



山东大学 · 泰山学堂

计算机网络实验

多队列网络调度与 802.11 协议模拟

王愚德 (201300301257)

张昊文 (201300130123)

冯宗浩 (201300130013)

指导老师 刘磊

2016 年 1 月 1 日

目录

1	多队列网络调度模拟	3
1.1	实验要求	3
1.2	实验环境	3
1.3	实验步骤及结果	3
1.3.1	公平判定标准	3
1.3.2	仿真算法描述	3
1.4	实验结果	4
1.4.1	等权重, 等包长	4
1.4.2	不等权重, 不等包长	4
1.5	结论分析与体会	6
2	简化版 802.11 协议模拟	7
2.1	实验要求	7
2.2	实验环境	7
2.3	实验步骤及结果	7
2.3.1	初始化	7
2.3.2	仿真算法描述	7
2.3.3	仿真参数	8
2.3.4	结果	8
2.4	结论分析与体会	8

1. 多队列网络调度模拟

1.1. 实验要求

创建 3 个包长与到达时间间隔均满足泊松分布的队列，每个队列有对应的权重。有 1 个服务端，其服务速率恒定，服务时间与包长成正比。选择一个公平判定标准，并实现一个公平队列调度算法，使得无论如何选取每个队列的平均到达时间与平均包长，队列调度都是公平的。

1.2. 实验环境

- Windows 10
- gcc 4.9.3
- Matlab R2015b

1.3. 实验步骤及结果

1.3.1. 公平判定标准

本次实验中，我们采用的公平判定标准是：队列占用服务资源的比例近似于队列权重的比例。此处，服务资源占用的量化标准为每个队列占用服务端的总时间。

1.3.2. 仿真算法描述

我们采用加权公平队列 (Weighted Fair Queueing) [1] 调度算法。以下是该算法完整的实现以及模拟流程。

1. 初始化阶段，设置各变量的初值。
 - (a) 设当前时间为 0，模拟时间在本次实验中取 10^6 秒。
 - (b) 设下次到达时间为 0，下次服务时间为 ∞ 。
2. 若当前时间大于模拟时间，结束模拟。否则，根据下次到达时间与下次服务时间的先后，决定执行到达或服务操作。
 - (a) 若下次到达时间小于下次服务时间，跳转至 2.
 - (b) 若下次到达时间大于下次服务时间，跳转至 3.
3. 对每次到达，置当前时间为下次到达时间。
 - (a) 到达包的包长（即服务该包所需的时间）按泊松分布生成。
 - (b) 计算包的虚拟开始时间与虚拟完成时间。
 - i. 虚拟开始时间 = $\max\{\text{当前时间}, \text{该队列上次虚拟完成时间}\}$ 。
 - ii. 虚拟完成时间 = 虚拟开始时间 + $\frac{\text{包长}}{\text{该队列的权重}}$ 。
 - (c) 将包压入队列中。
 - (d) 若压入该包之前，所有队列中都为空，则将下次处理时间设为当前时间加上包长。

- (e) 按泊松分布生成该队列的下次到达时间。选择所有队列的下次到达时间中最小的一个，作为下次到达时间。
- (f) 跳转至 2.
- 4. 对每次服务，置当前时间为下次服务时间。
 - (a) 将完成处理的包从其所属队列中弹出。若弹出后所有队列为空，则将下次处理时间设为 ∞ 。
 - (b) 从所有队列顶部的包中选择虚拟完成时间最小的包。将下次处理时间加上该包的包长。
 - (c) 跳转至 2.

1.4. 实验结果

1.4.1. 等权重，等包长

仿真参数：

1. 平均包长（以平均服务时间记）：1.0, 1.0, 1.0
2. 每个队列包的到达时间：1.25
3. 队列权重之比：1 : 1 : 1

模拟结果：

队列序号	1	2	3
完成包数	334149	333658	333312
服务时间	333333.72	333331.52	333334.57

可以看出，各队列的服务时间比例约为 1 : 1 : 1，近似于队列权重之比。

等待时间分布见图. 1。

队列长度分布见图. 2。

由于队列的各个参数相等，因此得出的图线基本重合。

1.4.2. 不等权重，不等包长

仿真参数：

1. 平均包长（以平均服务时间记）：0.7, 1.0, 1.3
2. 每个队列包的到达时间：1.25
3. 队列权重之比：2 : 3 : 5

模拟结果：

队列序号	1	2	3
完成包数	286127	300454	385046
服务时间	199997.94	300000.56	500000.52

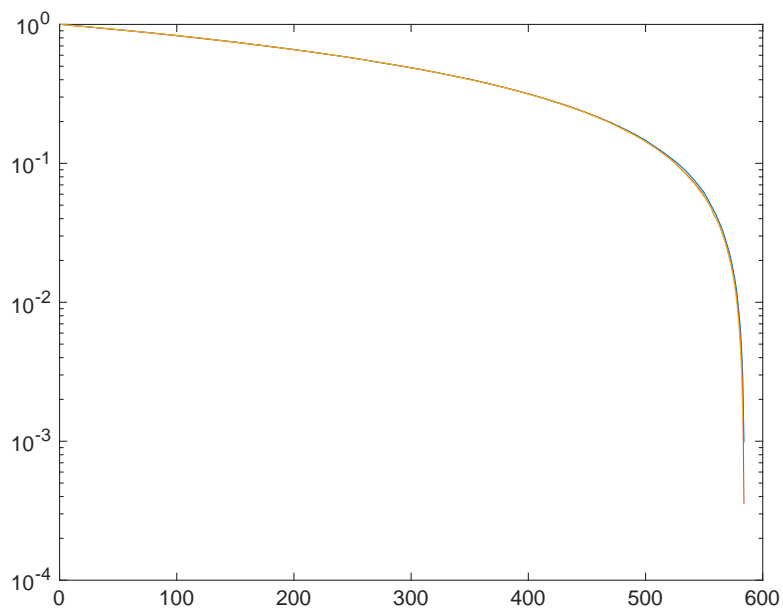


图. 1. 等待时间分布

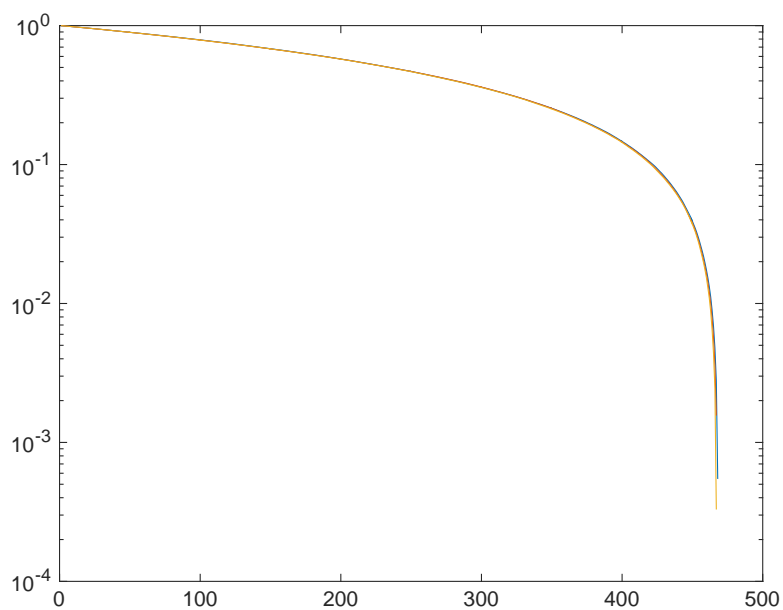


图. 2. 队列长度分布

可以看出，各队列的服务时间比例约为 $2:3:5$ ，近似于队列权重之比。

等待时间分布见图. 3。

队列长度分布见图. 4。

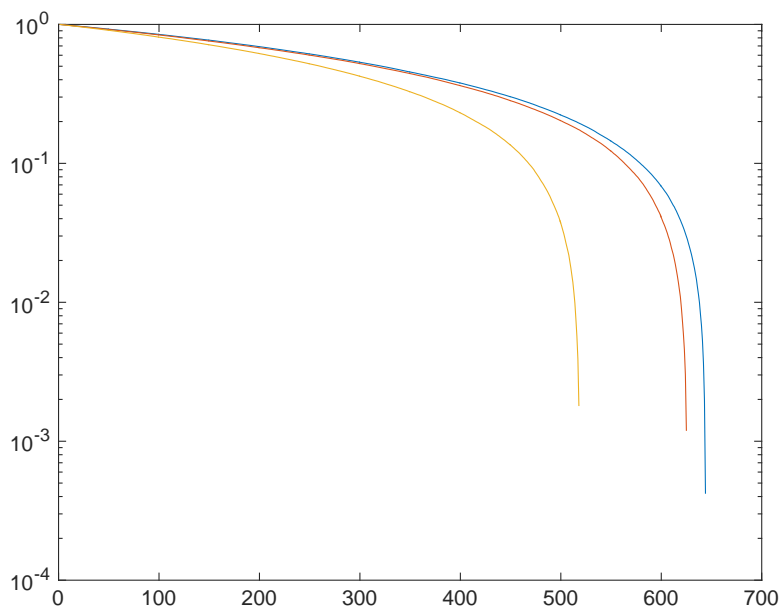


图. 3. 等待时间分布

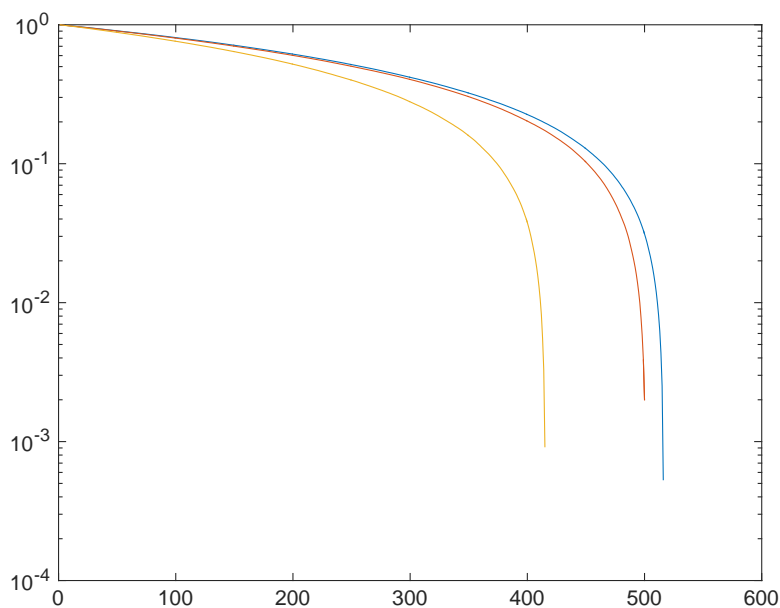


图. 4. 队列长度分布

1.5. 结论分析与体会

本次实验中我们实现了加权公平队列算法，模拟结果也与我们的期望基本吻合。在实验的模拟过程中，仿真参数的选取十分重要。假设队列的总到达速率慢于服务速率，则所有队列的包都能被服务完，但考虑到我们的公平标准是检查服务资源分配的比例，这种情况下不能体现出调度算法的公平性。因此我们将到达速率固定，仅修改队列的权重与平均包长，形成一定的队列积压，来观察服务资源分配情况。

2. 简化版 802.11 协议模拟

2.1. 实验要求

创建 10 个到达速率分布均满足泊松分布的队列，且它们的平均到达速率相同。有 1 个服务端，其服务速率满足泊松分布。实现一个简化版的 802.11 协议（不考虑 SIFS 和 AIFS₁ 两个时间间隔）。

2.2. 实验环境

- Windows 7
- Visual Studio 2012
- Matlab R2015a

2.3. 实验步骤及结果

2.3.1. 初始化

1. 设置初始竞争窗口和竞争窗口的最大值。
2. 设置每个队列的平均到达速率，平均包长，具体下一个包的到达时间和包长由泊松过程获得。
3. 设置时间片长度。

2.3.2. 仿真算法描述

当前时间未超过仿真总时间时：

1. 判断队列是否有包在本时间片内到达，若有，此包进入队列，并计算下一个包的到达时间，循环步骤 1.，直到下一个包的到达时间不在时间片内，跳转至 2.
2. 在时间片头侦听信道，若被占用，则将所有本时间片内请求服务的队列请求时间设为下一个时间片头，跳转至 5.；若未被占用，跳转至 3.
3. 统计在时间片头请求服务的队列数。若个数大于一，进行二元指数回退，对第 i 次冲突，每个发生冲突的队列随机等待 $0 - 2^{i+3}$ (2^{i+3} 最大为 2^{10}) 个时间片并跳转至 5. 若个数等于一，对其服务，出队列并判断队列是否为空，为空则将下一个请求服务时间设为下一个包到达时间；不为空则将下一个请求服务时间设为下一个时间片头。
4. 若本时间片中有队列请求服务，将请求服务时间设为下一个时间片头。跳转至 5.
5. 时间片计数加一，跳转至 1.

2.3.3. 仿真参数

1. 初始竞争窗口: $2^4=16$ 个
2. 竞争窗口最大值: $2^{10}=1024$ 个
3. 每个时间片大小: 0.01
4. 平均包长 (以平均服务时间记): 0.15
5. 每个队列包的平均到达时间 (与到达速率成倒数): 3

2.3.4. 结果

1. 信道利用率, 即 $\frac{\text{碰撞时间片个数} + \text{发送包时间片个数}}{\text{总时间片}}$: 52.4073%
2. 碰撞概率, 即 $\frac{\text{碰撞包数}}{\text{碰撞包数} + \text{发送包数}}$:

队列序号	1	2	3	4	5
碰撞概率	25.5907%	25.5688%	25.5628%	25.6101%	25.5855%
队列序号	6	7	8	9	10
碰撞概率	25.5810%	25.5928%	25.5901%	25.5998%	25.6045%

3. 吞吐率, 即 $\frac{\text{成功发包数}}{\text{总发包数}}$: 0.744385, 其中 总发包数 = 成功发包数 + 碰撞包数。

等待时间分布见图. 5。

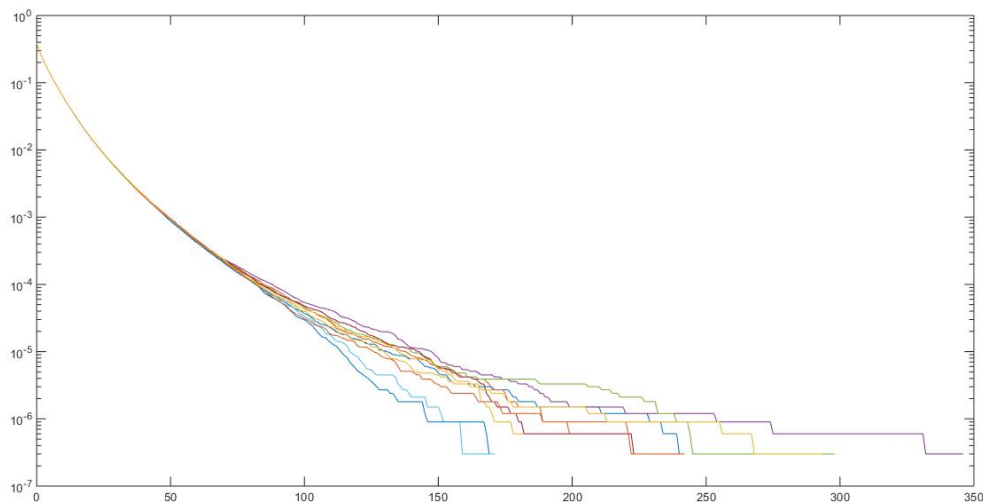


图. 5. 等待时间分布

队列长度分布见图. 6。

2.4. 结论分析与体会

1. 我们对初始竞争窗口大小设置进行了探究: 我们绘制了小于 $2^5 = 32$ 的初始竞争窗口的碰撞概率变化图, 如图. 7所示。可以看出, 随着初始竞争窗口的增大, 碰撞概率有一定的减

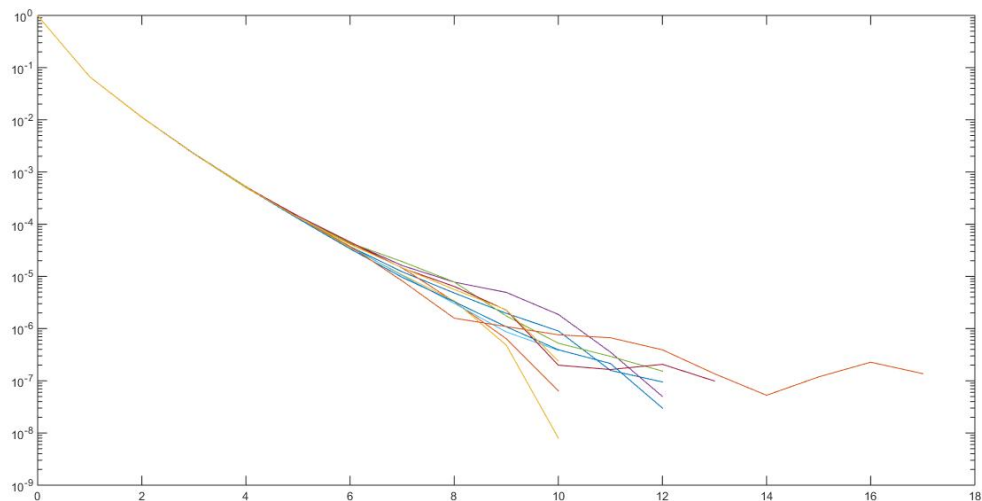


图. 6. 队列长度分布

小；而当初始竞争窗口大小超过 32 时，碰撞概率减小变得不明显。因此，将初始竞争窗口大小置为 32 比较合适，这也与 802.11b 的标准一致。

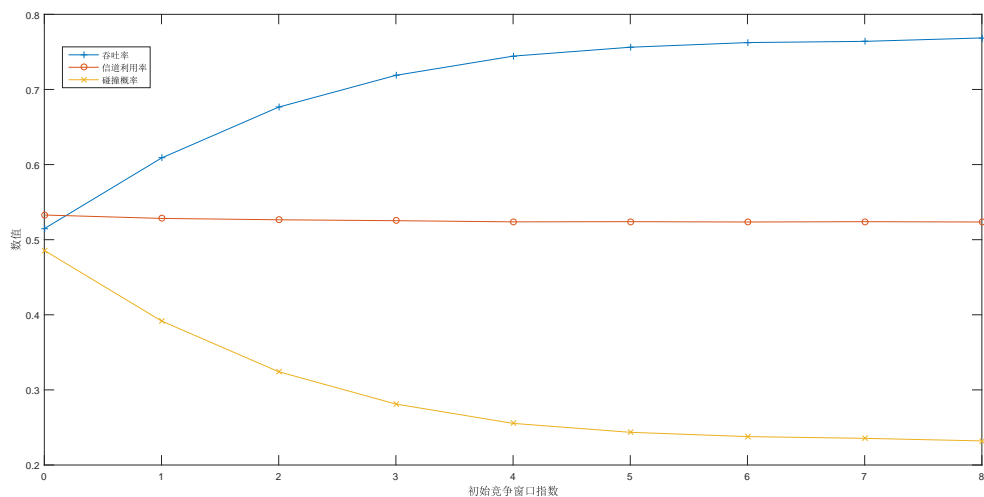


图. 7. 不同初始竞争窗口大小的碰撞概率

- 我们对时间片大小的选取进行了研究：在前面的仿真实验中，我们时间片的大小均设置为 0.01。而现在，我们又进行了 7 次实验，每次时间片的大小增加 0.01，观察信道利用率和碰撞概率的变化，绘制了图. 8。

我们发现，在一定范围内，随着时间片的增大，信道利用率不断升高，而碰撞概率出现了先升高，后降低的现象，吞吐量基本不变。

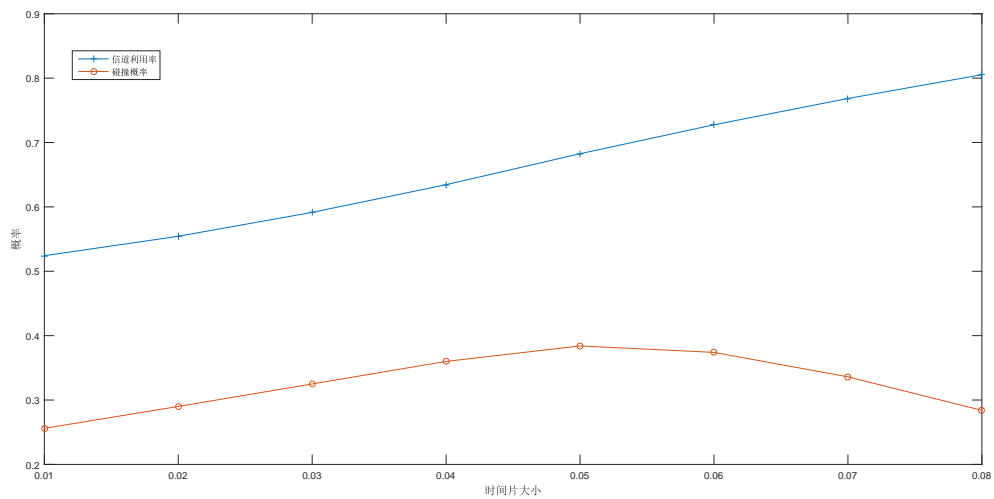


图. 8. 不同大小时间片的信道利用率和碰撞概率

引用

1. Alan Demers, Srinivasan Keshav, and Scott Shenker. Analysis and simulation of a fair queueing algorithm. In *ACM SIGCOMM Computer Communication Review*, volume 19, pages 1–12. ACM, 1989.