



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 北京理工大学

参赛队号 23100070222

1.孙琦

队员姓名 2.贾雅雯

3.杨子硕

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 chain 的信道接入机制建模及系统性能分析

摘 要：

随着无线局域网技术的不断成熟，WLAN (Wireless Local Area Network) 逐渐成为支撑互联网和物联网发展的关键技术。在当今数字化时代，WLAN 已经渗透到了我们生活的方方面面，从家庭到企业，从公共场所到工业应用。因此，如何正确评估和提升 WLAN 系统的吞吐能力，减少信道传输冲突，对于提高无线通信的效率、可靠性和覆盖范围具有非常重要的意义和价值。本文分析在 WLAN 的 DCF 机制下不同参数、不同干扰环境的多 BSS 系统的吞吐，建立了基于马尔可夫链的数值计算系统吞吐模型并对每个问题设计了相应的仿真器去验证模型精确度。

对于问题一，即 2 BSS 系统互听且 SIR 较低的情况：首先，我们以退避计数及退避阶数为二维马尔可夫链的状态参数建立了数值计算模型，计算出稳定状态，进而根据定义计算出了系统的吞吐为 67.17Mbps。随后，我们设计了仿真器算法并用 MATLAB 实现，得到的仿真结果为 65.3Mbps。

对于问题二，即 2 BSS 系统互听且 SIR 较高的情况：注意到此时并行传输会传输成功。因此模型不再有退避阶数，二维马尔可夫链转变成了一维马尔可夫链，仿照问题一的数值求解过程和仿真过程，我们计算出 2 BSS 系统吞吐的数值解为 70.56Mbps，重新设计了仿真器得到 2BSS 系统吞吐的仿真解为 68.9Mbps。

对于问题三，即 2 BSS 系统不互听且 SIR 较低及存在 10%丢包率的情况：我们首先考虑没有丢包率的模型，由于系统不互听， $E(\psi)$ 不能再用原来的 P_{tr} 去计算，而是需要找到节点能够探测到至少有一个节点在传输的概率来计算。这很大程度上提升了模型的精确度，也是本文的一个创新点。随后我们设计了一种双线交替更新两个系统的状态的算法去仿真系统，即用一个系统的状态去判断另一个系统未来的状态。最终得到的数值解为 21.56Mbps，仿真解为 23.21Mbps。

对于问题四，即 3 BSS 系统，其中 AP2 与 AP1 和 AP3 互听，AP1 和 AP3 不互听且 AP1 与 AP3 间 SIR 较高的情况：首先我们将三个系统分别考虑，仿照前几问分别写出它们的马尔可夫链，找到每个系统的参数之间的关系，然后联立方程组求解。随后我们设计了一种基于状态转移的算法来仿真这个系统，即通过当前系统的状态来判断系统的更新情况，并且判断系统的下一个状态。最终得到的数值解为 112.59Mbps，仿真解为 115.26Mbps。

本文通过分析数值解和仿真解的绝对误差和相对误差，得到了数学模型精度高，稳定性强的结论。本文通过分析仿真图像，证明了 DCF 机制下的信号接入系统的有效性和稳定性。不同干扰情景下的 BSS 系统吞吐计算结果也很好的反映了现实生活中的信号干扰现象。

关键词： WLAN； DCF 机制； 马尔可夫链； 退避算法； 仿真器

目录

1. 问题描述	5
1.1 问题背景	5
1.2 问题提出	5
2. 模型假设	6
3. 符号说明及参数计算	7
3.1 符号说明	7
3.2 参数计算	8
4. 问题一：2 BSS 互听且 SIR 较高的场景	9
4.1 问题一模型的建立	9
4.1.1 DCF 机制	9
4.1.2 基于 Markov chain 的 DCF 机制建模	10
4.2 问题一数值求解	13
4.3 问题一仿真算法	14
4.4 问题一模型的结果分析	15
5. 问题二：2 BSS 互听且 SIR 较低的场景	16
5.1 问题二模型的建立	16
5.2 问题二数值求解	17
5.3 问题二仿真算法	18
5.4 问题二模型的结果分析	19
6. 问题三：2 BSS 不互听、SIR 较小且存在信道质量丢包	20
6.1 问题三模型的建立	20
6.1.1 不考虑因信道质量丢包的情况	20
6.1.2 考虑因信道质量丢包的情况	23
6.2 问题三数值求解	23
6.3 问题三仿真算法	24
6.4 问题三模型的结果分析	25
7. 问题四：3 BSS 混合互听的场景	27
7.1 问题四模型的建立	27
7.2 问题四数值求解	29

7.3 问题四仿真算法.....	29
7.3.1 节点状态	29
7.3.2 状态转移函数	30
7.3.3 仿真算法设计	31
7.4 问题四模型的结果分析.....	33
8. 模型与仿真的讨论与改进	35
8.1 模型与仿真的优点.....	35
8.2 模型与仿真的缺点及改进.....	36
参考文献.....	37
附录 A MATLAB 程序 q1.m	38

1. 问题描述

1.1 问题背景

近年来，电脑、手机等已成为人们日常生活中的一部分，随着现代科学技术的快速发展，特别是无线局域网技术（WLAN, wireless local area network）的不断成熟，打破了过去只能依靠网线才能连接局域网络的局限，使 WLAN 技术得到广泛的运用。由于 WLAN 有着灵活性、移动性、高吞吐、应用普及等优势，逐渐成为支撑互联网和物联网发展的重要承载技术。因此有必要让 WLAN 提高系统的吞吐量和降低网络的传输时延，改进并优化 WLAN 网络信道接入机制对 WLAN 有着重要的意义与价值。

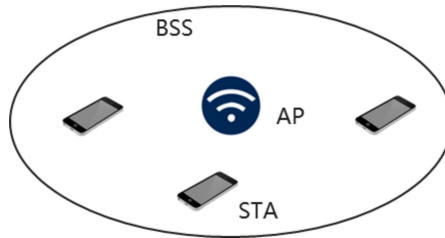


图 1 WLAN 网络图

如图 (1) 所示，基本服务集（BSS, basic service set）是 WLAN 的基本组成部分。而 BSS 是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）组成的。此时，当 STA 和 AP 在同一个 BSS 中时，则称 STA 关联到了 AP。本文将 AP 和 STA 统称为节点，各节点共享信道。考虑到每个节点的发送和接收不能同时发生，本文通过载波侦听多址接入/退避的机制来避免冲突。

对于 WLAN 组网中的单 BSS 建模问题已经被很好地解决，但在考虑多 BSS 建模的同时考虑同频干扰影响，目前学术研究有限，所处的情况较为复杂。因此，本文考虑分布式信道接入和二进制指数退避，同时针对发送节点间是否互听、不同信道质量表现等不同情况，构建 WLAN 网络信道接入机制模型，评估不同状态下系统的吞吐情况。

1.2 问题提出

本赛题要求就以下四种情形对 WLAN 网络信道接入机制问题建立模型，并通过数值分析方法进行求解，并编写仿真器评估模型精确度。

问题 1： 本题考虑 2 个 BSS 互听的场景，仅下行，且存在同频干扰影响，并发时两个终端接收到数据的 SIR 较低，导致两个 AP 的数据传输都失败。根据给定的 AP 发送包的载荷长度、PHY 头时长、MAC 头、物理层速率等数据，对该

系统进行建模，用数值分析方法求解，评估系统的吞吐，并编写仿真器评估模型精确度。

问题 2：本题考虑 2 个 BSS 互听的场景，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。根据给定的 AP 发送包的载荷长度、PHY 头时长、MAC 头、物理层速率等数据，对该系统进行建模，用数值分析方法求解，评估系统的吞吐，并编写仿真器评估模型精确度。

问题 3：本题考虑 2 个 BSS 不互听的场景，并发时两个终端接收到数据的 SIR 较小，会导致两个 AP 的发包均失败。当仅有一个 AP 发送数据时，即便不存在邻 BSS 干扰，也会有 10% 不同程度的丢包。根据给定的 AP 发送包的载荷长度、PHY 头时长、MAC 头、物理层速率等数据，对该系统进行建模，对该 2 BSS 系统进行建模，用数值分析方法求解，评估系统的吞吐，并编写仿真器评估模型精确度。

问题 4：本题考虑 3 个 BSS 场景，AP1 与 AP3 不互听，AP2 与两者都互听。假设 AP1 和 AP3 发包时间交叠时，SIR 较大，两者发送均成功。根据给定的 AP 发送包的载荷长度、PHY 头时长、MAC 头、物理层速率等数据，对该系统进行建模，用数值分析方法求解，评估系统的吞吐，并编写仿真器评估模型精确度。

2. 模型假设

- 假设一：在问题三中，一个 AP 在发包过程中从 PHY 到 ACK 之间任意时刻被另一系统的信号干扰都会导致发包失败；
- 假设二：在问题四中，AP1 与 AP2、AP1 与 AP3 的 SIR 均较大，会导致并行传输的失败；

3. 符号说明及参数计算

3.1 符号说明

符号	意义
L_{pl}, L_{MAC}	载荷长度,MAC 头长度 (<i>Bytes</i>)
$t_{phy}, t_{SIFS}, t_{DIFS}, t_{slot}$	phy,SIFS,DIFS,slotTime 的时长 (μs)
$t_{ACK}, t_{ACKtimeout}$	ACK 时长, 等待超时时长 (μs)
T, T_c, T_s	总时长, 碰撞时长, 成功时长 (μs)
$E[P]$	数据帧的有效载荷传输时长 (μs)
$rate$	物理层速率 (<i>Mbps</i>)
CW_i, CW_{min}, CW_{max}	系统 i 的竞争窗口值, 和其下限上限
r	最大重传次数
P_e	丢包率
$collision_i$	系统 i 的碰撞次数
S	吞吐量 (<i>Mbps</i>)
$b(t), s(t)$	t 时刻一个节点退避计数, 退避阶数
i	一个数据的发送次数
m	最大退避阶数
p	某个时隙发生碰撞的概率
τ / τ_1	节点在一个时隙发送数据帧的概率
τ_2	节点在脆弱时间内能发送数据帧的概率
t_{VP}	脆弱时期 (μs)
P_{tr}	某一时隙至少有一个节点在传输的概率
P_{tri} / P'_{tr}	节点 i 能探测到至少有一个节点在传输的概率
P_{si}	系统 i 成功传输的条件概率

3.2 参数计算

题干中给出的载荷长度、MAC 头长度的单位是 *Bytes*, 物理层速率给出的单位是 *Mbps*, 已知 $1\text{Bytes} = 8\text{bits}$, $1\text{Mbps} = 10^6\text{bits/s}$, 为了便于计算, 将载荷长度、MAC 头长度的单位转换为 *bits*, 将物理层速率的单位转换为 *bit*, 根据式 (1-2), 计算得到以 μs 单位的 MAC 头时长 t_{MAC} 、 $E[P]$ 。

$$t_{MAC} = \frac{L_{MAC}}{rate} \quad (1)$$

$$E[P] = \frac{L_{pl}}{rate} \quad (2)$$

其中, t_{MAC} 、 $E[P]$ 分别为 MAC 层头和有效载荷传输的时长, L_{MAC} 、和 L_{pl} 为 MAC 层头和有效载荷传输的字节长度, $rate$ 为物理层速率。

根据式 (3,4), 计算得到某一 DCF 帧间距内节点成功传输时长 T_s 和发生碰撞时长 T_c 。

$$T_s = t_{MAC} + t_{phy} + E[P] + t_{SIFS} + t_{ACK} + t_{DIFS} \quad (3)$$

$$T_c = t_{MAC} + t_{phy} + E^*[P] + t_{DIFS} + t_{ACKTimeout} \quad (4)$$

其中, t_{MAC} 、 t_{phy} 分别是 MAC 层头和物理层头时长, $E[P]$ 为数据帧的有效载荷传输时长, $E^*[P]$ 为发生冲突时较长数据帧的有效载荷传输时长。在本文中, 由于考虑的所有问题有效载荷长度都为 1500Bytes , 所以 $E[P]$ 与 $E^*[P]$ 始终相等, 我们在后文中统称 $E[P]$ 。并且在本文中, 考虑的所有问题以下参数均为常量:

表 1 参数表

常量	含义	值	单位
t_{phy}	物理层传输时长	13.6	μs
t_{ACK}	确认帧时长	32	μs
$t_{ACKTimeout}$	等待超时时长	65	μs
t_{slot}	时隙时长	9	μs
t_{SIFS}	短帧帧间距时长	16	μs
t_{DIFS}	载波侦听时长	43	μs
L_{mac}	媒体控制数据长度	30	Bytes
L_{pl}	载荷数据长度	1500	Bytes

4. 问题一：2 BSS 互听且 SIR 较高的场景

4.1 问题一模型的建立

如前所述，问题一只需要考虑 2 BSS 互听场景。假设两个 AP 同时回退到 0 而发送数据时，存在同频干扰。由于节点并发时的 SIR 较低，导致两个 AP 的数据传输都失败，但不因信道质量差而丢包。本节从 DCF 机制、基于 Markov chain 的模型建立两个方面来展开介绍。

4.1.1 DCF 机制

为了避免多个节点同时访问网络所带来的冲突问题，在 WiFi 协议中设置了分布式协调功能（DCF, distributed coordination function）。本题将从信道可用评估（CCA, clear channel assessment）、随机回退、数据传输这三个阶段依次介绍。

阶段一：CCA 机制

当一个节点打算发送时，首先进行一个 DCF 帧间距内（DIFS, DCF inter-frame space）的载波侦听。如果在该时段内接收到的信号能量强度（RSSI, received signal strength indication）低于 CCA 门限，判断信道为空闲，否则，判断信道为繁忙。

阶段二：随机回退

本题采用了随机回退机制，避免信道空闲时多个节点同时准备数据传输^[1]。设 CW 为竞争窗口，节点从 $[0, CW-1]$ 的均匀分布选取一个随机数作为回退数，等待该回退数个时隙长度 $slotTime$ ，随机回退时段时长为回退数 cw 乘以 $slotTime$ 。

为了降低节点再次发生碰撞的概率，本题采用二进制指数退避算法确定回退时间。设 CW 的初始值为 CW_{min} ，当每次数据传输失败后重传数据帧时，则 CW 翻倍。如果 CW 达到了 CW_{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回退。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。竞争窗口 CW 可表示为：

$$\begin{cases} W_i = 2^i W_0 & 0 \leq i \leq m \\ W_i = 2^m W_0 & m \leq i \leq r \end{cases} \quad (5)$$

其中， i 表示一个数据的发送次数， r 为最大重传次数， m 是最大退避阶数。

阶段三：数据传输

回退到 0 的节点发送一个数据帧，接收节点成功接收到数据之后等待短帧帧间距（SIFS, Short inter-frame space）后，回复 ACK 确认帧。如果发送节点收到 ACK，则数据发送成功。如果发送数据帧没有被接收节点成功接收，或者 ACK 发送失败，或者 ACK 没有被发送节点收到，则数据传输失败，发送节点需要在等待超时时间 ACKTimeout 后重传数据。

4.1.2 基于 Markov chain 的 DCF 机制建模

本题利用马尔可夫链理论对 DCF 机制建模。本题假设信道不因信道质量差而丢包，但当 2 个及以上节点同时回退到 0 发送数据时，节点会因碰撞而丢包。因此，信道可能处于三种状态，即：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。将退避器所处的阶数和随机回退数用二维 Markov chain 表示^[2]。

首先，令 $b(t)$ 和 $s(t)$ 代表 t 时刻一个节点退避随机过程的退避计数和退避阶数，这里的 t 是一个离散的虚拟时隙的开始时刻。二维 $\{b(t), s(t)\}$ 随机过程可以用二维 Markov chain 表示。 $b_{i,k} = \lim_{t \rightarrow \infty} P\{s(t) = i, b(t) = k\}$ 代表二维 Markov chain 的稳态解， $i \in [0, m], k \in [0, W_i - 1]$ 。

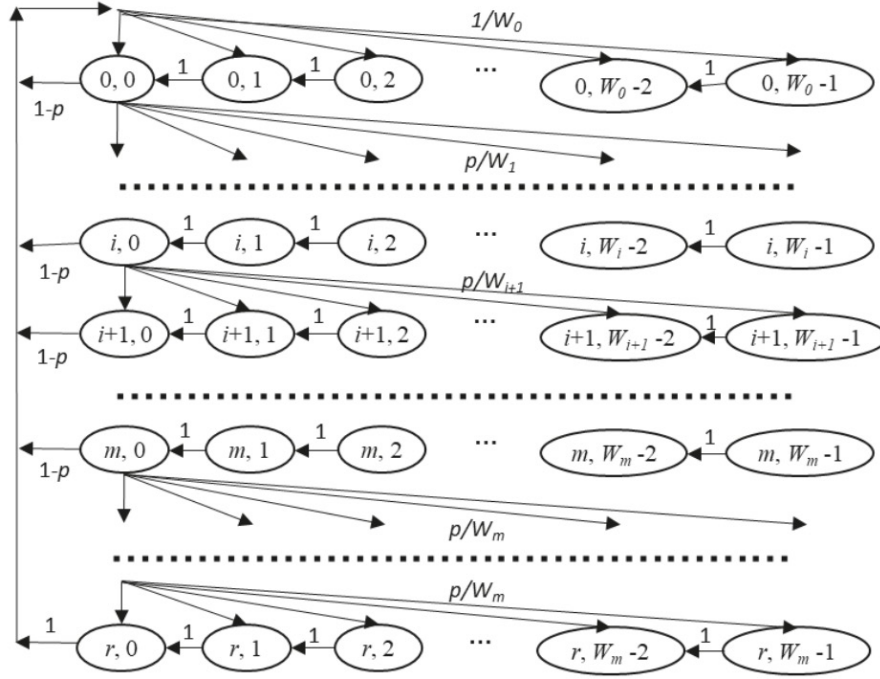


图 2 问题一的 Markov 链模型

p 为某个时隙发生碰撞的概率，Markov chain 一步状态转移概率可以表示为：

$$\begin{cases} P\{i, k|i, k+1\} = 1 & k \in [0, W_i - 2], i \in [0, r] \\ P\{0, k|i, 0\} = (1-p)/W_0 & k \in [0, W_0 - 1], i \in [0, r] \\ P\{i, k|i-1, 0\} = p/W_i & k \in [0, W_i - 1], i \in [1, r] \\ P\{0, k|r, 0\} = 1/W_0 & k \in [0, W_0 - 1] \end{cases} \quad (6)$$

式 (6) 中每个式子分别代表一定的物理含义。第一个等式代表，未达到重传上限时，退避计数器在每个空闲时隙的开始时刻减 1 的概率是 1。第二个等式代表，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 $[0, W_0 - 1]$ 中等概率选一个随机数进行回退。第三个等式代表，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 阶回退过程，并在 $[0, W_i - 1]$ 中等概率选一个随机数进行回退。最后一个等式代表，当节点到达最大的传输次数以后，无论成功还是失败，CW 都会重置。

该 Markov chain 的任意状态之间可达，是不可约的。任意状态到另一状态的步长不存在周期。从任何状态出发，都能到达另一状态，具有常返性。因此该二进制退避过程的非周期不可约 Markov chain 具有稳态解，且所有稳态的概率之和为 1。

令 $b_{i,k} = \lim_{t \rightarrow \infty} P\{s(t) = i, b(t) = k\}$, $i \in [0, r]$, $k \in [0, W_i - 1]$ 表示 Markov chain 的稳态解，对于一次发送失败的情况，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,

$$\begin{aligned} b_{i,0} &= b_{i-1,0} \times (p \times \frac{1}{W_i} + p \times \frac{1}{W_i} \times 1 + \dots + p \times \frac{1}{W_i} \times 1^{W_i-1}) \\ &= p \times b_{i-1,0} \quad (0 < i \leq r) \end{aligned} \quad (7)$$

进而有:

$$b_{i,0} = p^i \times b_{0,0} \quad (0 \leq i \leq r) \quad (8)$$

同理, 对于任一状态, 若 $0 < i < r$, 则是从一次发送失败的状态, 通过竞争窗口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,

$$b_{i,k} = \begin{cases} b_{i-1,0} \times p \times \frac{W_i - k}{W_i} & 0 < i < r, \\ (1-p) \times \frac{W_i - k}{W_i} \times \sum_{j=0}^{r-1} b_{j,0} + \frac{W_i - k}{W_i} b_{r,0} & i = 0. \end{cases} \quad (9)$$

将式 (7) 代入式 (9), 可得,

$$b_{i,k} = \frac{W_i - k}{W_i} \times b_{i,0} \quad (0 \leq i \leq r, 0 \leq k \leq W_i - 1) \quad (10)$$

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

$$\begin{aligned} 1 &= \sum_{k=0}^{W_i-1} \sum_{i=0}^r b_{i,k} \\ &= \sum_{i=0}^r b_{i,0} \sum_{k=0}^{W_i-1} \frac{W_i - k}{W_i} \\ &= \sum_{i=0}^r b_{i,0} \frac{W_i + 1}{2} \end{aligned} \quad (11)$$

根据式 (5)、式 (7) 和式 (11), 可以求得:

$$b_{0,0} = \begin{cases} \frac{2(1-p)(1-2p)}{(1-2p)(1-p^{r+1}) + W_0(1-p)(1-(2p)^{r+1})} & r \leq m \\ \frac{2(1-p)(1-2p)}{W_0(1-(2p)^{m+1})(1-p) + (1-2p)(1-p^{r+1}) + W_0 2^m p^{m+1}(1-p^{r-m})(1-2p)} & m < r \end{cases} \quad (12)$$

节点随机回退到 0 时发送数据, 因此节点在一个时隙发送数据帧的概率为:

$$\tau = \sum_{i=0}^r b_{i,0} = b_{0,0} \times \frac{1-p^{r+1}}{1-p} \quad (13)$$

传输数据发生冲突时，至少有另外一个节点也传输数据，若有 N 个节点，条件碰撞概率 p 可表示为

$$p = 1 - (1 - \tau)^{N-1} \quad (14)$$

由于本题仅考虑 2 个节点，即 $N = 2$ ：

$$p = 1 - (1 - \tau) = \tau \quad (15)$$

式 (12)、(13)、(15) 是关于 p 和 τ 的二元非线性方程，联立可求解。

设任一时隙时间中至少有一个传输的概率 P_{tr} 。在已知至少有一个节点在传输的情况下，节点能够传输成功当且仅当其中一个节点在传输且另一个节点不在传输，故至少有一个传输的概率 P_{tr} 与节点传输成功的条件概率 P_s 可表示为

$$\begin{aligned} P_{tr} &= 1 - (1 - \tau)^2 \\ P_s &= \frac{C_2^1 \tau (1 - \tau)}{P_{tr}} = \frac{2\tau(1 - \tau)}{P_{tr}} \end{aligned} \quad (16)$$

我们用 $E[\Psi]$ 表示信道上两个连续传输之间的空闲时隙个数的期望。对于 2 BSS 互听的情况， $E[\Psi]$ 可表示为：

$$E[\Psi] = \frac{1}{P_{tr}} - 1 \quad (17)$$

本题通过吞吐量这一重要指标来评价系统性能，吞吐 S 由信道的利用率与物理层速率（单位 $Mbps$ ）的乘积表示：

$$\begin{aligned} S &= \frac{E[\text{一个时隙内传输的有效载荷发送时长}]}{E[\text{一个时隙长度}]} \times \text{物理层速率} \\ &= \frac{P_s E[P]}{t_{slot} E[\Psi] + P_s T_s + (1 - P_s) T_c} \times \text{rate} \end{aligned} \quad (18)$$

4.2 问题一数值求解

在问题一中， $\text{rate} = 455.8$ 。代入式 (1-2) 中可得：

$$\begin{aligned} t_{MAC} &= \frac{L_{MAC}}{\text{rate}} = \frac{30 \times 8}{455.8} = 0.53 \\ E[P] &= \frac{L_{pl}}{\text{rate}} = \frac{1500 \times 8}{455.8} = 26.33 \end{aligned} \quad (19)$$

进而有

$$\begin{aligned} T_s &= t_{MAC} + t_{phy} + E[P] + t_{SIFS} + t_{ACK} + t_{DIFS} = 131.45 \\ T_c &= t_{MAC} + t_{phy} + E[P] + t_{DIFS} + t_{ACKTimeout} = 148.45 \end{aligned} \quad (20)$$

将 $CW_{min} = 16, CW_{max} = 1024, r = 32$ 代入式(12)中,并联立式(12)、(13)、(15),解得:

$$p = \tau = 0.1046 \quad (21)$$

将式(20)分别代入式(16)、(17)、(18)中,求得

$$S = 67.1744 \quad (22)$$

中间代入、联立的过程以及其它变量的值详见附件 Q1.py, 此处不全部展示。

4.3 问题一仿真算法

本小节我们设计了一种仿真算法来验证模型的精确度, 其基本思想是在设定的时间内, 模拟 AP 向各自关联的 STA 发送数据的情景, 算法流程如算法 1 所示:

在步骤 1 中, 我们初始化一些变量以便于后续的计算。其中 T 是当前模拟的时间, TTs 是总成功时间, cw 是当前竞争窗口大小。 cw_1 与 cw_2 指系统 1 与系统 2 的剩余回退时隙个数, $collision$ 是已经碰撞的次数。在步骤 2 中, 我们判断下一阶段两个系统会不会发生碰撞, 从而确定整个信道下一阶段的状态, 步骤 2 (a) 代表碰撞的情况, 步骤 2 (b) 代表成功传输的情况, 我们根据不同情况用对应的逻辑更新相关参数。在步骤 4 中, 我们根据吞吐公式计算出吞吐量并输出。

相关参数的更新逻辑在附件 q1.m 中有清晰展示。

算法 1: 2 BSS 互听且 SIR 较低的场景——DCF 机制下的仿真模拟算法

输入: 物理层速率 $rate$, 最小竞争窗口 CW_{min} , 最大竞争窗口 CW_{max} , 最大重传次数 r , 总模拟时间 $t_{simulate}$;

输出: 吞吐量 S ;

1、初始化

$T = 0$; $TTs = 0$; $cw = CW_{min}$; $cw_1 = rand([0, cw - 1])$; $cw_2 = rand([0, cw - 1])$; $collision = 0$;

2、模拟 AP 向各自关联的 STA 发送数据情景

(a) *if* $cw_1 == cw_2$:

if $collision > r$:

$collision \leftarrow 0$; $CW \leftarrow CW_{min}$; 生成随机回退数 cw_1, cw_2 ;

else, 更新 T 、 TTs 、 CW 、 $collision$, 生成随机回退数 cw_1, cw_2 ;

(b) *else*,

if $cw_1 < cw_2$: 更新 T 、 TTs 、 cw_2 , 重置 CW 、 $collision$, 生成随机回退数 cw_1 ;

else, 更新 T 、 TTs 、 cw_1 , 重置 CW 、 $collision$, 生成随机回退数 cw_2 ;

3、若 $T < t_{simulate}$, 回到步骤 2;

4. 输出吞吐量 $S = t_{pl} \times TTs \times rate / (10^6 \times Ts \times T)$

4.4 问题一模型的结果分析

通过问题一中建立的数值分析模型, 利用 Python 软件求解, 本文得出在 SIR 较低条件下, 2 BSS 系统吞吐的数值解为 $67.17Mbps$ 。通过问题一中建立的仿真模型, 利用 MATLAB 软件求解, 本文得出在 SIR 较低条件下, 2 BSS 系统吞吐的仿真解为 $65.3Mbps$ 。数值解与仿真解的绝对误差为 $1.87Mbps$, 相对误差为 2.78% 。数值解与仿真解非常接近, 这验证了使用本文的数值模型计算 SIR 较低且互听时 2 BSS 系统的吞吐, 精度是比较高的。

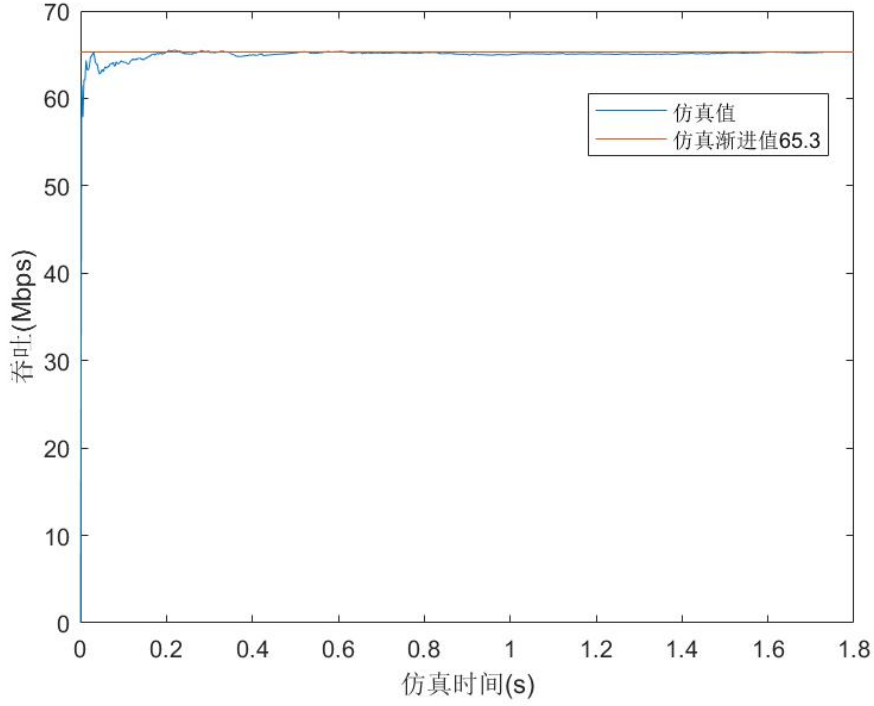


图3 问题一的仿真图

本文分析在问题一中建立的数值分析模型性质，得到问题一模型中马氏链是有稳态解的。本文分析问题一的仿真图（图3）得到随着仿真时间的推移，系统的吞吐渐进趋于一个稳定值。这验证了问题一模型中马氏链具有稳定解的性质。

5. 问题二：2 BSS 互听且 SIR 较低的场景

5.1 问题二模型的建立

本题考虑两个 AP 并发传输时两个终端接收到信号的 SIR 较高，两个 AP 的数据传输都能成功，物理层速率 $\text{rate}=275.3\text{Mbps}$ 。其他条件同问题 1，因此，本题将问题一构建的基于 Markov chain 的模型进行调整。

根据本题的 2 BSS 建模，我们可以知道，不存在因节点同时发送数据发送碰撞而丢包的可能性，一个数据的发送次数仅为 1。因此，问题二是问题一的特殊情况，在问题一的 DCF 机制模型基础上，我们可以不用关注 t 时刻一个节点退避随机过程的退避计数，只需考虑退避阶数 $s(t)$ ^[3]。

$s(t)$ 随机过程可以用一维 Markov chain 表示，如图 (4) 所示。 $b_{0,k} = \lim_{t \rightarrow \infty} P\{b(t) = k\}$ 代表一维 Markov chain 的稳态解， $k \in [0, W_0 - 1]$ 。

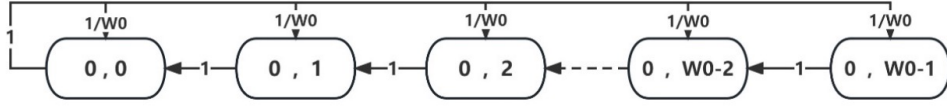


图 4 问题二的 Markov 链模型

根据式 (10)，我们可以得出，

$$b_{0,k} = \frac{W_0 - k}{W_0} \times b_{0,0} \quad 0 \leq k \leq W_0 - 1 \quad (23)$$

根据 Markov chain 的性质，所有稳态的概率之和为 1，因此满足以下式子：

$$\begin{aligned} 1 &= \sum_{k=0}^{W_0-1} b_{0,k} \\ &= b_{0,0} \sum_{k=0}^{W_0-1} \frac{W_0 - k}{W_0} \\ &= b_{0,0} + b_{0,0} \frac{1}{W_0} + b_{0,0} \frac{2}{W_0} + \cdots + b_{0,0} \frac{W_0 - 1}{W_0} \end{aligned} \quad (24)$$

可以求得：

$$b_{0,0} = \frac{2}{W_0 + 1} \quad (25)$$

节点随机回退到 0 时发送数据，因此节点在一个时隙发送数据帧的概率 $\tau = b_{0,0}$ 。传输数据发生冲突时，至少有另外一个节点也传输数据，共有 N 个节点，因此条件碰撞概率 p 可表示为：

$$p = 1 - (1 - \tau)^{N-1} \quad (26)$$

由于本题仅考虑 2 个节点，即 $N = 2$ ，本题最终得出条件碰撞概率 p 为：

$$p = \tau = \frac{2}{W_0 + 1} = \frac{2}{17} \quad (27)$$

5.2 问题二数值求解

在问题二中， $rate = 275.3$ 。代入式 (1-2) 中可得：

$$\begin{aligned} t_{MAC} &= \frac{L_{MAC}}{rate} = \frac{30 \times 8}{275.3} = 0.87 \\ E[P] &= \frac{L_{pl}}{rate} = \frac{1500 \times 8}{275.3} = 43.59 \end{aligned} \quad (28)$$

进而有

$$\begin{aligned} T_s &= t_{MAC} + t_{phy} + E[P] + t_{SIFS} + t_{ACK} + t_{DIFS} = 149.06 \\ T_c &= t_{MAC} + t_{phy} + E[P] + t_{DIFS} + t_{ACKTimeout} = 166.06 \end{aligned} \quad (29)$$

将 $p = \tau = \frac{2}{17}$ 代入式 (16)、(17)、(18) 中, 求得吞吐

$$S = 70.56 \quad (30)$$

中间代入、联立的过程以及其它变量的值详见附件 Q2.py。

5.3 问题二仿真算法

针对 SIR 较低这一条件, 本题在算法 1 基础上对 2 BSS 互听场景的模拟仿真进行了调整。注意到问题 2 中两个系统永远不会传输失败, 所以在步骤 2 (a) 中更新逻辑产生了变化, 并且不需要更新竞争窗口 CW 与碰撞次数 $collision$ 。其余步骤与算法 1 均相同。

相关参数的更新逻辑在附件 q2.m 中有清晰展示。

算法 2: 2 BSS 互听且 SIR 较高的场景——DCF 机制下的仿真模拟算法

输入: 物理层速率 $rate$, 最小竞争窗口 CW_{min} , 最大竞争窗口 CW_{max} , 最大重传次数 r ;

输出: 吞吐量 S ;

1、初始化

$T = 0$; $TTs = 0$; $cw = CW_{min}$; $cw_1 = rand([0, cw - 1])$; $cw_2 = rand([0, cw - 1])$;

2、模拟 AP 向各自关联的 STA 发送数据情景

(a) *if* $cw_1 == cw_2$:

更新 T 、 TTs , 生成随机回退数 cw_1, cw_2 ;

(b) *else*, cw 较小的节点优先进入信道, 另一个保持不变

if $cw_1 < cw_2$

更新 T 、 TTs 、 cw_2 , 生成随机回退数 cw_1 ;

else, 更新 T 、 TTs 、 cw_1 , 生成随机回退数 cw_2

3、若 $T < t_{simulate}$, 回到步骤 2;

4、输出吞吐量 $S = t_{pl} \times TTs \times rate / (10^6 \times Ts \times T)$

5.4 问题二模型的结果分析

通过问题二中建立的数值分析模型, 利用 Python 软件求解, 本文得出在 SIR 较高的条件下, 2 BSS 系统吞吐的数值解为 $70.56Mbps$ 。通过问题二中建立的仿真模型, 利用 MATLAB 软件求解, 本文得出在 SIR 较高的条件下, 2 BSS 系统吞吐的仿真解为 $68.9Mbps$ 。数值解与仿真解的绝对误差为 $1.66Mbps$, 相对误差为 2.35% 。数值解与仿真解非常接近, 这验证了使用本文的数值模型计算 SIR 较高且互听时 2 BSS 系统的吞吐, 精度是比较高的。

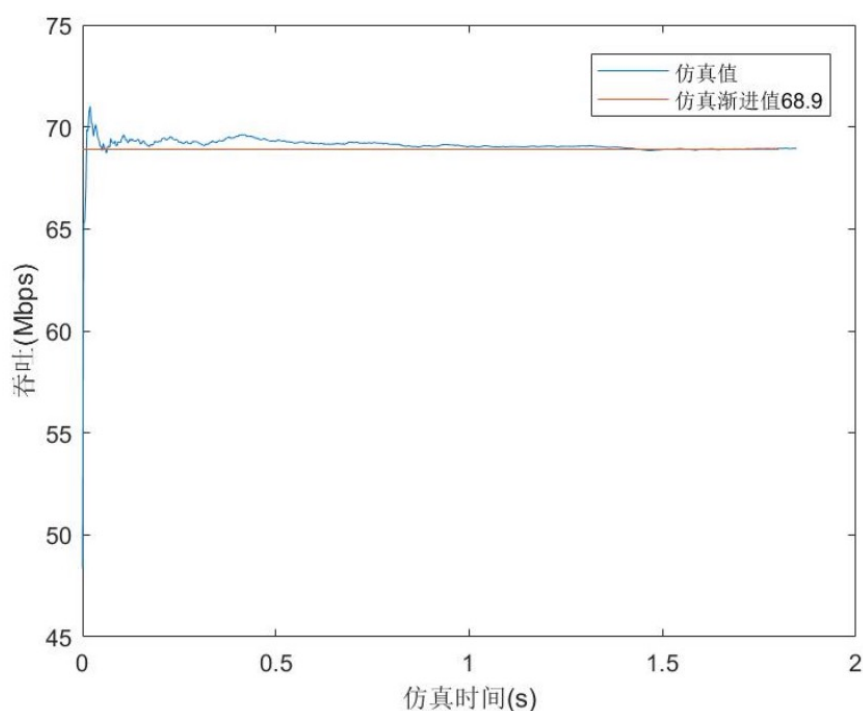


图5 问题2的仿真图

本文分析问题二中建立的数值分析模型的性质，得到问题二中马氏链是有稳态解的。本文分析问题二的仿真图（图5）得到随着仿真时间的推移，系统的吞吐渐进趋于一个稳定值。这验证了问题二模型中马氏链具有稳定解的性质。

与问题一相比较，在参数不变的2 BSS系统下（与问题一参数保持一致），SIR较高的系统吞吐（76.2Mbps）比SIR较低的系统吞吐（65.3Mbps）高。这很好的反映了现实中信号多且杂的场景会导致信号干扰和丢失数据的问题。

6. 问题三：2 BSS 不互听、SIR 较小且存在信道质量丢包

6.1 问题三模型的建立

本题假设两个AP并发时两个终端接收到数据的SIR较小，两个AP同时数据传输会失败，且考虑了因信道质量丢包的情况，其他条件同问题1，因此，本题将基于问题一的Markov chain模型进行改进。我们先在6.1.1中考虑不丢包的情况建立模型，然后在6.1.2节以此为基础再加入丢包的影响对模型进行调整。

6.1.1 不考虑因信道质量丢包的情况

与问题一不同的是，本题考虑的是2 BSS不互听的场景。假设一个AP回退至0准备开始发送数据，由于同频AP之间距离较远，另一个AP认为信道空闲，

因此总是在退避和发送数据，此时两节点会发生碰撞^[4]。因此，本题定义了脆弱时期 t_{VP} ，代表两个节点可能碰撞的时期，即：节点回退至 0 准备开始发送数据至完成数据传输这一时期。节点成功传输数据的时长还包括了侦听信道 DIFS 时长，在这个阶段其他节点传输数据是不会发生碰撞的，因此我们可以减去这部分的时长。脆弱时期 t_{VP} 的表达式为：

$$t_{VP} = Ts - t_{DIFS} \quad (31)$$

接下来，根据脆弱时期 VP 和等待回退数个时隙长度 t_{slot} ，确定不可发送的时隙个数 V ， V 也可以称以 $slotTime$ 为单位的脆弱周期长度，表示为：

$$V = \left\lfloor \frac{t_{VP}}{t_{slot}} \right\rfloor \quad (32)$$

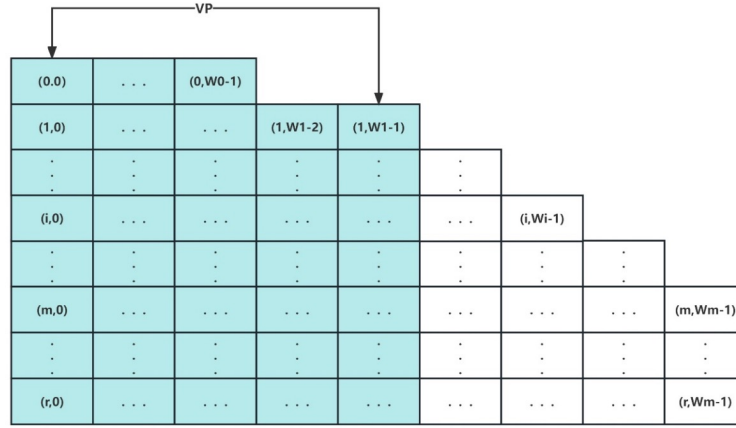
注意到问题 3 中系统的 Markov chain 并没有变化，于是式 (10) 对本题仍然成立，此时的 τ 指的是节点回退结束的概率，在接下来的讨论中，我们记为 τ_1 。

节点在脆弱时期传输数据与其他节点传输发送碰撞的概率 τ_2 。 τ_2 即为蓝框的状态概率之和，如图 (6b) 所示，因为在脆弱时期内，这些状态可以回退成功。可以观察到， τ_2 的表达式可以分为三个情况，即： $V < W_0$ ， $W_{X-1} < V < W_X$ ($1 \leq X \leq m$)， $V > W_m$ 。

$$\tau_2 = \sum_{i=0}^r \sum_{k=0}^{\min(V, W_k-1)} = \begin{cases} \sum_{i=0}^r \sum_{k=0}^V & V < W_0 \\ \sum_{i=0}^X \sum_{k=0}^{W_k-1} + \sum_{i=X+1}^r \sum_{k=0}^V & W_{X-1} < V < W_X \\ & (1 \leq X \leq m) \\ \sum_{i=0}^r \sum_{k=0}^{W_k-1} = 1 & V > W_m \end{cases} \quad (33)$$

(0,0)	...	(0,W0-1)							
(1,0)	(1,W1-2)	(1,W1-1)					
.
.
(i,0)	(i,Wi-1)			
.
.
(m,0)	(m,Wm-1)	
.
.
(r,0)	(r,Wm-1)	

a. 问题三的状态集合图



b. 在脆弱时期内能回退成功的状态集合图

图 6 问题三状态示意图

传输数据发生碰撞时，至少有另外一个节点也传输数据，共有 2 个节点，因此条件碰撞概率 p 可表示为：

$$p = 1 - (1 - \tau_2) = \tau_2 \quad (34)$$

式 (12)、(13)、(34) 是关于 P 和 τ 的二元非线性方程，联立可求解。

设 P_{tr} 是在考虑的时隙时间内至少有一个传输的概率， P'_{tr} 是节点能够检测到信道繁忙的概率，由于存在隐藏节点的问题，节点只能探测到自己的情况，无法知道其他节点是否在传输数据，因此可以得出

$$P_{tr} = 1 - (1 - \tau_1)^2 \quad (35)$$

$$P'_{tr} = 1 - (1 - \tau_1)^1 = \tau_1 \quad (36)$$

当至少有一个节点传输，恰好有一个节点在信道上传输成功，另一个节点不在传输数据的概率 P_s 可表示为：

$$P_s = \frac{C_2^1 \tau_1 (1 - \tau_2)}{P_{tr}} = \frac{2\tau_1 (1 - \tau_2)}{P_{tr}} \quad (37)$$

$E[\Psi]$ 表示信道上两个连续传输之间的空闲时隙个数的期望。在 BSS 不互听的条件下， $E[\Psi]$ 可表示为：

$$E[\Psi] = \frac{1}{P_{tr'}} - 1 = \frac{1}{\tau_1} - 1 \quad (38)$$

系统吞吐量 S 同式 (18)。

6.1.2 考虑因信道质量丢包的情况

本题假设因信道质量可能导致丢包，丢包率为 P_e 。此时在 Markov chain 中传向下一层的概率 p 不再是条件碰撞概率，而是因条件碰撞或丢包导致的失败概率。因此 p 与 τ 的关系和节点传输成功的概率 P_s 的表达式需要调整，其他关系式不变。调整后的式子变为：

$$p = \tau_2 + P_e(1 - \tau_2) \quad (39)$$

$$P_s = (1 - P_e) \times \frac{C_2^1 \tau_1 (1 - \tau_2)}{P_{tr}} = \frac{2(1 - P_e)\tau_1(1 - \tau_2)}{P_{tr}} \quad (40)$$

其中，式 (39) 右侧的含义是因碰撞而失败的概率与不碰撞但发生丢包的概率之和。

联立式 (12)、(13)、(39)，可求出 p, τ_1, τ_2 ，剩余步骤与 6.1.1 相同。

6.2 问题三数值求解

在本节我们只计算 $rate = 455.8, r = 32, CW_{min} = 16, CW_{max} = 1024$ 的情况，此时 T_s, T_c 等参数与问题一相同。其他情况的计算结果请见 6.4 节。

当不考虑因信道质量而丢包时，即丢包率 $P_e = 0$ 时将参数代入 (34) 中，联立式 (12)、(13)、(39)，解得 $p = 0.395, \tau_1 = 0.05, \tau_2 = 0.39$

代入式 (16)、(17)、(18) 中，得

$$S = 24.14 \quad (41)$$

当考虑因信道质量而丢包，且丢包率 $P_e = 0.1$ 时将参数代入式 (39) 中，联立式 (12)、(13)、(40)，解得 $p = 0.42, \tau_1 = 0.04, \tau_2 = 0.36$

代入式 (16)、(17)、(18) 中，得

$$S = 21.56 \quad (42)$$

中间代入、联立的过程以及其它变量的值详见附件 Q3.py。

6.3 问题三仿真算法

本题在问题一的算法基础上，针对 2 BSS 不互听且信道质量丢包这一条件，我们设计了新的算法。我们分别考虑两个系统的时间线，判断两个系统的已经确定状态的时间长短，则已经确定状态的时间较短的系统的状态可由另一个系统状态推出，由此确定该系统下一阶段的状态，重复以上过程。我们实现了两个系统的交替更新，直至达到模拟时间，如算法 3 所示。

在步骤 1 中，我们初始化一些变量以便于后续的计算。其中 T 是当前模拟的时间， T_i 是系统 i 当前模拟的时间， TTs 是总成功时间， CW_i 是系统 i 当前竞争窗口大小。 cw_i 是系统 i 的剩余回退时隙个数， $collision_i$ 是系统 i 已经碰撞的次数， Tb_i 是用于判断下一个状态的辅助变量 ($i = 1, 2$)。在步骤 2 中，我们判断接下来该更新哪个系统的状态，并确定该系统下一阶段的状态，根据下一阶段状态用对应逻辑调整参数。在步骤 4 中，我们根据吞吐公式计算出吞吐量并输出。

相关参数的更新逻辑在附件 q1.m 中有清晰展示。

算法 3: 2 BSS 不互听、SIR 较小且存在信道质量丢包的场景——仿真模拟

输入: 物理层速率 $rate$, 最小竞争窗口 CW_{min} , 最大竞争窗口 CW_{max} , 最大重传次数 r , 模拟时间 $t_{simulation}$;

输出: 吞吐量 S ;

1、初始化

$CW_1 = CW_{min}; CW_2 = CW_{min}; T_1 = 0; T_2 = 0; T = 0; TTs = 0; collision_1 = 0; collision_2 = 0; cw_1 = rand([0, CW_1 - 1]); cw_2 = rand([0, CW_2 - 1]); T_1 = T_1 + cw_1 \times t_{slot}; T_2 = T_2 + cw_2 \times t_{slot}; Tb_1 = 0; Tb_2 = 0;$

2、模拟 AP 向各自关联的 STA 发送数据情景

(a) *if* $T_1 \leq T_2$:

if $collision_1 < r$:

if 满足传输成功条件同时不因信道质量丢包:

成功, 重置 $CW_1, collision_1$; 随机生成回退数 cw_1 ; 更新 TTs, T_1 ;

elseif 满足传输成功条件但因信道质量丢包:

失败, 调整 $CW_1, collision_1$; 随机生成回退数 cw_1 ; 更新 T_1 ;

elseif 传输成功条件不满足:

失败, 调整 $CW_1, collision_1$; 随机生成回退数 cw_1 ; 更新 T_1 ;

else, 达到最大重传次数, 重置 $CW_1, collision_1$

更新系统时间 $T \leftarrow \min(T_1, T_2)$;

(b) *else*, 与 (a) 对称, 略

3、若 $T < t_{simulate}$, 回到步骤 2;

4、输出吞吐量 $S = t_{pl} \times TTs \times rate / (10^6 \times Ts \times T)$

6.4 问题三模型的结果分析

通过问题三中建立的数值分析模型, 利用 Python 软件求解, 本文在 SIR 较低且不互听的条件下, 分别求出了会随机丢包和不会随即丢包情形下, 2 BSS 系统吞吐的数值解 (见表2)。通过问题三中建立的仿真模型, 利用 Python 软件求

解，本文在 SIR 较低且不互听的条件下，分别求出了会随机丢包和不会随即丢包情形下，2 BSS 系统吞吐的仿真解。表2还显示了数值解与仿真解的绝对误差和相对误差。结果表明，数值解与仿真解的绝对误差和相对误差都较小，这说明了我们的数值模型有较好的精确度。

表 2 问题三结果展示表

	参数 1	参数 2	参数 3	参数 4	参数 5	参数 6	参数 7
CW_{min}	16	32	16	16	32	16	16
CW_{max}	1024	1024	1024	1024	1024	1024	1024
最大重传次数	6	5	32	6	5	32	32
物理层速率	286.8	286.8	286.8	158.4	158.4	158.4	455.8
m	6	5	6	6	5	6	6
随机丢包							
仿真	19.71	14.31	21.47	16.6	12.51	18.41	23.21
数值	18.98	16.91	19.03	15.51	13.69	15.63	21.56
绝对误差	0.73	2.6	2.44	1.09	1.18	2.78	1.65
相对误差	3.7%	15.4%	11.4%	6.6%	8.6%	15.1%	7.1%
不随机丢包							
仿真	23.1	16.79	25.27	19.33	14.7	21.52	27.41
数值	21.09	19.27	21.19	17.12	15.4	17.28	24.14
绝对误差	2.01	2.48	4.08	2.21	0.7	4.24	3.27
相对误差	8.7%	12.9%	16.1%	11.4%	4.5%	19.7%	11.9%

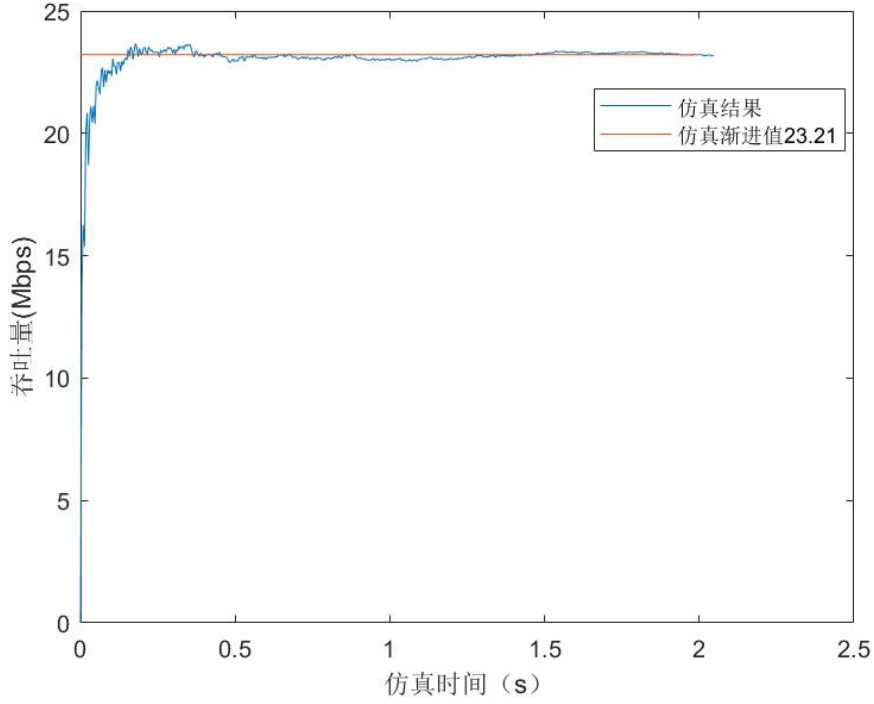


图 7 问题三的仿真图

本文分析问题三中建立的数值分析模型的性质，得到问题三中马氏链是有稳态解的。本文分析问题三的仿真图（图7），得到随着仿真时间的推移，系统的吞吐渐进趋于一个稳定值。这验证了问题三模型中马氏链具有稳定解的性质。

与问题一、二相比较，在处于参数 7 且 AP、STA 数量不变的 2 BSS 系统情况下，在不互听时且 SIR 较小时，系统传输效率会大大降低。这很好的反映了现实生活中，隐藏干扰信号对于信号的传输影响相较于明显干扰信号破坏力更强的情景。

问题三在加入随机丢包影响因素后，传输效率有所降低。这很好的反映了现实中因信道质量差导致传输效率低的情景。

7. 问题四：3 BSS 混合互听的场景

本题考虑了 3 BSS 场景，AP1 与 AP3 不互听，AP2 与两者都互听。假设 AP1 和 AP3 发包时间交叠时，SIR 较大，两者发送均成功。本题不考虑因信道质量丢包的情况。本题将基于问题一的 Markov chain 模型进行修改。

7.1 问题四模型的建立

根据本题的 3 BSS 建模，将三个节点分别用二维 Markov chain 表示^[5]，满足式 (12) 和式 (13)。则节点 j 的稳态解 $b_{0,0}^j$ 表示为：

$$b_{0,0}^j = \begin{cases} \frac{2(1-p_j)(1-2p_j)}{(1-2p_j)(1-p_j^{r+1})+W_0(1-p_j)(1-(2p_j)^{r+1})} & r \leq m \\ \frac{2(1-p_j)(1-2p_j)}{W_0(1-(2p_j)^{m+1})(1-p_j)+(1-2p_j)(1-p_j^{r+1})+W_0 2^m p_j^{m+1}(1-p_j^{r-m})(1-2p_j)} & m < r \end{cases} \quad (43)$$

节点 j 在一个时隙发送数据帧的概率可表示为：

$$\tau_j = \sum_0^r b_{i,0}^j = b_{0,0}^j \times \frac{1-p_j^{r+1}}{1-p_j} \quad (44)$$

由于 AP1 与 AP3 不互听，AP2 与两者都互听，各节点的条件碰撞概率 p 分别表示为：

$$\begin{cases} p_1 = \tau_2 \\ p_2 = 1 - (1 - \tau_1)(1 - \tau_3) \\ p_3 = \tau_2 \end{cases} \quad (45)$$

式 (44)、(45) 是关于 p_i 和 τ_i 的非线性方程组，联立可求解。 ($i = 1.2.3$)

在考虑 3 个节点的情况下，时隙时间中至少有一个传输的概率 P_{tr} , 系统 i 能够探测到传输的概率 P_{tri} 为： ($i = 1.2.3$)

$$\begin{aligned} P_{tr} &= 1 - (1 - \tau_1)(1 - \tau_2)(1 - \tau_3) \\ P_{tr1} &= 1 - (1 - \tau_1)(1 - \tau_2) \\ P_{tr2} &= P_{tr} \\ P_{tr3} &= 1 - (1 - \tau_3)(1 - \tau_2) \end{aligned} \quad (46)$$

各节点传输成功的概率 P_s 为：

$$\begin{cases} P_{s1} = \tau_1(1 - \tau_2) \\ P_{s2} = (1 - \tau_1)(1 - \tau_3) \\ P_{s3} = \tau_3(1 - \tau_2) \end{cases} \quad (47)$$

对于 3 BSS 混合互听的情况，各节点的 $E[\Psi]$ 可表示为：

$$\begin{cases} E_1[\Psi] = \frac{1}{P_{tr1}} - 1 \\ E_2[\Psi] = \frac{1}{P_{tr2}} - 1 \\ E_3[\Psi] = \frac{1}{P_{tr3}} - 1 \end{cases} \quad (48)$$

根据式 (18)，计算各节点的吞吐来评估系统性能。

7.2 问题四数值求解

在本节我们只计算 $rate = 455.8, r = 32, CW_{min} = 16, CW_{max} = 1024$ 的情况，此时 T_S, T_c 等参数与问题一相同。其他情况的计算结果请见 7.4 节。

将参数代入式 (44)、(45) 并联立求解，解得

$$\begin{aligned} p_1 &= p_3 = 0.089 \\ p_2 &= 0.202 \\ \tau_1 &= \tau_3 = 0.107 \\ \tau_2 &= 0.089 \end{aligned} \tag{49}$$

代入式 (16)、(17)、(18) 中，得

$$\begin{aligned} S_1 &= S_3 = 46.98 \\ S_2 &= 18.62 \end{aligned} \tag{50}$$

从而

$$S = S_1 + S_2 + S_3 = 112.59 \tag{51}$$

中间代入、联立的过程以及其它变量的值详见附件 Q1.py, 此处不全部展示。

7.3 问题四仿真算法

7.3.1 节点状态

本题考虑的是 3 BSS 混合互听的场景，那么节点可能处于三种状态：传输中、回退中、停在原处。我们令这三种状态对应的是 1, 0, -1，则节点的状态编码为： $C = [C_1, C_2, C_3]$ ，其中， $C_1, C_2, C_3 \in \{1, 0, -1\}$ 。

在所有情况中，但只有 8 种情况会出现。其中，有 4 种是存在节点数据传输成功的情况，3 种是节点发生碰撞而导致传输失败的情况，其余 1 种是三个节点同时回退的情况。具体编码如下表所示。

表 3 节点编码类别

类型	编码
节点同时回退	(0,0,0)
节点传输成功	(1,-1,1), (-1,1,-1), (1,-1,0), (0,-1,1)
节点传输失败	(1,1,1), (1,1,-1), (-1,1,1)

7.3.2 状态转移函数

在本节，我们定义了一个状态转移函数 *updatestate*，通过输入当前系统的三个节点的剩余回退数以及剩余传输时长，输出下一个阶段的节点状态。

函数 *updatestate*

输入： 节点 i 的剩余回退数 BO_i , 剩余传输时长 $rest_i, i = 1, 2, 3$;

输出： 下一阶段的节点状态;

函数的运行过程如图8

函数的具体实现参见附件 *updatestate.m*。

举例说明， $BO_2 = 0 \rightarrow r_2 > 0 \rightarrow BO_1 > 0 \rightarrow BO_3 = 0 \rightarrow r_3 > 0$ ，输出的结果是 $(-1, 1, 1)$ 。这个过程意思是，由于 AP2 与两者都互听，因此优先判断 AP2 的剩余回退数 BO_2 和剩余传输时长 $rest_2$ ，此时 AP2 正在传输数据；AP1 的剩余回退数 BO_1 非负，说明该节点仍需回退，但由于 AP1 与 AP2 互听，AP2 必须停下等待；AP3 的剩余回退数 BO_3 为 0，且剩余传输时长 $rest_3$ 大于 0，说明该节点也在传输数据。综上，AP1 的状态为 -1，AP2, AP3 的状态为 1。

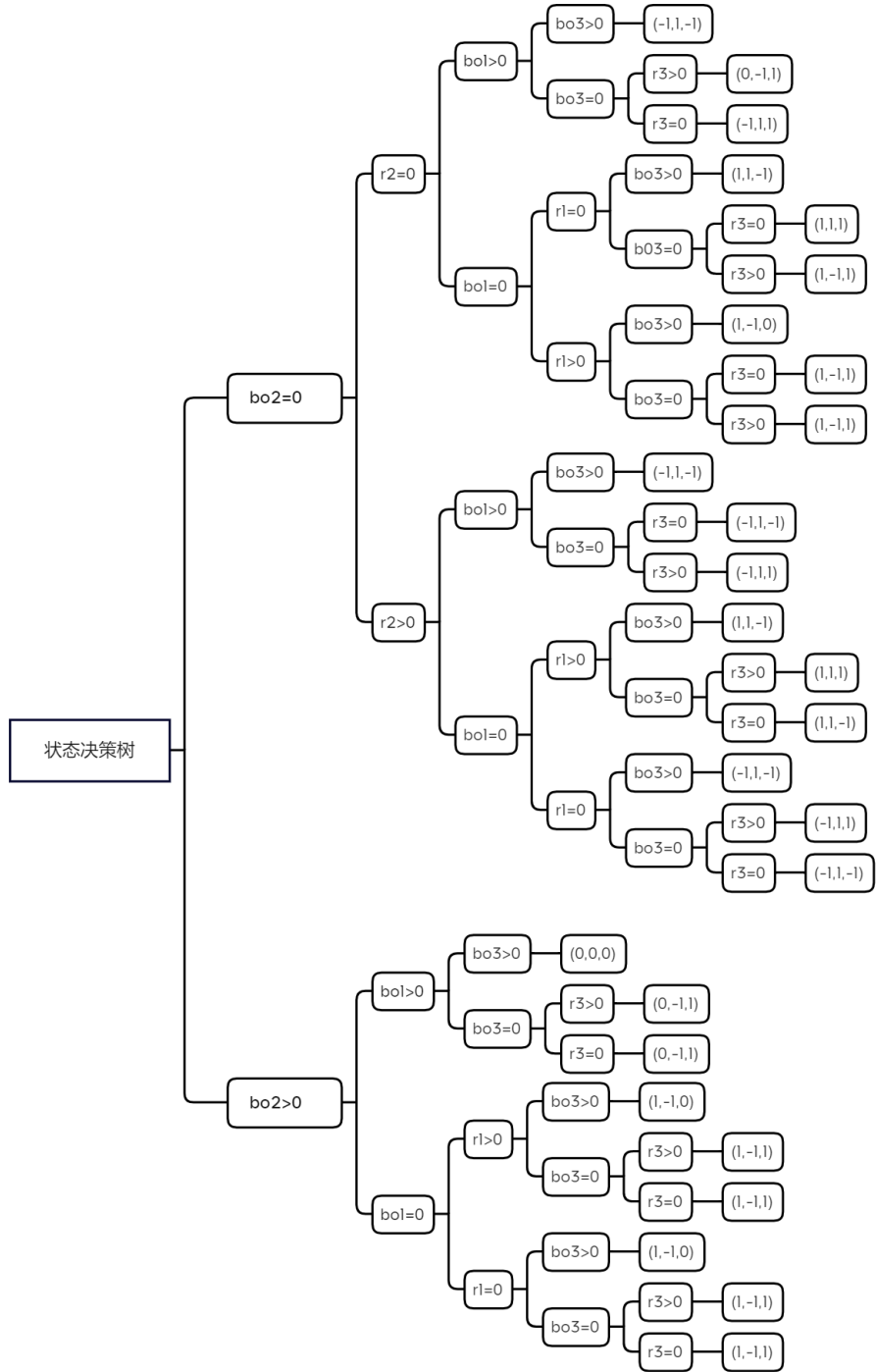


图 8 状态转移决策树

7.3.3 仿真算法设计

本题针对 3 BSS 混合互听这一条件，基于函数 *updatestate* 设计了算法。算法的基本思想是判断当前系统处于什么状态，根据不同的状态进行不同的操作，

然后根据参数判断系统的下一个状态，如此循环，算法四描述了这一过程。

在步骤 1 中，我们初始化一些变量以便于后续的计算。其中 T 是当前模拟的时间， TTs 是总成功时间， CW_i 是系统 i 当前竞争窗口大小。 BO_i 是系统 i 的剩余回退时隙个数， $rest_i$ 是系统 i 剩余传输时间 ($i = 1, 2$)。在步骤 2 中，我们判断当前节点的状态，对于每种状态匹配对应的逻辑去更新变量，并调用 *updatestate* 函数确定节点下一阶段的状态。在步骤 4 中，我们根据吞吐公式计算出吞吐量并输出。

每种节点状态对应的更改变量的逻辑清晰的体现在 q4.m 中，这里不展开列举讲解，事实上，q4.m 实现了分别计算出每个节点的吞吐。

算法 4: 3 BSS 混合互听的仿真模拟算法

输入： 物理层速率 $rate$, 最小竞争窗口 CW_{min} , 最大竞争窗口 CW_{max} , 最大重传次数 r , 模拟时间 $t_{simulation}$;

输出： 吞吐量 S ;

1、初始化

$CW_1 = CW_{min}; CW_2 = CW_{min}; T = 0; TTs = 0; cw_i = rand([0, CW_i - 1]), BO_i = CW_i, rest_i = 0, i = 1, 2, 3$

2、根据当前系统状态，更新系统下一个状态，并更新参数

if $state = [0, 0, 0]$: BO_{min} 的节点优先传输数据，更新 $T, BO_i, state$;

$state = updatestate(BO_1, BO_2, BO_3, rest_1, rest_2, rest_3)$

elseif $state = [1, -1, 0]$: 节点传输数据成功，更新 $T, BO_i, rest, state$;

$state = updatestate(BO_1, BO_2, BO_3, rest_1, rest_2, rest_3)$

elseif $state = [-1, 1, 1]$: 节点传输数据失败，更新 $T, BO_i, rest, state$;

$state = updatestate(BO_1, BO_2, BO_3, rest_1, rest_2, rest_3)$

..... 这里不再列举，请详见附录代码;

3、若 $T < t_{simulate}$ ，回到步骤 2;

4、输出吞吐量 $S = t_{pl} \times TTs \times rate / (10^6 \times Ts \times T)$

7.4 问题四模型的结果分析

通过问题四中建立的数值分析模型,利用 Python 软件求解,本文得出在 AP1 与 AP2 之间,AP2 与 AP3 之间均互听,AP1 与 AP3 之间不互听且 AP1 和 AP3 SIR 较大的情境中,3BSS 系统吞吐的数值解(见表4)。通过问题四中建立的仿真模型,利用 MATLAB 软件求解,本文得出在 AP1 与 AP2 之间,AP2 与 AP3 之间均互听,AP1 与 AP3 之间不互听且 AP1 和 AP3 SIR 较大的情境中,3 BSS 系统吞吐的仿真解(见表4)。表4显示了数值解与仿真解的绝对误差和相对误差。单个系统的吞吐和总吞吐的绝对误差和相对误差都较小,这验证了使用本文的数值模型计算 AP1 与 AP2 之间,AP2 与 AP3 之间均互听,AP1 与 AP3 之间不互听且 AP1 和 AP3 SIR 较大的情境中,3 BSS 系统吞吐精度是比较高的。并且观察到 AP2 的吞吐明显小于 AP1 与 AP3,这符合了“AP2 的发送机会被挤占”的预期。

表 4 问题四的结果展示表

	参数 1	参数 2	参数 3	参数 4	参数 5	参数 6	参数 7
CW_{min}	16	32	16	16	32	16	16
CW_{max}	1024	1024	1024	1024	1024	1024	1024
最大重传次数	6	5	32	6	5	32	32
物理层速率	286.8	286.8	286.8	158.4	158.4	158.4	455.8
m	6	5	6	6	5	6	6
不随机丢包							
仿真	106.38	83.26	106.37	91.26	72.95	91.26	115.26
S_1	45.97	33.48	45.96	40.19	29.96	40.19	49.27
S_2	14.44	16.28	14.45	10.88	13.03	10.88	16.72
S_3	45.97	33.48	45.97	40.19	29.97	40.19	49.27
数值	103.45	87.69	103.46	87.87	76.12	87.88	112.59
S_1	43.21	35.2	43.22	36.77	30.65	36.78	46.98
S_2	17.03	17.29	17.02	14.33	14.82	14.32	18.62
S_3	43.21	35.2	43.22	36.77	30.65	36.78	46.98
绝对误差	2.93	4.43	2.91	3.39	3.17	3.38	2.67
相对误差	2.8%	5.1%	2.7%	3.7%	4.2%	3.7%	2.3%

本文分析在问题四中建立的数值分析模型的性质，得到问题四中马氏链是有稳态解的。本文分析问题四的仿真图（图9），得到随着仿真时间的推移，系统的吞吐渐进趋于一个稳定值。这验证了问题四模型中马氏链具有稳定解性质。

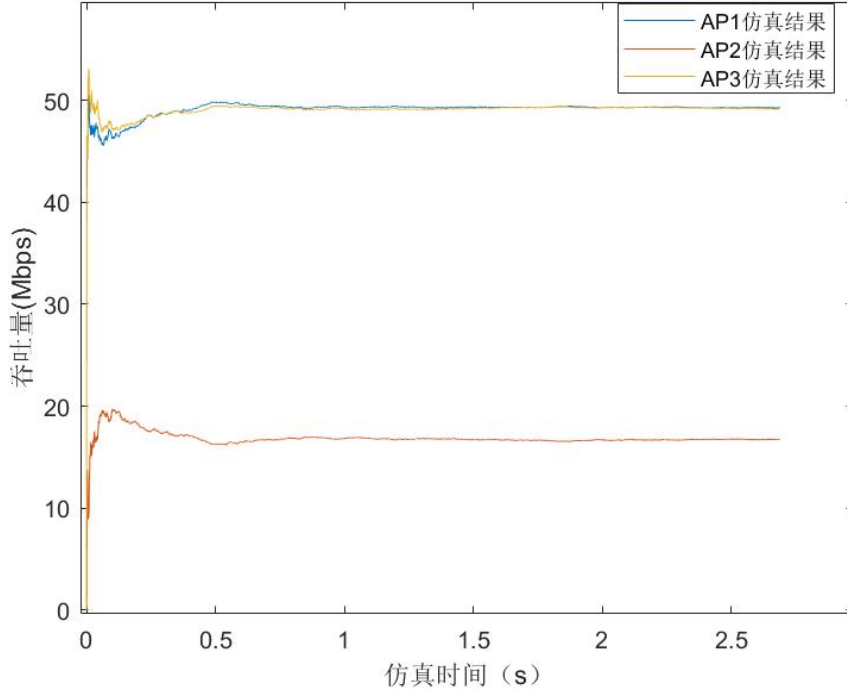


图9 问题四的仿真图

观察在问题四情形下 3 BSS 系统吞吐，本文得出在一个系统被多个干扰信号干扰的时候，传输效率会大大降低的特点。这很好的反映了现实中处于信号传输密集区域信号传输效率低的情景。

本文分析图 (9) 及表 (4)，得到对于参数及干扰环境均相同的 BSS 系统在稳态平均意义下，系统吞吐是相等的。如问题四中 AP1,AP3 所在 BSS 系统。

8. 模型与仿真的讨论与改进

8.1 模型与仿真的优点

1、本题设计的模拟仿真算法非常接近数值分析结果，也符合常识，说明模型的精确度较高；

2、本文在问题一中建立了基于 Markov chain 的信道接入机制模型，分析各节点的状态转移以及数据传输情况。并根据不同问题的条件，不断调整基于 Markov chain 的信道接入机制模型，避免了“一问题一方法”；

3、在问题三中，我们考虑到了节点不互听的情况下， $E[\Psi]$ 不能再原来的 P_{tr} 去计算，而是需要找到节点能够探测到至少有一个节点在传输的概率 P'_{tr} 从而去计算 $E[\Psi]$ 。这很大程度上提升了模型的精确度，也是本文的一个创新点。

4、在问题四中，本文对节点可能处于的三种状态进行编码，确定了 5 种节

点数据传输成功的状态编码和 3 种节点数据传输失败的状态编码；并且我们定义了一种状态转移函数，使得我们只需要知道节点的剩余目标数以及剩余传输时长，就可以判断三个节点的下一个状态，在保证精确度的前提下大大简化了编程复杂度。

8.2 模型与仿真的缺点及改进

我们的模型是一个离散 Markov 过程，即以一个时隙为步长，然而现实的情况是连续情况，导致数值解还不够精准。未来我们可以建立连续 Markov 模型去提升模型精确度。

参考文献

- [1] Bianchi G, IEEE 802.11 - Saturation Throughput Analysis, IEEE COMMUNICATIONS LETTERS, 1998.
- [2] 刘次华, 随机过程. 第 4 版, 随机过程. 第 4 版, 2008.
- [3] Chatzimisios P, Vitsas V, 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.
- [4] Hung F Y, Marsic I, Performance analysis of the IEEE 802.11 DCF in the presence of the hidden stations, Computer Networks, 54(15):2674-2687, 2010.
- [5] Chen D R, Zhang Y J, Is Dynamic Backoff Effective for Multi-Rate WLANs?, IEEE Communications Letters, 11(8):647-649, 2007.

附录 A MATLAB 程序 q1.m

```
% 模拟参数
payload_length = 1500 * 8; % 载荷长度, 单位比特
phy_duration = 13.6e-6; % PHY头时长, 单位秒
mac_header = 30 * 8; % MAC头长度, 单位比特
data_rate = 455.8e6; % 物理层速率, 单位比特每秒
ack_duration = 32e-6; % ACK时长, 单位秒
sifs_duration = 16e-6; % SIFS时长, 单位秒
difs_duration = 43e-6; % DIFS时长, 单位秒
slot_duration = 9e-6; % SLOT时长, 单位秒
ack_timeout = 65e-6; % ACK超时时长, 单位秒
cw_min = 16; % 最小竞争窗口
cw_max = 1024; % 最大竞争窗口
max_retries = 32; % 最大重传次数
cw = cw_min; % 初始化CW
simulation_duration = 1000; % 模拟时间

% 进一步计算参数
mac_duration = mac_header / data_rate; % MAC时长
payload_duration = payload_length / data_rate; % E(P)时长
Ts = mac_duration + phy_duration + payload_duration +
    sifs_duration + ack_duration + difs_duration; % Ts时长
Tc = mac_duration + phy_duration + payload_duration +
    ack_timeout + difs_duration; % Tc时长

% 开始模拟
T = 0; % 初始化时间
TTe = 0;
TTs = 0;
TTc = 0;
cw1 = randi([0, cw - 1]); % 生成随机回退数cw1.cw2
cw2 = randi([0, cw - 1]);
collision_count = 0; % 记录冲突次数

while T < simulation_duration % 设置模拟的时间上限
    if cw1 == cw2
        % 发生碰撞
        if collision_count > max_retries % 达到最大重试次数上限
```

```

collision_count = 0;
TTe = TTe + cw1 * slot_duration;
TTc = TTc + Tc;
T = T + Tc + cw1 * slot_duration; % 更新时间
collision_count = collision_count + 1;
cw = min( 2^collision_count * cw_min , cw_max); %
    调整竞争窗口大小, 不超过cw_max
cw1 = randi([0, cw - 1]); % 重新生成随机回退数cw1
cw2 = randi([0, cw - 1]); % 重新生成随机回退数cw2
else
TTe = TTe + cw1 * slot_duration;
TTc = TTc + Tc;
T = T + Tc + cw1 * slot_duration; % 更新时间
collision_count = collision_count + 1;
cw = min(2^collision_count * cw_min , cw_max); %
    调整竞争窗口大小, 不超过cw_max
if collision_count > max_retries % 达到最大重试次数上限
    cw = cw_min;
    cw1 = randi([0, cw - 1]); % 重新生成随机回退数cw1
    cw2 = randi([0, cw - 1]); % 重新生成随机回退数cw2
else
    cw1 = randi([0, cw - 1]); % 重新生成随机回退数cw1
    cw2 = randi([0, cw - 1]); % 重新生成随机回退数cw2
end
end
else
    % 比较cw1和cw2的大小, 确定哪一个进入信道
    if cw1 < cw2
        collision_count = 0;
        cw = cw_min;
        % cw1进入信道, cw2保持不变
        TTe = TTe + cw1 * slot_duration;
        TTs = TTs + Ts;
        T = T + Ts + cw1 * slot_duration; % 更新时间
        cw2 = cw2 - cw1;
        cw1 = randi([0, cw - 1]); % 重新生成随机回退数cw1
    else
        cw = cw_min;
    end
end

```



```

        collision_count = 0;
        % cw2进入信道, cw1保持不变
        TTe = TTe + cw2 * slot_duration;
        TTs = TTs + Ts;
        T = T + Ts + cw2 * slot_duration; % 更新时间
        cw1 = cw1 - cw2;
        cw2 = randi([0, cw - 1]); % 重新生成随机回退数cw2

    end
end
end

TP = payload_duration * TTs * data_rate / ( 10^6 * Ts * T );

% 输出结果
fprintf('T (总时长): %.6f 秒\n', T);
fprintf('TTe (空闲总时长): %.6f 秒\n', TTe);
fprintf('TTc (冲突总时长): %.6f 秒\n', TTc);
fprintf('TTs (成功传输总时长): %.6f 秒\n', TTs);
fprintf('吞吐量: %.6f Mbps\n', TP);

```

其余代码均打包在附件中，这里不一一列举。