

## 实验二 MATLAB 编程实现 ALOHA 算法

尹达恒

(江南大学物联网工程学院, 江苏 无锡)

### 1 实验目的

通过本次实验, 进一步了解 ALOHA 算法, 并将理论知识与实际相结合。

### 2 实验设备

MATLAB 编程软件

### 3 实验原理

ALOHA 法在多路存取方法中是最简单的, 只要有一个数据包提供使用, 这个数据包就被立即发送给射频读写器。ALOHA 法是射频电子标签控制的, 它只适用于只读射频电子标签。通常, 这类射频电子标签只有一些数据传输给射频读写器, 并且是在一个周期循环中将这些数据发送给射频读写器。数据传输时间只是循环周期的一小部分, 所以在传输之间产生相当大的间隙; 同时, 各个射频电子标签的循环周期的差别可以忽略不计, 各个射频电子标签的重复时间之间的差别是微不足道的。所以存在着一定的概率, 两个射频电子标签可以在不同的时间段传输数据, 使数据包不相互碰撞。将时间分为离散的小段, 每一段称为时隙, 每个时隙都足够让一个标签发送完信息;  $N$  个时隙合为一帧 ( $N$  是一个默认值); 发射端随机选择一帧中的一个时隙向接收端发送信息, 一旦发生碰撞, 就在下一帧中随机选择一个时隙重新发送, 如图 1。平均交换的数据包量  $G$  可以用最简单的方法从一个数据包的传输持续时间  $\tau$  计算出来:

$$G = \sum_{i=1}^n \frac{\tau_i}{T} r_i$$

其中:  $n$  是系统中的标签数量,  $r_i$  为观察时间  $T$  内由应答器发送的数据包的数量。传输信道的平均吞吐率  $S$  可由交换的数据包量  $G$  得出:

$$S = Ge^{-2G}$$

可以得出当  $G = 0.5$  时, 最大吞吐率  $S = 1/(2e) = 18.4\%$ 。

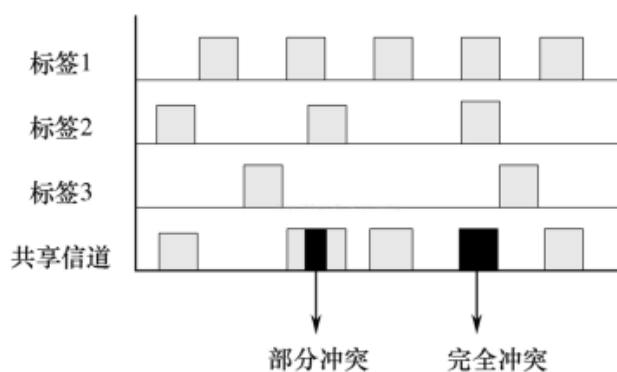


图 1. ALOHA 算法碰撞图

## 4 实验步骤

编写 ALOHA 模拟算法，依据第 3 节所介绍的 ALOHA 算法原理，可以得到 ALOHA 模拟算法流程如算法 1 所示。

---

### 算法 1. ALOHA 模拟算法流程

---

**Require:** 标签数量  $m$

**Require:** 各标签发送数据次数  $n$

**Require:** 各标签发送信息的时间  $B = \{b_{ij}\}$

**Require:** 各标签的观察时间  $T$

**Require:** 数据包宽度  $T_0$

1: 将  $B$  各元素按大小排序以便于查找冲突

2: **for** 按时间顺序遍历  $B$  中每个时间点  $b_{ij}$  **do**

3:     **if**  $b_{ij} \leq T$  **then**

4:         发送数据

5:         **if**  $b_{ij} - \min \{b | b \in B \wedge b < b_{ij}\} \geq T_0$  且  $\max \{b | b \in B \wedge b > b_{ij}\} - b_{ij} \geq T_0$  **then**

6:             发送成功

7:         **else**

8:             发送失败

9:         **end if**

10:     **end if**

11: **end for**

**Ensure:** 统计发送成功和发送失败的次数并绘图

---

按照算法 1 所示的算法流程编写 matlab 程序，运行结果如图 2 和 3。

## 5 实验结果分析

1、由图 2 可以看出随着数据包交换量的增大，ALOHA 网络的数据吞吐量在数据包交换量较小时上升，数据包交换量到达一定值后又开始下降，这是由于在数据包交换量较小时网络可以成功发送所有的数据包，因此数据表越多网络吞

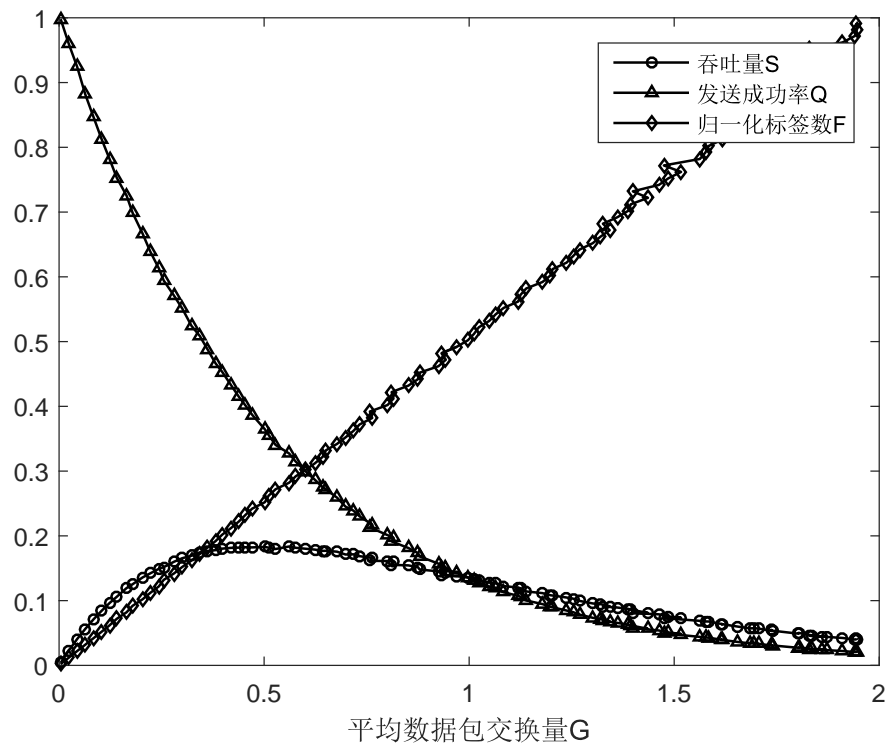


图 2. ALOHA 算法模拟结果

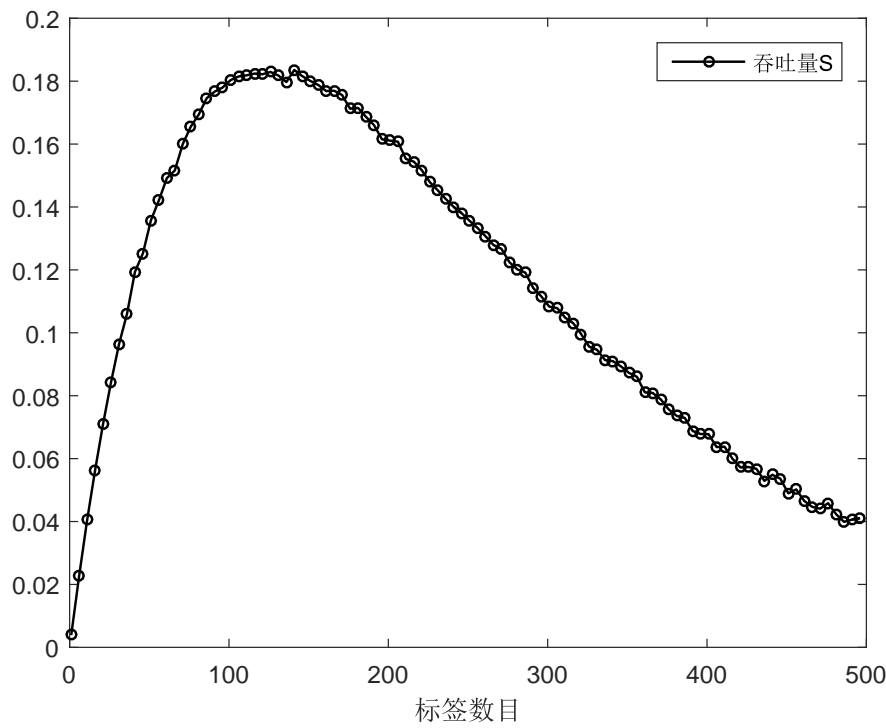


图 3. ALOHA 算法模拟结果

吐量越大，而当数据包交换量达到一定值后，网络出现拥塞，要发送的数据包越多，互相之间的干扰也越大，发送成功率下降，从而网络的吞吐量下降；

2、由图 3 可以看出，随着标签数目的增大，ALOHA 网络的数据吞吐量也呈先上升后下降的趋势，其原因类似，是由于标签量较少时不容易出现拥塞，而标签量较大时使网络出现拥塞所致。

# 附录

## A 实验代码

```

Tag_num=500;
ym=1:5:Tag_num;
G=zeros(size(ym));
S=zeros(size(ym));
5 Q=zeros(size(ym));
F=zeros(size(ym));
k=1;
for m=ym
    n=1000;
10 A=rand(m,n);
    A1=0.5*A;
    B=cumsum(A1,2);
    T=B(1,n);
    C=1:1:(m*n);
15 for i=1:m
        for j=1:n
            C(1,(i-1)*n+j)=B(i,j);
        end
    end
20 D=sort(C);
    E=diff(D);
    T0=0.001;
    N=0;
    M=0;
25 for i=1:(m*n-1)
        if D(1,i)<=T
            M=M+1;
            if i==1&&E(1,1)>=T0
                N=N+1;
30 elseif i==(m*n-1)&&E(1,(m*n-1))>=T0
                N=N+1;
            elseif i~=1&&i~=(m*n-1)&&E(1,i)>=T0&&E(1,i-1)>=T0
                N=N+1;
            end
35 else continue
        end
    end
    G(k)=T0/T*M;
    S(k)=T0/T*N;
40 Q(k)=S(k)/G(k);
    F(k)=m/Tag_num;
    k=k+1;

```

```
end
markersize=4;
45 figure(1)
plot(G,S,'k-o','linewidth',1,'markersize',markersize);
hold on
plot(G,Q,'k-^','linewidth',1,'markersize',markersize);
hold on
50 plot(G,F,'k-d','linewidth',1,'markersize',markersize);
hold on
xlabel('平均数据包交换量G');
legend('吞吐量S','发送成功率Q','归一化标签数F');
figure(2)
55 plot(ym,S,'k-o','linewidth',1,'markersize',markersize)
legend('吞吐量S')
xlabel('标签数目')
```