Leader 到 Initializer 的切换。

通路 3, 为协调者状态到非协调者状态切换。当协调者收到竞选帧后，若协调者同意非协调者的竞选请求，则协调者会给该非协调者回复一个肯定的竞选响应帧。此时协调者的状态从 Leader 变为 Follower。

通路 4, 为非协调者状态到协调者状态切换。当某个非协调者，发现自身的性能比当前的协调者强的时候，会主动向协调者发送竞选帧，如果协调者回复了肯定的竞选响应帧，则该非协调者的状态会从 Follower 成为 Leader。另外，当前协调者突然消失的情况：如果，非协调者 1 倍协调帧间隔内没有收到协调帧，则性能次强的非协调者会成为新的协调者，即状态机状态从 Follower 切换为 Leader，并且立即发送协调帧，宣示主权。

通路 5, 为非协调者状态到初始状态切换。暂不存在 Follower 到 Initializer 的状态切换。

通路 5, 为初始状态到非协调者状态切换。新入网的 STA 处于 Initializer 状态。监听 1.5 倍协调帧间隔后，如果收到了协调帧，这说明当前已经存在一个协调者，则新入网的 STA 状态机切换为 Follower 状态。

4.2.2.3 状态职责（格式调整）

(删除，回见 20220324 群里第一个版本)

4.2.3 基于 Raft 的分布式 Leader 选举

4.2.3.1 领导者丢失问题

章节 4.2.2 介绍了 Leader/Follower 状态机的状态集，状态切换准则以及各个状态的职责。其中有 Leader 状态到 Follower 状态的切换，还有 Follower 状态到 Leader 状

态的切换，其中都需要检测当前领导者是否存在。在领导者发送的广播协调帧中会携带，当前家庭 Wi-Fi 网络中存在的所有终端（STA）的信息，即所有设备都会维护一个涵盖所有 STA 的属性的特性表。某个 STA 退出家庭 Wi-Fi 网络后，领导者会及时更新自身的 STA 特性表，并通知给其他 STA。这样来看思路很清晰，可以应对大多数网络状况。但实际上存在这样一个很严重的问题，领导者退出了怎么办？这其实是一个典型的分布式一致性的问题。

为了解决上述问题，借鉴了服务器集群中常用的解决分布式一致性的问题的方法。对于一个服务器集群来说，多个节点的协同和管理很重要。其中选举一个主节点来确保其他节点的有序运行，以及各节点的数据一致性问题是有必要的。这里的分布式一致性的问题是指，每个集群节点的数据是一致的。分布式选举算法有 Bully 算法、Raft 算法以及 ZAB 算法等，Raft 算法选举速度快，算法复杂度低、易于实现，算法稳定性比 Bully 算法更好，ZAB 算法性能很高，但是选举所需时间较长，不适合智能家居场景，最终借鉴了 Raft 算法。

4.2.3.2 Raft 算法介绍（格式调整）

首先介绍 Raft 算法的原理，特殊说明：介绍算法过程中出现的缩写，英文简称，仅限于此，与上下文无关：

Raft 算法是典型的“少数服从多数”的投票选举算法，它的选举机制与实际生活中的民主投票机制相似。也就是说，Raft 算法中，获得投票最多的节点成为主。

采用 Raft 算法选举，集群节点的角色有 3 种：

- Leader，即主节点，同一时刻只有一个 Leader，负责协调和管理其他节点；
- Candidate，即候选者，每一个节点都可以成为 Candidate，节点在该角色下才可以被选为新的 Leader；
- Follower，Leader 的跟随者，不可以发起选举。

Raft 选举的流程，如图 4-3，可以分为以下几步：

1. 初始化时，所有节点均为 Follower 状态。
2. 开始选举主节点时，所有节点的状态首先由 Follower 转化为 Candidate，然后向其他节点发起选举请求。

3. 其他节点收到选举请求时，会记录选举请求的先后顺序，按照先后顺序，回复是否同意其成为主节点。这里需要注意的是，在每一轮选举中，任意一个节点都只能投出一张票。
4. 如果某个发起选举请求的节点，收到了超过一半的选票，则成为主节点，其状态转化为 Leader，而其他节点的状态会从 Candidate 降为 Follower。Leader 节点与 Follower 节点之间会定期进行心跳包的交互，以知晓主节点是否活着。
5. 当 Leader 节点的任期到了，即发现其他服务器开始新一轮选主周期时，Leader 节点的状态会从 Leader 降级为 Follower，进入新一轮选主

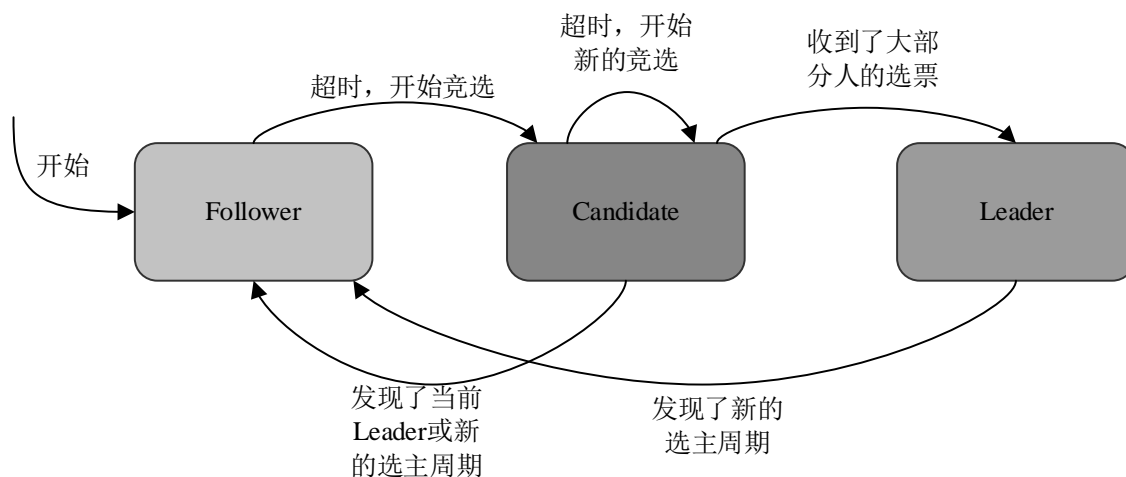


图 4-3 Raft 算法角色状态转换

每一轮选举，每个节点只有一次投票权。这种选举和人大代表选举很相似，一般情况下每一任人大代表都有一定的任期，任期结束后会触发重新选举，且投票者只能把自己在本轮选举中唯一的票投给当前某个候选者。对应到 Raft 算法中，选举主节点是周期进行的，包括选主和任职两个时间段，选主阶段对应投票阶段，任职阶段对应节点成为主之后的任期。但也有例外，如果主节点发生了故障，会立马发起选举，重新选出一个主节点。

从算法的逻辑上来讲很简单，但是所有成员之间需要进行大量的信息交互，对于半双工的 Wi-Fi 信道来讲，是一种很大的负荷，所以本文只是借鉴其算法的思想。接下来介绍在集中式受控并行随机接入的技术方案中，领导者离开的处理方式。

4.2.3.3 分布式选举

章节 4.2.2 介绍了 STA 状态机的工作原理，在家庭 Wi-Fi 网络中，从路由器刚启动时，集中式方案的运行机理，首先来描述一下在最初几个 STA 入网时，各自的状态机是怎么转换的：

首先，第一个终端 STA1 到来时，状态机 state1 处于 initializer 状态，表示为 state1=Initializer，入网首先会检测局域网空口环境中是否存在领导者发送的广播协调帧，如果协调帧周期 T_{leader} 内没有收到协调帧，则 STA1 会认为当前局域网内没有协调者，自己将成为协调者，此时 state1=Leader，并且以周期 T_{leader} 发送广播协调帧；第二个终端 STA2 入网后，在周期 T_{leader} 内收到了 STA1 发送的协调帧，则 STA2 知道了当前局域网内存在领导者，STA2 状态机的状态 state2 从 initializer 变为了 Follower，即此时 state2=Follower，但是 STA2 发现 STA1 所发出的协调帧中，没有自身的特性信息，此时 STA2 向 STA1 发送一个单播特性帧，使得 STA1 得知 STA2 的存在，下一个 T_{leader} 周期，构造新的特性帧时，会携带 STA2 的信息；后续 STA3, STA4, STA5 都和 STA2 的经历了同样的流程，成为了非领导者，被协调者 STA1 “管理”着，就这样稳定的运转。其中当前局域网内的五个 STA 都会维护一个特性表。每个特性表的内容是一致的，而且特性表中包含一个根据 STA 特性计算出来的权值 Weight。

如果此时作为协调者的 STA1 离开了，此时，其他 4 个 STA 应该应该怎么办呢，此时会执行分布式选举算法。

首先 4 个 STA 状态切换为 Initializer，即 state2 = state3 = state4 = state5=Initializer，类似于 Raft 算法中的候选者状态。权值 Weight 最大的 STA，比如是 STA2，向其他 3 个 STA 发送选举请求，3 个 STA 都向 STA2 投票时，STA2 会成为新的 Leader，此时，4 个 STA 的状态会变为 state2=Leader，state3=state4= state5=Follower。如果 $0.25T_{\text{leader}}$ 时间内没有收到 STA2 的选举请求，STA3 会发起选举请求，重复上述过程，成为新的领导者。

上述过程是以 5 个 STA 场景下，协调者退出局域网的情况表述，现在由具体到抽象的描述此过程，假设当前家庭网络里存在 N 个 STA，标号为 STA_i，其中 $i=1,2,...N$ ，假设 STA1 是最初的领导者，其他都是非领导者，此时 STA1 退出了当前局域网，则

剩下的 (N-1) 个 STA 首先全部把状态机的状态切换为 Initializer，然后按照特性表中的权重 Weight 从大到小排序，排序后的标号为 j，其中 $j=1, 2, \dots, (N-1)$ ，第 j 个 STA 分别等待 $(j * T_{leader} / N)$ 时间后，发起选举，如果 j 号 STA 选举成功，则成为领导者，其他(N-2)个 STA 成为非领导者。

4.2.4 受控并行随机接入机制的自适应门限

第三章中对受控并行随机接入机制的理论性能做了分析，其中对于平均分组来说，改进后接入机制与原始 DCF 信道接入机制相比，在终端数量大的时候，吞吐量得到了很大的提升，但是对于终端数量较少的情况来说，虽然冲突率降低了，但是吞吐量却没有原始 DCF 机制更高。

为此，在受控并行随机接入机制中引入了自适应受控接入，假设终端数量门限为 $n_{threshold}$ ，当终端总数量 N 大于该门限时，执行受控接入，小于等于该门限时，执行原始信道接入机制。由图 3-14 可知，门限值可以通过联立式(3-30)和式(3-33)求得。即

$$S'_{total} = S_{total} \Rightarrow n_{threshold} \quad (4-1)$$

当 $n = n_{threshold}$ 时，即满足上式。每个分组周期，每个 STA 根据当前使用物理层参数，重新估算阈值 $n_{threshold}$ 。

4.2.5 基于终端类型的联合流量预测

智能家居业务传输优化方案中，在执行 STA 分组时，为了做到每个分组中流量均衡，需要预测每个 STA 在本次分组周期内的流量大小。一般是根据过去周期内的流量大小来预测，为了降低流量预测实现的复杂度，提升预测的准确度，本章节提出了基于终端类型的联合流量预测。

首先，对不同类型的终端进行了参考权值定义，如表 4-1，权值范围是[0,255]，权值越大终端的平均流量越大，比如网络电视在开机状态时一般比空调、冰箱等设备平均流量大。

表 4-1 终端类型权值定义

编号	终端类型	参考权值
0	网络电视	220
1	笔记本电脑	200
2	手机	180
3	平板	170
4	网络摄像头	150
5	智能音箱	100
6	冰箱	50
7	空调	40
8	扫地机器人	30
9	烟雾报警器	20
...

现在用假设有一个终端设备，编号为 i ，参考权值为 $w_r(i)$ ，则终端的最终权值 $w(i)$ 为

$$w(i) = \eta \times w_r(i) + (1 - \eta) \times w_p(i) \quad (4-2)$$

其中 η 表示参考权值在最终权值中占的比例，范围 $[0,1]$ ， $w_p(i)$ 表示根据编号为 i 的终端设备在过去 3 个分组周期内流量值的归一化权值和 $w_p(i) \in [0, 255]$ ，可由式(4-3)得出。

$$w_p(i) = \frac{\alpha \times flow_1 + \beta \times flow_2 + \gamma \times flow_3}{flow_{max}} \times 255 \quad (4-3)$$

其中 $\alpha, \beta, \gamma \in [0,1]$ ，且 $\alpha + \beta + \gamma = 1, \alpha > \beta > \gamma$ ，当前是第 j 周期， $flow_1$ 表示上一个分组周期，即 $(j-1)$ 周期的流量值， $flow_2$ 表示 $(j-2)$ 个分组周期的流量值， $flow_3$ 表示 $(j-3)$ 个分组周期的流量值， $flow_{max}$ 表示该分组周期内终端的最大流量值，可以由式(4-4)求得。

$$flow_{max} = v_{PHY} \times T_{beacon} \quad (4-4)$$

其中 v_{PHY} 表示 STA 当前的物理发送速率，此式求出的 $flow_{max}$ 只是一个估值。

联立式(4-2)-式(4-4)可以求得编号 i 的终端的权值 $w(i)$ 。

4.2.6 负载均衡分组（重新梳理算法流程）

章节 4.2.1 中已经介绍了集中式受控并行随机接入的总体设计思想，把局域网中的 STA 分组之后，每个组占用一个时隙，组内的成员只能在指定时隙中发送数据帧。第三章已经对受控并行随机接入机制在智能家居场景中的吞吐量提升做了理论分析，理论上分组抢占信道机制对于多用户场景是有效的，本小节介绍时隙划分表示以及分组的具体实现。

4.2.6.1 时隙表示

以信标帧 Beacon 为基准，从信标帧周期 T_{beacon} 起始位置为起点，把 Beacon 周期划分为 M 个时隙，如图 4-4。

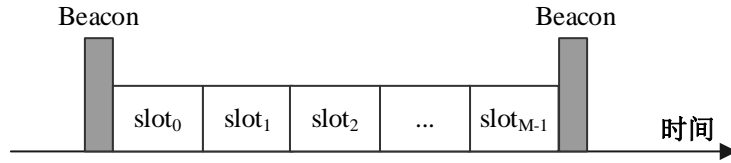


图 4-4 时隙划分图示

这些时隙大小满足的条件，如式(4-5)

$$\sum_{i=0}^{M-1} \text{slot}_i = T_{\text{beacon}} \quad (4-5)$$

其中 T_{beacon} 代表 Beacon 周期， slot_i 表示第 i 个时隙的大小， M 为最大时隙个数。

每个时隙中的终端个数要小于某个门限，用式(4-6)表示：

$$\text{StaNum}_i \leq n_{\max}, \text{ For } 0 \leq i \leq M-1 \quad (4-6)$$

其中 StaNum_i 表示时隙 i 中的 STA 数量， n_{\max} 为时隙中最大 STA 数量限制。

4.2.6.2 负载均衡分组流程

问题的条件和目标：存在 N 个 STA，STA 编号为 $0, 1, 2, \dots, N-1$ ，每个 STA 拥有一个属性权值，比如某个时隙内的流量大小的，每个时隙存在最大 STA 数量限制，见式(4-6)，共有 M 个时隙，时隙大小满足式(4-5)，每个时隙中的 STA 的权值和尽可能一致。

现在把实际问题抽象为数学问题，把条件和目标汇总如式(4-7)

$$\left\{ \begin{array}{l} \sum_{j=0}^{M-1} \text{slot}_j = T_{\text{beacon}} \\ \text{StaNum}_j \leq n_{\max}, j \in [0, M-1] \\ \sum_{j=0}^{M-1} \text{StaNum}_j = N \\ X_{i,j} = \{0,1\} \\ \sum_{j=0}^{M-1} X_{i,j} = 1, i \in [0, N-1], j \in [0, M-1] \\ \text{Weight}_j = \sum_{i=0}^{M-1} h(i), X_{i,j} = 1, j \in [0, M-1] \end{array} \right. \quad \begin{array}{l} \text{obj to min}(\delta^2(\text{Weight}_j)) \end{array} \quad (4-7)$$

其中 $h(i)$ 表示第 i 号 STA 的属性权值的大小， Weight_j 表示分配在第 j 个时隙的 STA 的权值之和。 $X_{i,j}$ 是一个二进制指标变量，它确定终端设备 i 是否加入第 j 个分组，如果终端设备 i 加入小组 j 则它为 1，否则为 0，公式(4-7)中第五个式子限制每个设备节点必须加入一个时隙，并且只能加入一个时隙。 $\delta^2(\text{Weight}_j)$ 表示所有时隙内的权值和的方差。

简而言之，就是把 N 个数字尽可能平均的分成 M 组。如果列出 N 个数字分成 M 组的所有划分情况，然后依次计算其方差，从而可以得出最平均的分组情况，这是典型的暴力解法，随着 N 的增长，这些组合数量会变得极为庞大，如 $N=1000$ ， $M=100$ ，仅列出他们所有的分成 M 组的排列组合就是一件很困难的事情，这种方法是典型的暴力解法，时间复杂度过高，是不可取的。实际上，这是一个典型的组合数学问题，该问题可以使用贪心算法或者动态规划算法，逐步划分成更小的子过程。本文所使用的的分组算法见图 4-5

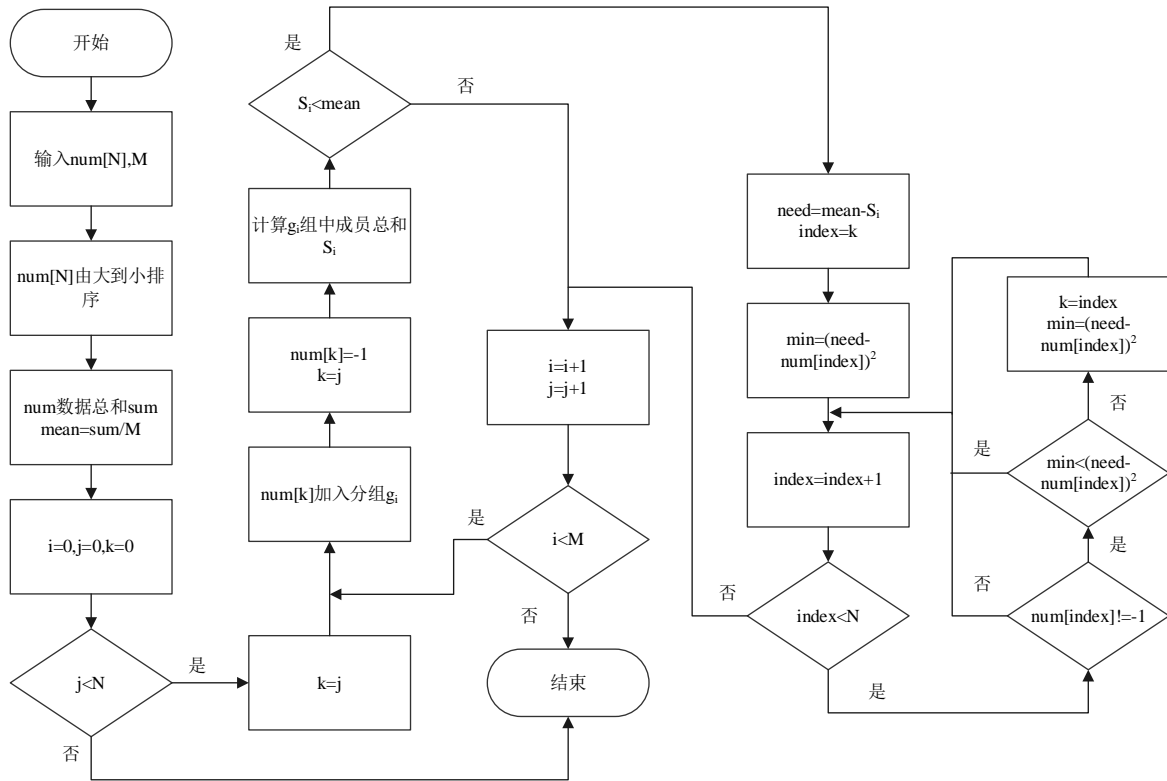


图 4-5 负载均衡分组算法流程

接下来结合一个例子来描述求解过程，假设现在有 10 个数，即 $N=10$ ，分别为 28, 25, 19, 18, 10, 9, 6, 4, 3, 1。现需要把其均分为 3 个组，即 $M=3$ 。求解过程如下：

(1) 首先计算 N 个数，分为 M 组的平均值 mean ，上述例子总和为 123，分成 3 个组的均值为 41；

(2) 对 N 个数进行从大到小的排序，得到 $\text{num}\{28, 25, 19, 18, 10, 9, 6, 4, 3, 1\}$ ；

(3) 从最大的数字 max 开始，下标为 j ，即 $\text{max} = \text{num}[j]$ ，把 max 放入一组 g_i 。如果 j 达到最大值，则分组结束。

(4) 计算组 g_i 中的数字之和 sum_i ，如果 sum_i 大于 mean ，则组 g_i 分组完成， $i+1$ ， $j+1$ ，执行 (3)；否则，转 (5)

(5) 从数组中选择下一个数字，即 $\text{num}[j+1]$ 开始，寻找最接近 $\text{need} = \text{mean} - \text{sum}_i$ 的数字，即寻找 $(\text{need} - \text{num}[j+1])^2$ 最小的数字，下标为 k ，把 $\text{num}[k]$ 加入 g 中，执行 (4)。

对于例子来讲，10 个数的总和为 123，分成 3 组的平均值为 41；从大到小开始

进行分组，首先将 28 纳入一个组 g ，因为 $28 < 41$ ，所以在剩余的数中继续寻找与 $\text{delta0} = (41 - 28) = 13$ 最接近的数；从大到小遍历的过程中，寻找离 13 最近的数，得到 18 和 10，显然 $(18 - 13)^2 > (13 - 10)^2$ ，因此将 10 纳入 g ；接着，继续再剩余的数中寻找与 $(13 - 10) = 3$ 最接近的数，最后得到一个分组 $(28, 10, 3)$

同理，按照此方法，可以得到另外的几个组 $(25, 9, 6, 1)$ ， $(19, 18, 4)$

其中，在步骤（5）中寻找最接近 need 值的时候，如果是在数组中遍历，时间复杂度会达到 $O(N)$ ，为了降低时间复杂度，把原始数据以平衡二叉树来存储，在寻找最接近 need 的数字时，只需记住上一个数字的位置，而不需要从头遍历二叉树，划分一个分组，其时间复杂度为 $O(N * \log N)$ ， M 个分组的时间复杂度为 $O(M * N * \log N)$ 。

4.2.7 基于 TSF 定时器的同步机制

各 STA 之间的同步问题是基于 TSF 定时器完成的。在 802.11 协议中有两种同步机制（TSF），一类是 BSS 网络模式中的同步机制，另一类是 IBSS 网络模式中的同步机制。本文主要使用前者。在 BSS 网络中，AP 是整个局域网的同步源，AP 和每个 STA 中都有一个 64 位 TSF 定时器，其单位是 μs 。定时同步是通过 Beacon 帧来完成的。在 Beacon 中有一个 Timestamp 字段，该字段是 64 位，并且在 Beacon 中不是以 Element 形式进行存放的，所以每一个 Beacon 帧中必定会有一个 Timestamp。

Beacon 帧是由 AP 以 T_{beacon} 为周期定时构造发送的，当 AP 在构造本周期的 Beacon 帧时，会把 AP 自身的 TSF 定时器的值填入 Timestamp 字段，当与此 AP 关联的 STA 收到 Beacon 帧时，会解析其中的 Timestamp 字段，然后对该值进行矫正，将天线与收发器的处理时间纳入其中，然后与本 STA 的 TSF 定时器进行对比，如果不同，本地 STA 的 TSF 定时器会被更新为矫正后的 Timestamp 字段，这样一来，当前局域网内的 AP 和 STA 的 TSF 定时器都是同步的。在漏失掉某个 Beacon 帧的情况下，通过本地 TSF 计时器也可以与整体的 TSF 维持同步。知晓每次 Beacon 到来的时机， T_{beacon} 以及当前 TSF 定时器的值即可实现同步。

4.3 集中式受控并行随机接入机制的软件实现

4.3.1 开发平台与架构

本章节前半部分，介绍了基于第三章提出的流量均衡的受控并行随机接入的智能家居业务传输优化方案，提出的集中式的方案设计，为了验证所提出集中式方案的性能，在国内知名半导体公司的用于智能家居场景的无线网卡芯片上进行了软件实现。该无线网卡是一款 Linux 平台上的网卡，其驱动主要分为 UMAC 和 LMAC 两部分，其中 UMAC 是开源的，为了更好的维护代码，也便于应用于其他型号的芯片，将整个优化方案在 UMAC 中设计成了一个单独的模块，称它为智能家居业务优化模块简称智能家居优化模块。优化模块与整个驱动以及协议栈之间的层次关系如图 4-6 所示

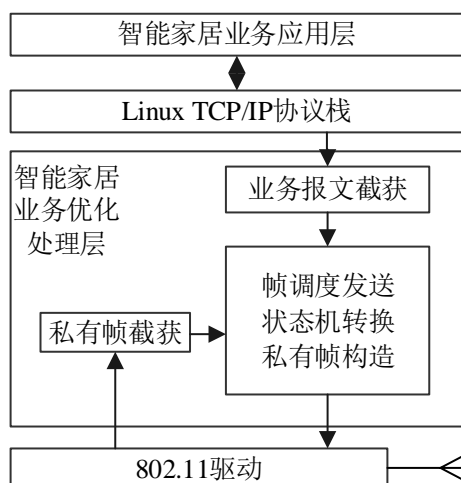


图 4-6 智能家居优化方案实现架构

智能家居业务优化模块实现于网卡驱动的 UMAC 层，实际功能层次位于 Linux 内核 TCP/IP 协议栈和整个无线网卡驱动之间，相当于一个中间层，这样设计使逻辑更清晰。

对于每个智能家居终端来讲，它们的状态机角色发生切换的时机是随机的，所以每个终端设备都要运转完整的 Leader/Follower 状态机，因此，每个终端设备的优化模块都是一致的并且完整的。下一章节介绍智能家居优化模块的具体实现。

4.3.2 智能家居业务优化模块的具体实现

智能家居优化模块作为一个单独的模块,也需要资源分配、线程和定时器创建等初始化操作,将所有的模块初始化操作封装成一个函数,是在驱动程序初始化是调用执行,驱动卸载时释放相关资源。在 Linux 内核中,每个网卡设备都抽象为 `struct net_device` 结构体,并且网卡驱动通过其中的结构体 `struct net_device_ops` 向内核注册与上层 TCP/IP 协议栈交互的接口,比如,其中的函数指针 `ndo_start_xmit` 主要负责把协议栈中的数据帧送到无线网卡驱动进行发送。而在网卡驱动中注册的 `ndo_start_xmit` 对应的函数报文发送函数为 `wal_bridge_vap_xmit()`,通过修改该函数,就可以实现协议栈待发送智能家居业务数据的过滤和截获。

同样的, Linux 内核提供了从驱动向上层协议栈提交报文的接口 `netif_rx_ni()`,驱动程序中把接收到的空口数据,经过一定的函数流程,最终调用 `netif_rx_ni()` 接口提交协议栈。因此通过对此过程中的调用流程就可以实现对接收数据帧的过滤和截获。

除了上述资源初始化和去初始化部分,从业务流程上来看,整个智能家居业务优化模块主要包括五大部分,高精度定时器 `Sh_timer`、状态机线程 `FSM_thread`、帧发送线程 `Tx_thread`、智能家居业务报文截获 `Data_Packet_filter` 以及私有帧截获 `Private_data_filter`,各个部分的具体工作流程以及它们之间的关系见图 4-7。

其中高精度定时器和状态机线程构成了整个状态机,高精度定时器的中断处理函数里周期性唤醒状态机线程,执行状态机的具体逻辑。`state` 不同状态机线程所执行的具体功能不同,具体功能[章节 4.2.2](#) 已经介绍,主要是通过触发完成量来唤醒发送线程,使得发送线程构造并发送私有帧。除此之外,发送线程还负责发送业务报文。而私有帧截获部分主要用来截获私有帧,从私有帧中解析出其具体内容,并更新对应的状态或者标志位。

整个模块里只使用了一个高精度定时器,当状态机处于 `Leader` 状态时,怎么周期性的发送协调帧呢? 其实是这样的,假设定时器的周期是 t_{htimer} ,协调帧周期为 t ,且 $t_{htimer} < t$,则只需要设定一个计数变量 `Cnt`,初始值为 0,每次状态机被高精度定时

器唤醒，将 Cnt 加 1，则当 $Cnt = Cnt_{max}$ 时，经历的时间即为 t ，其他情况也类似，其中 Cnt_{max} 可以由式(4-8)求得。这样做可以减少所使用定时器的个数。

$$Cnt_{max} = \frac{t}{t_{timer}} \quad (4-8)$$

另外，流量预测是 Leader 状态的 STA1 和 Follower 状态的 STA2 联合完成的。首先 Follower，在构造特性帧的时候，会把 STA 本身发送速率，设备类型，前三个周期内的流量大小填入特性帧，Leader 收集当前所有的 Follower 的特性信息，并汇总到特性表中，然后在更新分组信息时使用流量预测章节中的公式求得权值，然后根据此权值，使用负载均衡分组的方式，进行 STA 的分组。

至于受控并行随机接入机制的自适应门限，是在状态机线程中实现的，处于 Leader 状态的 STA 在每次构造协调帧时都会重新更新门限值，如果此时 STA 数量 n 小于门限，则将告知所有 STA 全时隙可发送，即直接执行传统接入机制，而暂停受控并行随机接入机制。

针对 [章节 4.2.7](#) 中所介绍的同步机制，是在 Sh_timer 函数中和 Tx_thread 中实现的。在本文使用的网卡芯片中，TSF 同步是 LMAC 自动完成的，并且 HMAC 很容易可以获取其值。当 Beacon 帧到来时，LMAC 告知 HMAC 当前 TSF 定时器的值，智能家居优化模块以该值为基准，即完成同步，另外 Sh_timer 函数会周期性更新 TSF 值。

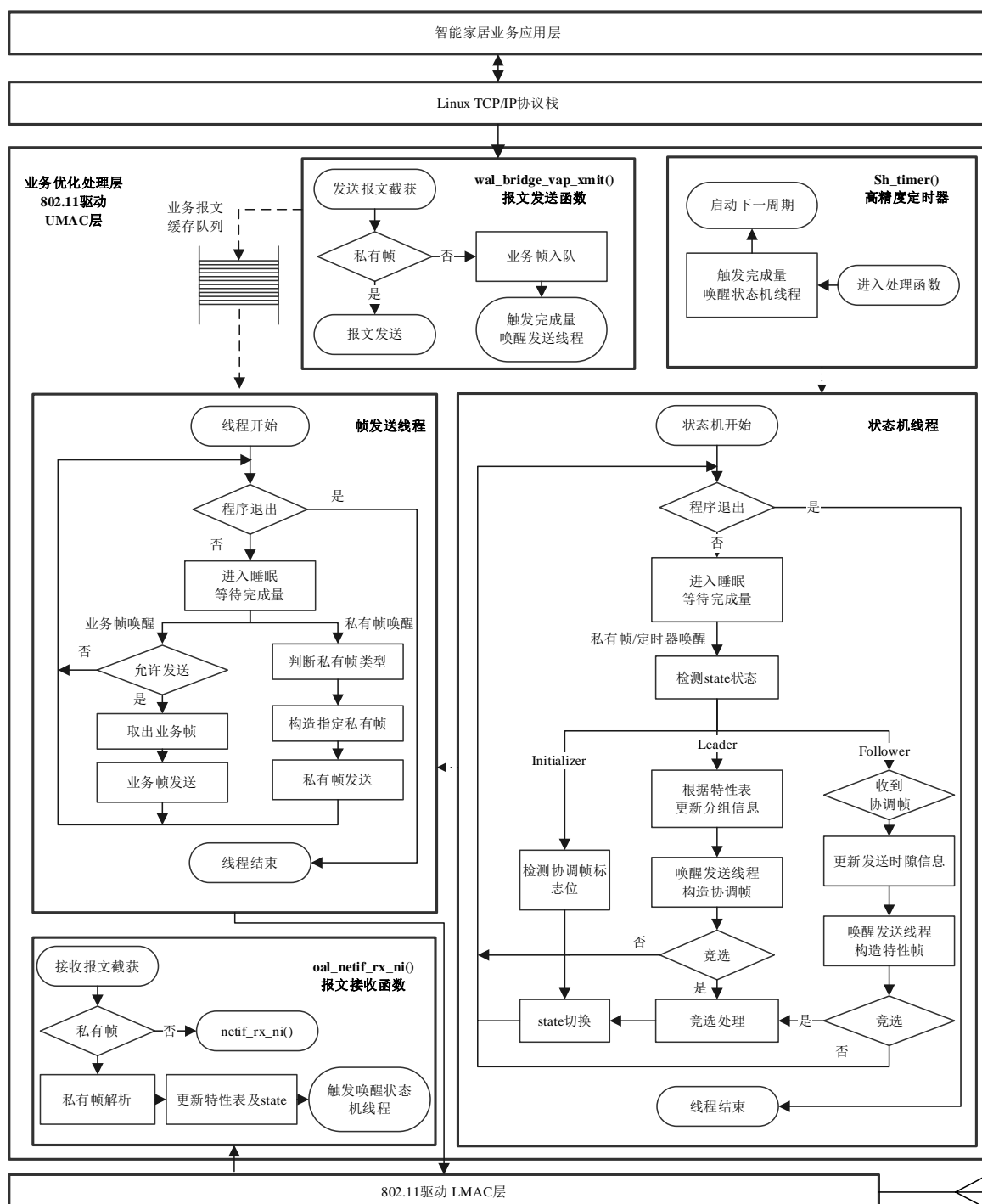


图 4-7 集中式智能家居业务传输优化模块各部分工作流程及关系图

4.4 分布式受控并行随机接入机制的方案设计

章节 4.2 中，阐述了集中式的受控并行随机接入的对路由器透明的实现方案，核

心思想在当前家庭 Wi-Fi 局域网中的所有 STA 中，选取一个参考节点，参考节点根据一个准则，对 STA 进行分组，每个分组占用一个或多个时隙，每个组内的 STA 仅在指定时隙内抢占信道资源，发送数据帧。但是集中式方案中，STA 之间需要进行频繁的私有信息帧的交互，占用空口资源，在智能家居规模过大时，这种私有信息的交互是一种开大的开销，牺牲了一定的带宽。所以提出了以尽可能少的私有信息交互为设计目标的受控并行随机接入的实现方案，仍然是对路由器透明的。

4.4.1 分布式方案设计概述

分布式的受控接入方案中，家庭 Wi-Fi 网络中的终端设备 STA 之间不再像集中式方案需要私有信息的交互，节省了交互私有帧所牺牲的带宽，这样一来每个 STA 就无法得知当前局域网内 STA 的数量，也无法使用“领导者”进行集中式的 STA 分组，分组个数和时隙大小也不能像集中式的方案，进行自适应。在这种情况下，为了减少大量用户场景下的信道接入冲突概率，让所有 STA 约定好时隙的划分规则，每个 STA 依据自身设备类型，随机选择时隙位置和数量。

4.4.2 随机分组概述

本小节主要讲述分布式方案中，STA 如何进行分组。有两个前提，对路由器透明以及无法进行私有帧的交互。对此提出了随机分组的机制。首先一个 Beacon 周期被划分为 M 个小时隙，依据 STA 的 AID 进行分组，每个组随机占用一个时隙，如图 4-8。

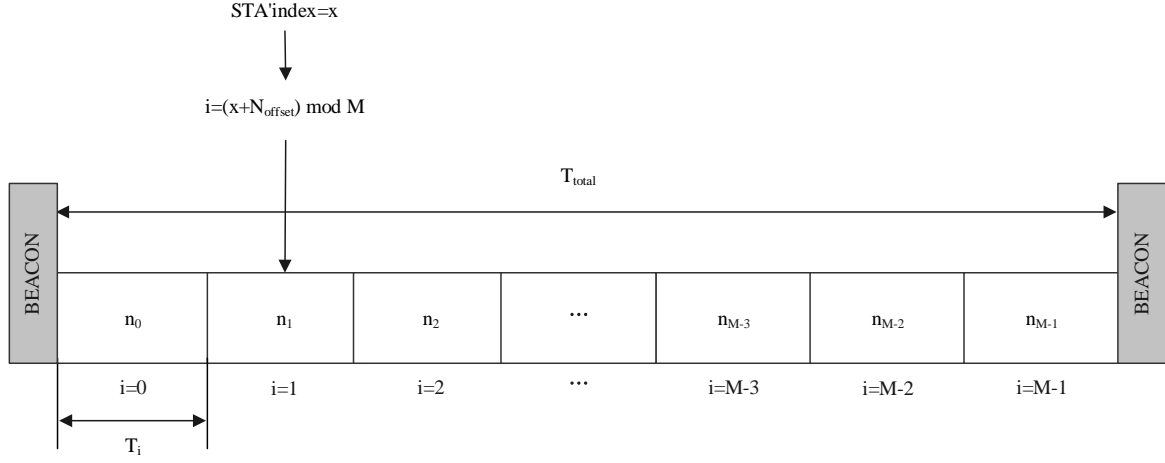


图 4-8 随机分组图示

其中 x 代表终端设备 STA 的标号 AID，是 STA 与路由器建立连接过程中，由路由器为 STA 分配的在局域网内的唯一编号； N_{offset} 是由 STA 产生的随机数，每个 Beacon 周期，都会重新生成一次，时隙编号可以由式(4-9)获得。 N_{offset} 的存在可以有效预防 STA 在发送上行数据帧时，出现固定时延的情况，在一定程度上提升 STA 信道接入的公平性。

$$i = (x + N_{offset}) \bmod M \quad (4-9)$$

4.4.3 与集中式方案的对比

相同点：集中式和分布式方案都是对路由器透明的，不需要路由器的参与就可以正常运转；两者都是采用第三章中所提出的受控并行随机接入的思想，对 STA 进行分组。

首先一个很重要的区别，局域网内的 STA 不在区分 Leader/Follower 不需要状态机机制，每个 STA 独立运转；分布式方案中，STA 不再是集中式管理，STA 的分组由自身完成，节省了原本集中式方案中私有信息帧的交互，节省了空口资源。

所以分布式方案的系统复杂度比集中式方案的复杂度低很多，但是因为没有私有帧的交互，故无法知晓当前局域网内所有 STA 的流量情况，无法按照更合理的流量进行 STA 的分组，这其实是典型的优化折中，如图 4-9，为一般通信系统的三个重要指标。

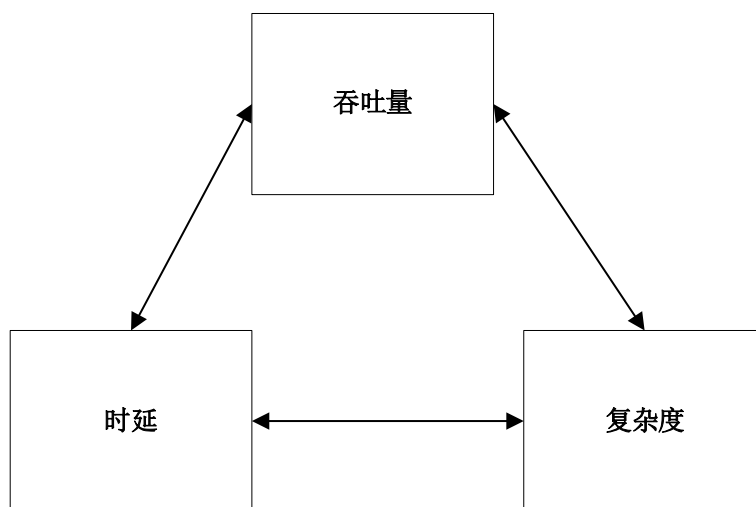


图 4-9 通信系统优化折中

对于一个通信系统来讲，吞吐量越高，时延越低，系统复杂度越低越好，这是通信系统一直在追求的设计目标。

但是实际上，图示三个指标是相互制衡的。

集中式方案和原始 DCF 相比，牺牲了复杂度和时延，提升了系统吞吐量；分布式方案牺牲了时延，提升了系统吞吐量。

分布式方案系统复杂度比集中式方案低很多，但是无法按照更合理的流量对 STA 进行分组，所以分布式方案的系统吞吐量会比集中式方案差一些。反过来讲，和集中式方案相比，分布式方案牺牲了一小部分吞吐量，降低了系统复杂度。

分布式方案系统复杂度低，实现起来更容易，同样采用了 STA 分组的思想，相对于原始 DCF 机制，密集用户的智能家居场景下，对降低冲突率，提升系统吞吐量有很大改善。对 STA 分组的思想，给降低密集用户场景下冲突率提供了新的解决思路。

4.5 分布式受控并行随机接入机制的软件实现

分布式方案的软件实现和集中式方案一样，都是同一开发平台上进行的，都是在 Linux 平台的无线网卡的 UMAC 层实现的。智能家居优化模块也是作为 UMAC 层的一个单独的模块。线程、定时器等资源的初始化与去初始化分别在驱动的加载和卸载过程中执行。和集中式方案不同的是，从业务功能上来讲主要三个部分：高精度定

时器 Sh_timer、帧发送线程 Tx_thread 以及智能家居业务报文截获 Data_Packet_filter, 三个部分的具体工作流程以及其模块之间的关系如图 4-10 所示。

其中定时器周期性唤醒发送线程, 与此同时定时器周期性产生随机数 N_{offset} , 通过式(4-9)即可求得时隙号。发送线程通过时隙号判断当前时隙是否可以发送数据, 如果此时不可以发送, 则等待下一个时隙。另外 STA 自身的 AID 可以在 STA 关联完成时获得。

将图 4-10 与图 4-7 对比即可明显看出, 分布式方案的系统复杂度比集中式方案的系统复杂度低很多。

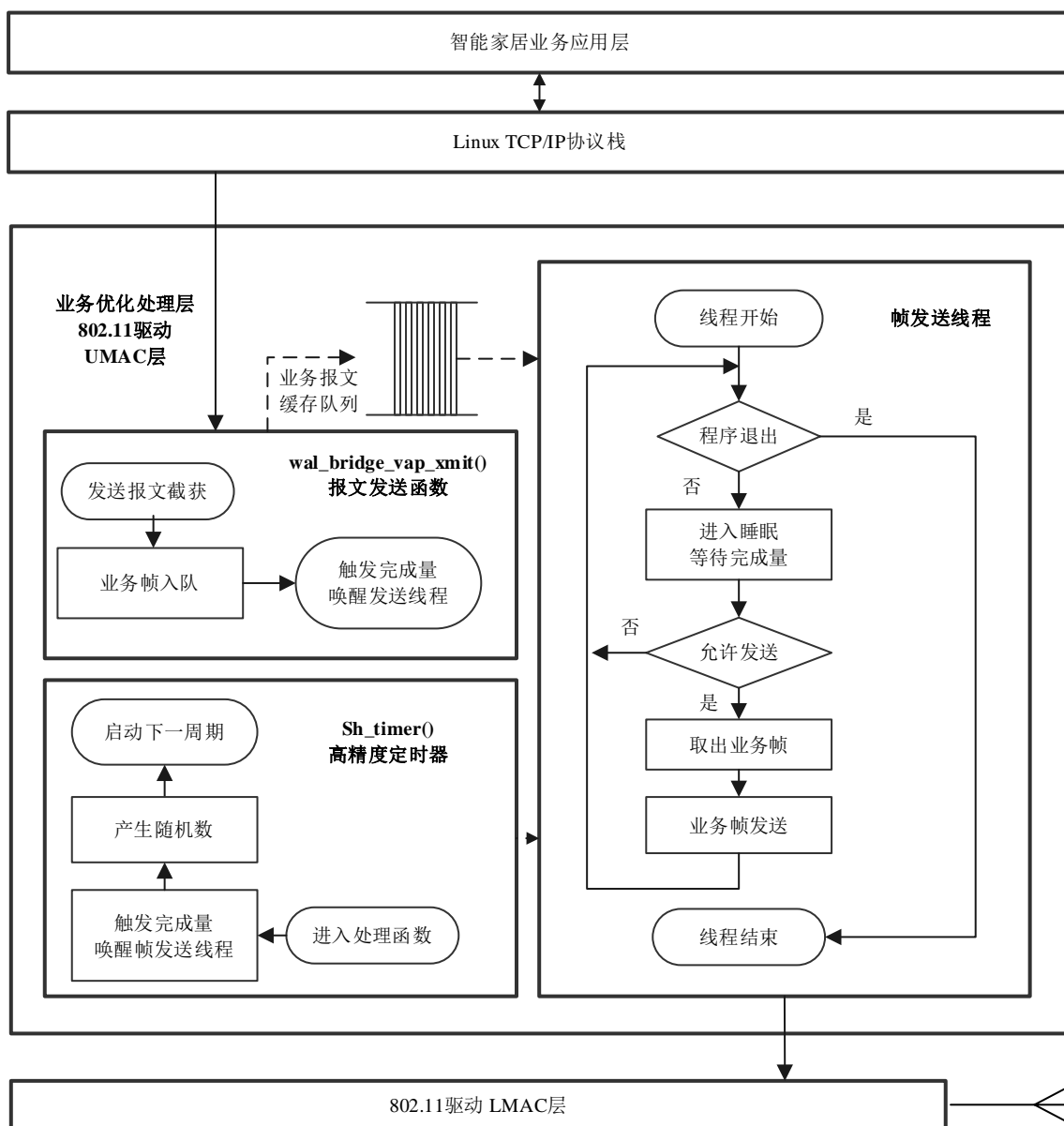


图 4-10 分布式智能家居业务传输优化模块各部分工作流程及关系图

4.6 两种实现方案的实验结果与分析（需更新）

4.6.1 实验测试环境概述

为了验证智能家居场景下 Wi-Fi 的性能，使用某网卡芯片厂商的用于物联网以及智能家居场景的 Wi-Fi 芯片单板，在单个无线路由器场景下，如图 4-11 所示，分别使用单板个数 n 为 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60 等几种不同规模的网络，统计了系统吞吐量随单板数量变化的情况，其中单板都是通过无线方式

连接单板，另外还有一普通台式机通过网线连接路由器。为了完全展示各方案的优缺点，没有开启[章节 4.2.4](#) 中的自适应门限

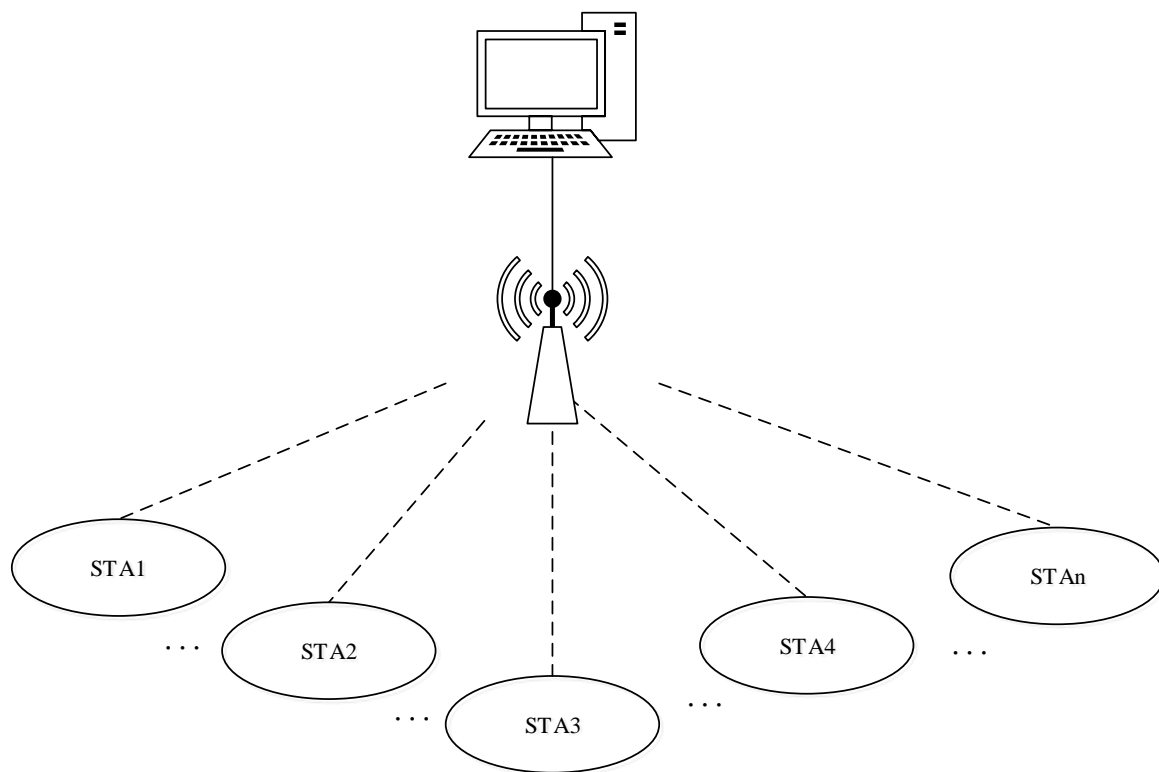


图 4-11 实际测试网络拓扑图 (改)

实验测试环境实景图如图 4-12 和图 4-13 所示



图 4-12 实验测试环境全景

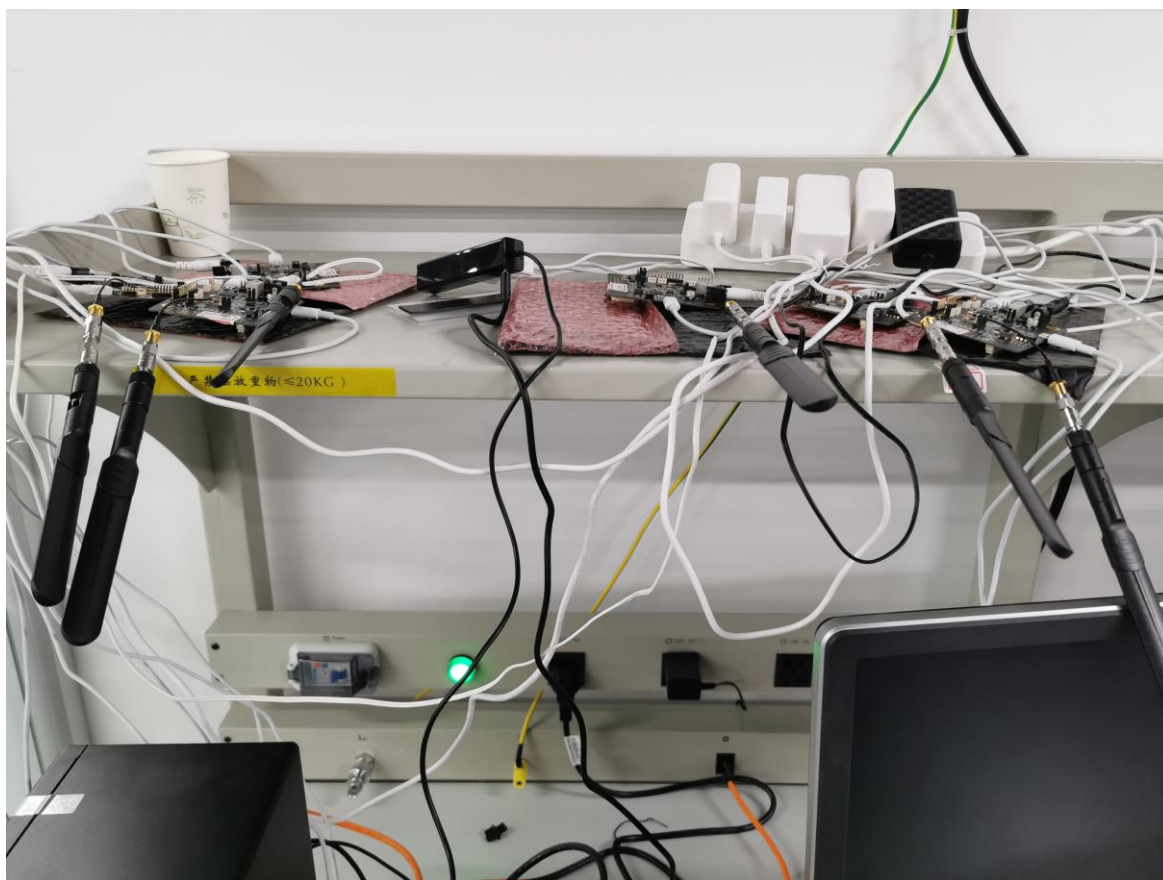


图 4-13 实验测试单板

图 4-12 为整个实验环境，图 4-13 为实际单板。

4.6.2 实际测试结果分析

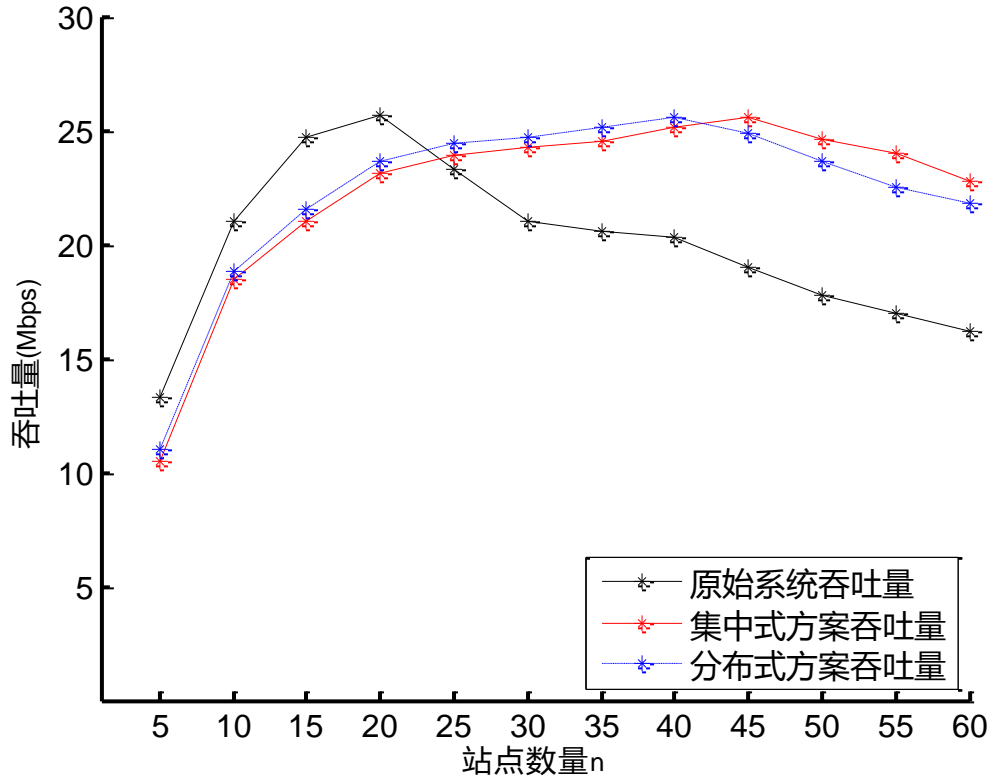


图 4-14 集中式方案和分布式方案实验结果对比

图 4-14 中散点为实际试验结果，连线只是为了表示变化趋势。由图 4-14 可知在 STA 数量较少时($n < 25$)，集中式优化方案和分布式优化方案都比原始系统吞吐量差一些，此时信道处于非繁忙状态，原始 DCF 机制完全可以承受现有的负载，使用优化方案，反而使得其信道空闲时间变多，信道利用率降低，此时集中式优化方案比分布式优化方案性能更差一些，主要是因为集中式方案存在私有帧的交互，浪费了部分带宽。

随着 STA 数量变大($25 < n < 45$)，在原始 DCF 机制吞吐量明显降低时，优化后的方案吞吐量仍在提升，此时分布式优化方案仍然比集中式优化方案吞吐量要大，但是当 $n > 45$ 时集中式方案的吞吐量逐渐超过分布式方案，此时系统处于繁忙状态，两种优化方案的吞吐量均有下降趋势，但是集中式方案的流量均衡带来的性能提升，要高过私有帧的性能损耗。但是可以预见的是如果 n 继续变大，私有帧的性能损耗要超

过流量均衡带来的性能提升，此时分布式方案的性能会超过集中式方案。

综上所述，在某些大规模 STA 网络中，集中式方案中所使用的私有帧所占用的信道资源是不可忽略的。但是两种基于受控并行随机接入的实现方案在 STA 数量较大的时候很有优势，在所测试网络规模范围内，相对于原始 DCF 机制，最大吞吐量提升达到 33.85%。

4.7 本章小结（需更新）

本章主要针对第三章提出的流量均衡的受控并行随机接入机制，结合章节 1.3 节中总结的智能家居业务场景的四个重要特征，提出了集中式和分布式两个对路由器透明的具体实现方案并进行了实际单板测试。首先对集中式方案各个模块的工作原理进行了详细的介绍，主要有状态机的转换，分布式选举，自适应门限、流量预测以及负载均衡分组等，集中式方案章节的最后通过介绍整体软件实现框架，介绍了方案各部分之间的关系以及如何协调工作；然后对分布式方案进行了详细的阐述。介绍了与集中式方案不同的随机分组机制，并分析了与集中式方案的不同之处，紧接着在对分布式的软件架构介绍的同时，讲述了分布式方案中各模块之间的关系；最后对上述两种实现方案进行了实际测试与对比分析，分布式方案实现复杂度比集中式方案低，在某些规模范围内的网络中，性能提升的效果比集中式方案稍差，但是随着规模的进一步变大，集中式方案中的私有帧交互损耗的带宽将是不可忽略的，总体来讲，两种优化方案在数量较大的智能家居场景很有优势，在所测试范围内，最大吞吐量提升达 33.85%。

5 基于地理分区的智能家居隐藏终端优化研究

5.1 引言

基于 Wi-Fi 的智能家居场景，除了 [章节 3.1](#) 中提到的两个特点之外，还有一个重要的特点：网络环境复杂。存在各种射频干扰，如其他家庭的无线通信信号等，而且家中终端设备排布很分散，有的两个设备之间虽然没有太远的距离，但是中间有墙等障碍物的阻挡。

而对于家庭中的 Wi-Fi，作为无线通信方式，介质访问采用完全分布式的 CSMA/CA 机制，在家庭网络中，终端设备准备发送数据的时候会通过载波监听，来判断当前局域网内是否有其他用户在占用信道，但是每个终端设备的检测范围是有限的，在距离太远，或者中间存在使得信号衰减严重的障碍物时，产生隐藏终端问题的概率会很高，如果两个用户长时间处于该状态，两者会不停的进行二进制退避算法，不停地重传数据帧，最后导致两者的数据发送成功的量几乎为零。

因此，本章提出了一种全新基于受控并行随机接入的隐藏终端问题优化方案，来降低设备分布复杂的智能家居场景下，隐藏终端事件发生的概率。

5.2 隐藏终端问题概述

无线传输中有两个重要的范围：传输范围和侦听范围。其中传输范围指接受帧的通信范围，取决于发送能量和无线电波的传输特性；而侦听范围是指可以检测到的传输范围，取决于接收器灵敏度和无线电波传输特性。

隐藏终端（Hidden Terminal）问题的拓扑图如图 5-1 所示，终端 STA1 向路由器发送数据，左侧的虚线圆是节点 STA1 能够检测到信道是否繁忙的最大侦听范围，右侧的虚线圆形是 STA2 能够监听到的最大通信范围，终端 STA2 不能监听到终端 STA1 正在发送数据。与此同时，如果终端 STA1 也准备向 AP 发送数据帧，就会出现终端 STA1 和终端 STA2 同时向路由器发送数据的场景，路由器无法正确接收两者发送的任何数据，这时称节点 STA1 和终端 STA2 发生了隐藏终端问题。

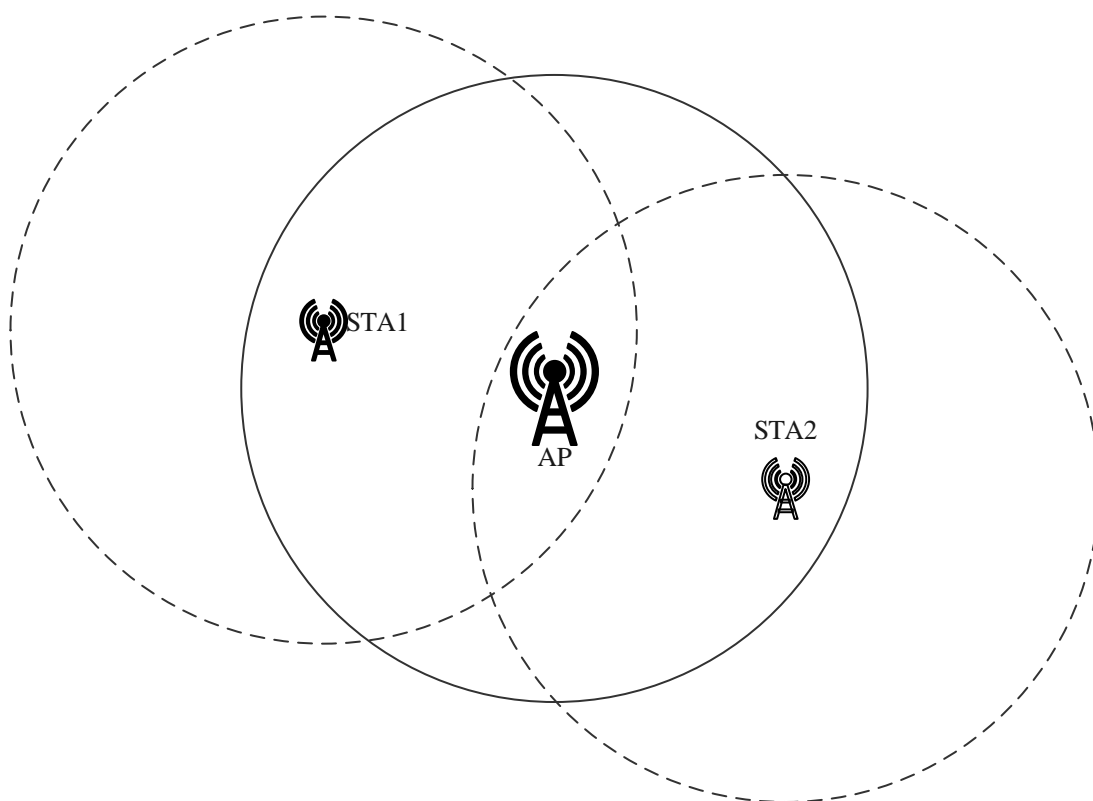


图 5-1 隐藏终端拓扑图

隐藏终端的本质是可视距离太远的两个终端在执行 CSMA/CA 机制抢占信道资源时，由于终端的检测范围，即侦听范围有限，两者都没有检测到信道繁忙，误认为信道空闲，结果路由器无法正确接收两者的数据。

5.3 现有隐藏终端问题研究分析

5.3.1 RTS/CTS 机制

802.11 协议引入 DCF 机制来解决 MAC 层共享无线信道接入问题，DCF 机制又分为以 CSMA/CA 机制为基础的 Basic 模式和为了避免隐藏终端问题而引入的 RTS/CTS (Request To Send/Clear To Send) 模式。在实际使用场景中，需要根据当前设置的阈值门限 (RTS_threshold) 来决定是否启用 RTS/CTS 机制，通常，当要发送的数据帧的字节数大于 RTS_threshold 时，会使用 RTS/CTS 机制，来降低产生冲突的概率，采用 RTS/CTS 机制可以有效减缓冲突带来的吞吐和时延损耗，提升用户体验。RTS/CTS 机制工作原理如图 5-2 所示，其中 RTS 帧和 CTS 帧都是没有加密的单播

帧。

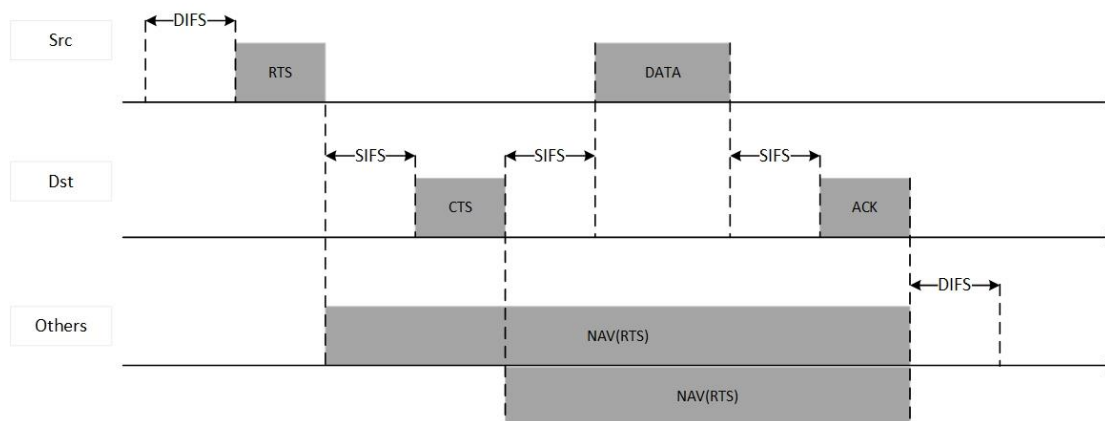


图 5-2 RTS/CTS 机制工作原理

假设图中源站 Src 为 AP，目的站 Dst 为某个 STA，当 AP 有数据帧待发送时，就会开始信道竞争，AP 在信道上发送 RTS 帧请求使用信道资源，目的 STA 会回应一个 CTS 帧，其他 STA 则会设置自身 NAV（Network Allocation Vector），在一定时间内不会抢占信道资源。通过 RTS/CTS 帧交互后，即可获得一次传输机会(TXOP)，AP 向 STA 发送数据帧，STA 接收数据后间隔 SIFS 之后向 AP 回复 ACK 帧。本轮数据帧发送结束，即可开始下一轮信道竞争。

与之类似的，对于图 5-1 中出现的隐藏终端问题，假设现在是 STA2 要向 AP 发送一个数据帧，RTS/CTS 机制避免其发生的过程如下：

首先，STA2 向 AP 发送一个不加密的 RTS 单播帧；如果此时 AP 是空闲的，即 AP 会成功解调出 STA2 的 RTS，AP 会等待 SIFS 之后发送 CTS 给 STA2。因为无线信道本身就是一个广播信道，如果帧没有加密的话，那么所有终端都可以解析其中的信息，所以 STA1 也可以收到 AP 给 STA2 发送的不加密的单播 CTS；STA1 收到 CTS 之后。发现不是发给自己的，那么 STA1 会把 CTS 帧里的 duration 字段解析出来，并设置本地 NAV。如果 NAV 没有倒数到 0，那么其会主动冻结其退避计数器，知道 NAV 等于 0，即这段时间内 STA1 不会再抢占信道；STA2 收到路由器回复的 CTS 帧后，发现是发给自己的帧。此时 STA2 已知信道空闲，在等待 SIFS 后，STA2 发送自身的数据，AP 正确接收 STA2 的数据之后，向 STA2 反馈 ACK，从而完成本次传输。

虽然 RTS/CTS 机制可以在某种程度上避免隐藏终端问题，但是在极端情况下，

比如终端数量太多的情况，RTS/CTS 会牺牲 10%的空口带宽，这在用户密集的智能家居场景是不可接受的。

5.3.2 需求分析

隐藏终端问题本质上，是两个或多个 STA，无法检测出对方在占用信道，误认为信道空闲，而进行数据帧发送后，发生空口冲突的现象。章节 1.3 中，总结了智能家居场景的四个特点，其中特点（3）家庭中有些设备相距较远，而且空口环境复杂，不像较为理想的空旷环境，家庭中存在墙体等物理障碍物，还存在产生电磁干扰的设备，使得某些设备载波检测范围变相缩小，容易出现隐藏终端问题。隐藏终端问题在 STA 数量大，数据报文发送频率高的局域网中，会严重影响网络系统的吞吐量，而且隐藏终端数目越多，对系统吞吐量影响越大^[16]。

RTS/CTS 是现在 Wi-Fi 网络中解决隐藏终端问题的重要机制，某个 STA 使用 RTS/CTS 机制发送数据帧的时候，其他所有 STA 都要暂停发送，会牺牲系统吞吐量，STA 数量较少时对吞吐量的影响不会太大。但是在这种 STA 数量多，数据帧发送频率高的场景下，对吞吐量的影响会很明显。受前面章节对 STA 分组思想的启发，对此，本章提出了基于地理分区的隐藏终端优化方案。

5.4 基于地理分区的隐藏终端优化方案设计

地理分区，是以路由器 AP 为中心的覆盖范围划分为多个扇形区域，每个扇形区域内的终端设备 STA 作为一组，不同组占用不同的发送时隙，这就使得可能出现隐藏终端的 STA 处于不同的发送时隙中，打破了出现隐藏终端的条件，大大降低了智能家居场景下隐藏终端出现的概率。

5.4.1 基于参考节点的地理分区模型

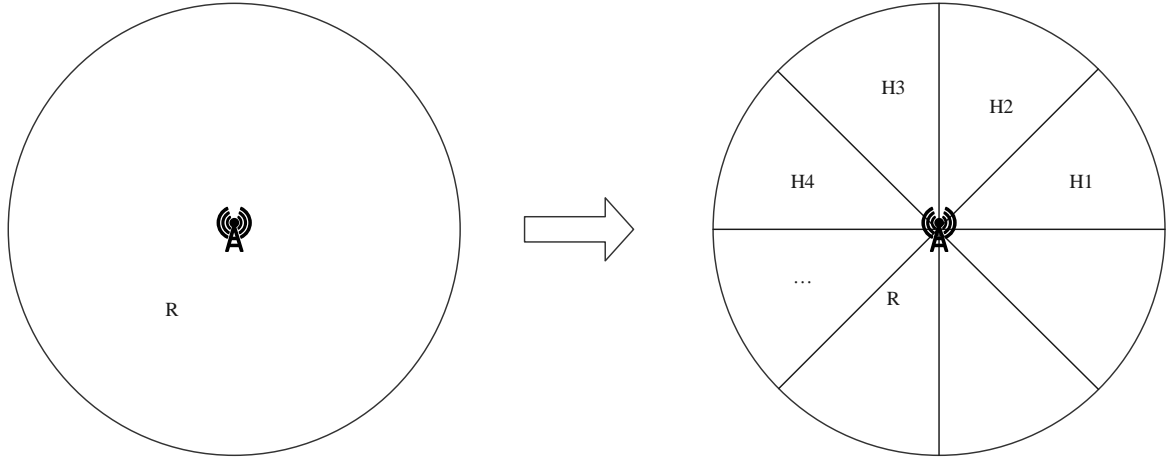


图 5-3 地理分区示意图

基于参考节点的地理分区模型如图 5-3 所示，假设路由器 AP 的覆盖区域为半径为 R 的圆形，现在把整个区域划分为 M 个扇形，编号 $0, 1, 2, \dots, M-1$ 。以圆形水平线为参考线，第 i 个区域的范围为 $(i \times \frac{360^\circ}{M}) \sim ((i+1) \times \frac{360^\circ}{M})$ 。每个区域内的 STA，划分为一个小组，每个小组占用是一个发送时隙。此时每个小组中的 STA 之间的最大距离由 $2R$ ，减小为 R ，而且只有同一扇形区域内的 STA 才可能会同时进行信道资源的抢占。

5.4.2 分组流程概述

每个 STA 的分组执行流程都是一致的，见图 5-4，在每个扇形区域距离路由器 $R/2$ 的扇形中心位置处设置一个参考节点 H ，参考节点的标号为 $0, 1, 2, \dots, M-1$ ，每个参考节点会周期性发送一个广播帧，广播帧中标识了参考节点自身的编号 $index$ ，以及一个随机数 $Random$ 。某一个 STA 在一个 Beacon 周期内，收到这些广播帧之后，会按照 LMAC 反馈的接收信号强度 RS ，从大到小排序，然后选择 RS 最大的参考节点 H 的广播帧，解析其中的两个值，如式(5-1)。

$$\begin{cases} x = index \\ N_{offset} = Random \end{cases} \quad (5-1)$$

此时，分组方式就与章节 4.4.2 一致，将式(5-1)代入式(4-9)可以求得该周期内应该占用的时隙号。

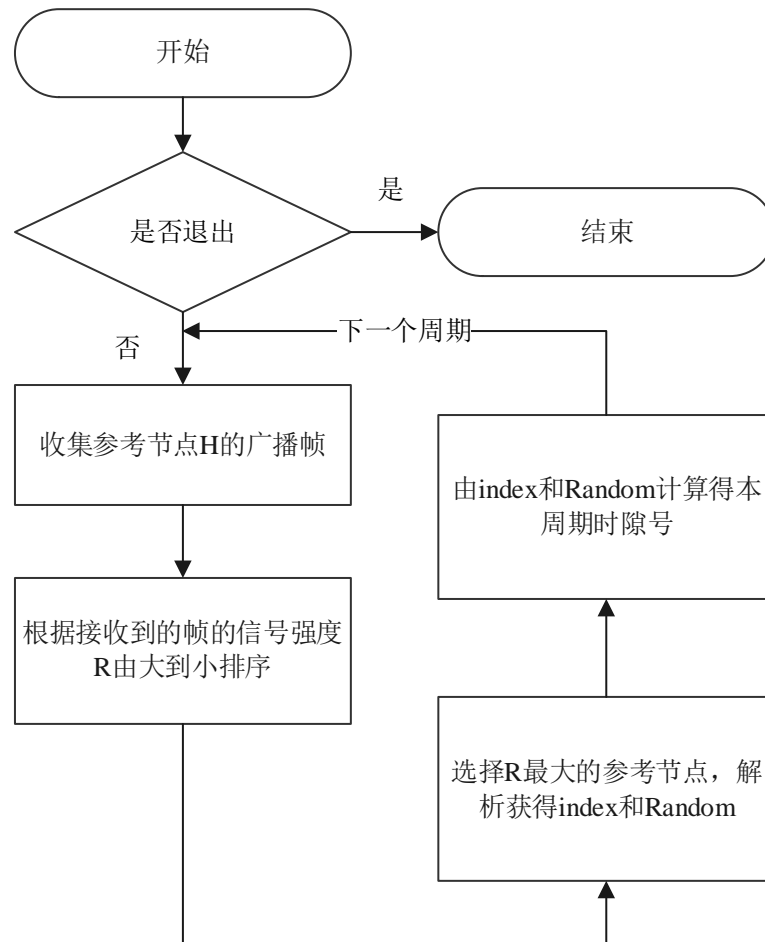


图 5-4 STA 地理分区分组执行流程

5.5 基于地理分区隐藏终端优化方案的仿真与分析（需更新）

5.5.1 仿真平台搭建

仿真平台是在国内某知名企业计算云上搭建的，Linux 内核版本为 4.14，NS-3 版本为 3.30。在 STA 侧的 MAC 层添加自定义模块。

STA 分为了两类，一种是参考节点，一种是普通节点。

参考节点中周期性构造广播帧，帧中包含两个参数，一个是参考节点的编号，另一个是一个随机数，使用传统 DCF 机制抢占信道资源。

普通节点按照图 5-4 的流程工作。屏蔽了数据帧的原始发送流程，在所属的时隙中发送数据帧，非所属时隙，有数据到来时，缓存在自定义的队列中，处于同一个

分组内的 STA 使用传统 DCF 机制抢占信道资源，关闭 RTS/CTS 机制。

使用上述修改后的模块，创建一个单接入点 AP， $(n + m)$ 个 STA 的 802.11 协议拓扑图，如图 5-3 所示， $(n + m)$ 个 STA 分布在 AP 在覆盖范围内，所有 STA 之间存在 k 个隐藏终端，关闭 RTS/CTS 机制。同样的使用修改前的模块构建与之相同的拓扑图，关闭 RTS/CTS 机制。其中 m 个 STA 是参考节点，仅周期性发送广播帧，不产生其他数据，另外 n 个 STA 是普通节点，每轮仿真统计系统平均吞吐量。

5.5.2 仿真与结果分析（需更新）

仿真参数见表 5-1

表 5-1 隐藏终端优化仿真参数

参数	取值
数据帧速率	54Mbps
控制帧速率	24Mbps
传播时延	2 μ s
时隙时长	9 μ s
SIFS	16 μ s
DIFS	34 μ s
最小竞争窗口	15
最大竞争窗口	1023
最大重传限制次数	6

创建的是单 AP，多 STA 的星型网络拓扑，关闭了 RTS/CTS 机制，分别进行了 $n = 5, 10, 20, 30, \dots, 50, 60, 70, 80, 90, 100$ 等几个网络规模的仿真，每种规模网络中含有 $(0.1n)$ 个隐藏终端，即 $k = \lceil n \times 10\% \rceil$ ，不足 1 的向上取整，其中 $m = 6$ 。

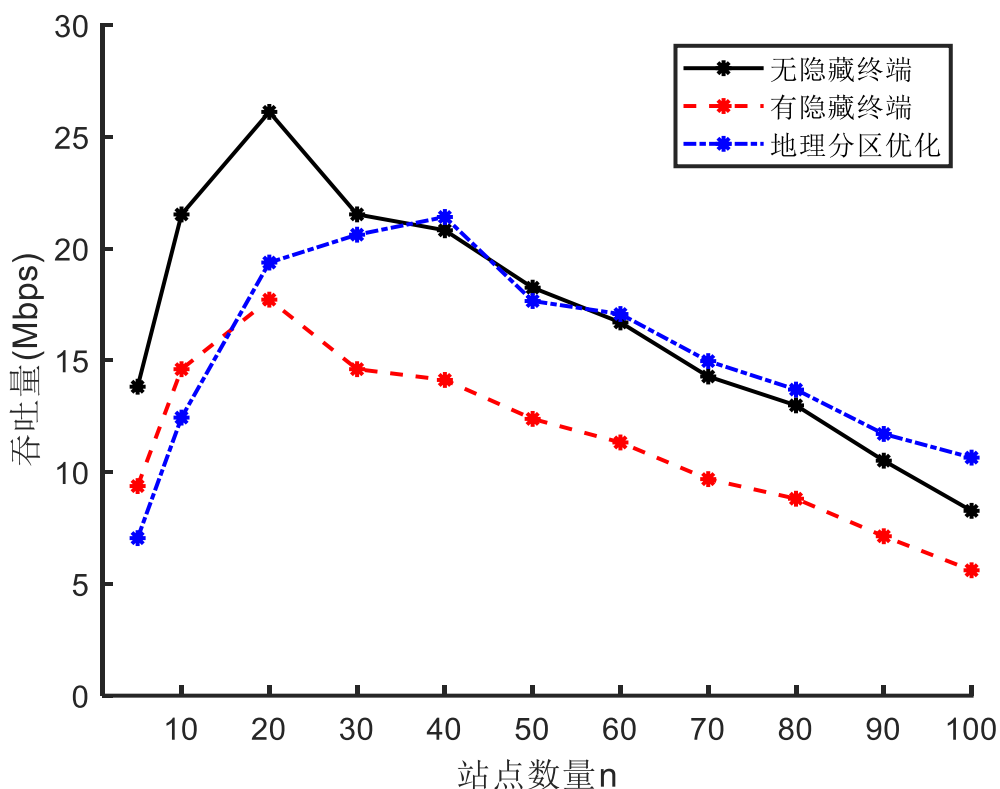


图 5-5 地理分区优化对比

图 5-5 中散点表示实际仿真结果，曲线仅表示变化趋势，其中黑色实线表示同等规模网络无隐藏终端问题时的系统吞吐量变化趋势，红色虚线表示存在隐藏终端问题时的吞吐量的变化趋势，而蓝色点横虚线表示地理分区优化后的吞吐量变化趋势。由图 5-5 可知有无隐藏终端问题吞吐量对比可知，隐藏终端问题的存在，会使得系统提前进入满负载状态，整体吞吐量包括最大吞吐量都比无隐藏终端问题时低。三个曲线对比可知，地理分区优化方案，在 STA 数量较少时，吞吐量和优化前相差不大，随着 STA 数量的增大，地理分区优化方案的吞吐量基本接近无隐藏终端的情况，随着 STA 数量的进一步增大，吞吐量甚至超过无隐藏终端的情况。地理分区优化方案相对于优化前，吞吐量最大提升近 40%。

5.6 本章小结（需更新）

本章主要在第三章所提出的受控并行随机接入机制的分组思想上，针对智能家居场景隐藏终端问题提出了一种全新的解决思路，基于地理分区的隐藏终端优化方

案。首先介绍了隐藏终端问题的具体发生条件，对现有应对策略 RTS/CTS 机制的工作原理以及优缺点进行了分析。然后对基于地理分区的隐藏终端优化方案进行了详细的阐述，讲述了分区的具体方式，以及具体实现方案。最后通过 NS-3 仿真平台，对优化方案进行了仿真分析，并对比了 Basic 模式无隐藏终端、Basic 模式有隐藏终端以及地理分区优化方案的系统性能。分区优化在终端数量较大，隐藏终端数目多的场景下，对系统性能的提升有很大优势，吞吐量最高提升近 40%。

6 总结与展望

6.1 本文主要内容及结论（需更新）

因为 Wi-Fi 网络在成本、吞吐量等方面有优势，使其得到了广泛普及，但是随着智能家居的兴起，家庭中智能终端设备数量急剧增长，使得家庭局域网逐渐演变为 Wi-Fi 密集用户场景，导致 Wi-Fi 的实际性能常常与设计性能相去甚远。

本论文主要研究内容是单个接入点情况下终端设备数量较多的家庭局域网场景，如何让各终端设备能够及时、公平的接入 Wi-Fi 网络，设计一种公平的信道接入解决方案，解决用户并发接入风暴的问题，优化智能家居场景下的系统性能和传输效率。主要研究内容和成果有：

1、分析了现在的 Wi-Fi 网络在智能家居场景下吞吐量下降、信道接入困难等现象产生的原因，指出当前智能家居场景的业务特点，以及传统 DCF 机制的不足之处和隐藏终端问题对整个家庭局域网系统性能的影响，并分别提出解决两类问题的思路。

2、基于现有的 802.11 协议，提出一种基于改进 DCF 机制的流量均衡的受控并行随机接入机制。对数量较大的终端设备“分而治之”。根据每个周期内的流量大小的预测值，将流量大小不同的终端均衡的划分到不同的分组，每个分组内的终端只能在指定的时隙中竞争信道资源，可以很好的限制直接参与信道竞争的终端数量，使得信道竞争有序化，达到了减小碰撞概率，提升系统性能的目的。

3、针对受控并行随机接入机制，基于 Bianchi 模型提出了改进的马尔科夫模型，通过理论数值分析和 NS-3 平台的实际仿真对比，在证实改进马尔科夫模型在分析 DCF 机制性能的准确性的同时，也证明了改进信道接入机制在智能家居等用户量较大场景下优势。同时提出了集中式和分布式两种具体实现方案，实际测试数据表明受控并行随机接入机制能够降低冲突概率，提升系统吞吐量，适用于密集用户场景。

4、针对智能家居隐藏终端问题，在分析了隐藏终端问题发生的原因之后，基于“分而治之”的思想，提出了基于地理分区的隐藏终端优化方案。把路由器覆盖范围划分为若干扇形区域，把每个区域内的 STA 分为一组，只有同组内的 STA 同时竞争

信道，同区域内的 STA 可视距离较小，打破了产生隐藏终端的条件。实际仿真结果表明该方案对减少隐藏终端的效果明显，适用于用户密集场景，在减少隐藏终端问题的同时还降低了发生并发接入风暴的概率。

通过以上研究分析与方案设计，使得基于 Wi-Fi 的智能家居系统能获得更大的系统吞吐量、更高的可靠性，可以让智能家居用户获得更加良好的使用体验，加快智能家居产业的发展。

6.2 本文的主要创新点（未完成）

6.3 课题展望（需更新）

虽然本文针对基于 Wi-Fi 的智能家居业务的传输性能做出了相应分析，并提出了一定的优化方案，但限于精力与能力的不足，仍存在一些不足之处，还需后续进行进一步研究与完善

1、本文基于 Bianchi 模型提出的改进 Markov 模型，在分析信道接入机制上考虑流量不饱和，退避计数器冻结以及最大重传次数限制，但是依旧是同构流量模型，假设了所有 STA 的流量模型是相同的，虽然相对于原始模型更符合实际情况，但是实际场景更多的是异构流量，后续应该探讨研究异构流量模型的建模分析，给出更加符合智能家居场景的数学模型，对优化方案做出更准确的性能评估；

2、本文提出的对智能家居业务传输优化方案，一定程度上会牺牲终端设备在当前网络环境中的传输时延，尤其是在终端数量较少的场景，这个时延会非常影响用户体验，另外优化主要考虑的上行业务，因此，后续可以开展进一步的研究，使用更新颖的技术，比如更好的分组规则等，同时考虑下行业务，在保证吞吐量的同时，尽可能不影响传输时延，进一步提升整体系统性能；

3、本文提出的优化虽然是针对智能家居场景的，但是根据网络负载及系统规模，优化传输性能，可以使其适应于其他密集用户场景下的业务传输优化，另外还需要考虑更好的与现有 802.11 标准的相兼容；

4、针对隐藏终端问题的优化方案，是单个 AP 的场景，而且需要提前设置参考节点，系统复杂度较高，后续可以考虑通过 GPS 定位系统来实现地理分区，后续可进一步分析多 AP 重叠覆盖场景下的优化问题。另外还应该考虑本文所提出的两个优化方案的融合，使其既能均衡流量、降低冲突概率，也能达到解决隐藏终端问题的目的。

致 谢

时间飞逝，三年的研究生生涯已经快要结束，回忆这三年的校园时光，每个片段都像昨天发生的一样。在此期间，我接触并参与了许多科研项目，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，极大的提升了自身的专业技能。如今，我的毕业论文即将完成，学习生涯也即将结束，踏上人生的新阶段，此时此刻，我想向这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阶段给予了极大的帮助，在他孜孜不倦的教导和帮助下，我得以逐渐将课堂上的理论知识不断应用于实践当中，不断地进步，提升自我。我还要感谢喻洪涛老师，感谢您在科研项目中给予我的指导和帮助。我也要感谢刘应状老师，他在我的论文写作期间提出了许多宝贵的意见，使我能够不断地进步，使我顺利完成论文工作。

然后，我要感谢实验室团队的周谋成、刘寺杰、郭志斌、罗声海、张萌、黄维、王子舒、马松林、刘依山、郭康、刘一雄、黎育才、杜子豪、刘振宇、王禹磊、朱锦、李舟、叶冲、金明浩等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，还在我的毕业设计上提供了大量的支持，这三年里留下了许多珍贵的记忆，愿我们各自安好，同学情谊延续一生。

我还要感谢我的父母和家人，你们是我坚强的后盾，也是驱动我不断前进的不竭动力。感谢赵晓朵在写论文这段时间给予我的陪伴，让我顺利地度过最后的挑战。

最后，衷心感谢所有在学习和生活中帮助和支持着我的人！感谢所有参加论文评审、答辩的专家和老师！

华中科技大学硕士学位论文

表目录

表 1-1 全国智能家居相关支持性政策文件.....	2
表 1-2 历代 Wi-Fi 标准命名及协议对应	5
表 1-3 Wi-Fi 技术特征发展	6
表 2-1 NS-2 与 NS-3 特征比较.....	27
表 2-2 NS-3 基本模块	27
表 3-1 IEEE 802.11e 的业务优先级	33
表 3-2 数值分析及仿真参数设置.....	38
表 4-1 终端类型权值定义.....	68
表 5-1 隐藏终端优化仿真参数.....	93

图目录

图 1-1 智能家居实现模型及分类.....	4
图 2-1 802.11 标准中 DCF 与 PCF 的关系	13
图 2-2 各种 IFS 示意图.....	14
图 2-3 CSMA/CA 原理图.....	15
图 2-4 二进制指数退避算法工作原理.....	16
图 2-5 PCF CFP 周期	17
图 2-6 排队系统的组成.....	18
图 2-7 Bianchi 模型的马尔科夫模型状态转换图	23
图 2-8 Linux 802.11 协议栈架构	26
图 2-9 NS-3 仿真基本模型	30
图 3-1 802.11e MAC 层协议结构示意图.....	33
图 3-2 EDCA 机制下的发送优先级对比.....	35
图 3-3 RAW 工作机制.....	37
图 3-4 非饱和负载下系统平均吞吐量.....	39

华中科技大学硕士学位论文

图 3-5 非饱和流量下 STA 平均误帧率	40
图 3-6 受控接入机制的网络拓扑图及接入规则	42
图 3-7 时隙划分	42
图 3-8 STA 分组效果图	43
图 3-9 改进 Markov 模型的状态转换图	46
图 3-10 不同流量模型下 CSMA/CA 的竞争成功概率边界	50
图 3-11 流量不饱和条件时不同站点数下 p 的值	52
图 3-12 改进后 DCF 机制和 DCF 机制冲突概率对比	54
图 3-13 非饱和负载下原始 DCF 机制吞吐量和 n 的关系图	55
图 3-14 非饱和负载下改进前后吞吐量对比	56
图 3-15 非饱和负载下理论与仿真对比	58
图 4-1 集中式受控并行随机接入整体方案示意图	61
图 4-2 Leader/Follower 状态机的状态转换图	62
图 4-3 Raft 算法角色状态转换	65
图 4-4 时隙划分图示	69
图 4-5 负载均衡分组算法流程	71
图 4-6 智能家居优化方案实现架构	73
图 4-7 集中式智能家居业务传输优化模块各部分工作流程及关系图	76
图 4-8 随机分组图示	78
图 4-9 通信系统优化折中	79
图 4-10 分布式智能家居业务传输优化模块各部分工作流程及关系图	81
图 4-11 实际测试网络拓扑图（改）	82
图 4-12 实验测试环境全景	83
图 4-13 实验测试单板	84
图 4-14 集中式方案和分布式方案实验结果对比	85
图 5-1 隐藏终端拓扑图	88
图 5-2 RTS/CTS 机制工作原理	89

图 5-3 地理分区示意图.....	91
图 5-4 STA 地理分区分组执行流程	92
图 5-5 地理分区优化对比.....	94

参考文献（未完成）

- [1] 康爱香, 郝枫, 宋旭阳. 不可再生自然资源对经济增长的福祸之辨[J]. 统计与信息论坛, 2021, 36(11): 95-106.
- [2] 王天源. 平均预期寿命的影响因素分析[J]. 现代经济信息, 2017(05): 24+26.
- [3] Majumder S, Aghayi E, Noferesti M, et al. Smart Homes for Elderly Healthcare—Recent Advances and Research Challenges[J]. Sensors, 2017, 17(11): 2496.
- [4] Nathan V, Paul S, Prioleau T, et al. A Survey on Smart Homes for Aging in Place Toward solutions to the specific needs of the elderly[J]. Ieee Signal Processing Magazine, 2018, 35(5): 111-119.
- [5] 贺丹, 刘厚莲. 中国人口老龄化发展态势、影响及应对策略[J]. 中共中央党校(国家行政学院)学报, 2019, 23(04): 84-90.
- [6] Mallinson D J, Shafi S. Smart home technology: Challenges and opportunities for collaborative governance and policy research[J]. Review of Policy Research, 2022.
- [7] Bhati A, Hansen M, Chan C M. Energy conservation through smart homes in a smart city: A lesson for Singapore households[J]. Energy Policy, 2017, 104: 230-239.
- [8] Bianchi G. Performance analysis of the IEEE 802.11 distributed coordination function[J]. Ieee Journal on Selected Areas in Communications, 2000, 18(3): 535-547.
- [9] 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.
- [10] Le-Trung Q. An Open-Source Framework for Integration of Wireless Ad-hoc Drivers into Linux Kernel[J]. Proceedings of the Seventh Symposium on Information and Communication Technology (Soict 2016), 2016: 202-209.
- [11] Feng W. Performance analysis of IEEE802.11e EDCA wireless networks under finite load[J]. Wireless Networks, 2020, 26(6): 4431-4457.
- [12] 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.
- [13] 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.
- [14] Erturk M A, Vollero L, Aydin M A. Optimal joint load balancing and EDCA configuration of IEEE 802.11 wireless hotspots[J]. International Journal of Communication Systems, 2018, 31(2).
- [15] Abu-Khadrah A I, Zakaria Z, Othman M, et al. Enhance the Performance of EDCA Protocol by Adapting Contention Window[J]. Wireless Personal Communications, 2017, 96(2): 1945-1971.
- [16] Khurana S, Kahol A, Jayasumana A P. Effect of hidden terminals on the performance of IEEE 802.11 MAC protocol[J]. 23rd Annual Conference on Local Computer Networks - Proceedings, 1998: 12-20.