



中国研究生创新实践系列大赛  
“华为杯”第二十届中国研究生  
数学建模竞赛

学 校

武汉大学

参赛队号

23104860166

队员姓名

1.

刘芮

2.

彭依妮

3.

赵子龙

# 中国研究生创新实践系列大赛

## “华为杯”第二十届中国研究生

### 数学建模竞赛

题 目： 基于 Markov 链的 WLAN 网络信道接入机制建模

#### 摘 要：

无线局域网能够提供低成本、高吞吐和便利的无线通信服务，被广泛应用于家庭、校园、公司以及如咖啡馆的公共场所等众多场合，为人们的生活和工作提供无数便利。当前，随着用户数的激增，设备数量、应用类型、网络流量随之飞速增长，导致 AP 部署日趋密集，而可用信道数相对有限。这导致不同的 BSS 需要复用同一个信道，用相同信道号的 AP 称为同频 AP。同频 AP 之间通信区域存在重叠时存在同频干扰，是 WLAN 组网最显著的干扰问题，将大大降低网络性能。如何在同频多 BSS 场景下对系统吞吐性能进行有效评估对网络部署具有指导意义，可以为未来更复杂的网络环境的基础设施建设奠定基础。

本文基于 DCF 机制，针对多 BSS 中存在的同频干扰问题，考虑在以两 BSS 和三 BSS 为代表的同频多 BSS 场景中同频 AP 能否互听以及数据并发传输是否成功的多种情况下，首先分析不同组网条件下的信道状态转换和状态时长；然后利用 Bianchi 模型对 MAC 协议下的两 BSS 互听场景数据传输中的随机退避过程进行建模，基于 Markov chain 模型求解节点在每个虚拟时隙发送数据帧的概率和发生碰撞的条件概率，最后基于概率分布构建系统吞吐解算模型，实现系统性能的评估，同时编写仿真器验证模型精确度。

针对问题一和问题二，需要对 WLAN 组网中 DCF 机制下的两 BSS 互听场景进行数据传输过程的分析和建模，实现对系统吞吐率的评估。其中，问题一假设在使用相同信道的情况下并发传输时的 SIR 较低，导致两个 AP 的数据传输都失败，问题二与之相反。这两节首先利用基于 Bianchi 模型对 MAC 协议下的两 BSS 互听场景数据传输中的随机退避过程进行建模，基于 Markov chain 求解节点在每个虚拟时隙发送数据帧的概率和发生碰撞的条件概率，从而分别推导单个时隙长度内的信道空闲时长、成功传输时长和碰撞时长，最后基于题目给定的各项参数解算问题一题设情况下系统的吞吐为 67.1743Mbps，问题二题设情况下系统的吞吐为 70.5585Mbps。同时，编写仿真器对两 BSS 互听场景的随机退避过程进行仿真模拟，分别在问题一和问题二题设下获得仿真条件下的系统吞吐为 65.2437Mbps 和 68.9498Mbps。

针对问题三，需要对 WLAN 组网中 DCF 机制下的两 BSS 不互听场景进行数据传输过程的分析和建模，实现对系统吞吐率的评估。本节首先利用基于 Bianchi 模型对 MAC 协

议下的随机退避过程进行建模，针对两 BSS 不互听场景，基于 Markov chain 求解节点在每个虚拟时隙发送数据帧的概率和出现信号干扰的条件概率，从而推导节点在单个时隙长度内的信道空闲时长、成功传输时长和失败传输时长，最后利用两个节点是否发送数据帧相互独立的条件，得到**系统的吞吐为 52.6231Mbps**，**仿真模拟得到系统吞吐为 54.7553Mbps**。同时针对题设和其余六组不同系统参数验证模型精度，**模型平均误差为 4.7%**。

针对问题四，需要对 WLAN 组网中 DCF 机制下的三 BSS 场景中 AP1 与 AP3 不互听，AP2 分别与两者互听情况进行数据传输过程的分析和建模，实现对系统吞吐率的评估。本节基于 Markov chain 求解节点在每个虚拟时隙发送数据帧的概率和发生碰撞的条件概率。针对题设三 BSS 场景，分析信道的全部占用情况，计算每种情况的发生概率并描述其对应时隙大小，从而得到系统信道利用率并计算**系统吞吐为 110.4150Mbps**，**仿真模拟得到系统吞吐为 111.0180Mbps**。同时针对题设和其余六组不同系统参数验证模型精度，**模型平均误差为 3.1%**。

**关键词：** WLAN 组网；系统吞吐；DCF 机制；Markov chain 模型；信道同频干扰

关注公众号：建模忠哥，获取更多资料

## 目录

<b>1</b>	<b>问题背景与问题重述</b>	<b>5</b>
1.1	问题背景	5
1.2	任务重述	6
1.2.1	任务要求	7
1.2.2	相关缩略语和已知参数说明	8
<b>2</b>	<b>模型假设</b>	<b>8</b>
<b>3</b>	<b>符号说明</b>	<b>9</b>
<b>4</b>	<b>问题一</b>	<b>10</b>
4.1	问题分析	10
4.2	模型建立	11
4.2.1	DCF 机制下的信道状态分析	11
4.2.2	基于 Markov chain 的两 BSS 互听场景数据传输过程建模	12
4.2.3	基于概率分布的系统吞吐解算模型	15
4.3	问题求解	16
4.3.1	基于 Markov chain 的系统吞吐评估	16
4.3.2	仿真验证及结果分析	17
<b>5</b>	<b>问题二</b>	<b>18</b>
5.1	问题分析	18
5.2	模型建立	18
5.2.1	考虑并发碰撞的信道状态时长分析	18
5.2.2	基于 Markov chain 的两 BSS 互听场景数据传输过程建模	19
5.2.3	基于概率分布的系统吞吐解算模型	20
5.3	问题求解	21
5.3.1	基于 Markov chain 的系统吞吐评估	21
5.3.2	仿真验证及结果分析	22
<b>6</b>	<b>问题三</b>	<b>23</b>
6.1	问题分析	23
6.2	模型建立	24

6.2.1	考虑传输时间上有重叠时的信道状态时长分析	24
6.2.2	基于 Markov chain 的两 BSS 不互听场景数据传输过程建模	24
6.2.3	基于概率分布的系统吞吐解算模型	28
6.3	问题求解	28
6.3.1	基于 Markov chain 的系统吞吐评估	28
6.3.2	仿真验证及结果分析	29
7	问题四	32
7.1	问题分析	32
7.2	模型建立	33
7.2.1	三 BSS 场景下的信道状态时长分析	33
7.2.2	基于 Markov chain 的三 BSS 场景数据传输过程建模	34
7.2.3	基于概率分布的系统吞吐解算模型	37
7.3	问题求解	39
7.3.1	基于 Markov chain 的系统吞吐评估	39
7.3.2	仿真验证及结果分析	40
8	工作总结与展望	43
8.1	结论	43
8.2	展望	43
	参考文献	44
	附录 A MATLAB 代码	45
A.1	问题一代码	45
A.2	问题二代码	49
A.3	问题三代码	52
A.4	问题四代码	58

# 1 问题背景与问题重述

## 1.1 问题背景

无线局域网 (WLAN, wireless local area network), 即 Wi-Fi, 能够提供低成本、高吞吐和便利的无线通信服务, 被广泛应用于家庭、校园、公司以及如咖啡馆的公共场所等众多场合, 为人们的生活和工作提供无数便利 [1]。常见的 WLAN 结构包括基本服务集 (BSS, basic service set)、独立基本服务集和延伸服务集。其中, BSS 结构是一种使用最广泛的 WLAN 结构, 由处于某一特定覆盖区域内的站点 (STA, station) 与一个专职管理 BSS 的无线接入点 (AP, access point) 组成。常见的 STA 有手机、笔记本、物联网设备等, 常见的 AP 有无线路由器、Wi-Fi 热点等。在 Wi-Fi 协议中, 若将 STA 和 AP 统称为节点, 各节点共享无线信道, 为了避免多个节点同时访问网络所带来的冲突问题, 引入载波侦听多址接入/退避 (CSMA/CA, carrier sense multi-access and collision avoidance) 机制 [2]。分布式协调功能 (DCF, distributed coordination function) 将 CSMA/CA 技术和确认字符 (ACK, acknowledge character) 技术结合, 采用二进制指数退避算法来避免碰撞 (随机退避过程), 是节点共享无线信道进行数据传输的基本接入方式 [3, 4]。

媒体接入控制 (MAC, medium access control) 协议被广泛地应用于传统的有线局域网 (LAN) 及目前流行的 WLAN 中。在 WLAN 中, MAC 所对应的标准为 IEEE 802.11, 其工作方式采用 DCF 和 PCF (中心控制), 它定义了无线信道的使用方式, 在传感器节点之间分配有限的无线通信资源, 用来构建传感器网络系统的底层基础结构 [5, 6]。Bianchi 建立 Markov chain 模型来表示 DCF 的退避过程, 将退避器所处的阶数和随机退避数用二维 Markov chain 表示, 推导节点在每个虚拟时隙的发送概率  $\tau$  和发生碰撞的条件概率  $p$ , 从而评估 BSS 的吞吐, 并数学步骤推导出系统饱和状态下的吞吐与稳态概率的关系, 具有较高精度 [7]。其他很多研究学者以此为基础开展 MAC 协议性能分析研究, Chatzimisios 研究了有最大重传次数限制的 MAC 层性能情况 [8]。Huang 和 Ivan Marsic 介绍了隐藏节点下网络模型和性能分析 [9]。Chen 分析了多速率 MAC 协议的性能 [10]。此外, 还有一些提出了对随机退避窗口初始值的改进算法 [11, 12]; 一些提出了对不同的业务采取不同的退避窗口和算法 [13]。吞吐是单位时间内发送数据有效载荷的比特数, 单位为 bps。吞吐  $S$  可以由信道的利用率与物理层速率 (单位为 bps) 的乘积表示:

$$S = \frac{E[\text{一个时隙内传输的有效载荷发送时长}]}{E[\text{一个时隙长度}]} \times \text{物理层速率} \quad (1.1)$$

当前, 随着用户数的激增, 设备数量、应用类型、网络流量随之飞速增长, 导致 AP 部署日趋密集, 而另一方面, 可用信道数是相对有限的。这势必导致不同的 BSS 需要复用同一个信道, 用相同信道号的 AP 称为同频 AP。同频 AP 之间通信区域存在重叠时, 存在

相互干扰问题，叫作同频干扰，即在某一个特定时间内，只有一个用户或者一个 AP 能够传输数据。同频干扰是 WLAN 组网最显著的干扰问题，由于同时收发数据时会有干扰和延时，将大大降低网络性能。在同频多 BSS 场景下，对系统吞吐性能进行有效评估需要考虑两个关键条件，即发送节点间能否互听，以及并发传输时是否成功。发送节点间能否互听与信号能量强度（RSSI, received signal strength indication）密切相关。节点收到信号后，根据 RSSI 是否高于信道可用评估（CCA, clear channel assessment）门限，来判断信道为忙或闲。节点间距离不同，信号衰减不同，最终 RSSI 也会随之不同，所以并不是所有的节点之间都能互听。当有多个 BSS 的节点同时发送数据（即并发传输）时，数据传输成功与否与信干比（SIR, signal to interference ratio）有关，SIR 即信号强度与干扰强度的比值，可以用信号 RSSI 与干扰信号 RSSI 的差值表示。若 SIR 足够高，则信号能被成功解调，即并发传输成功；若 SIR 很低，则信号解调失败，即并发传输失败。

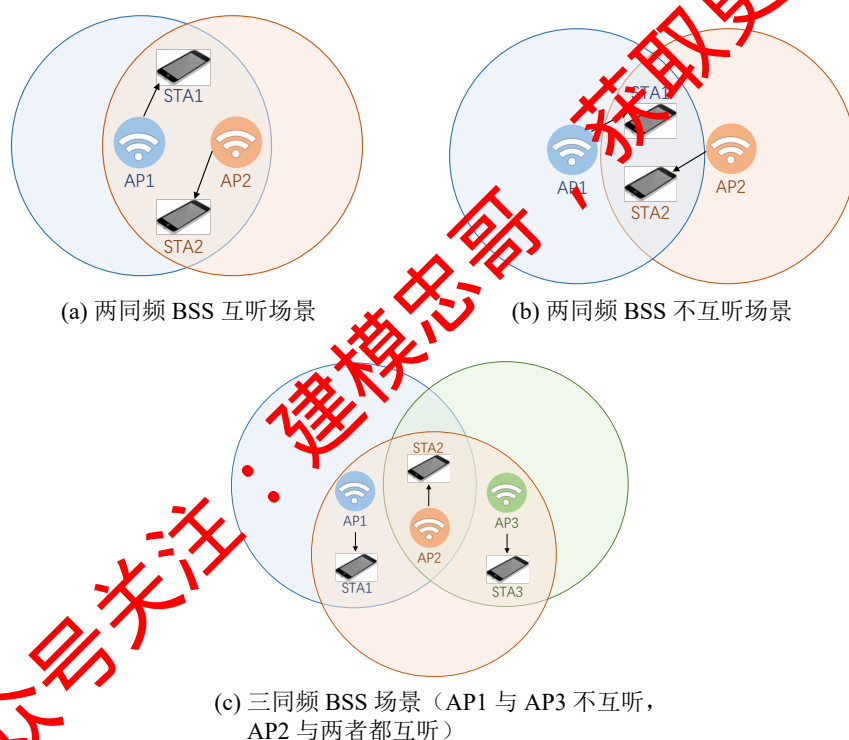


图 1.1 不同 BSS 场景的节点分布情况

## 1.2 任务重述

通过赛题背景和题目分析，本文需要解决 WLAN 网络信道接入机制建模问题。具体指在以两 BSS 和三 BSS 为代表的同频多 BSS 场景中，在同频干扰影响下，基于 DCF 机制对系统进行模型建立，并通过数值分析方法求解，评估系统的吞吐，从而实现系统性能的评估，同时利用仿真实验验证模型精度。考虑到发送节点间能否互听，并发传输时是否成功，是进行系统建模需要考虑的两个先决条件，任务分为四种情况求解：如图 1.1(a) 和



图 1.1(b) 所示, 在两 BSS 场景下, 存在互听和不互听两种情况。对于两 BSS 互听的场景, 大部分时候只有一个 AP 能够接入信道, 数据传输成功, 而在并发时需要根据节点之间的 RSSI 估算两个 AP 并发时的 SIR, 在问题一和问题二提出的并发时 SIR 较低和较高两种假设下, 分别在两个 AP 的数据并发传输都失败和成功的情况下对系统建模求解, 并通过仿真验证模型的精确度; 对于两 BSS 不互听的场景, 根据问题三题设以及并发时 SIR 比较小的假设, 在单个 AP 的发包因信道质量导致的丢包率  $P_e = 10\%$  和两个 AP 在时间上有重叠时的发包均失败情况下, 对系统建模求解, 评估系统吞吐, 并通过仿真验证模型的精确度; 对于三 BSS 场景, 如图 1.1(c) 所示, 根据问题四题设的 RSSI 值可得 AP1 与 AP3 不互听, AP2 与两者都互听。在 AP1 和 AP3 在时间上有重叠发包时 SIR 较大, 两者发送均成功的假设下, 对该系统进行建模, 利用数值分析方法求解并评估系统吞吐, 最终通过仿真验证模型性能。

### 1.2.1 任务要求

基于上述背景, 题目给出了多 BSS 场景下存在同频干扰影响时的四种场景假设, 本文需要研究解决以下问题:

**问题一:** 在两 BSS 互听的场景下, 根据本小题给定的参数, 结合单个 AP 的数据传输成功而并发时 SIR 较低, 即并发时两个 AP 的数据传输都失败的假设, 对该两 BSS 互听系统进行建模, 用数值分析方法求解, 评估系统的吞吐, 并编写仿真器验证模型精确度。

**问题二:** 在与问题一相同的场景和条件条件下, 根据本小题给定的参数, 结合并发时两个终端接收到数据的 SIR 较高, 即两个 AP 的数据传输都能成功的假设, 对该两 BSS 互听系统进行建模, 用数值分析方法求解, 评估系统的吞吐, 并编写仿真器验证模型精确度。

**问题三:** 在两 BSS 不互听的场景下, 根据本小题给定的参数, 在单个 AP 的发包因信道质量导致的丢包率  $P_e = 10\%$  和两个 AP 在时间上有重叠时的发包因 SIR 比较小而均失败的假设情况下, 对该两 BSS 不互听系统进行建模, 用数值分析方法求解, 评估系统的吞吐, 并编写仿真器验证模型精确度。

**问题四:** 在三 BSS 场景下, AP1 与 AP3 不互听, AP2 与两者都互听。根据本小题给定的参数, 结合 AP1 和 AP3 发包时间重叠时 SIR 较大, 两者发送均成功的假设, 对该三 BSS 系统进行建模, 用数值分析方法求解, 评估系统的吞吐, 并编写仿真器验证模型精确度。

基于上述问题分析, 为了实现网络传输系统的全面性能评估分析, 本文基于 DCF 机制, 主要针对多 BSS 中存在的同频干扰问题, 研究在以两 BSS 和三 BSS 为代表的多 BSS 场景中同频节点能否互听、以及数据并发传输是否成功等多种情况下的时隙状态分布和系统吞吐变化。针对不同问题中给出的信号传输条件依次进行信道状态转换和时长分析、基于 Markov chain 的数据传输过程建模和基于概率分布的系统吞吐建模。然后代入具体参数, 给出数值求解结果。最后, 基于 MATLAB 编写仿真器进行数值验证和误差分析, 进一步验证模型精确性。



### 1.2.2 相关缩略语和已知参数说明

表 1.1 相关缩略语列表

缩略语	全称	中文意义
WLAN	wireless local area network	无线局域网
BSS	basic service set	基本服务集
AP	access point	无线接入点
STA	station	站点
DCF	distributed coordination function	分布式协调功能
CCA clear	channel assessment	信道可用评估
CSMA/CA	carrier sense multi-access and collision avoidance	载波监听多址接入/退避
CW	contention window	竞争窗口
RSSI	received signal strength indication	接收信号能量强度
SIR	signal to interference ratio	信干比
MAC	medium access control	媒体控制
PHY	physical	物理层
DIFS	DCF inter-frame space	DCF 帧间距
SIFS	short inter-frame space	短帧间距
ACK	Acknowledgement	确认
ACKTimeout	Acknowledgement timeout	确认超时

## 2 模型假设

1. 本文数据传输仅考虑下行方向，即 BSS 中只有 AP 向其关联的 STA 发送数据；
2. 在问题没有另外说明丢包率的情况下，本文默认信道为理想信道，即不会因信道质量差导致丢包；
3. 在移动通信网中，干扰是限制系统性能的主要因素，包括同频干扰、邻道干扰、互调干扰等。本文仅考虑同频干扰下的建模问题；
4. 数据帧的有效载荷传输时长和发生冲突时较长数据帧的有效载荷传输时长，在本题中假设二者相等；
5. 本文中不考虑环境噪声的影响，干扰信号仅来自于相邻 BSS。

### 3 符号说明

符号	意义
$T_e$	空闲时隙的长度
$T_s$	成功传输时隙的长度
$T_c$	碰撞时隙的长度
$P_e$	某一时隙为空闲时隙的概率
$P_s$	某一时隙节点成功发送数据的概率
$P_c$	某一时隙节点未成功发送数据的概率
$P_{el}$	因信道质量影响导致丢包的概率
$E[P]$	数据帧的有效载荷传输时长
$PHY$	物理层头的传输时长
$MAC$	MAC 层头的传输时长
$DIFS$	DCF 帧间距时长
$SIFS$	短帧间距时长
$ACK$	确认时长
$ACK_{Timeout}$	确认超时时长
$W_0$	CW 初始值
$W_{min}$	CW 最小值
$W_{max}$	CW 最大值
$W_i$	数据发送 $i$ 次时的 CW 值
$r$	最大重传次数
$m$	最大退避阶数
$\tau$	节点在某一时刻发送数据帧的概率
$p$	条件碰撞概率
$N_c$	互听的节点总数
$N_{nonc}$	与某节点不互听的节点数
$S$	系统吞吐
$S'$	仿真得到的系统吞吐
$Rate_{PHY}$	信道的物理层速率
$V$	数据成功传输时长与单个空闲时隙的比值

## 4 问题一

### 4.1 问题分析

问题一针对 WLAN 组网中 DCF 机制下的两 BSS 互听场景进行数据传输过程的分析 and 建模，实现对系统吞吐率的评估。其中，两个 BSS 中的 AP 的通信区域存在重叠（如图 1.1(a)），在使用相同信道的情況下会存在同频干扰问题。本问题中，假设并发时的 SIR 较低，导致两个 AP 的数据传输都失败。因此在只考虑下行数据传输时，同频干扰情况出现于：BSS1 中的 AP1(->STA1) 和 BSS2 中的 AP2(->STA2) 在随机退避过程中同时退避到 0 并同时发送数据，AP2->STA1 对 AP1->STA1 产生干扰，同时 AP1->STA2 对 AP2->STA2 产生干扰，最终导致 AP1->STA1 和 AP2->STA2 的 SIR 都很低，信号解调失败，即发生碰撞丢包。

问题一的核心是对 AP1 和 AP2 的随机退避过程进行仿真模拟，计算出两个 AP 同时退避到 0 而同时发送数据的概率，然后推导单个时隙长度内系统成功发送数据的有效载荷发送时长的占比，从而解算对系统吞吐。因此，本节需要分别基于 DCF 机制和 MAC 协议分析信道状态转换情况，分别推导单个时隙长度内的信道空闲时长、成功传输时长和碰撞时长。然后，利用 Bianchi 模型对两 BSS 互听场景的数据传输过程进行建模，基于 Markov 链求解节点在每个虚拟时隙的发送概率和发生碰撞的条件概率，最后基于概率分布建立系统吞吐的解算模型。同时，编写仿真器对两 BSS 互听场景的随机退避过程进行仿真模拟，获得仿真条件下的系统吞吐，从而验证模型精确度。本问题求解的思路流程图如图 4.1 所示。

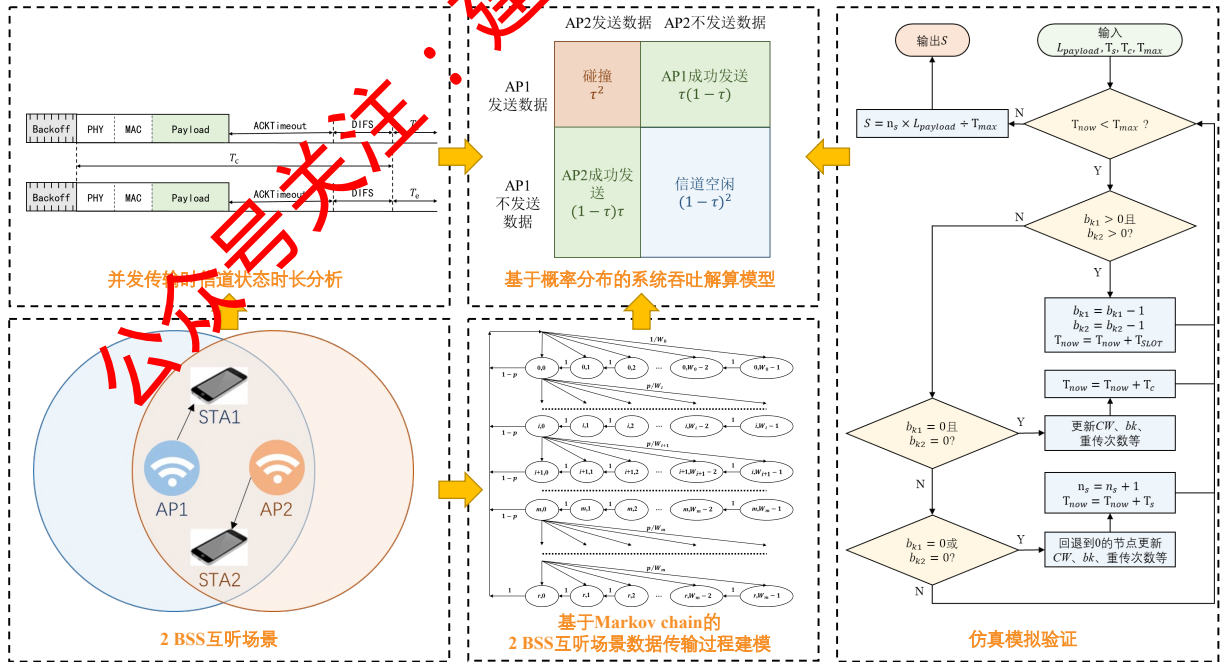


图 4.1 问题一的思路流程图

## 4.2 模型建立

### 4.2.1 DCF 机制下的信道状态分析

DCF 机制提供了一种分布式、基于竞争的信道接入功能。可将每个节点接入信道进行数据传输的过程分为 3 个阶段，信道可用评估（CCA）、随机退避、数据传输。其中，随机退避采用二进制指数退避算法确定退避时间从而有效避免冲撞，随机退避过程中信道状态时长包含如图 4.2 中所示的情况。当 AP 在随机退避过程中退避到 0，如果信道空闲，则节点开始一次数据传输，且数据帧的传输时长由 PHY 头、MAC 头和有效载荷 Payload 传输时长组成。若接收节点成功接收到数据，则回复 ACK 确认帧，AP 收到 ACK 后判断数据传输成功；若 AP 在随机退避过程中侦听到信道繁忙，则暂停随机退避，直到信道在一个时间段内被检测为空闲，再接着暂停前的退避数继续退避。退避到 0 时，开始发送数据。

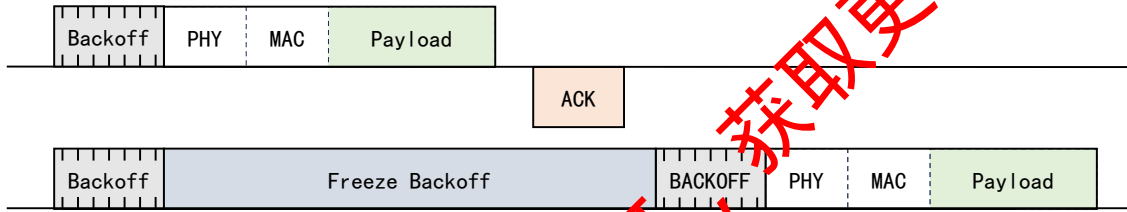


图 4.2 随机退避过程信道状态示意

根据 Bianchi 基于 Markov chain 的模型假设，DCF 机制下的信道可能处于三种状态：空闲、成功传输、碰撞。将每个状态看作一个虚拟时隙，那么信道在三种虚拟时隙中转化。信道在成功发射和冲突状态的空闲时隙、成功传输时隙和碰撞时隙的帧序列示意如图 4.3 所示。将信道处于空闲时隙、成功传输时隙和碰撞时隙的长度分别记作  $T_e$ 、 $T_s$  和  $T_c$ 。其中， $T_e$  即单个退避时隙长度  $SlotTime$ ， $T_s$  和  $T_c$  可分别表示为：

$$T_s = PHY + MAC + E[P] + PIFS + ACK + DIFS, \quad (4.1)$$

$$T_c = PHY + MAC + E^*[P] + DIFS + ACK_{Timeout}. \quad (4.2)$$

其中，PHY 和 MAC 即数据帧中的 PHY 头和 MAC 头传输时长， $E[P]$  和  $E^*[P]$  分别表示数据帧的有效载荷传输时长和发生冲突时较长数据帧的有效载荷传输时长，在本题中二者相等。DIFS 为 DCF 帧间距时长，PIFS 为短帧帧间距时长，ACK 为回复确认帧时长， $ACK_{Timeout}$  为等待超时时长。

在本节中，考虑并发碰撞导致数据传输失败，则信道状态时长参照图 4.4 所示。两个 AP 在随机退避过程中同时退避到 0，并同时发送数据，且数据帧的传输时长由 PHY 头、MAC 头和有效载荷 Payload 传输时长组成。由于冲突导致发送失败，接收节点将不会回复 ACK，两个 AP 在等待  $ACK_{Timeout}$  后判断数据发送失败，侦听信道 DIFS 时长判断信道空

闲后，选择随机数退避，进入空闲时隙  $T_e$ ，准备重新传送数据。

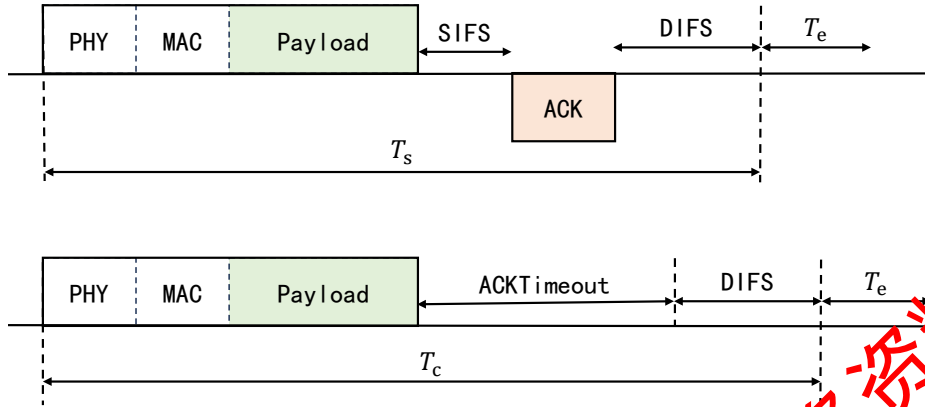


图 4.3 帧序列示意

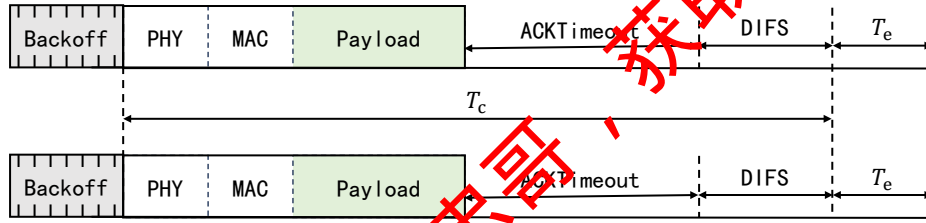


图 4.4 并发碰撞导致数据传输失败时信道状态时长示意图

#### 4.2.2 基于 Markov chain 的两 BSS 互听场景数据传输过程建模

对于由 AP1 和 STA1 构成的 BSS1 和由 AP2 和 STA2 构成的 BSS2 互听场景，在 AP1->STA1 和 AP2->STA2 的数据下传过程中，DCF 帧间距时长为  $DIFS$ ，短帧帧间距时长为  $SIFS$ ，单个退避时隙长度为  $SlotTime$ ，回复确认帧时长为  $ACK$ ，等待超时时长  $ACK_{Timeout}$ 。AP1 和 AP2 的 CW 初始值均为  $W_0$ ，CW 最大值为  $W_{max}$ ，每次数据传输失败后重传数据帧时，CW 翻倍。如果 CW 达到了  $W_{max}$ ，则保持此值，直到被重置为止。每次数据传输成功时 CW 重置，开始下一个数据帧的退避。若传输连续失败，重传次数达到  $r$  后，数据帧被丢弃，CW 重置传输下一个数据帧。可知，重传  $r$  次时，无论成功还是失败，CW 都会重置。因此，CW 可以用式 (4.3) 表示：

$$\begin{cases} W_i = 2^i W_0, & 0 \leq i \leq m, \\ W_i = 2^m W_0, & m \leq i \leq r. \end{cases} \quad (4.3)$$

其中， $i$  表示一个数据的发送次数，即阶数； $r$  为最大重传次数； $m$  为最大退避阶数，即  $m = \log_2 (W_{max}/W_0)$ 。

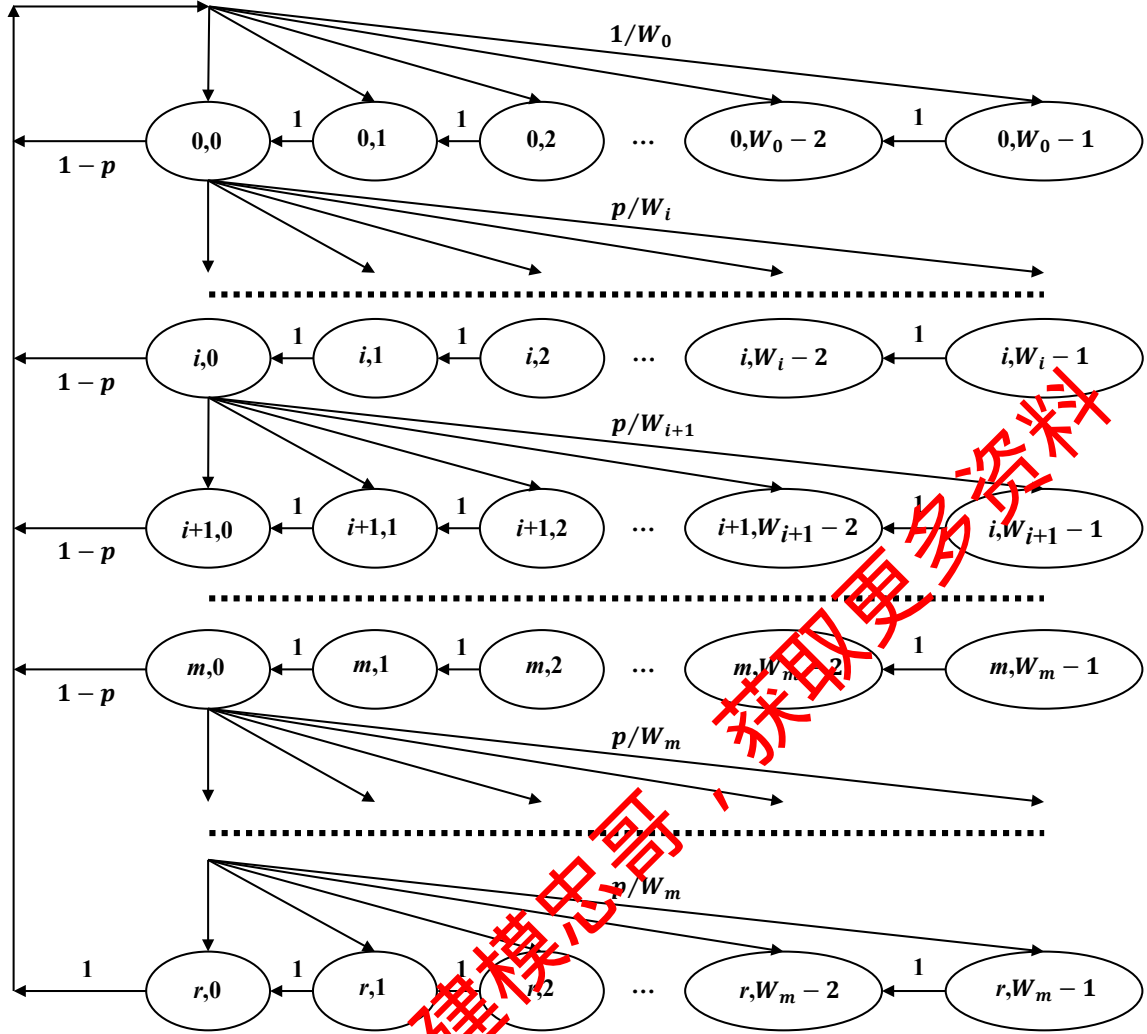


图 4.5 DCF 机制存在同频干扰的 Markov chain 模型

令  $s(t)$  表示 AP 在退避随机过程的退避阶数,  $b(t)$  表示 AP 在退避随机过程的退避计数, 其中  $t$  表示一个离散的虚拟时隙的开始时刻。二维  $\{s(t), b(t)\}$  随机过程可以用二维 Markov chain 表示, 如图 4.5 所示。定义  $b_{i,k}$  为某一时隙的任一状态, 其中  $i \in [0, m]$ ,  $k \in [0, W_i - 1]$ , 则  $b_{i,k} = \lim_{t \rightarrow \infty} P\{s(t) = i, b(t) = k\}$  代表二维 Markov chain 的稳态解。定义  $p$  为某个时隙发生碰撞的概率, Markov chain 中的一步状态转移情况可分为以下四种:

(1) 未达到重传上限时, AP 的退避计数器在每个空闲时隙的开始时刻减 1, 且概率为 1, 可表示为:

$$P\{i, k | i, k + 1\} = 1, k \in [0, W_i - 2], i \in [0, r]. \quad (4.4)$$

(2) 未达到重传上限时, 当一个数据成功传输后, 新到达的数据在  $[0, W_0 - 1]$  中选一个随机数进行退避, 每个随机数被选中的概率相等, 该状态转移情况的概率可以表示为:

$$P\{0, k | i, 0\} = \frac{1-p}{W_0}, k \in [0, W_0 - 1], i \in [0, r]. \quad (4.5)$$



(3) 未达到重传上限时，当一个数据第  $i-1$  次传输过程发生碰撞，节点进入第  $i$  阶退避过程，并在  $[0, W_i - 1]$  中选一个随机数进行退避，每个随机数被选中的概率相等，该状态转移情况的概率可以表示为：

$$P\{i, k \mid i-1, 0\} = \frac{p}{W_i}, k \in [0, W_i - 1], i \in [1, r]. \quad (4.6)$$

(4) 当 AP 到达最大的传输次数以后，无论数据成功传输还是发生碰撞，CW 都会重置，该状态转移情况的概率可以表示为：

$$P\{0, k \mid r, 0\} = \frac{1}{W_0}, k \in [0, W_0 - 1]. \quad (4.7)$$

对于两个 AP 同时退避到 0 而同时发送数据导致的同频干扰现象，从图4.5中可以看出，状态  $b_{i-1,0}$  到状态  $b_{i,0}$  的步长包括： $b_{i-1,0} \rightarrow b_{i,0}$ ,  $b_{i-1,0} \rightarrow b_{i,1} \rightarrow b_{i,0}$ ,  $\dots$ ,  $b_{i-1,0} \rightarrow b_{i,W_i-1} \rightarrow \dots \rightarrow b_{i,1} \rightarrow b_{i,0}$ ，共  $W_i$  种，求和可得：

$$\begin{aligned} b_{i,0} &= b_{i-1,0} \times \left( p \times \frac{1}{W_i} + p \times \frac{1}{W_i} \times 1 + \dots + p \times \frac{1}{W_i} \times 1^{W_i-1} \right) \\ &= p \times b_{i-1,0} \\ &= p^i \times b_{0,0}. \end{aligned} \quad (4.8)$$

其中， $0 < i \leq r$ 。

同理，对于任一状态  $b_{i,k}$ ，若  $0 < i \leq r$ ，则是从一次同频干扰导致的发送失败的状态，通过 CW 加倍之后转移过来的。若  $i = 0$ ，则是从任一阶发送成功，或达到重传次数限制后转移过来的。因此有

$$b_{i,k} = \begin{cases} b_{i-1,0} \times p \times \frac{W_i-k}{W_i}, & 0 < i < r, \\ (1-p) \times \frac{W_i-k}{W_i} \times \sum_{j=0}^{r-1} b_{j,0} + \frac{W_i-k}{W_i} \times b_{r,0}, & i = 0. \end{cases} \quad (4.9)$$

将公式(4.8)代入公式(4.9)，可得：

$$b_{i,k} = \frac{W_i - k}{W_i} \times b_{i,0}, 0 < i \leq r, 0 \leq k \leq W_i - 1. \quad (4.10)$$

根据 Markov chain 的性质，所有稳态的概率之和为 1，因此有：

$$\sum_{k=0}^{W_i-1} \sum_{i=0}^r b_{i,k} = \sum_{i=0}^r b_{i,0} \sum_{k=0}^{W_i-1} \frac{W_i - k}{W_i} = \sum_{i=0}^r b_{i,0} \frac{W_i + 1}{2} = 1. \quad (4.11)$$

根据式(4.3)和式(4.11)可以求得：

$$b_{0,0} = \begin{cases} \frac{2(1-p)(1-2p)}{(1-2p)(1-p^{r+1})+W_0(1-p)(1-(2p)^{r+1})}, & r \leq m, \\ \frac{2(1-p)(1-2p)}{W_0(1-(2p)^{m+1})(1-p)+(1-2p)(1-p^{r+1})+W_0 2^m p^{m+1}(1-p^{r-m})(1-2p)}, & m < r. \end{cases} \quad (4.12)$$

节点随机退避到 0 时发送数据，因此节点在一个时隙发送数据帧的概率为：

$$\tau = \sum_{i=0}^r b_{i,0} = b_{0,0} \times \frac{1-p^{r+1}}{1-p}. \quad (4.13)$$

当该节点在一个时隙发送数据帧时，若还有至少有另外一个节点也在该时隙传输数据，则会发生碰撞导致丢包。定义节点数为  $N$ ，则条件碰撞概率  $p$  可表示为：

$$p = 1 - (1 - \tau)^{N-1}. \quad (4.14)$$

#### 4.2.3 基于概率分布的系统吞吐解算模型

针对本问题研究的两 BSS 共听场景，节点总数  $N = 2$ ，基于公式(4.14)则知任一节点在一个时隙发送数据帧时的碰撞概率  $p = \tau$ 。因此，可获得如图4.6所示的节点传输数据状态概率分布图。其中，当 AP1 和 AP2 同时在某一个时隙发送数据时，发生同频干扰导致碰撞丢包，概率  $P_c = \tau^2$ ；当某一个时隙仅有 AP1 或 AP2 其中一个节点发送数据时，可实现数据的有效传输，概率  $P_s = 2\tau(1-\tau)$ ；当某一个时隙 AP1 和 AP2 均不发送数据帧时，信道处于空闲状态，概率  $P_e = (1-\tau)^2$ 。

		AP2发送数据	AP2不发送数据
AP1发送数据		碰撞 $\tau^2$	AP1成功发送 $\tau(1-\tau)$
AP1不发送数据		AP2成功发送 $(1-\tau)\tau$	信道空闲 $(1-\tau)^2$

图 4.6 AP1 和 AP2 传输数据状态概率分布图

因此，在两 BSS 共听场景中系统的吞吐  $S$  可以表示为：

$$S = \frac{P_s \cdot E[P]}{P_c \cdot T_c + P_s \cdot T_s + P_e \cdot T_e} \times Rate_{PHY} \quad (4.15)$$

$$= \frac{2\tau(1-\tau) \cdot E[P]}{\tau^2 \cdot T_c + 2\tau(1-\tau) \cdot T_s + (1-\tau)^2 \cdot T_e} \times Rate_{PHY}.$$

其中， $Rate_{PHY}$  为信道的物理层速率， $\tau$  可由公式(4.13)和(4.14)构成的非线性方程组求得唯一解。

### 4.3 问题求解

#### 4.3.1 基于 Markov chain 的系统吞吐评估

基于问题一给定信息和相关参数表可知，在该两 BSS 互听场景中，DCF 帧间距时长  $DIFS = 43\mu s$ ，短帧帧间距时长  $SIFS = 16\mu s$ ，单个退避时隙长度  $SlotTime = 9\mu s$ ，回复确认帧时长  $ACK = 32\mu s$ ，等待超时  $ACK_{Timeout} = 65\mu s$ ，PHY 头时长  $PHY = 13.6\mu s$ ，物理层速率  $Rate_{PHY} = 455.8Mbps$ 。

则有 AP 发送包的载荷发送时长：

$$E[P] = \frac{1500 \times 8}{455.8} = 26.3273\mu s. \quad (4.16)$$

MAC 头时长：

$$MAC = 30 \times 8 / 455.8 = 0.5265\mu s. \quad (4.17)$$

则信道空闲时长  $T_e$ 、成功传输时长  $T_s$  和碰撞的传输时长  $T_c$  分别为：

$$T_e = SlotTime = 9\mu s. \quad (4.18)$$

$$T_s = PHY + MAC + E[P] + SIFS + ACK + DIFS = 131.4538\mu s. \quad (4.19)$$

$$T_c = PHY + MAC + E[P] + DIFS + ACK_{Timeout} = 148.4538\mu s. \quad (4.20)$$

然后，DCF 机制的随机退避过程中，最大重传次数给定为  $r = 32$ ，AP 的 CW 初始值给定为  $W_0 = 16$ ，CW 最大值给定为  $W_{max} = 1024$ ，则 AP 的最大退避阶数为：

$$m = \log_2 \frac{W_{max}}{W_0} = 6. \quad (4.21)$$

因此，在本问题中， $m < r$ ，信道总 AP 节点数为  $N = 2$ ，根据公式 (4.12)，(4.13) 和 (4.14) 构成的非线性方程组，可以解算节点在一个时隙发送数据的概率  $\tau$  和条件碰撞概率  $p$  的唯

一解为：

$$\tau = p = 0.1046. \quad (4.22)$$

可知，AP1 和 AP2 同时在一个时隙发生同频干扰导致碰撞丢包的概率为  $P_c = \tau^2 = 0.0109$ ；某一个时隙仅有 AP1 或 AP2 其中一个节点实现数据的有效传输的概率为  $P_s = 2\tau(1 - \tau) = 0.1874$ ；某一个时隙 AP1 和 AP2 均不发送数据帧使信道处于空闲状态的概率为  $P_e = (1 - \tau)^2 = 0.8017$ 。

因此，基于公式(4.15)可算得系统的吞吐：

$$S = \frac{P_s \cdot E[P]}{P_c \cdot T_c + P_s \cdot T_s + P_e \cdot T_e} \times Rate_{PHY} = 67.1743 \text{Mbps}. \quad (4.23)$$

#### 4.3.2 仿真验证及结果分析

进一步编写仿真器对 AP1 和 AP2 的随机退避过程和信道的状态转换进行仿真模拟。具体仿真算法流程如下。根据算法 1 基于 MATLAB 编写仿真程序，对时长  $T_{\max} = 1000\text{s}$  内进行节点数据传输的仿真模拟，详细代码见附录A.1。基于仿真模拟结果计算的成功传输传输时长  $T_s = 131.4539\mu\text{s}$ ，碰撞传输时长  $T_c = 148.4539\mu\text{s}$ ，系统吞吐  $S = 65.2780 \text{Mbps}$ 。系统吞吐仿真结果与模型解算结果的误差仅为 2.9%，可以说明系统吞吐解算模型的具有一定的精确性。

---

##### 算法 1 问题一仿真算法

---

数据：系统通用参数

输入：载荷长度  $L_{\text{payload}}$ ，成功传输传输时长  $T_s$ ，碰撞传输时长  $T_c$ ，仿真时长  $T_{\max}$

输出：吞吐  $S$

变量：当前仿真时刻  $T_{\text{now}}$ ，成功传输次数  $n_s$

```

1: while  $T_{\text{now}} < T_{\max}$  do
2:   if  $b_{k1} > 0$  and  $b_{k2} > 0$  then
3:      $b_{k1} \leftarrow b_{k1} - 1$ 
4:      $b_{k2} \leftarrow b_{k2} - 1$ 
5:      $T_{\text{now}} \leftarrow T_{\text{now}} + T_{\text{SLOT}}$ 
6:   else if  $b_{k1} = 0$  and  $b_{k2} = 0$  then
7:     更新  $CW$ 、 $bk$ 、重传次数等
8:      $T_{\text{now}} \leftarrow T_{\text{now}} + T_c$ 
9:   else if  $b_{k1} = 0$  or  $b_{k2} = 0$  then
10:    退避到 0 的节点更新  $CW$ 、 $b_k$ 、重传次数等
11:     $n_s \leftarrow n_s + 1$ 
12:     $T_{\text{now}} \leftarrow T_{\text{now}} + T_s$ 
13:  $S \leftarrow n_s \times L_{\text{payload}} \div T_{\max}$ 
14: return  $S$ 

```

---

## 5 问题二

### 5.1 问题分析

问题二同问题一，需要考虑在两个 BSS 中 AP 存在通信区域重叠情况下，针对 WLAN 组网中 DCF 机制下的两 BSS 互听场景（如图 1.1(a)）进行数据传输过程的分析和建模，实现对系统吞吐率的评估。与问题一不同的是，在本小题中，假设并发时两个终端接收到数据的 SIR 较高，故而两个 AP 的数据传输都能成功。因此在只考虑下行数据传输时，两个 AP 并发时数据传输不受同频干扰影响，即 BSS1 中的 AP1( $\rightarrow$ STA1) 和 BSS2 中的 AP2( $\rightarrow$ STA2) 在随机退避过程中同时退避到 0 并发送数据，信号都能被成功解调，从而两个 AP 的数据都能成功传输。

问题二的核心与问题一类似，对 AP1 和 AP2 的随机退避过程建立数学模型，计算出两个 AP 在随机退避过程中同时退避到 0 发送数据的概率，然后推导出单个时隙长度内系统成功发送数据的有效载荷发送时长的占比，从而求解系统的吞吐。与问题一不同的是，在本小题中假设两个 AP 并发时数据传输都能成功。本问题求解的思路流程图如图 5.1 所示。

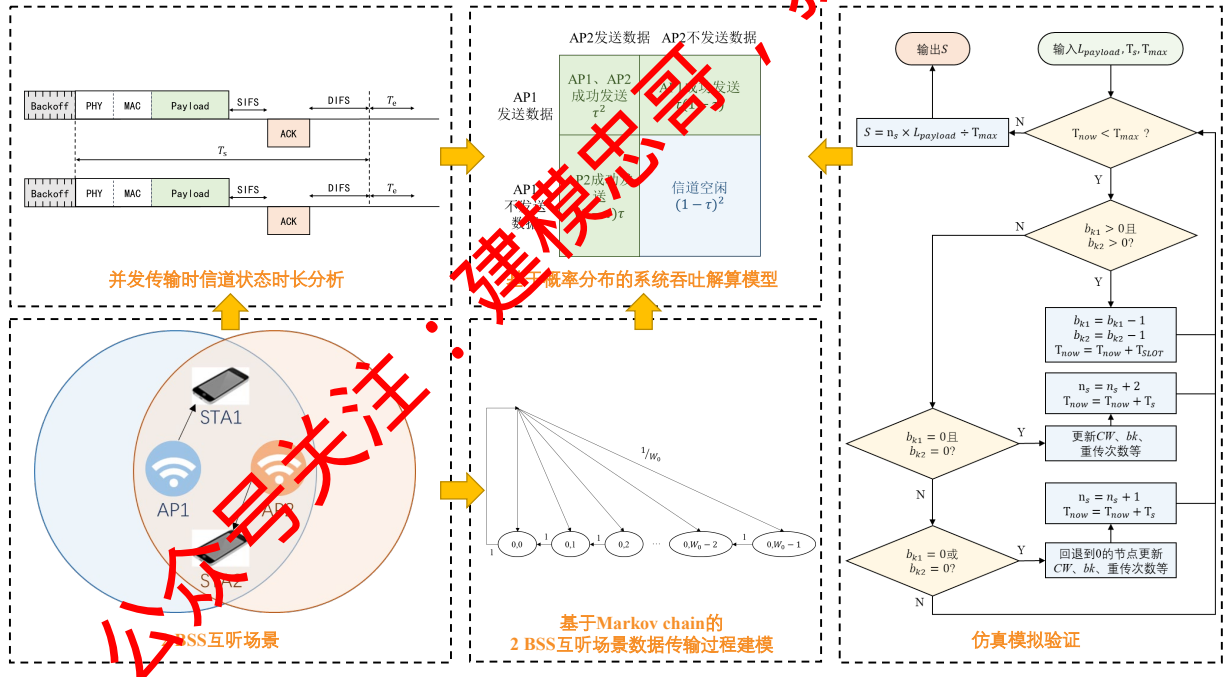


图 5.1 问题二的思路流程图

### 5.2 模型建立

#### 5.2.1 考虑并发碰撞的信道状态时长分析

在本节中，考虑并发碰撞时两个 AP 都能成功传输数据，则信道状态时长参照图 5.2 所示。两个 AP 在随机退避过程中同时退避到 0，并同时发送数据，且数据帧的传输时长由 PHY 头、MAC 头和有效载荷 Payload 传输时长组成。由于发生碰撞后二者都能成功传输

数据，对应接收节点成功接收到数据之后等待  $SIFS$  后，回复 ACK 确认帧。AP 收到 ACK 后判断数据传输成功，开始侦听信道 DIFS 时长判断信道空闲后，选择随机数退避，进入空闲时隙  $T_e$ ，准备下一次数据传送。

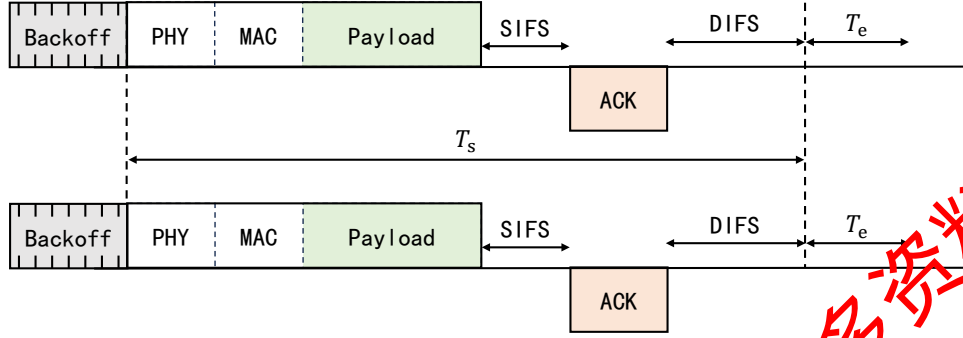


图 5.2 并发碰撞数据传输成功时信道状态时长示意图

### 5.2.2 基于 Markov chain 的两 BSS 互听场景数据传输过程建模

问题二在问题一的基础上假设：并发时两个终端接收到数据的 SIR 较高，两个 AP 的数据传输都能成功。因此可知，每个 AP 在随机退避过程中退避到 0 时一定能成功发送数据，不存在同频干扰导致的数据传输失败。所以，本问题中 AP1 和 AP2 的 CW 窗长度始终保持为  $W_0$ 。

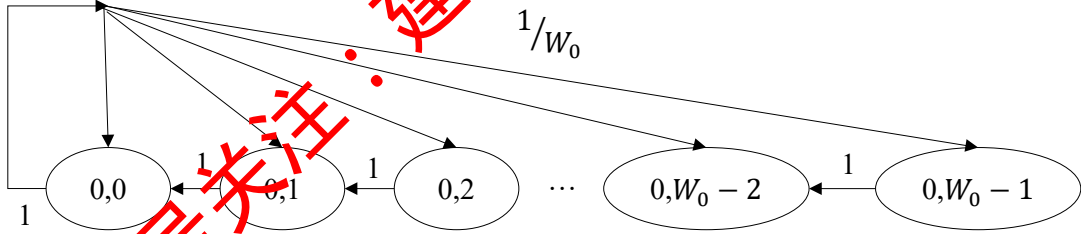


图 5.3 DCF 机制不存在同频干扰的 Markov chain 模型

由于本问题中 AP 一定不会发生数据重传，则 AP 的退避阶数  $s(t) = 0$ ，退避计数  $b(t) \in [0, W_0 - 1]$ . 重新定义  $b_{0,k}$  为某一时隙的任一状态，其中  $k \in [0, W_0 - 1]$ . 该随机过程的 Markov chain 如图 5.3 表示，二维 Markov chain 的稳态解为  $b_{0,k} = \lim_{t \rightarrow \infty} P\{b(t) = k\}$ . Markov chain 中的一步状态转移情况可分为以下两种：

- (1) 退避计数器在每个空闲时隙的开始时刻减 1，该状态转移的概率为 1，可表示为

$$P\{0, k \mid 0, k + 1\} = 1, k \in [0, W_0 - 2]. \quad (5.1)$$

- (2) 当一个数据成功传输后，新到达的数据在  $[0, W_0 - 1]$  中选一个随机数进行退避，每



个随机数被选中的概率相等，该状态转移情况的概率可以表示为：

$$P\{0, k \mid 0, 0\} = \frac{1}{W_0}, k \in [0, W_0 - 1]. \quad (5.2)$$

任一状态  $b_{0,k}$  都是由 AP 退避到 0 并成功发送数据后转移过来的。状态  $b_{0,0}$  到状态  $b_{0,k}$  的步长包括： $b_{0,0} \rightarrow b_{0,k}$ ,  $b_{0,0} \rightarrow b_{0,k+1} \rightarrow b_{0,k}$ ,  $\dots$ ,  $b_{0,0} \rightarrow b_{0,W_0-1} \dots \rightarrow b_{0,k+1} \rightarrow b_{0,k}$ , 共  $W_0 - k$  种，求和可得：

$$\begin{aligned} b_{0,k} &= b_{0,0} \times \left( \frac{1}{W_0} + \frac{1}{W_0} \times 1 + \dots + \frac{1}{W_0} \times 1^{W_0-1-k} \right) \\ &= \frac{W_0 - k}{W_0} \times b_{0,0}. \end{aligned} \quad (5.3)$$

其中， $0 \leq k \leq W_0 - 1$ .

根据 Markov chain 的性质，所有稳态的概率之和为 1，因此有：

$$\sum_{k=0}^{W_0-1} b_{0,k} = \sum_{k=0}^{W_0-1} \frac{W_0 - k}{W_0} b_{0,0} = 1. \quad (5.4)$$

根据公式(5.4)可以求得：

$$b_{0,0} = \frac{2}{W_0 + 1}. \quad (5.5)$$

节点随机退避到 0 时发送数据，因此节点在一个时隙发送数据帧的概率为：

$$\tau = b_{0,0} = \frac{2}{W_0 + 1}. \quad (5.6)$$

### 5.2.3 基于概率分布的系统吞吐解算模型

图5.4为节点传输数据状态概率分布图。其中，当 AP1 和 AP2 同时在某一个时隙发送数据时，均可成功发送数据，概率  $P_{s1} = \tau^2$ ；当某一个时隙仅有 AP1 或 AP2 其中一个节点发送数据时，可实现数据的有效传输，概率  $P_{s2} = 2\tau(1 - \tau)$ ；当某一个时隙 AP1 和 AP2 均不发送数据帧时，信道处于空闲状态，概率  $P_e = (1 - \tau)^2$ 。其中，需要注意的是，与仅有 AP1 或 AP2 其中一个节点发送数据相比，当 AP1 和 AP2 同时在某一个时隙发送数据时，信道传输的有效载荷是双倍的。

因此，在两 BSS 共听场景中系统的吞吐  $S$  可以表示为：

$$\begin{aligned} S &= \frac{(2P_{s1} + P_{s2}) \cdot E[P]}{(P_{s1} + P_{s2}) \cdot T_s + P_e \cdot T_e} \times Rate_{PHY} \\ &= \frac{2\tau \cdot E[P]}{\tau(2 - \tau) \cdot T_s + (1 - \tau)^2 \cdot T_e} \times Rate_{PHY}. \end{aligned} \quad (5.7)$$

	AP2发送数据	AP2不发送数据
AP1 发送数据	AP1、AP2 成功发送 $\tau^2$	AP1成功发送 $\tau(1 - \tau)$
AP1 不发送 数据	AP2成功发 送 $(1 - \tau)\tau$	信道空闲 $(1 - \tau)^2$

图 5.4 AP1 和 AP2 传输数据状态概率分布图

其中， $Rate_{PHY}$ 、 $E[P]$ 、 $T_s$  和  $T_e$  的计算模型均与问题一相同。

### 5.3 问题求解

#### 5.3.1 基于 Markov chain 的系统吞吐评估

问题二在问题一给定的参数基础上，修改了物理层速率  $Rate_{PHY} = 275.3\text{Mbps}$ ，其他参数不变。则有 AP 发送包的载荷发送时长：

$$E[P] = \frac{4500 * 8}{275.3} = 43.5888\mu\text{s}. \quad (5.8)$$

MAC 头时长：

$$MAC = 30 * 8 / 275.3 = 0.8718\mu\text{s}. \quad (5.9)$$

则成功传输时长  $T_s$  为：

$$T_s = T_{PHY} + MAC + E[P] + SIFS + ACK + DIFS = 149.0606\mu\text{s}. \quad (5.10)$$

根据公式(5.6)可以解算节点在一个时隙发送数据的概率  $\tau$  为：

$$\tau = p = 0.1176. \quad (5.11)$$

可知，AP1 和 AP2 同时在某一个时隙均成功发送数据的概率为  $P_{s1} = \tau^2 = 0.0138$ ；某一个时隙仅有 AP1 或 AP2 其中一个节点实现数据的有效传输的概率为  $P_{s2} = 2\tau(1 - \tau) = 0.2076$ ；某一个时隙 AP1 和 AP2 均不发送数据帧使信道处于空闲状态的概率为  $P_e = (1 -$

$$\tau)^2 = 0.7785.$$

因此，基于公式(5.7)可算得系统的吞吐：

$$S = \frac{(2P_{s1} + P_{s2}) \cdot E[P]}{(P_{s1} + P_{s2}) \cdot T_s + P_e \cdot T_e} \times Rate_{PHY} = 70.5585 \text{Mbps}. \quad (5.12)$$

### 5.3.2 仿真验证及结果分析

进一步编写仿真器对 AP1 和 AP2 的随机退避过程和信道的状态转换进行仿真模拟。具体仿真算法流程如下。根据算法 2 基于 MATLAB 编写仿真程序，对时长  $T_{\max} = 1000\text{s}$  内进行节点数据传输的仿真模拟，详细代码见附录。基于仿真模拟结果计算的成功传输传输时长  $T_s = 149.0606\mu\text{s}$ ，碰撞传输时长  $T_e = 166.0606\mu\text{s}$ ，系统吞吐  $S = 68.9442\text{Mbps}$ 。系统吞吐仿真结果与模型解算结果的误差仅为 2.34%，可以说明系统吞吐解算模型的具有一定的精确性。

---

#### 算法 2 问题二仿真算法

---

**数据:** 系统通用参数

**输入:** 载荷长度  $L_{\text{payload}}$ , 成功传输传输时长  $T_s$ , 仿真时长  $T_{\max}$

**输出:** 吞吐  $S$

**变量:** 当前仿真时刻  $T_{\text{now}}$ , 成功传输次数  $n_s$

```

1: while  $T_{\text{now}} < T_{\max}$  do
2:   if  $b_{k1} > 0$  and  $b_{k2} > 0$  then
3:      $b_{k1} \leftarrow b_{k1} - 1$ 
4:      $b_{k2} \leftarrow b_{k2} - 1$ 
5:      $T_{\text{now}} \leftarrow T_{\text{now}} + T_{\text{SLOT}}$ 
6:   else if  $b_{k1} = 0$  and  $b_{k2} = 0$  then
7:     更新  $CW$ 、 $b_k$ 、重传次数等
8:      $n_s \leftarrow n_s + 2$ 
9:      $T_{\text{now}} \leftarrow T_{\text{now}} + T_s$ 
10:  else if  $b_{k1} = 0$  or  $b_{k2} = 0$  then
11:    退避到 0 的节点更新  $CW$ 、 $b_k$ 、重传次数等
12:     $n_s \leftarrow n_s + 1$ 
13:     $T_{\text{now}} \leftarrow T_{\text{now}} + T_s$ 
14:   $S \leftarrow n_s \times L_{\text{payload}} \div T_{\max}$ 
15: return  $S$ 

```

---

## 6 问题三

### 6.1 问题分析

问题三考虑在 AP 密集部署时同频 AP 之间的距离远，导致 AP 间 RSSI 低于 CCA 门限，即 WLAN 组网中 DCF 机制下的两 BSS 不互听场景下进行数据传输过程的分析和建模，实现对系统吞吐率的评估。具体而言，由于两 BSS 不互听，在下行数据传输时，AP1 和 AP2 之间通信区域无交集（如图 1.1(b)），向各自关联的 STA1 和 STA2 传输数据时有很大概率出现二者同时或先后开始发送数据的情况，即二者发包在时间上有重叠。若 AP1 或 AP2 位于 STA2 或 STA1 的通信区域内，后两者可能先后接收到信号包和干扰包。此时，节点 AP1 或 AP2 相对于 STA2 或 STA1 形成隐藏节点，当 SIR 较小时信号包无法解调，干扰数据传输，即存在邻 BSS 干扰。本问题中，假设仅有单个 AP 发送数据时存在因信道质量导致的丢包，且丢包率  $P_e = 10\%$ ，两个 AP 发送数据在时间上有重叠时 SIR 比较小，导致两个 AP 的发包均失败。因此在只考虑下行数据传输时，当 BSS1 中的 AP1( $\rightarrow$ STA1) 和 BSS2 中的 AP2( $\rightarrow$ STA2) 在随机退避过程中退避到 0 并发送数据的时间有重叠时，存在邻 BSS 干扰，AP1 $\rightarrow$ STA1 和 AP2 $\rightarrow$ STA2 的 SIR 都比较小，信号解调失败，从而导致两个 AP 的发包均失败。在不存在邻 BSS 干扰时，AP1 $\rightarrow$ STA1 或 AP2 $\rightarrow$ STA2 由于信道质量存在概率  $P_e = 10\%$  的丢包。问题三的核心是对 AP1 和 AP2 的随机退避过程建立数学模型，在计算出两个 AP 在随机退避过程中发生并发传输重叠概率的基础上，推导出单个时隙长度内系统成功发送数据的有效载荷发送时长的占比，从而求解系统的吞吐。本问题求解的思路流程图如图 6.1 所示。

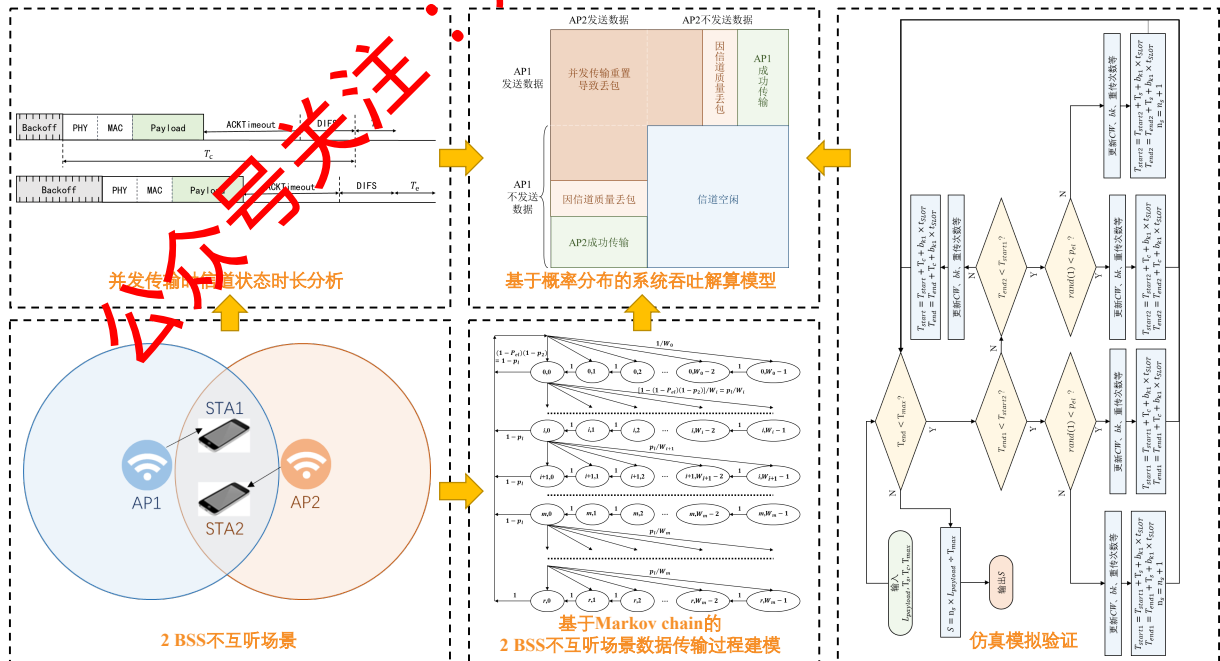


图 6.1 问题三的思路流程图

## 6.2 模型建立

### 6.2.1 考虑传输时间上有重叠时的信道状态时长分析

在本节中，考虑两个 AP 在数据传输的时间有重叠时，SIR 比较小，导致两个 AP 的发包均失败，则信道状态时长参照图 6.2 所示。两个 AP 在随机退避过程中先后退避到 0，且在重叠的时间内依次发送数据，且数据帧的传输时长由 PHY 头、MAC 头和有效载荷 Payload 传输时长组成。由于有时间重叠时两者发包均失败，接收节点将不会回复 ACK，两个 AP 在分别等待  $ACK_{Timeout}$  后判断数据发送失败，侦听信道 DIFS 时长判断信道空闲后，选择随机数退避，进入空闲时隙  $T_e$ ，准备重新传送数据。

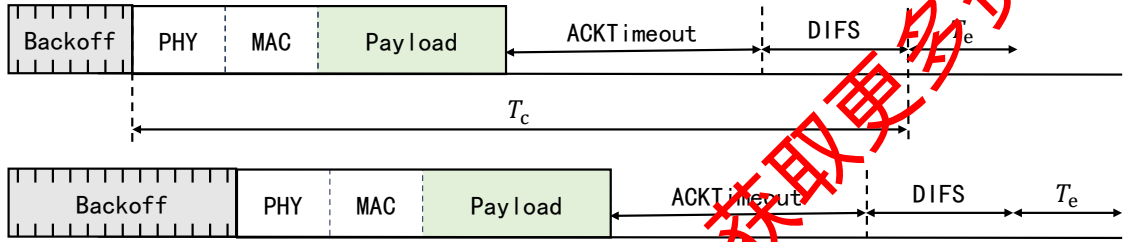


图 6.2 并发传输重叠时信道状态时长示意图

### 6.2.2 基于 Markov chain 的两 BSS 不互听场景数据传输过程建模

对于问题三中研究的两 BSS 不互听场景，在 AP1->STA1 和 AP2->STA2 的数据下传过程中，每次数据传输失败后重传数据帧时，CW 翻倍。如果 CW 达到了  $W_{max}$ ，则保持此值，直到被重置为止。每次数据传输成功时 CW 重置，开始下一个数据帧的退避。若传输连续失败，重传次数达到  $r$  后，数据帧被丢弃，CW 重置传输下一个数据帧。因此，CW 仍然可以用问题一中的式 (4.3) 表示。

本问题中的二维  $\{s(t), b(t)\}$  随机过程的 Markov chain 模型如图 6.3 所示。定义  $b_{i,k}$  为某一时隙的任一状态，其中  $i \in [0, m]$ ， $k \in [0, W_i - 1]$ ，定义  $P_{el}$  为因信道质量导致的丢包概率， $p_2$  为某一时隙两个 AP 发包时间重叠导致发包失败的概率，Markov chain 中的一步状态转移情况可分为以下四种：

(1) 未达到重传上限时，AP 的退避计数器在每个空闲时隙的开始时刻减 1，且概率为 1，可表示为：

$$P\{i, k \mid i, k + 1\} = 1, k \in [0, W_i - 2], i \in [0, r]. \quad (6.1)$$

(2) 未达到重传上限时，当一个数据成功传输后，新到达的数据在  $[0, W_0 - 1]$  中选一个随机数进行退避，每个随机数被选中的概率相等，该状态转移情况的概率可以表示为：

$$P\{0, k \mid i, 0\} = \frac{(1 - P_{el})(1 - p_2)}{W_0}, k \in [0, W_0 - 1], i \in [0, r]. \quad (6.2)$$

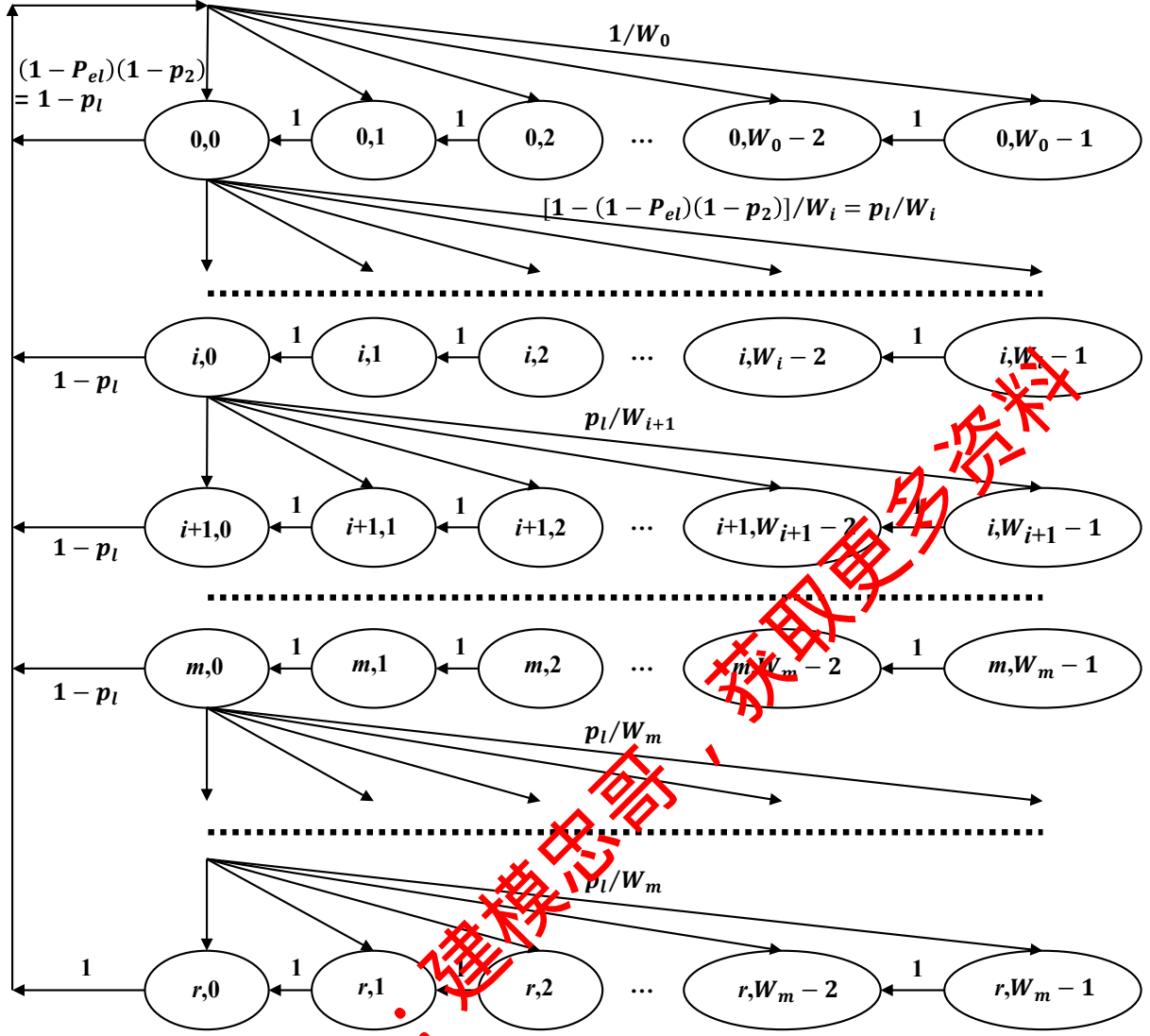


图 6.3 两 BSS 不互听场景的 Markov chain 模型

(3) 未达到重传上限时，当一个数据第  $i - 1$  次传输过程发生并发传输重叠或因信道质量影响而丢包，节点进入第  $i$  阶退避过程，并在  $[0, W_i - 1]$  中选一个随机数进行退避，每个随机数被选中的概率相等，该状态转移情况的概率可以表示为：

$$P\{i, k \mid i - 1, 0\} = \frac{1 - (1 - P_{el})(1 - p_2)}{W_i}, k \in [0, W_i - 1], i \in [1, r]. \quad (6.3)$$

(4) 当 AP 到达最大的传输次数以后，无论数据成功传输还是发生丢包，CW 都会重置，该状态转移情况的概率可以表示为：

$$P\{0, k \mid r, 0\} = \frac{1}{W_0}, k \in [0, W_0 - 1]. \quad (6.4)$$

对于两个 AP 因信道质量导致丢包或因发包时间重叠导致发包失败，状态  $b_{i-1,0}$  到状



态  $b_{i,0}$  的步长包括:  $b_{i-1,0} \rightarrow b_{i,0}$ ,  $b_{i-1,0} \rightarrow b_{i,1} \rightarrow b_{i,0}$ ,  $\dots$ ,  $b_{i-1,0} \rightarrow b_{i,W_i-1} \dots \rightarrow b_{i,1} \rightarrow b_{i,0}$ , 共  $W_i$  种, 求和可得:

$$\begin{aligned} b_{i,0} &= b_{i-1,0} \times \frac{1 - (1 - P_{el})(1 - p_2)}{W_i} \times (1 + 1 + \dots + 1^{W_i-1}) \\ &= [1 - (1 - P_{el})(1 - p_2)] \times b_{i-1,0} \\ &= [1 - (1 - P_{el})(1 - p_2)]^i \times b_{0,0}. \end{aligned} \quad (6.5)$$

其中,  $0 < i \leq r$ .

同理, 对于任一状态  $b_{i,k}$ , 若  $0 < i \leq r$ , 则是从一次因信道质量导致丢包或因发包时间重叠导致发包失败的状态, 通过 CW 加倍之后转移过来的。若  $i = 0$ , 则是从任一阶发送成功, 或达到重传次数限制后转移过来的。因此有,

$$b_{i,k} = \begin{cases} b_{i-1,0} \times [1 - (1 - P_{el})(1 - p_2)] \times \frac{W_i - k}{W_i}, & 0 < i < r, \\ (1 - P_{el})(1 - p_2) \times \frac{W_i - k}{W_i} \times \sum_{j=0}^{r-1} b_{j,0} + \frac{W_i - k}{W_i} \times b_{r,0}, & i = 0. \end{cases} \quad (6.6)$$

将公式 (4.8) 带入公式 (4.9), 可得:

$$b_{i,k} = \frac{W_i - k}{W_i} \times b_{i,0}, \quad 0 < i \leq r, \quad 0 \leq k \leq W_i - 1. \quad (6.7)$$

根据 Markov chain 的性质, 所有稳态的概率之和为 1, 因此有:

$$\sum_{k=0}^{W_i-1} \sum_{i=0}^r b_{i,k} = \sum_{i=0}^r b_{i,0} \sum_{k=0}^{W_i-1} \frac{W_i - k}{W_i} = \sum_{i=0}^r b_{i,0} \frac{W_i + 1}{2} = 1. \quad (6.8)$$

为简化公式表述, 定义条件丢包概率为  $p_l = 1 - (1 - P_{el})(1 - p_2)$ . 根据式(4.3)和式(6.8)可以求得:

$$b_{i,0} = \begin{cases} \frac{2(1-p_l)(1-2p_l)}{(1-2p_l)(1-p_l^{r+1}) + W_0(1-p_l)(1-(2p_l)^{r+1})}, & r \leq m, \\ \frac{2(1-p_l)(1-2p_l)}{W_0(1-(2p_l)^{m+1})(1-p_l) + (1-2p_l)(1-p_l^{r+1}) + W_0 2^m p_l^{m+1} (1-p_l^{r-m})(1-2p_l)}, & m < r. \end{cases} \quad (6.9)$$

节点随机退避到 0 时发送数据, 因此节点在一个时隙发送数据帧的概率为:

$$\tau_1 = \sum_{i=0}^r b_{i,0} = b_{0,0} \times \frac{1 - p_l^{r+1}}{1 - p_l}. \quad (6.10)$$

但由于两个 AP 不互听, 存在二者同时或先后开始发送数据的情况, 当两个 AP 出现如图所示的并发传输重叠时, 题目假设 SIR 比较小, 会导致两个 AP 的发包均失败。因此,

节点在一个时隙发送的数据因并发传输重叠而丢包的概率可以表征为 [9]:

$$\tau_2 = \sum_{i=0}^r \sum_{k=0}^V b_{i,k} = \begin{cases} \left[ (V+1) \frac{1-p_l^{r+1}}{1-p_l} - \frac{V(V+1)}{2W_0} \cdot \frac{1-(\frac{p_l}{2})^{r+1}}{1-\frac{p_l}{2}} \right] \cdot b_{0,0}, & V < W_0, \\ \left[ \frac{1-p_l^X}{2(1-p_l)} + \frac{W_0}{2} \cdot \frac{1-(2p_l)^X}{1-2p_l} + (V+1) \frac{p_l^X - p_l^{r+1}}{1-p_l} \right. \\ \quad \left. - \frac{V(V+1)}{2W_0} \cdot \frac{(\frac{p_l}{2})^X - (\frac{p_l}{2})^{r+1}}{1-\frac{p_l}{2}} \right] \cdot b_{0,0}, & W_{X-1} < V < W_X, \quad (6.11) \\ 1, & 1 \leq X \leq m, \\ & V > W_m. \end{cases}$$

其中,  $V$  是信号包传输时长与单个空闲时隙  $SlotTime$  的比值, 阶数  $X$  对应的窗口数  $W_X$  是比  $V$  更大的最小窗口数。公式(6.11)的三个分段分别表征三种情况:

(1) 当  $V < W_0$  时, 两个 AP 发生并发传输重叠时只会导致一次数据重传。其中,  $\tau_1$  是该情况下的一种特殊状态。因此,  $\tau_1$  可以认为是信号包传输时长小于单个空闲时隙  $SlotTime$ , 即取  $V = 0$ 。

(2) 当  $W_{X-1} < V < W_X, 1 \leq X \leq m$  时, 信号包传输时长大于初始 CW 值, 两个 AP 发生并发传输重叠可能会导致多次数据重传;

(3) 当  $V > W_{\max}$  时, 信号包传输时长大于最大 CW 值, AP 的数据发送一定会因为并发传输重叠而丢包。

当该节点在一个时隙发送数据帧时, 若还有至少另外一个互听节点也在该时隙传输数据, 则会发生碰撞导致丢包。若该节点发送数据时, 至少有另外一个不互听节点也该时隙先后发送数据, 产生并发传输重叠, 也会丢包。定义该节点的互听节点数为  $N_c$ , 不互听节点数为  $N_{\text{nonc}}$ , 则因发包时间重叠而失败的概率  $p_2$  可表示为:

$$p_2 = 1 - (1 - \tau_1)^{N_c} (1 - \tau_2)^{N_{\text{nonc}}}. \quad (6.12)$$

针对本问题研究的两 BSS 不共听场景, 任一节点的互听节点数为  $N_c = 0$ , 不互听节点数为  $N_{\text{nonc}} = 1$ , 即节点只会因为与互听节点(隐藏节点)并发传输重叠而丢包。基于公式(6.12)则知任一节点在一个时隙发送数据帧时的条件传输失败概率为:

$$\begin{aligned} p_1 &= 1 - (1 - P_{\text{el}})(1 - \tau_1)^{N_c} (1 - \tau_2)^{N_{\text{nonc}}} \\ &= 1 - (1 - P_{\text{el}})(1 - \tau_2). \end{aligned} \quad (6.13)$$

由公式(6.9)至公式(6.12)构成的关于  $\tau_1, \tau_2$  和  $p_1$  的三元非线性方程组可以求解出  $\tau_1, \tau_2$  和  $p_1$  的唯一解。

### 6.2.3 基于概率分布的系统吞吐解算模型

节点在任一时隙的状态可分为传输数据中和空闲中，由于 AP1 和 AP2 不互听，则两个节点的状态相互独立。两个节点处于相等地位，其通过信道的吞吐相同。

某一个节点在某一个时隙发送数据，不发生并发传输重叠且不因为信道质量丢包时，该节点可在这一时隙实现成功数据传输，此事件的概率为：

$$P_s = \tau_1(1 - \tau_2)(1 - P_{cl}). \quad (6.14)$$

某一个节点在某一个时隙发送数据，发生并发传输重叠，导致传输失败，此事件的概率为：

$$P_{cl} = \tau_1\tau_2. \quad (6.15)$$

某一个节点在某一个时隙发送数据，因信道质量影响差而丢包，导致传输失败，此事件的概率为：

$$P_{c2} = \tau_1(1 - \tau_2)P_{cl}. \quad (6.16)$$

某一个节点在某一个时隙不发送数据帧，此事件的概率为：

$$P_e = (1 - \tau_1). \quad (6.17)$$

某一个节点在所设场景内的吞吐  $S_0$  可以表示为：

$$S_0 = \frac{P_s \cdot E[P]}{(P_{cl} + P_{c2}) \cdot T_c + P_s \cdot T_s + P_e \cdot T_e} \times Rate_{PHY}. \quad (6.18)$$

其中， $Rate_{PHY}$ 、 $E[P]$ 、 $T_s$ 、 $T_c$  和  $T_e$  的计算模型均与问题一相同。

在两 BSS 不互听场景中，系统共包含两个不互听的节点，两个节点地位相同，则系统的吞吐  $S$  可以表示为：

$$S = 2S_0. \quad (6.19)$$

## 6.3 问题求解

### 6.3.1 基于 Markov chain 的系统吞吐评估

首先基于问题三题目中的给定信息和附录 4 的参数表进行模型求解。在该问题的两 BSS 不互听场景中，DCF 机制参数、节点各状态的时长、发送数据的载荷长度和物理层速率均与问题一中的参数一致。

$V$  的定义为节点的数据成功传输时长与单个空闲时隙的比值：

$$V = \left\lceil \frac{T_s}{SlotTime} \right\rceil_{up} = 7, \quad (6.20)$$

其中， $[\cdot]_{up}$  表示向上取整。

该系统状态下  $m < r$ ，由公式 (6.9) 至式 (6.13) 构成的关于  $\tau_1$  和  $\tau_2$  的非线性方程组可求得：

$$\tau_1 = 0.0458, \quad (6.21)$$

$$\tau_2 = 0.3524. \quad (6.22)$$

针对某一节点，由公式 (6.14) 至 (6.17) 其在某一时隙发生对应事件的概率为：

$$P_s = 0.0298, \quad (6.23)$$

$$P_{c1} = 0.0164, \quad (6.24)$$

$$P_{c2} = 0.0038, \quad (6.25)$$

$$P_e = 0.9504. \quad (6.26)$$

因此，在两 BSS 不互听场景中系统的吞吐  $S$  为：

$$S = \frac{2 \cdot P_s \cdot P_s}{(P_{c1} + P_{c2}) \cdot T_c + P_s \cdot T_s + P_e \cdot T_c} \times Rate_{PHY} = 52.6231 \text{Mbps}. \quad (6.27)$$

同上述计算过程，根据附录 6 中的参数对系统的吞吐及关键指标进行解算，详细结果如表 6.1 所示。

### 6.3.2 仿真验证及结果分析

进一步编写仿真器对 AP1 和 AP2 的随机退避过程和信道的状态转换进行仿真模拟。具体仿真算法流程如下。根据算法 3 基于 MATLAB 编写仿真程序，对时长  $T_{max} = 1000s$  内进行节点数据传输的仿真模拟，详细代码见附录。

---

**算法 3** 问题三仿真算法

---

**数据:** 系统通用参数

**输入:** 载荷长度  $L_{\text{payload}}$ , 成功传输传输时长  $T_s$ , 碰撞传输时长  $T_c$ , 仿真时长  $T_{\text{max}}$

**输出:** 吞吐  $S$

**变量:** 节点传输数据起始时间  $T_{\text{start}}$ , 传输结束时间  $T_{\text{end}}$ , 成功传输次数  $n_s$

```
1: while  $T_{\text{end}} < T_{\text{max}}$  do
2:   if  $T_{\text{end1}} < T_{\text{start2}}$  then
3:     if  $\text{rand}(1) < p_{\text{el}}$  then
4:       更新  $CW$ 、 $b_k$ 、重传次数等
5:        $T_{\text{start1}} \leftarrow T_{\text{start1}} + T_c + b_{k1} \times t_{\text{SLOT}}$ 
6:        $T_{\text{end1}} \leftarrow T_{\text{end1}} + T_c + b_{k1} \times t_{\text{SLOT}}$ 
7:     else
8:       更新  $CW$ 、 $b_k$ 、重传次数等
9:        $T_{\text{start1}} \leftarrow T_{\text{start1}} + T_s + b_{k1} \times t_{\text{SLOT}}$ 
10:       $T_{\text{end1}} \leftarrow T_{\text{end1}} + T_s + b_{k1} \times t_{\text{SLOT}}$ 
11:       $n_s \leftarrow n_s + 1$ 
12:    else if  $T_{\text{end2}} < T_{\text{start1}}$  then
13:      与上同理
14:    else
15:      更新  $CW$ 、 $b_k$ 、重传次数等
16:       $T_{\text{start}} \leftarrow T_{\text{start}} + T_c + b_k \times t_{\text{SLOT}}$ 
17:       $T_{\text{end}} \leftarrow T_{\text{end}} + T_c + b_k \times t_{\text{SLOT}}$ 
18:   $S \leftarrow n_s \times L_{\text{payload}} \div T_{\text{max}}$ 
19: return  $S$ 
```

---

模型解算结果和仿真器模拟结果如表 6.1 所示, 其中,  $\text{State}_0$  表示题目和附录 4 给定参数状态,  $\text{State}_1$  至  $\text{State}_6$  分别表示附录 6 给定的 6 种参数状态。系统吞吐仿真结果与模型解算结果的平均误差仅为 4.7%, 可以说明系统吞吐解算模型的具有一定的精确性。

表 6.1 问题三中系统吞吐和各关键参数的模型解算结果和仿真器模拟结果

对应变量	State <sub>0</sub>	State <sub>1</sub>	State <sub>2</sub>	State <sub>3</sub>	State <sub>4</sub>	State <sub>5</sub>	State <sub>6</sub>
$CW_{\min}$	16	16	32	16	16	32	16
$CW_{\max}$	1024	1024	1024	1024	1024	1024	1024
$r$	32	6	5	32	6	5	32
$Rate_{PHY}(\text{Mbps})$	455.8	286.8	286.8	286.8	158.4	158.4	158.4
$m$	6	6	5	6	6	5	6
$m$ 与 $r$ 关系	$m < r$	$m = r$	$m = r$	$m < r$	$m = r$	$m = r$	$m < r$
$E[P](\mu\text{s})$	26.3273	41.8410	41.8410	41.8410	75.7576	75.7576	75.7576
$MAC(\mu\text{s})$	0.5265	0.8368	0.8368	0.8368	1.5152	1.5152	1.5152
$V$	5	7	7	7	11	11	11
窗口判断	$V < CW_{\min}$	$V < CW_{\min}$	$V < CW_{\min}$	$V < CW_{\min}$	$V < CW_{\min}$	$V < CW_{\min}$	$V < CW_{\min}$
$T_e(\mu\text{s})$	9	9	9	9	9	9	9
$T_s(\mu\text{s})$	131.4539	147.2778	147.2778	147.2778	181.8727	181.8727	181.8727
$T_c(\mu\text{s})$	148.4539	164.2778	164.2778	164.2778	198.8727	198.8727	198.8727
$p_1$	0.0451	0.0505	0.0344	0.0496	0.0427	0.0296	0.0416
$p_2$	0.3524	0.3380	0.2506	0.3315	0.3871	0.3086	0.3759
$P_e$	0.8908	0.9495	0.9656	0.9504	0.9573	0.9704	0.9584
$P_s$	0.0712	0.0301	0.0232	0.0298	0.0236	0.0184	0.0234
$P_{c1}$	0.0302	0.0171	0.0086	0.0164	0.0165	0.0091	0.0156
$P_{c2}$	0.0079	0.0033	0.0026	0.0033	0.0026	0.0020	0.0026
$S(\text{Mbps})$	<b>52.6231</b>	<b>44.2267</b>	<b>39.8915</b>	<b>44.2191</b>	<b>33.8528</b>	<b>30.9206</b>	<b>33.9910</b>
$S'(\text{Mbps})$	<b>54.7553</b>	<b>45.0926</b>	<b>37.2831</b>	<b>45.3556</b>	<b>34.0235</b>	<b>27.7993</b>	<b>36.2978</b>



## 7 问题四

### 7.1 问题分析

问题四考虑如图 1.1(c) 所示的三 BSS 场景，由于其中 AP1 与 AP2 之间，AP2 与 AP3 之间 RSSI 均为-70dBm，即高于 CCA 门限，因此 AP2 与 AP1 和 AP3 两者都互听；AP1 与 AP3 之间 RSSI 为-96dBm，即低于 CCA 门限，因此 AP1 与 AP3 不互听。因此问题四需要针对 WLAN 组网中 DCF 机制下的三 BSS 场景中存在的 AP1 与 AP3 不互听，AP2 与两者都互听的情况进行数据传输过程的分析和建模，实现对系统吞吐率的评估。

具体而言，由于 AP1 与 AP3 不互听，在下行数据传输时，AP1 和 AP3 之间通信区域无交集，向各自关联的 STA1 和 STA3 传输数据时有很大概率出现二者同时或先后开始发送数据的情况，即二者发包在时间上有重叠。若 AP1 或 AP3 位于 STA3 或 STA1 的通信区域内，后两者可能先后接收到信号包和干扰包。此时，节点 AP1 或 AP3 相对于 STA3 或 STA1 形成隐藏节点。根据问题四题设，AP1 和 AP3 发送数据在时间上有重叠时 SIR 较大，两个 AP 的发包均成功，因此不存在邻 BSS 干扰。由于 AP2 与 AP1 和 AP3 两者都互听，即 AP2 与 AP1 和 AP3 分别存在通信区域的重叠，在使用相同信道的前提下可能存在同频干扰问题。当并发时终端接收到数据的 SIR 较低时，信号解调失败，从而数据传输失败。假设此时发生并发碰撞的为 AP1 和 AP2，存在同频干扰，BSS1 中的 AP1(->STA1) 和 BSS2 中的 AP2(->STA2) 在随机退避过程中同时退避到 0 发送数据，AP2->STA1 对 AP1->STA1 产生干扰，同时 AP1->STA2 对 AP2->STA2 产生干扰，最终导致 AP1->STA1 和 AP2->STA2 的 SIR 都很低，信号解调失败，即发生并发碰撞丢包。本问题求解的思路流程图如图 7.1 所示。

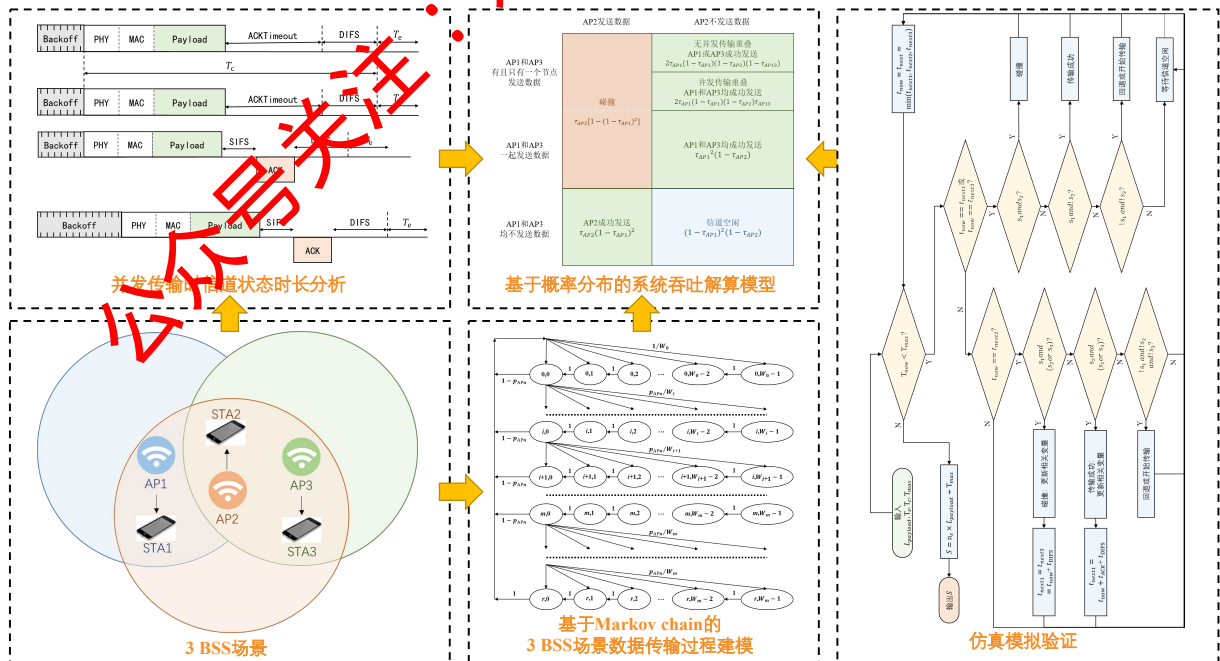
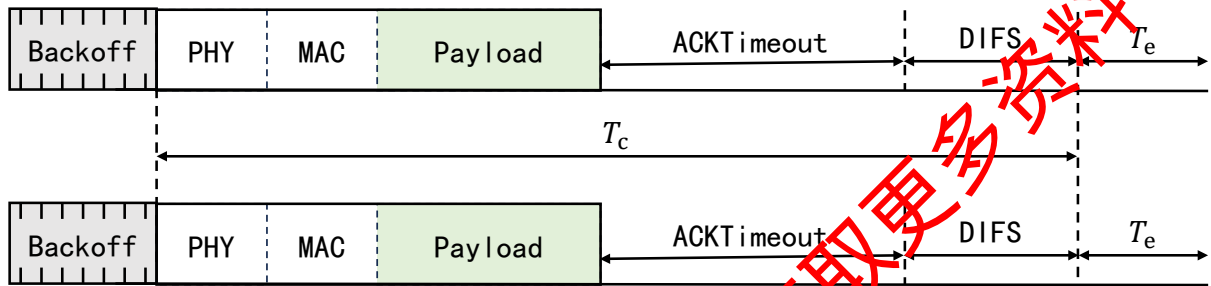


图 7.1 问题四的思路流程图

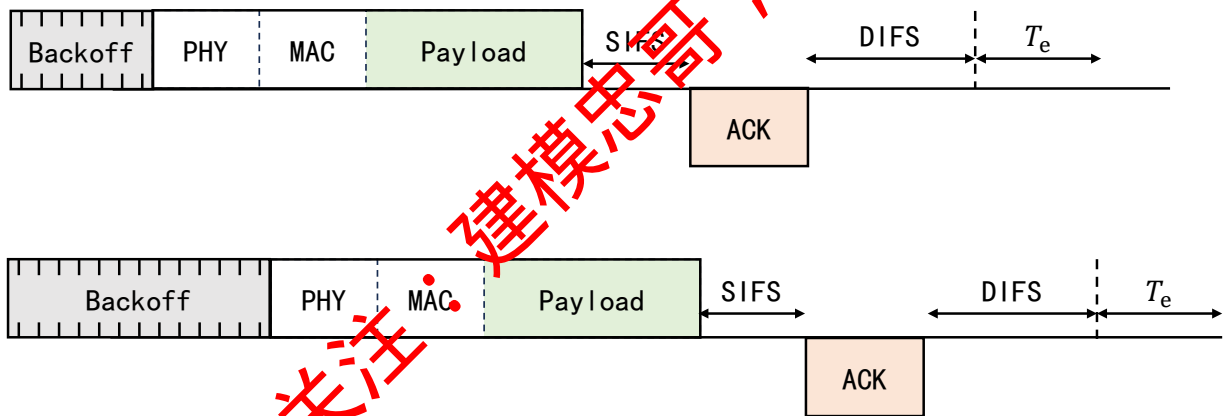
## 7.2 模型建立

### 7.2.1 三 BSS 场景下的信道状态时长分析

在本节中，考虑在三 BSS 场景中 AP2 与 AP1 和 AP3 两者都互听而 AP1 与 AP3 不互听情况，AP2 与 AP1 或 AP3 可能发生并发碰撞，且二者并发传输时发包均失败；AP1 与 AP3 可能出现并发传输重叠，此时 SIFS 较大，两者发送均成功，则信道状态时长参照图 7.2 所示。



(a) AP2 与 AP1 或 AP3 并发碰撞导致数据传输失败时信道状态时长示意图



(b) AP1 与 AP3 发生并发传输重叠时信道状态时长示意图

图 7.2 三 BSS 场景下的信道状态时长示意图

对于图 7.2 (a) 所示的情况，两个 AP 在随机退避过程中同时退避到 0，并同时发送数据，且数据帧的传输时长由 PHY 头、MAC 头和有效载荷 Payload 传输时长组成。由于冲突导致发送失败，接收节点将不会回复 ACK，两个 AP 在等待 ACKTimeout 后判断数据发送失败，侦听信道 DIFS 时长判断信道空闲后，选择随机数退避，进入空闲时隙  $T_e$ ，准备重新传送数据。对于图 7.2(b) 所示的情况，两个 AP 在随机退避过程中先后退避到 0，且在重叠的时间内依次发送数据，且数据帧的传输时长由 PHY 头、MAC 头和有效载荷 Payload 传输时长组成。由于有时间重叠时两者发包均成功，对应接收节点依次成功接收到数据之后等待 SIFS 后，回复 ACK 确认帧。AP 依次收到 ACK 后判断数据传输成功，开始侦听

信道 DIFS 时长判断信道空闲后，选择随机数退避，依次进入空闲时隙  $T_e$ ，准备下一次数据传送。

### 7.2.2 基于 Markov chain 的三 BSS 场景数据传输过程建模

对于问题四中研究的三 BSS 场景，AP1 与 AP3 不互听，AP2 与两者都互听。题目假设 AP1 和 AP3 出现并发传输重叠时，SIR 较大，两者发送均成功。进一步假设当 AP1 和 AP2，或 AP2 和 AP3 发生并发碰撞时，SIR 较小，AP 均发包失败。在此情况下，三个 AP 的数据下传过程只会因为并发碰撞而导致数据传输失败，然后重传数据帧时，CW 翻倍。如果 CW 达到了  $W_{\max}$ ，则保持此值，直到被重置为止。每次数据传输成功时 CW 重置，开始下一个数据帧的退避。若传输连续失败，重传次数达到  $r$  后，数据帧被丢弃，CW 重置传输下一个数据帧。因此，本问题中的 CW 仍然可以用问题一中的式 (4.3) 表示。

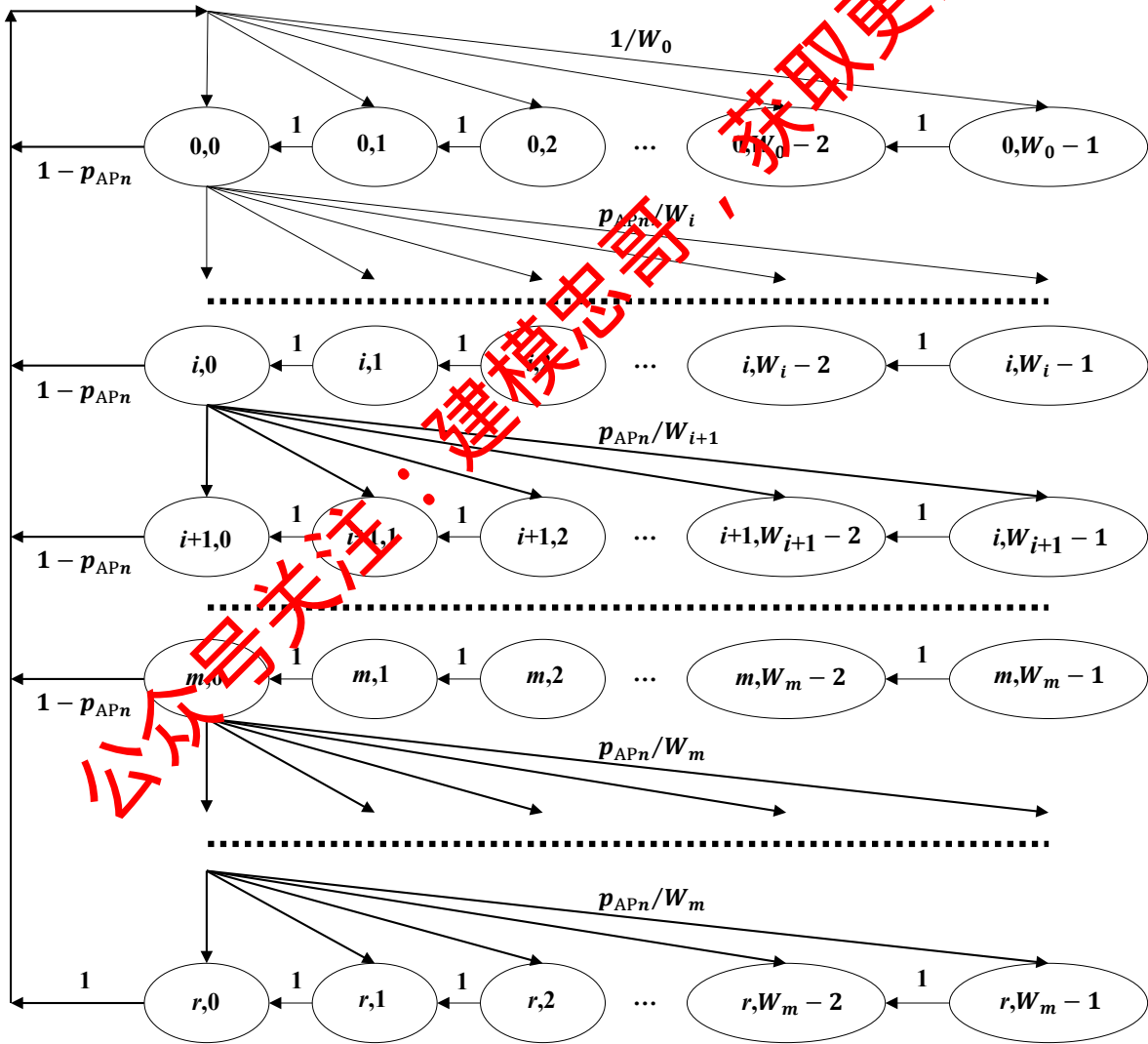


图 7.3 三 BSS 场景中各 AP 的 Markov chain 模型

本问题中的三个 AP 的二维  $\{s(t), b(t)\}$  随机过程 Markov chain 模型如图 7.3 所示。  $b_{i,k}$

为某一时隙的任一状态，其中  $i \in [0, m]$ ,  $k \in [0, W_i - 1]$ . 定义  $p_{APn}$  为某个时隙  $APn$  因并发碰撞导致数据传输失败的概率，其中  $n = 1, 2, 3$ . 该 Markov chain 中的一步状态转移情况可分为以下四种：

(1) 未达到重传上限时， $APn$  的退避计数器在每个空闲时隙的开始时刻减 1，且概率为 1，可表示为：

$$P\{i, k | i, k + 1\} = 1, k \in [0, W_i - 2], i \in [0, r]. \quad (7.1)$$

(2) 未达到重传上限时，当  $APn$  的一个数据成功传输后，新到达的数据在  $[0, W_0 - 1]$  中选一个随机数进行退避，每个随机数被选中的概率相等，该状态转移情况的概率可以表示为：

$$P\{0, k | i, 0\} = \frac{1 - p_{APn}}{W_0}, k \in [0, W_0 - 1], i \in [0, r] \quad (7.2)$$

(3) 未达到重传上限时，当  $APn$  的一个数据第  $i - 1$  次传输过程发生碰撞，节点进入第  $i$  阶退避过程，并在  $[0, W_i - 1]$  中选一个随机数进行退避，每个随机数被选中的概率相等，该状态转移情况的概率可以表示为：

$$P\{i, k | i - 1, 0\} = \frac{p_{APn}}{W_i}, k \in [0, W_i - 1], i \in [1, r]. \quad (7.3)$$

(4) 当  $APn$  到达最大的传输次数以后，无论数据成功传输还是发生碰撞， $CW$  都会重置，该状态转移情况的概率可以表示为：

$$P\{0, k | r, 0\} = \frac{1}{W_0}, k \in [0, W_0 - 1]. \quad (7.4)$$

对于任意两个  $AP$  因并发碰撞导致发包失败，状态  $b_{i-1,0}$  到状态  $b_{i,0}$  的步长包括： $b_{i-1,0} \rightarrow b_{i,0}$ ,  $b_{i-1,0} \rightarrow b_{i,1} \rightarrow b_{i,0}$ , ...,  $b_{i-1,0} \rightarrow b_{i,W_i-1} \rightarrow b_{i,1} \rightarrow b_{i,0}$ ，共  $W_i$  种，求和可得：

$$\begin{aligned} b_{i,0} &= b_{i-1,0} \times \frac{p_{APn}}{W_i} \times (1 + 1 + \dots + 1^{W_i-1}) \\ &= p_{APn} \times b_{i-1,0} \\ &= p_{APn}^i \times b_{0,0}. \end{aligned} \quad (7.5)$$

其中， $0 < i \leq r$ .

同理，对于每个  $AP$  的任一状态  $b_{i,k}$ ，若  $0 < i \leq r$ ，则是从一次并发碰撞导致发包失败的状态，通过  $CW$  加倍之后转移过来的。若  $i = 0$ ，则是从任一阶发送成功，或达到重传

次数限制后转移过来的。因此有：

$$b_{i,k} = \begin{cases} b_{i-1,0} \times p_{APn} \times \frac{W_i - k}{W_i}, & 0 < i < r, \\ (1 - p_{APn}) \times \frac{W_i - k}{W_i} \times \sum_{j=0}^{r-1} b_{j,0} + \frac{W_i - k}{W_i} \times b_{r,0}, & i = 0. \end{cases} \quad (7.6)$$

将公式 (7.5) 带入公式 (7.6)，可得：

$$b_{i,k} = \frac{W_i - k}{W_i} \times b_{i,0}, \quad 0 < i \leq r, 0 \leq k \leq W_i - 1. \quad (7.7)$$

根据 Markov chain 的性质，所有稳态的概率之和为 1，因此有：

$$\sum_{k=0}^{W_i-1} \sum_{i=0}^r b_{i,k} = \sum_{i=0}^r b_{i,0} \sum_{k=0}^{W_i-1} \frac{W_i - k}{W_i} = \sum_{i=0}^r b_{i,0} \frac{W_i + 1}{2} = 1. \quad (7.8)$$

根据式 (4.3) 和式 (7.8) 可以求得：

$$b_{0,0} = \begin{cases} \frac{2(1-p_{APn})(1-2p_{APn})}{(1-2p_{APn})(1-p_{APn}^{r+1}) + W_0(1-p_{APn})(1-(2p_{APn})^{r+1})}, & r \leq m, \\ \frac{2(1-p_{APn})(1-2p_{APn})}{W_0(1-(2p_{APn})^{m+1})(1-p_{APn}) + (1-2p_{APn})(1-p_{APn}^{r+1})}, & m < r. \end{cases} \quad (7.9)$$

节点 APn 随机退避到 0 时发送数据，因此节点 APn 在一个时隙发送数据帧的概率为：

$$\tau_{APn} = \sum_{i=0}^r b_{i,0} = b_{0,0} \times \frac{1 - p_{APn}^{r+1}}{1 - p_{APn}}. \quad (7.10)$$

当某一节点在一个时隙发送数据帧时，若还有至少有另外一个节点也在该时隙传输数据，则会发生碰撞导致丢包。定义互听节点总数为  $N_n$ ，则条件碰撞概率  $p_{APn}$  可表示为：

$$p_{APn} = 1 - (1 - \tau_{APn})^{N_n - 1}. \quad (7.11)$$

对于 AP1，其所在区域互听节点总数  $N_1$  为 2，则 AP1 的条件碰撞概率为  $p_{AP1} = \tau_{AP1}$ 。对于 AP2，其所在区域互听节点总数  $N_2$  为 3，则 AP2 的条件碰撞概率为  $p_{AP2} = 1 - (1 - \tau_{AP2})^2$ 。对于 AP3，其所在区域互听节点总数  $N_3$  为 2，则 AP3 的条件碰撞概率为  $p_{AP3} = \tau_{AP3} = \tau_{AP1}$  (AP1 和 AP3 的状态是对等的，所以  $p_{AP3} = p_{AP1} = \tau_{AP1}$ )。

另一方面，由于 AP1 和 AP3 不互听，存在二者并发传输重叠的情况，此时两个 AP 的

发包均成功。因此，AP1 和 AP3 在一个时隙发生并发传输重叠的概率可以表征为：

$$\tau_{AP13} = \sum_{i=0}^r \sum_{k=0}^V b_{i,k} = \begin{cases} \left[ (V+1) \frac{1-\tau_{AP1}^{r+1}}{1-\tau_{AP1}} - \frac{V(V+1)}{2W_0} \cdot \frac{1-(\frac{\tau_{AP1}}{2})^{r+1}}{1-\frac{\tau_{AP1}}{2}} \right] \cdot b_{0,0}, & V < W_0, \\ \left[ \frac{1-\tau_{AP1}^X}{2(1-\tau_{AP1})} + \frac{W_0}{2} \cdot \frac{1-(2\tau_{AP1})^X}{1-2\tau_{AP1}} + \frac{\tau_{AP1}^X - \tau_{AP1}^{r+1}}{1-\tau_{AP1}} \right] & W_{X-1} < V < W_X, \\ \times (V+1) - \frac{V(V+1)}{2W_0} \cdot \frac{(\frac{\tau_{AP1}}{2})^X - (\frac{\tau_{AP1}}{2})^{r+1}}{1-\frac{\tau_{AP1}}{2}} \right] \cdot b_{0,0}, & 1 \leq X \leq m, \\ 1, & V > W_m. \end{cases} \quad (7.12)$$

由公式 (7.9) 和式 (7.10) 构成的关于  $p_{APn}$  和  $\tau_{APn}$  的非线性方程组可求得  $\tau_{APn}$  的唯一解。然后将  $\tau_{AP1}$  的结算结果带入公式 (7.12) 即可结算出  $\tau_{AP13}$ 。

### 7.2.3 基于概率分布的系统吞吐解算模型

	AP2发送数据	AP2不发送数据
AP1和AP3 有且只有一个节点 发送数据	碰撞 $\tau_{AP2}[1-(\tau_{AP1})^2]$	无并发传输重叠 AP1或AP3成功发送 $\tau_{AP1}(1-\tau_{AP1})(1-\tau_{AP2})(1-\tau_{AP13})$
AP1和AP3 一起发送数据		并发传输重叠 AP1和AP3均成功发送 $2\tau_{AP1}(1-\tau_{AP1})(1-\tau_{AP2})\tau_{AP13}$
AP1和AP3 均不发送数据	AP2成功发送 $\tau_{AP2}(1-\tau_{AP1})^2$	AP1和AP3均成功发送 $\tau_{AP1}^2(1-\tau_{AP2})$
		信道空闲 $(1-\tau_{AP1})^2(1-\tau_{AP2})$

图 7.4 三 BSS 的传输数据状态概率分布图

图7.4为节点传输数据状态的概率分布图。可以看出，在某一时隙有节点开始发送数据并能成功发送数据的情况包含以下四种：



(1) AP2 发送数据，且 AP1 和 AP3 均不发送数据，该情况发生的概率为：

$$P_{s1} = \tau_{AP2}(1 - \tau_{AP1})^2. \quad (7.13)$$

(2) AP2 不发送数据，AP1 和 AP3 一起发送数据。该情况发生的概率为：

$$P_{s2} = \tau_{AP1}^2(1 - \tau_{AP2}). \quad (7.14)$$

(3) AP2 不发送数据，AP1 和 AP3 中有且仅有一个节点发送数据，且不发生并发传输重叠，该情况发生的概率为：

$$P_{s3} = 2\tau_{AP1}(1 - \tau_{AP1})(1 - \tau_{AP2})(1 - \tau_{AP13}). \quad (7.15)$$

(4) AP2 不发送数据，AP1 和 AP3 中有且仅有一个节点发送数据，且发生并发传输重叠，该情况发生的概率为：

$$P_{s4} = 2\tau_{AP1}\tau_{AP13}(1 - \tau_{AP1})(1 - \tau_{AP2}). \quad (7.16)$$

可知，在某一时刻成功发送数据的总概率为：

$$P_s = P_{s1} + P_{s2} + P_{s3} + P_{s4}. \quad (7.17)$$

值得注意的是，在情况 (2) 中，AP1 和 AP3 一起发送数据且均可成功发送，所以在一段信号成功传输时长  $T_s$  内可发送 2 倍的有效载荷。在情况 (4) 中，AP1 和 AP3 发生并发传输重叠，则平均占用的信号成功传输时长是  $1.5T_s$ ，并在此时长内发送 2 倍的有效载荷。

当 AP2 发送数据，且 AP1 和 AP3 中至少有一个节点发送数据时，会发生并发碰撞，概率为：

$$P_c = \tau_{AP2} [1 - (1 - \tau_{AP1})^2]. \quad (7.18)$$

当三个节点均不发送数据时，信道处于空闲状态，概率为：

$$P_e = (1 - \tau_{AP1})^2(1 - \tau_{AP2}). \quad (7.19)$$

因此，在三 BSS 场景中系统的吞吐  $S$  可以表示为：

$$S = \frac{(2P_{s1} + P_{s2} + P_{s3} + 2P_{s4}) \cdot E[P]}{P_c \cdot T_c + P_s \cdot T_s + 0.5P_{s4} \cdot T_s + P_e \cdot T_e} \times Rate_{PHY}. \quad (7.20)$$

其中,  $Rate_{PHY}$ 、 $E[P]$ 、 $T_c$ 、 $T_s$  和  $T_e$  的计算模型均与问题一相同。

### 7.3 问题求解

#### 7.3.1 基于 Markov chain 的系统吞吐评估

首先基于问题四题目中的给定信息和附录 4 的参数表进行模型求解。在该问题的三 BSS 场景中, DCF 机制参数、节点各状态的时长、发送数据的载荷长度和物理层速率均与问题一中的参数一致。

因此, 该系统状态下,  $m < r$ , 由公式 (7.9) 和式 (7.10) 构成的关于  $p_{APn}$  和  $\tau_{APn}$  的非线性方程组可求得:

$$\tau_{AP1} = 0.1046, \tau_{AP2} = 0.0934. \quad (7.21)$$

$V$  的定义为节点的数据成功传输时长与单个空闲时隙的比值, 因此可知:

$$V = \left\lceil \frac{T_s}{SlotTime} \right\rceil_{up} = 15 \quad (7.22)$$

其中,  $[\cdot]_{up}$  表示向上取整。可知  $V < W_0$ , 则将公式 (7.21) 和公式 (7.22) 代入公式 (7.12) 的第一种情况可以求得:

$$\tau_{AP13} = 0.9326. \quad (7.23)$$

然后, 在某一时刻有节点开始发送数据并能成功发送数据的四种情况的发生概率分别如下所示。

(1) AP2 发送数据, 且 AP1 和 AP3 均不发送数据, 该情况发生的概率为:

$$P_{s1} = \tau_{AP2}(1 - \tau_{AP1})^2 = 0.0749. \quad (7.24)$$

(2) AP2 不发送数据, AP1 和 AP3 一起发送数据。该情况发生的概率为:

$$P_{s2} = \tau_{AP1}^2(1 - \tau_{AP2}) = 0.0099. \quad (7.25)$$

(3) AP2 不发送数据, AP1 和 AP3 中有且仅有一个节点发送数据, 且不发生并发传输重叠, 该情况发生的概率为:

$$P_{s3} = 2\tau_{AP1}(1 - \tau_{AP1})(1 - \tau_{AP2})(1 - \tau_{AP13}) = 0.0115. \quad (7.26)$$

(4) AP2 不发送数据, AP1 和 AP3 中有且仅有一个节点发送数据, 且发生并发传输重

叠，该情况发生的概率为：

$$P_{s4} = 2\tau_{AP1}\tau_{AP13}(1 - \tau_{AP1})(1 - \tau_{AP2}) = 0.1584. \quad (7.27)$$

可知，在某一时刻成功发送数据的总概率为：

$$P_s = P_{s1} + P_{s2} + P_{s3} + P_{s4} = 0.2546. \quad (7.28)$$

AP2 发送数据，且 AP1 和 AP3 中至少有一个节点发送数据导致并发碰撞的概率为：

$$P_c = \tau_{AP2} [1 - (1 - \tau_{AP1})^2] = 0.0185. \quad (7.29)$$

三个节点均不发送数据导致信道处于空闲状态的概率为：

$$P_e = (1 - \tau_{AP1})^2(1 - \tau_{AP2}) = 0.7768. \quad (7.30)$$

因此，在三 BSS 场景中系统的吞吐  $S$  为：

$$S = \frac{(2P_{s1} + P_{s2} + P_{s3} + 2P_{s4}) \cdot E[T_s] + P_c \cdot T_c + P_s \cdot T_s + 0.5P_{s4} \cdot T_s + P_e \cdot T_e}{P_c \cdot T_c + P_s \cdot T_s + 0.5P_{s4} \cdot T_s + P_e \cdot T_e} \times Rate_{PHY} = 110.415 \text{Mbps}. \quad (7.31)$$

同上述计算过程，根据附录 6 中的参数对系统的吞吐及关键指标进行解算，详细结果如表 7.1 所示。

### 7.3.2 仿真验证及结果分析

进一步编写仿真程序对 AP1 和 AP2 的随机退避过程和信道的状态转换进行仿真模拟。具体仿真算法流程如下：根据算法 4 基于 MATLAB 编写仿真程序，对时长  $T_{\max} = 1000\text{s}$  内进行节点数据传输的仿真模拟，详细代码见附录。模型解算结果和仿真器模拟结果如表 7.1 所示，其中  $\text{State}_0$  表示题目和附录 4 给定参数状态， $\text{State}_1$  至  $\text{State}_6$  分别表示附录 6 给定的 6 种参数状态。系统吞吐仿真结果与模型解算结果的平均误差仅为 3.1%，可以说明系统吞吐解算模型的具有一定的精确性。

---

**算法 4** 问题四仿真算法

---

**数据:** 系统参数

**输入:** 载荷长度  $L_{\text{payload}}$ , 成功传输传输时长  $T_s$ , 碰撞传输时长  $T_c$ , 仿真时长  $T_{\text{max}}$

**输出:** 吞吐  $S$

**变量:** 当前仿真时刻  $T_{\text{now}}$ , 下一仿真时刻  $T_{\text{next}}$ , 节点传输状态  $s$ , 成功传输次数  $n_s$

```
1: while  $T_{\text{now}} < T_{\text{max}}$  do
2:   if  $t_{\text{now}} == t_{\text{next1}}$  then
3:     if  $s_1$  and  $s_2$  then
4:       碰撞
5:     else if  $s_1$  and  $!s_2$  then
6:       传输成功
7:     else if  $!s_1$  and  $!s_2$  then
8:       退避或开始传输
9:     else
10:      等待信道空闲
11:    if  $t_{\text{now}} == t_{\text{next3}}$  then
12:      与上同理
13:    if  $t_{\text{now}} == t_{\text{next2}}$  then
14:      if  $s_2$  and ( $s_1$  or  $s_3$ ) then
15:        碰撞: 更新相关变量
16:         $t_{\text{next1}} \leftarrow t_{\text{next3}} \leftarrow t_{\text{now}} + t_{\text{DIFS}}$ 
17:      else if  $s_2$  and ( $s_1$  or  $s_3$ ) then
18:        传输成功: 更新相关变量
19:         $t_{\text{next}} \leftarrow t_{\text{now}} + t_{\text{ACK}} + t_{\text{DIFS}}$ 
20:      else if  $!s_1$  and  $!s_1$  and  $!s_3$  then
21:        退避或开始传输
22:      else
23:        等待信道空闲
24:       $t_{\text{now}} \leftarrow t_{\text{next}} \leftarrow \min(t_{\text{next1}}, t_{\text{next2}}, t_{\text{next3}})$ 
25: return  $S \leftarrow n_s \times L_{\text{payload}} \div T_{\text{max}}$ 
```

---

表 7.1 问题四中系统吞吐和各关键参数的模型解算结果和仿真器模拟结果

对应变量	State <sub>0</sub>	State <sub>1</sub>	State <sub>2</sub>	State <sub>3</sub>	State <sub>4</sub>	State <sub>5</sub>	State <sub>6</sub>
$CW_{\min}$	16	16	32	16	16	32	16
$CW_{\max}$	1024	1024	1024	1024	1024	1024	1024
$r$	32	6	5	32	6	5	32
$Rate_{PHY}(\text{Mbps})$	455.8	286.8	286.8	286.8	158.4	158.4	158.4
$m$	6	6	5	6	6	5	6
$m$ 与 $r$ 关系	$m < r$	$m = r$	$m = r$	$m < r$	$m = r$	$m = r$	$m < r$
$E[P](\mu s)$	26.3273	41.841	41.841	41.841	75.7576	75.7576	75.7576
$MAC(\mu s)$	0.5265	0.8368	0.8368	0.8368	1.5152	1.5152	1.5152
$V$	15	17	17	17	21	21	21
$V$ 的取值判断	$V < W_0$	$W_0 < V < W_1$	$V > CW_{\min}$	$W_0 < V < W_1$	$W_0 < V < W_1$	$V < W_0$	$W_0 < V < W_1$
$T_e(\mu s)$	9	9	9	9	9	9	9
$T_s(\mu s)$	131.4538	147.2778	147.2778	147.2778	181.8727	181.8727	181.8727
$T_c(\mu s)$	148.4538	164.2778	164.2778	164.2778	198.8727	198.8727	198.8727
$\tau_{AP1}$	0.1046	0.1046	0.057	0.1046	0.1046	0.057	0.1046
$\tau_{AP2}$	0.0934	0.0934	0.0537	0.0934	0.0537	0.0537	0.0934
$\tau_{AP13}$	0.9326	0.9438	0.7621	0.9438	0.9624	0.8553	0.9624
$P_s$	0.2546	0.2546	0.1526	0.2546	0.2307	0.1526	0.2546
$P_c$	0.0185	0.0185	0.0059	0.0185	0.0106	0.0059	0.0185
$P_e$	0.7268	0.7268	0.8415	0.7268	0.7587	0.8415	0.7268
$S(\text{Mbps})$	<b>110.415</b>	<b>98.8122</b>	<b>84.2736</b>	<b>98.8122</b>	<b>84.2225</b>	<b>71.9727</b>	<b>80.3004</b>
$S'(\text{Mbps})$	<b>111.018</b>	<b>102.1904</b>	<b>83.0276</b>	<b>102.1508</b>	<b>88.0104</b>	<b>72.1273</b>	<b>87.9889</b>

## 8 工作总结与展望

无线局域网的广泛应用带来大量同频干扰的问题。对同频多 BSS 场景建模,探究其工作状态和系统性能对指导通信技术发展具有重要意义。本文基于 Bianchi 模型,针对多种场景建立 Markov chain 模型,求解信道状态及概率,得到信道最大利用率和系统性能等指标参数。

### 8.1 结论

本文针对两 BSS 和三 BSS 部分典型场景的系统吞吐计算进行了研究,做出的主要贡献包括:

1. 建立了两 BSS 和三 BSS 互听和不互听典型场景下的 Markov chain 模型,给出了求解节点状态概率的求解公式。
2. 划分了两 BSS 和三 BSS 部分经典场景下的系统信道占用情况及其概率,建立了基于概率分布的系统吞吐解算模型。
3. 构建了两 BSS 和三 BSS 部分经典场景下多 AP 下行传输的软件仿真模拟模型。

### 8.2 展望

由于现代无线通信技术发展繁荣、进步迅速,本文对无线通信系统的描述不够全面,可能在许多方面存在疏漏,仍有许多细节需要进一步开展探究。在此提出本文的不足之处以及未来展望。

1. 本文描述的是同频理想信道的传输过程,忽略了环境等因素对信号传输各环节的影响。因此,还需探究不同频率信道和同频非理想信道的传输过程。
2. 本文仅针对下行传输过程进行建模,忽略了回复报文以及上行传输过程及其影响。因此,后续还需要考虑上行传输对系统的影响。
3. 本文对同频 BSS 之间统一假设 SIR 足够大和 SIR 足够小两种假设,未探究变化 SIR 对传输过程的影响。因此,后续还需要考虑不同 SIR 对不同节点传输结果的影响。

## 参考文献

- [1] Farhad A, Pyun J Y, Resource Management for Massive Internet of Things in IEEE 802.11 ah WLAN: Potentials, Current Solutions, and Open Challenges, *Sensors*, 22(23):9509, 2022.
- [2] Cao B, Li M, Zhang L, et al., How does CSMA/CA affect the performance and security in wireless blockchain networks, *IEEE Transactions on Industrial Informatics*, 16(6):4270-4280, 2019.
- [3] Refaet A, Amed M A, Abed W, et al., WLAN performance evaluation in different wireless access techniques (DCF, PCF, HCF), *Periodicals of Engineering and Natural Sciences*, 8(3): 1297-1308, 2020.
- [4] Joshi N, Yadav P, Sharma S, IEEE 802.11 DCF MAC Protocol for CF-enabled WLAN in AdHoc and Infrastructure modes, 2023 International Conference on Device Intelligence, Computing and Communication Technologies,(DICCT), IEEE, 586-590, 2023.
- [5] Anuar A S M, Muhamad W N W, Ali D M, et al., A review on link adaptation techniques for energy efficiency and QoS in IEEE802. 11 WLAN, *Indonesian Journal of Electrical Engineering and Computer Science*, 17(1):331-339, 2020.
- [6] Khalaf S, Hasan H M, Simulation and performance evaluation of IEEE802. 11 WLAN under different operating conditions, *Bulletin of Electrical Engineering and Informatics*, 11(4): 2196-2203, 2022.
- [7] Bianchi, G., IEEE 802.11-saturation throughput analysis, *IEEE Communications Letters*, 2 (12):318-320, 2002.
- [8] Chatzimisios P, Vitsas Y, Boucouvalas A, Throughput and delay analysis of IEEE 802.11 protocol, *Proceedings 3rd IEEE International Workshop on System-on-Chip for Real-Time Applications*, 168-174, 2002.
- [9] Hung F Y, Morris I, Performance analysis of the IEEE 802.11 DCF in the presence of the hidden stations, *Computer Networks*, 54(15):2674-2687, 2010.
- [10] Chen D R, Zhang Y J, Is Dynamic Backoff Effective for Multi-Rate WLANs?, *IEEE Communications Letters*, 11(8):647-649, 2007.
- [11] Aad I, Ni Q, Barakat C, et al., Enhancing IEEE 802.11 MAC in congested environments, 2004 4th Workshop on Applications and Services in Wireless Networks, 82-91, 2004.
- [12] Ni Q, Aad I, Barakat C, et al., Modeling and analysis of slow CW decrease IEEE 802.11 WLAN, 14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications: volume 2, 1717-1721, 2003.
- [13] Xiao Y, Backoff-based priority schemes for IEEE 802.11, *IEEE International Conference on Communications: volume 3*, 1568-1572, 2003.



## 附录 A MATLAB 代码

### A.1 问题一代码

问题一模型求解程序

```
clear all
close all

% 通用参数
t_ACK = 32;
t_SIFS = 16;
t_DIFS = 43;
t_SLOT = 9;
t_ACKTimeout = 65;
t_phy = 13.6;
CWmax = 1024;
L_mac = 30 * 8;
L_payload = 1500 * 8;

% 给定参数
W0 = 16;
v_physicalDataRate = 286.8 * 1e6;
r = 32;
m = 6;

% 计算参数
t_MAC = L_mac / v_physicalDataRate * 1e6;
t_payload = L_payload / v_physicalDataRate * 1e6;
t_s = t_phy + t_MAC + t_payload + t_SIFS + t_ACK + t_DIFS;
t_c = t_phy + t_MAC + t_payload + t_ACKTimeout + t_DIFS;

% 节点发送数据概率曲线求解
p = 0:0.001:1;
f1 = (W0 * p - (2^(m+1)) * (p^(m+2))) * (1-p);
f2 = (CWmax * (p^(m+1)) - CWmax * (p^(r+2))) * (1-2*p);
f3 = (CWmax * p + p - 2) * (1-2*p) * (1-p^(r+1));
f4 = (1-p^(r+1)) * 2 * (1-2*p);
f = f1 + f2 + f3 - f4;
figure
plot(p, f)
```

```

tt = 0.10462;           %通过函数曲线判断数值解
% tt = 0.51;
P2 = tt*(1-tt);
Ptr = 1-(1-tt)*(1-tt);
Ps = P2*2/Ptr;

%系统吞吐计算
S1 = Ptr*Ps*t_payload;
S2 = (1-Ptr)*t_SLOT+Ptr*Ps*t_s+(1-Ps)*Ptr*t_c;
S = S1/S2*v_physicalDataRate;

```

#### 问题一仿真器程序

```

clear;clc
%% 参数定义
% 数据速率-bps
% 时间      -us
% 通用参数
t_ACK = 32;
t_SIFS = 16;
t_DIFS = 43;
t_SLOT = 9;
t_ACKTimeout = 65;
p_el = 0.1;

% 给定参数
CWmin = 16;
CWmax = 1024;
c_retrymax = 32;
v_physicalDataRate = 455.8 *1e6;
t_phy = 13.6;
L_mac = 30 *8;
L_payload = 1500 *8;

% 计算参数
m = log2(CWmax/CWmin);
r = c_retrymax;
W0 = CWmin;

```

```

t_MAC = L_mac/v_physicalDataRate*1e6;
t_payload = L_payload/v_physicalDataRate*1e6;
t_s = t_phy+t_MAC+t_payload+t_SIFS+t_ACK+t_DIFS;
t_c = t_phy+t_MAC+t_payload+t_ACKTimeout+t_DIFS;
t_t = t_phy+t_MAC+t_payload;

```

```
%% 变量
```

```

CW_1 = CWmin;
CW_2 = CWmin;
bi_1 = randi([0,CW_1-1]);
bi_2 = randi([0,CW_2-1]);
c_retry1 = 0;
c_retry2 = 0;

```

```
%% 仿真
```

```

c_success1 = 0;
c_success2 = 0;
c_collision = 0;
t_now = t_DIFS;
T_max = 1.0e9;
t_next = 0.0;
t_next1 = 0.0;
t_next2 = 0.0;
state = 0;
state1 = 0;
state2 = 0;

```

```

while(t_now < T_max)
    if(bi_1==0 && bi_2~=0) % 回退
        bi_1 = bi_1-1;
        bi_2 = bi_2-1;
        t_next = t_now+t_SLOT;
    elseif(bi_1==0 && bi_2==0) % 碰撞
        t_next = t_now+t_c;
        CW_1 = CW_1*2;
        CW_2 = CW_2*2;
        if(CW_1>CWmax)
            CW_1 = CWmax;
        end
    end
end

```

```

        if(CW_2>CWmax)
            CW_2 = CWmax;
        end
        c_retry1 = c_retry1+1;
        c_retry2 = c_retry2+1;
        if(c_retry1>c_retrymax)
            c_retry1 = 0;
            CW_1 = CWmin;
        end
        if(c_retry2>c_retrymax)
            c_retry2 = 0;
            CW_2 = CWmin;
        end
        bi_1 = randi([0,CW_1-1]);
        bi_2 = randi([0,CW_2-1]);
        c_collision = c_collision+1;
    elseif(bi_1==0) % AP1开始传输
        t_next = t_now+t_s;
        c_retry1 = 0;
        CW_1 = CWmin;
        bi_1 = randi([0,CW_1-1]);
        c_success1 = c_success1+1;
    elseif(bi_2==0) % AP2开始传输
        t_next = t_now+t_s;
        c_retry2 = 0;
        CW_2 = CWmin;
        bi_2 = randi([0,CW_2-1]);
        c_success2 = c_success2+1;
    end
    % channel
    t_now = t_next;
end

%% 计算结果
S = (c_success1+c_success2)*1500*8/(T_max/1e6);
print1 = (c_success1+c_success2)*(t_s)/T_max;
print2 = S/v_physicalDataRate;
fprintf(' 吞吐量S    传输占比    信道利用率\n%9d    %0.4f    %0.4f\n', S
,print1,print2)

```

## A.2 问题二代码

### 问题二模型求解程序

```
clear all
close all

% 通用参数
t_ACK = 32;
t_SIFS = 16;
t_DIFS = 43;
t_SLOT = 9;
t_ACKTimeout = 65;
t_phy = 13.6;
CWmax = 1024;
L_mac = 30 * 8;
L_payload = 1500 * 8;

% 给定参数
W0 = 16;
v_physicalDataRate = 286.8 * 1e6;
r = 32;
m = 6;

% 计算参数
t_MAC = L_mac / v_physicalDataRate * 1e6;
t_payload = L_payload / v_physicalDataRate * 1e6;
t_s = t_phy + t_MAC + t_payload + t_SIFS + t_ACK + t_DIFS;
t_c = t_phy + t_MAC + t_payload + t_ACKTimeout + t_DIFS;

% 系统吞吐计算
tt = 1 / W0 + 1;

Ptr = (1 - tt) * (1 - tt);
S1 = 2 * tt * t_payload;
S2 = tt * (2 - tt) * t_s + Ptr * t_SLOT;
S = S1 / S2 * v_physicalDataRate;
```

### 问题二仿真器程序

```
clear;clc
```

```

%% 参数定义
% 数据速率-bps
% 时间      -us
% 通用参数
t_ACK = 32;
t_SIFS = 16;      % 短帧帧间距
t_DIFS = 43;
t_SLOT = 9;
t_ACKTimeout = 65;
p_el = 0.1;

% 给定参数
CWmin = 16;
CWmax = 1024;
c_retrymax = 32;
v_physicalDataRate = 275.3 *1e6;
t_phy = 13.6;
L_mac = 30 *8;      % MAC长度
L_payload = 1500 *8;

% 计算参数
m = log2(CWmax/CWmin);
r = c_retrymax;
W0 = CWmin;
t_MAC = L_mac/v_physicalDataRate*1e6;
t_payload = L_payload/v_physicalDataRate*1e6;
t_s = t_phy+t_MAC+t_payload+t_SIFS+t_ACK+t_DIFS;
t_c = t_phy+t_MAC+t_payload+t_ACKTimeout+t_DIFS;
t_t = t_phy+t_MAC+t_payload;

%% 仿真
% 变量
CW_1 = CWmin;
CW_2 = CWmin;
bi_1 = randi([0,CW_1-1]);
bi_2 = randi([0,CW_2-1]);
c_retry1 = 0;
c_retry2 = 0;

```

```

% 仿真
c_success1 = 0;
c_success2 = 0;
c_collision = 0;
t_now = t_DIFS;
T_max = 1.0e9;
t_next = 0.0;
while(t_now<T_max)
    if(bi_1~=0 && bi_2~=0) % 回退
        bi_1 = bi_1-1;
        bi_2 = bi_2-1;
        t_next = t_now+t_SLOT;
    else
        if(bi_1==0 && bi_2==0) % 碰撞
            c_collision = c_collision+1;
        end
        if(bi_1==0) % AP1传输
            t_next = t_now+t_s;
            CW_1 = CWmin;
            bi_1 = randi([0,CW_1-1]);
            c_success1 = c_success1+1;
        end
        if(bi_2==0) % AP2传输
            t_next = t_now+t_s;
            CW_2 = CWmin;
            bi_2 = randi([0,CW_2-1]);
            c_success2 = c_success2+1;
        end
    end
    % channel
    t_now = t_next;
end

%% 计算结果
S = (c_success1+c_success2)*L_payload/(T_max/1e6);
print1 = (c_success1+c_success2)*(t_s)/T_max;
print2 = S/v_physicalDataRate;
fprintf(' 吞吐量S    传输占比    信道利用率\n%9d    %0.4f    %0.4f\n',S,
        print1,print2)

```



### A.3 问题三代码

#### 问题三模型求解程序

```
clear;clc
%%
% 通用参数
t_ACK = 32;
t_SIFS = 16;
t_DIFS = 43;
t_SLOT = 9;
t_ACKTimeout = 65;
p_el = 0.1;
% 给定参数
CWmin = 16;
CWmax = 1024;
c_retrymax = 32;
v_physicalDataRate = 455.8 *1e6;
t_phy = 13.6;
L_mac = 30 *8;
L_payload = 1500 *8;
% 计算参数
m = log2(CWmax/CWmin);
r = c_retrymax;
W0 = CWmin;
t_MAC = L_mac/v_physicalDataRate*1e6;
t_payload = L_payload/v_physicalDataRate*1e6;
t_s = t_phy+t_MAC+t_payload+t_SIFS+t_ACK+t_DIFS;
t_c = t_phy+t_MAC+t_payload+t_ACKTimeout+t_DIFS;
t_t = t_phy+t_MAC+t_payload;
v = ceil(t_t./t_SLOT);
x_min = 0.01;
x_max = 0.49;
[y_min,~] = t3_tau2(x_min);
[y_max,~] = t3_tau2(x_max);
while(x_max-x_min>1e-5)
    x = (x_min+x_max)./2;
    [y,~] = t3_tau2(x);
    if(y*y_min<0)
        y_max = y;
        x_max = x;
```

```

else
    y_min = y;
    x_min = x;
end
end

tau2 = x;
[y,tau1] = t3_tau2(tau2);
p_s = tau1.*(1-tau2).*(1-p_el);
p_c1 = tau1.*tau2;
p_c2 = tau1.*(1-tau2).*p_el;
p_e = 1-tau1;

print1 = 2*(p_s.*t_payload)./(p_c1.*t_c+p_c2.*t_c+p_s.*t_s+p_e.*
    t_SLOT);
S = print1.*v_physicalDataRate;
fprintf(' 吞吐量S   信道利用率\n%9d  %0.4f\n', S,print1)

function [y,tau1] = t3_tau2(tau2)
    % 通用参数
    t_ACK = 32;
    t_SIFS = 16;
    t_DIFS = 43;
    t_SLOT = 9;
    t_ACKTimeout = 64;
    p_el = 0.1;

    % 给定参数
    CWmin = 16;
    CWmax = 1024;
    c_retrymax = 32;
    v_physicalDataRate = 286.8 *1e6;
    t_phy = 13.6;
    L_mac = 30 *8;
    L_payload = 1500 *8;

    % 计算参数
    W0 = CWmin;
    r = c_retrymax;

```

```

m = log2(CWmax./CWmin);
t_MAC = L_mac/v_physicalDataRate*1e6;
t_payload = L_payload/v_physicalDataRate*1e6;
t_s = t_phy+t_MAC+t_payload+t_SIFS+t_ACK+t_DIFS;
t_c = t_phy+t_MAC+t_payload+t_ACKTimeout+t_DIFS;
t_t = t_phy+t_MAC+t_payload;
v = ceil(t_t./t_SLOT);

%%
p_l = 1-(1-p_el).*(1-tau2);
if(r<=m)
    b00b = 2*(1-p_l).*(1-2*p_l);
    b00a = (1-2*p_l).*(1-p_l.^(r+1))+W0*(1-p_l).*(1-(2*p_l).^(r+1));
    b00 = b00b./b00a;
else
    b00b = 2*(1-p_l).*(1-2*p_l);
    b00a = W0*(1-(2*p_l).^(m+1)).*(1-p_l)+(1-2*p_l).*(1-p_l.^(r+1))+W0*(2^m).*(p_l.^(m+1)).*(1-p_l.^(r-m)).*(1-2*p_l);
    b00 = b00b./b00a;
end
tem1b = (v+1).*(1-p_l.^(r+1));
tem1a = 1-p_l;
tem2b = v.*(v+1).*(1-(p_l./2).^(r+1));
tem2a = 2*W0*(1-p_l./2);
y = tau1.*tem1a.*tem2a.*b00a-tem1b.*tem2a.*b00b+tem1a.*tem2b.*b00b;
tau1 = b00.*(1-p_l.^(r+1))./(1-p_l);
end

```

### 问题三仿真器程序

```

clear;clc
%% 参数定义
% 数据速率-bps
% 时间 -us
% 通用参数
t_ACK = 32;
t_SIFS = 16; % 短帧帧间距

```

```

t_DIFS = 43;
t_SLOT = 9;
t_ACKTimeout = 65;
p_el = 0.1;

% 给定参数
CWmin = 32;
CWmax = 1024;
c_retrymax = 5;
v_physicalDataRate = 158.4 *1e6;
t_phy = 13.6;
L_mac = 30 *8;
L_payload = 1500 *8;

% 计算参数
m = log2(CWmax/CWmin);
r = c_retrymax;
W0 = CWmin;
t_MAC = L_mac/v_physicalDataRate*1e6;
t_payload = L_payload/v_physicalDataRate*1e6;
t_s = t_phy+t_MAC+t_payload+t_SIFS+t_ACK+t_DIFS;
t_c = t_phy+t_MAC+t_payload+t_ACKTimeout+t_DIFS;
t_t = t_phy+t_MAC+t_payload;

%% 变量
CW_1 = CWmin;
CW_2 = CWmin;
bi_1 = randi([0,CW_1-1]);
bi_2 = randi([0,CW_2-1]);
c_retry1 = 0;
c_retry2 = 0;

%%
c_success1 = 0;
c_success2 = 0;
c_failure1 = 0;
c_failure2 = 0;
t_now = 0.0;
T_max = 1.0e9;

```

```

t_next = 0.0;
t_start1 = t_DIFS+bi_1*t_SLOT;
t_start2 = t_DIFS+bi_2*t_SLOT;
t_end1 = t_start1+t_phy+t_MAC+t_payload;
t_end2 = t_start2+t_phy+t_MAC+t_payload;
t_start_next1 = t_start1;
t_start_next2 = t_start2;
t_end_next1 = t_end1;
t_end_next2 = t_end2;

while(t_end1<T_max || t_end2<T_max)
    if(t_end1<t_start2)
        if(rand<p_el) % AP1传输失败
            if(c_retry1<c_retrymax)
                c_retry1 = c_retry1+1;
                CW_1 = 2*CW_1;
                if(CW_1>CWmax)
                    CW_1 = CWmax;
                end
            else
                c_retry1 = 0;
                CW_1 = CWmin;
            end
            bi_1 = randi([0,CW_1-1]);
            c_failure1 = c_failure1+1;
            t_start_next1 = t_end1+t_ACKTimeout+t_DIFS+bi_1*t_SLOT;
            t_end_next1 = t_start_next1+t_phy+t_MAC+t_payload;
        else % AP1传输成功
            c_retry1 = 0;
            CW_1 = CWmin;
            bi_1 = randi([0,CW_1-1]);
            c_success1 = c_success1+1;
            t_start_next1 = t_end1+t_SIFS+t_ACK+t_DIFS+bi_1*t_SLOT;
            t_end_next1 = t_start_next1+t_phy+t_MAC+t_payload;
        end
    elseif(t_end2<t_start1)
        if(rand<p_el) % AP2传输失败
            if(c_retry2<c_retrymax)
                c_retry2 = c_retry2+1;

```

```

        CW_2 = 2*CW_2;
        if(CW_2>CWmax)
            CW_2 = CWmax;
        end
    else
        c_retry2 = 0;
        CW_2 = CWmin;
    end
    bi_2 = randi([0,CW_2-1]);
    c_failure2 = c_failure2+1;
    t_start_next2 = t_end2+t_ACKTimeout+t_DIFS+bi_2*t_SLOT;
    t_end_next2 = t_start_next2+t_phy+t_MAC+t_payload;
else % AP2 传输成功
    c_retry2 = 0;
    CW_2 = CWmin;
    bi_2 = randi([0,CW_2-1]);
    c_success2 = c_success2+1;
    t_start_next2 = t_end2+t_SIFS+t_ACK+t_DIFS+bi_2*t_SLOT;
    t_end_next2 = t_start_next2+t_phy+t_MAC+t_payload;
end
else
    if(c_retry1<c_retrymax)
        c_retry1 = c_retry1+1;
        CW_1 = 2*CW_1;
        if(CW_1>CWmax)
            CW_1 = CWmax;
        end
    else
        c_retry1 = 0;
        CW_1 = CWmin;
    end
    bi_1 = randi([0,CW_1-1]);
    c_failure1 = c_failure1+1;
    t_start_next1 = t_end1+t_ACKTimeout+t_DIFS+bi_1*t_SLOT;
    t_end_next1 = t_start_next1+t_phy+t_MAC+t_payload;
    if(c_retry2<c_retrymax)
        c_retry2 = c_retry2+1;
        CW_2 = 2*CW_2;
        if(CW_2>CWmax)

```

```

        CW_2 = CWmax;
    end
else
    c_retry2 = 0;
    CW_2 = CWmin;
end
bi_2 = randi([0,CW_2-1]);
c_failure2 = c_failure2+1;
t_start_next2 = t_end2+t_ACKTimeout+t_DIFS+bi_2*t_SLOT;
t_end_next2 = t_start_next2+t_phy+t_MAC+t_payload;
end
t_start1 = t_start_next1;
t_start2 = t_start_next2;
t_end1 = t_end_next1;
t_end2 = t_end_next2;
end

%% 计算结果
S = (c_success1+c_success2)*L_payload/(T_max/1e6);
print1 = (c_success1+c_success2)*L_payload/T_max;
print2 = S/v_physicalDataRate;
fprintf(' 吞吐量S    传输时延    信道利用率\n%9d    %0.4f    %0.4f\n', S
    ,print1,print2)

```

#### A.4 问题四代码

##### 问题四模型求解程序

```

clear all
close all

% 通用参数
t_ACK = 32;
t_SIFS = 16;    % 短帧帧间距
t_DIFS = 43;
t_SLOT = 9;
t_ACKTimeout = 65;
t_phy = 13.6;
CWmax = 1024;
L_mac = 30 * 8;    % MAC长度

```



```

L_payload = 1500 *8;

% 给定参数
W0 = 16;
v_physicalDataRate = 286.8 *1e6;
r = 32;
m = 6;

% 计算参数
t_MAC = L_mac/v_physicalDataRate*1e6;
t_payload = L_payload/v_physicalDataRate*1e6;
t_s = t_phy+t_MAC+t_payload+t_SIFS+t_ACK+t_DIFS;
t_c = t_phy+t_MAC+t_payload+t_ACKTimeout+t_DIFS;
t_s1 = (3/2)*(t_phy+t_MAC+t_payload+t_SIFS+t_ACK+t_DIFS);
t_t = t_phy+t_MAC+t_payload;
v = ceil((t_s)./t_SLOT);

tau1_0 = 0:0.00001:1;
tau2_0 = 0:0.00001:1;
[y1,tau1]=t4_tau1(tau1_0);
[y2,tau2]=t4_tau2(tau2_0);
tau1_0 = 0.10462;
% tau1_0 = 0.057;
tau2_0 = 0.09339;
% tau2_0 = 0.0537;
tau13 = t4_tau13(tau1); %AP1和AP3并发传输重叠的概率

%系统吞吐计算
p_s1 = 1-tau1.*(1-tau1).*(1-tau2).*(1-tau13)+tau2.*((1-tau1).^2);
p_s2 = (tau1.^2).*(1-tau2);
p_s3 = 2*tau1.*(1-tau1).*(1-tau2).*tau13;
p_s = p_s1+p_s2+p_s3;
p_e = ((1-tau1).^2).*(1-tau2);
p_c1 = 1-p_s-p_e;
p_c2 = tau2-tau2.*((1-tau1).^2);

print = ((p_s1+2*p_s2+2*p_s3).*t_payload)./(p_c1.*t_c+p_s3.*t_s1+(
    p_s1+p_s2).*t_s+p_e.*t_SLOT);
S = print.*v_physicalDataRate/1e6;

```

```

function [y,tau1] = t4_tau1(tau1_0)
    %通过曲线零点判断求解tau1的函数
    % 给定参数
    W0 = 32;
    v_physicalDataRate = 158.4 *1e6;
    r = 5;
    m = 5;

    p_1 = tau1_0;
    if(r<=m)
        b00b = 2*(1-p_1).*(1-2*p_1);
        b00a = (1-2*p_1).*(1-p_1.^(r+1))+W0*(1-p_1).*(1-(2*p_1).^(r
            +1));
        b00 = b00b./b00a;
    else
        b00b = 2*(1-p_1).*(1-2*p_1);
        b00a = W0*(1-(2*p_1).^(m+1)).*(1-p_1)+(1-2*p_1).*(1-p_1.^(r
            +1))+W0*(2^m).*(p_1.^(m+1)).*(1-p_1.^(r-m)).*(1-2*p_1);
        b00 = b00b./b00a;
    end

    tem1b = 1-p_1.^(r+1);
    tem1a = 1-p_1;
    y = tau1_0.*tem1a.*b00a-tem1b.*b00b;
    tau1 = tau1_0;
    figure;plot(tau1_0,y);
end

function [y,tau1] = t4_tau2(tau1_0)
    %通过曲线零点判断求解tau2的函数
    % 给定参数
    W0 = 32;
    v_physicalDataRate = 286.8 *1e6;
    r = 5;
    m = 5;

    p_1 = 1-(1-tau1_0).*(1-tau1_0);
    if(r<=m)

```

```

        b00b = 2*(1-p_1).*(1-2*p_1);
        b00a = (1-2*p_1).*(1-p_1.^(r+1))+W0*(1-p_1).*(1-(2*p_1).^(r
            +1));
        b00 = b00b./b00a;
    else
        b00b = 2*(1-p_1).*(1-2*p_1);
        b00a = W0*(1-(2*p_1).^(m+1)).*(1-p_1)+(1-2*p_1).*(1-p_1.^(r
            +1))+W0*(2^m).*(p_1.^(m+1)).*(1-p_1.^(r-m)).*(1-2*p_1);
        b00 = b00b./b00a;
    end
    tem1b = 1-p_1.^(r+1);
    tem1a = 1-p_1;
    y = tau1_0.*tem1a.*b00a-tem1b.*b00b;
    tau1 = tau1_0;
    figure;plot(tau1_0,y);
end

function [tau13] = t4_tau13(tau1)
    %%%求解tau13的函数
    % 通用参数
    t_ACK = 32;
    t_SIFS = 16; % 短帧间隔
    t_DIFS = 43;
    t_SLOT = 9;
    t_ACKTimeout = 64;
    t_phy = 13.6;
    CWmax = 1024;
    L_mac = 30*8; % MAC长度
    L_payload = 1500*8;
    % 给定参数
    W0 = 16;
    v_physicalDataRate = 286.8*1e6;
    r = 32;
    m = 6;

    % 计算参数
    t_MAC = L_mac/v_physicalDataRate*1e6;
    t_payload = L_payload/v_physicalDataRate*1e6;

```

```

t_t = t_phy+t_MAC+t_payload+t_SIFS+t_ACK+t_DIFS;
v = ceil((t_t)./t_SLOT);

%计算b00
p_l = tau1;
if(r<=m)
    b00b = 2*(1-p_l).*(1-2*p_l);
    b00a = (1-2*p_l).*(1-p_l.^(r+1))+W0*(1-p_l).*(1-(2*p_l).^(r+1));
    b00 = b00b./b00a;
else
    b00b = 2*(1-p_l).*(1-2*p_l);
    b00a = W0*(1-(2*p_l).^(m+1)).*(1-p_l)+(1-2*p_l).*(1-p_l.^(r+1))+W0*(2^m).*(p_l.^(m+1)).*(1-p_l.^(r-m)).*(1-2*p_l);
    b00 = b00b./b00a;
end

%计算t_AP13
if v<W0
    Q1 = (v+1)*(1-tau1^(r+1))/(1-tau1);
    Q2 = v*(v+1)/(2*W0);
    Q3 = (1-(tau1/2)^(r+1))/(1-tau1/2);
    Q = Q1-Q2*Q3;
    tau13 = Q*b00;
elseif v<CWmax
    for X = 1:m
        if v<2^X*W0
            break
        end
    end
    Q1 = 0.5*(1-tau1^(X))/(1-tau1);
    Q2 = 0.5*W0*(1-(2*tau1)^(X))/(1-2*tau1);
    Q3 = (v+1)*(tau1^(X)-tau1^(r+1))/(1-tau1);
    Q4 = v*(v+1)/(2*W0);
    Q5 = ((tau1/2)^(X)-(tau1/2)^(r+1))/(1-tau1/2);
    Q = Q1+Q2+Q3-Q4*Q5;
    tau13 = Q*b00;
else
    tau13 = 1;
end

```

```
end  
end
```

#### 问题四仿真器程序

```
clear;clc  
%% 参数定义  
% 数据速率-bps  
% 时间      -us  
% 通用参数  
t_ACK = 32;  
t_SIFS = 16;      % 短帧帧间距  
t_DIFS = 43;  
t_SLOT = 9;  
t_ACKTimeout = 65;  
  
% 给定参数  
CWmin = 16;  
CWmax = 1024;  
c_retrymax = 32;  
v_physicalDataRate = 455.8 * 1e6;  
t_phy = 13.6;  
L_mac = 30 * 8;      % MAC长度  
L_payload = 1500 * 8;  
  
% 计算参数  
t_MAC = L_mac / v_physicalDataRate * 1e6;  
t_payload = L_payload / v_physicalDataRate * 1e6;  
t_t = t_phy + t_MAC + t_payload;  
t_s = t_phy + t_MAC + t_payload + t_SIFS + t_ACK + t_DIFS;  
t_s_res = t_SIFS + t_ACK + t_DIFS;  
t_c = t_phy + t_MAC + t_payload + t_ACKTimeout + t_DIFS;  
  
%% 变量  
CW_1 = CWmin;  
CW_2 = CWmin;  
CW_3 = CWmin;  
bi_1 = randi([0, CW_1-1]);  
bi_2 = randi([0, CW_2-1]);
```

```

bi_3 = randi([0,CW_3-1]);
c_retry1 = 0;
c_retry2 = 0;
c_retry3 = 0;

%% 仿真器
c_success1 = 0;
c_success2 = 0;
c_success3 = 0;
c_collision1 = 0;
c_collision3 = 0;
t_now = t_DIFS;
T_max = 1.0e9;
t_next = 0.0;
t_next1 = t_DIFS;
t_next2 = t_DIFS;
t_next3 = t_DIFS;
state = 0;
state1 = 0;
state2 = 0;
state3 = 0;
state_next1 = 0;
state_next2 = 0;
state_next3 = 0;

while(t_now<T_max)
    if(t_now==t_next1) % AP1 响应
        if(state1 && state2) % AP1 AP2 碰撞
            state_next1 = 0;
            c_collision1 = c_collision1+1;
            CW_1 = CW_1*2;
            if(CW_1>CWmax)
                CW_1 = CWmax;
            end
            c_retry1 = c_retry1+1;
            if(c_retry1>c_retrymax)
                c_retry1 = 0;
                CW_1 = CWmin;
            end
        end
    end
end

```

```

        bi_1 = randi([0,CW_1-1]);
        t_next1 = t_now+t_ACKTimeout+t_DIFS;
    elseif(state1 && ~state2)    % AP1传输完成
        state_next1 = 0;
        t_next1 = t_now+t_SIFS+t_ACK+t_DIFS;
        c_retry1 = 0;
        CW_1 = CWmin;
        bi_1 = randi([0,CW_1-1]);
        c_success1 = c_success1+1;
    elseif(~state1 && ~state2)    % 信道空闲
        if(bi_1==0)                % AP1开始传输
            state_next1 = 1;
            t_next1 = t_now+t_t;
        else                        % AP1回退
            bi_1 = bi_1-1;
            t_next1 = t_now+t_SLOT;
        end
    else % ~state1 && state2        % AP2占用信道
        t_next1 = 2*T_max;
    end
end
if(t_now==t_next3)                % AP3响应
    if(state2 && state3)            % AP2 AP3碰撞
        state_next3 = 0;
        c_collision3 = c_collision3+1;
        CW_3 = CW_3*2;
        if(CW_3>CWmax)
            CW_3 = CWmax;
        end
        c_retry3 = c_retry3+1;
        if(c_retry3>c_retrymax)
            c_retry3 = 0;
            CW_3 = CWmin;
        end
        bi_3 = randi([0,CW_3-1]);
        t_next3 = t_now+t_ACKTimeout+t_DIFS;
    elseif(~state2 && state3)    % AP3传输完成
        state_next3 = 0;
        t_next3 = t_now+t_SIFS+t_ACK+t_DIFS;
    end
end

```



```

        c_retry3 = 0;
        CW_3 = CWmin;
        bi_3 = randi([0,CW_3-1]);
        c_success3 = c_success3+1;
elseif(~state2 && ~state3) % AP3
    if(bi_3==0) % AP3开始传输
        state_next3 = 1;
        t_next3 = t_now+t_t;
    else % AP3回退
        bi_3 = bi_3-1;
        t_next3 = t_now+t_SLOT;
    end
else % AP2占用信道
    t_next3 = 2*T_max;
end
end
if(t_now==t_next2) % AP2响应
    if(state2 && (state1 || state3)) % 碰撞
        state_next2 = 0;
        CW_2 = CW_2*2;
        if(CW_2>CWmax)
            CW_2 = CWmax;
        end
        c_retry2 = c_retry2+1;
        if(c_retry2>c_retrymax)
            c_retry2 = 0;
            CW_2 = CWmin;
        end
        bi_2 = randi([0,CW_2-1]);
        if(t_next1>T_max)
            t_next1 = t_now+t_DIFS;
        elseif(t_next3>T_max)
            t_next3 = t_now+t_DIFS;
        end
        t_next2 = t_now+t_ACKTimeout+t_DIFS;
    elseif(state2 && ~state1 && ~state3) % AP2传输完成
        state_next2 = 0;
        c_retry2 = 0;
        CW_2 = CWmin;
    end
end

```

```

        bi_2 = randi([0,CW_2-1]);
        c_success2 = c_success2+1;
        t_next1 = t_now+t_SIFS+t_ACK+t_DIFS;
        t_next2 = t_now+t_SIFS+t_ACK+t_DIFS;
        t_next3 = t_now+t_SIFS+t_ACK+t_DIFS;
    elseif(~state1 && ~state2 && ~state3) % 信道空闲
        if(bi_2==0) % AP2开始传输
            state_next2 = 1;
            t_next2 = t_now+t_t;
        else % AP2回退
            bi_2 = bi_2-1;
            t_next2 = t_now+t_SLOT;
        end
    else % ~state2 && (state1 || state3) % 信道忙
        t_next2 = min([t_next1, t_next3])+t_SIFS+t_ACK+t_DIFS;
    end
end
% channel
state1 = state_next1;
state2 = state_next2;
state3 = state_next3;
t_next = min([t_next1, t_next2, t_next3]);
t_now = t_next;
end

%% 计算结果
S = (c_success1+c_success2+c_success3)*L_payload/(T_max/1e6);
print1 = (c_success1+c_success2+c_success3)*(t_s)/T_max;
print2 = S/v_physicalDataRate;
fprintf('吞吐量s    传输占比    信道利用率\n%9d    %0.4f    %0.4f\n', S
,print1,print2)

```