

基于Python Anaconda的物联网ALOHA算法仿真研究

杨群伟¹, 郭丽清², 王宁^{1,3}

(1. 厦门华夏学院 信息与智能机电学院; 2. 厦门华夏学院 通识教育中心;
3. 信息通信技术与智慧教育福建省高校工程研究中心, 福建 厦门 361021)

摘要: ALOHA算法是RFID技术解决多标签冲突的关键方法, 传统多采用MATLAB对其进行仿真验证。但受中美两国竞争影响, 高校使用受限, 现Python大数据分析工具已成为MATLAB的有效替代。首先介绍ALOHA算法工作原理, 其次使用排列组合和极限思想推导其理论吞吐率的数学表达式, 最后引入Python大数据分析/科学计算平台Anaconda, 实现了经典ALOHA算法的碰撞模拟和吞吐率分析。该方法有助于改进ALOHA算法的理论教学和研究工作, 并解决了MATLAB使用受限的问题。

关键词: 物联网 ALOHA 算法; 数据可视化; 计算机仿真; Python Anaconda

DOI: 10.11907/rjcdk.231650

开放科学(资源服务)标识码(OSID):



中图分类号: TP391.9

文献标识码: A

文章编号: 1672-7800(2024)008-0151-05

Simulation Research of IoT ALOHA Algorithm Based on Python Anaconda

YANG Qunwei¹, GUO Liqing², WANG Ning^{1,3}

(1. School of Information and Intelligent Mechatronics, Xiamen Huaxia University; 2. General Education Center, Xiamen Huaxia University; 3. Information Communication Technology and Smart Education of Fujian Engineering Research Center, Xiamen 361021, China)

Abstract: ALOHA algorithm is a key method for RFID technology to solve multi-tag collision, and traditionally MATLAB is often used for simulation verification. However, due to the competition between China and the United States, it has been restricted in universities, and the Python big data analysis tool has now become an effective alternative to MATLAB. This article begins by introducing the working principles and related conclusions of the ALOHA algorithm. The theoretical throughput rates are then derived using concepts of permutations, combinations, and limits to facilitate clear understanding. Finally, the article incorporates the Python-based data analysis and scientific computing platform, Anaconda, to implement collision simulation and throughput analysis for the ALOHA algorithm. By introducing these new tools and methods, this article aims to improve the theoretical teaching and research of the ALOHA algorithm while addressing the limitations associated with MATLAB usage.

Key Words: ALOHA algorithm for the Internet of Things; data visualization; computer simulation; Python Anaconda

0 引言

射频识别(Radio Frequency Identification, RFID)技术作为物联网感知层的核心技术之一, 是一种通过电磁波自动识别目标的技术。其基本原理是利用天线间的近场耦

合或反向调制耦合来传递信息, 以实现目标物体的认证。与传统的识别技术相比, 射频识别技术具有操作简单、穿透性强和自动化程度高等优点, 在物流、门禁、工业生产、智能仓库以及车联网等领域得到了广泛应用^[1]。在实际应用中, 经常会有大量标签等待识别, 当读写器发出请求命令后, 标签会立刻响应该读写器, 标签碰撞问题不

收稿日期: 2023-06-20

扫描二维码阅读全文:



基金项目: 厦门华夏学院人才引进校内科研项目(HX202203)

作者简介: 杨群伟(1986-), 男, 硕士, 厦门华夏学院信息与智能机电学院讲师, 研究方向为云计算、软件工程、计算机应用技术; 郭丽清(1988-), 女, 硕士, 厦门华夏学院通识教育中心讲师、高级工程师, 研究方向为大数据分析、计算机教育; 王宁(1979-), 男, 博士, 厦门华夏学院信息与智能机电学院教授, 研究方向为大数据分析、计算机教育。本文通讯作者: 王宁。

可避免,ALOHA 算法是解决该问题的主要方法之一。ALOHA 算法分为纯 ALOHA 和时隙 ALOHA,时隙 ALOHA 是对纯 ALOHA 算法的改进。纯 ALOHA 算法与时隙 ALOHA 算法的基本特点对比如表 1 所示。

Table 1 Comparison of ALOHA and slotted ALOHA algorithms
表 1 ALOHA 与时隙 ALOHA 算法对比

算法特点	纯 ALOHA	时隙 ALOHA
发送时刻	任意时间数据都可以发送	只有在时隙开始处才能发送
时间是否同步	不同步	全局同步
是否划分时隙	否,连续时间	是,离散时间
脆弱时间	2τ	τ
吞吐率	最大吞吐量出现在 $G=1/2$ 时,吞吐率为 18.4%	最大吞吐量出现在 $G=1$ 时,吞吐率为 36.8%,时隙 ALOHA 将碰撞次数减少到一半,因此效率提高了一倍

为了证明以上结论,可以采用纯数学推导和计算机模拟实现的方式。计算机模拟具有可视化效果仿真的优势,对研究和教学非常有帮助。

1 相关研究

当前在对物联网 ALOHA 算法的相关研究中,主要采用 MATLAB 软件进行模拟验证。例如,杨灵等^[3]采用 MATLAB 软件对 ALOHA 算法过程及效率进行分析,在帧长和标签数目相同的条件下求出系统的吞吐率最大值;于子甲等^[4]在对 ALOHA 协议原理进行深入研究的基础上,结合仿真实验平台,基于 MATLAB 设计了一组相关的 ALOHA 协议仿真实验用于教学示范;王冬云等^[5]基于改进 ALOHA 算法的 RFID 标签防碰撞研究,提出一种新的改进算法,采用动态预测权值估计标签数目,使标签数目与数据帧长度大致相等,并基于 MATLAB 进行算法的仿真验证;何金栋等^[6]提出一种新型的基于 Q-learning 与神经网络的帧长优化算法,采用 MATLAB 进行验证;魏福禄等^[7]运用 MATLAB 辅助工具,建立基于动态帧时隙 ALOHA 算法的危险品运输 RFID 防碰撞模型。查看近年来高校通信相关专业的硕博论文,如文献^[8-10]也都采用 MATLAB 软件进行实验仿真和验证。MATLAB 是商业软件,价格昂贵,并且随着中美竞争加剧,美国对科技产品的出口限制越来越严,导致越来越多高校无法使用该软件。

Anaconda 是 Python 语言下用于数据分析的一整套工具集合(包含 Numpy、matplotlib、scipy),具有开源、免费,不受任何组织限制的优点^[11]。借助于 Python 大数据分析工具进行数据收集、清洗、可视化分析,不仅仅在计算机领域,在跨学科、跨行业的应用也越来越广泛。陈路等^[12]基于 Python 设计一个学生成绩统计与分析系统,提出使用 Matplotlib、Pandas、Numpy 等库实现对学生成绩的计算与统计分析;林郁仓等^[13]通过 Python+Pandas+Matplotlib 对医院抗菌药物数据进行可视化输出,以解决药学部门的痛点问题;关成斌^[14]提出并实现了基于 Python 及其 numpy、mat-

plotlib 等模块的超限经济订货批量数据分析法;刘学琳等^[15]以测定三棱镜色散的实验数据处理为例,应用 Python+Scipy+Matplotlib+PyQt5 编写了物理实验数据处理系统;吴涛等^[16]基于 Jupyter-lab 平台对大学物理的人机交互式在线教学模式进行研究,分析该教学模式的优势和特点,并以简谐振动和分子热运动为例进行基于该平台的互动式、可视化教学实践探索。总之,基于 Python 科学计算/大数据的可视化场景设计越来越多,有逐渐取代 MATLAB 之势。

与先前基于 MATLAB 的研究不同,本文通过借鉴参考文献^[11-16]所述的方法引入 Python 大数据分析/科学计算平台 Anaconda,实现了对 ALOHA 算法的可视化分析^[3-10]。该方法较为新颖,提供了一种用于该领域仿真研究的新方法和思路,可规避该领域使用 MATLAB 潜在受限的问题。通过公布源代码,本文提供了以此为基础的案例,可直接应用于相关本科专业教学。并且相较于前人的研究,本文通过柱形图进行了 ALOHA 冲突效果演示,在可视化效果方面更加突出^[4]。另外,本文详细整理归纳了 ALOHA 算法的完整推导过程,弥补了相关教材^[17]和文献^[3-10]的不足。

2 ALOHA 算法工作原理

在纯 ALOHA 中,传输时间是连续的,数据的发送也是独立、随机的。每当电台(station)有可用帧时,其都会发送该帧。在发送帧之后,站点等待有限的时间来接收确认。如果在这段时间内没有收到确认,则站点会假定该帧由于冲突而被破坏,并重新发送该帧。如果发生冲突并且帧被破坏,发送方将在重新传输之前随机等待一段时间。如果多个帧试图同时占用通道,则会发生冲突。纯 ALOHA 算法数据传输过程示例如图 1 所示。

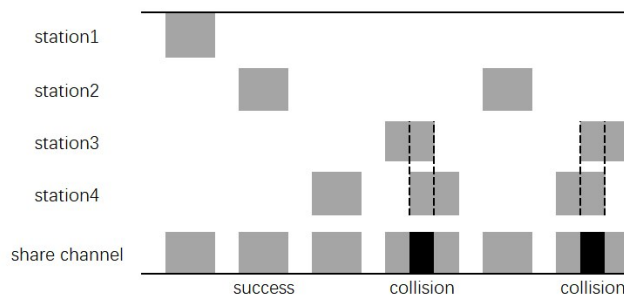


Fig. 1 Example of ALOHA algorithm data transmission process

图 1 ALOHA 算法数据传输过程示例

时隙 ALOHA 是为了提高纯 ALOHA 的性能而开发的,因为纯 ALOHA 中发生碰撞的几率非常高。在这种 ALOHA 中,系统时间被划分为多个时隙,因此系统只能向一个时隙发送一个帧,而且该帧只能在时隙的开始时刻发送。如果系统无法在时隙的开头发送帧,则必须等待下一个时隙启动。如果两个系统尝试在一个时隙的开头传输帧,则

会产生冲突。但其比纯 ALOHA 更好,因为碰撞的可能性更小。时隙 ALOHA 算法数据传输过程示例如图 2 所示。

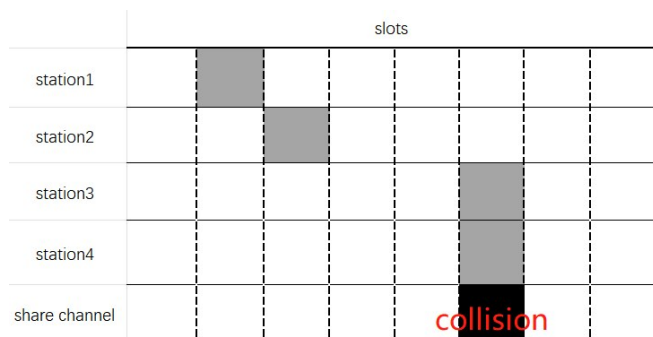


Fig. 2 Example of the slotted ALOHA algorithm data transmission process

图 2 时隙 ALOHA 算法数据传输过程示例

信道吞吐量取决于无碰撞数据帧的通过数量。因为纯 ALOHA 算法在帧传输的任意时间内都可能发生碰撞,所以其脆弱时间为 2τ , τ 定义为单个帧传输时间。时隙 ALOHA 算法只有在帧开始时可能发生碰撞,其脆弱时间缩短为 τ ,碰撞概率缩小了一半,因此吞吐率提升了一倍。

3 ALOHA 算法数学分析

3.1 数学基础

自然对数公式如下:

$$\lim_{n \rightarrow \infty} \left(1 + \frac{x}{n}\right)^n = e^x \quad (1)$$

对式(1)求倒数,可得式(2):

$$\lim_{n \rightarrow \infty} \left(1 - \frac{x}{n}\right)^n = e^{-x} \quad (2)$$

泰勒公式如下:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots, -\infty < x < \infty \quad (3)$$

二项式分布如下:

$$P(X = i) = \binom{n}{i} \left(\frac{\lambda}{n}\right)^i \left(1 - \frac{\lambda}{n}\right)^{n-i} \quad (4)$$

对于二项式分布,当 i 较小,取 $n \rightarrow \infty$ 极限时,有:

$$\frac{\binom{n}{i}}{n^i} \rightarrow \frac{1}{i!} \quad (5)$$

根据式(2)可得:

$$\left(1 - \frac{\lambda}{n}\right)^{n-i} \rightarrow e^{-\lambda} \quad (6)$$

根据式(5)、式(6)可得 Poisson 分布:

$$P(X = i) = \binom{n}{i} \left(\frac{\lambda}{n}\right)^i \left(1 - \frac{\lambda}{n}\right)^{n-i} = \frac{e^{-\lambda} \lambda^i}{i!} \quad (7)$$

根据式(7)和式(1)求 Poisson 分布期望,可得:

$$E(X) = \sum_{i=0}^{\infty} i \frac{\lambda^i}{i!} e^{-\lambda} = \lambda e^{-\lambda} \sum_{i=1}^{\infty} \frac{\lambda^{i-1}}{(i-1)!} = \lambda e^{-\lambda} \sum_{k=0}^{\infty} \frac{\lambda^k}{k!} = \lambda e^{-\lambda} e^{\lambda} = \lambda \quad (8)$$

3.2 推导

定义一段观察时间为 T , 电子标签 i 传输一个数据帧的时间为 τ_i , 观察时间内系统中同时出现的电子标签数量为 n , 在观察时间内由电子标签 i 发送的数据包数量为 r_i , 平均交换的数据包量为 $G^{[17]}$ 。有:

$$G = \sum_{i=1}^n \frac{\tau_i}{T} r_i \quad (9)$$

一般情况下, r_i 相等且为 1 时, τ_i 相等, 因此有:

$$G = \frac{n\tau}{T} \quad (10)$$

G 的逻辑意义是单位时隙时间 τ 内出现的要传输的数据包数, 即数据帧产生速度。

令 $N=T/\tau$, 则:

$$G = \frac{n}{N} \quad (11)$$

N 是根据单位帧传输时间 τ 划分的时隙总数, 在 N 一定的情况下, 同时出现的标签数量越多, G 越大。

3.2.1 时隙 ALOHA 算法

时隙 ALOHA 算法将 T 时间划分为 N 个时隙, 在一个时隙时间范围内出现 k 个帧的概率服从 Poisson 分布。根据式(7)、式(9), 可得:

$$P(k) = \frac{G^k e^{-G}}{k!} \quad (12)$$

令 $k=0$, 则 $P(0) = e^{-G}$ 。

该公式表示在一个时隙时间内没有帧要传输的概率是 e^{-G} , 该值也是无碰撞概率。

e^{-G} 的另一种理解方法为: 在 N 个时隙内, 总共出现了 n 个帧, 在某个时隙内不产生碰撞的条件是其他 $n-1$ 个帧都不选择该时隙。根据式(2)、式(12)可得:

$$q = \left(1 - \frac{1}{N}\right)^{n-1} = \left(1 - \frac{G}{n}\right)^{n-1} = e^{-G} \quad (13)$$

因此, 吞吐率为:

$$S = Gq = Ge^{-G} \quad (14)$$

3.2.2 纯 ALOHA 算法

纯 ALOHA 算法的脆弱时间为时隙 ALOHA 算法的两倍, 因此无碰撞概率为:

$$q = \left(1 - \frac{2}{N}\right)^{n-1} = \left(1 - \frac{2G}{n}\right)^{n-1} = e^{-2G} \quad (15)$$

吞吐率为:

$$S = Gq = Ge^{-2G} \quad (16)$$

4 基于 Python 工具的 ALOHA 算法仿真实现

4.1 实验环境

硬件: CPU 为 Intel i5-12400, Mem 为 8G。

操作系统: Windows 11。

软件: Anaconda3-5.3.0-Windows-x86_64。

代码编辑器: Jupyter Notebook。

4.2 标准吞吐率曲线

功能目标:对式(14)、式(16)进行数值模拟和可视化。

效果:显示出纯 ALOHA 算法和时隙 ALOHA 算法的吞吐率函数曲线,如图3所示。

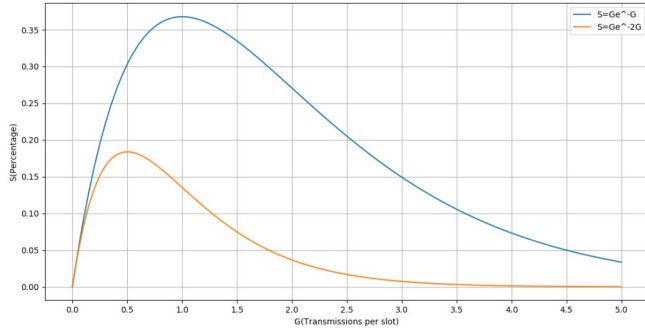


Fig. 3 Function curve of throughput rate

图3 吞吐率函数曲线

关键代码如下:

引入模块:

```
import numpy as np
import matplotlib.pyplot as plt
from math import e
```

通过 numpy 生成 x 和 y 轴数据序列, x 数字范围为 0~5, 总共 100 个点。

```
G = np.linspace(0, 5, 100)
```

```
S = x*pow(e, -x)
```

```
Ss = x*pow(e, -2*x)
```

可视化作图,通过 matplotlib 绘制曲线:

```
plt.plot(G, S, label="S=Ge^-G")
```

```
plt.plot(G, Ss, label="S=Ge^-2G")
```

4.3 碰撞模拟

功能目标:可视化演示时隙 ALOHA 算法碰撞现象。

效果:如图4所示,图示中有绿色柱、红色柱和空白区。空白区表示所在时隙没有数据传输;绿色柱表示有一个帧传输,没有发生冲突;红色柱表示该时隙出现了传输冲突,且柱体高度表示发生碰撞的帧数量。

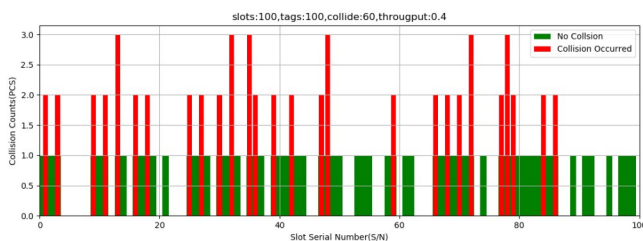


Fig. 4 Display of the conflict phenomena

图4 冲突现象展示

算法说明:

(1)通过初始化,设置时隙总数和标签数为 $N=100$,单位时隙长度为 1。

(2)通过 numpy 模块产生随机数列 time_seq , 数列长度为 N , 随机数范围为 $[0-N]$, time_seq 表示各个标签进行数据传输所选定的时隙编号,接下来对 time_seq 进行排序。

(3)针对随机数列 time_seq 检查冲突,若有冲突记录所在时隙及冲突数量。

例如某个 time_seq 检查结果如下:

[0 1 2 4 4 7 7 7 8 8 8 ... 90 90 91 93 93 94 96 98 98 98]

在 0、1、2 等时隙位置上没有发生碰撞,在 3、5、6 等时隙位置上没有数据传输,而在 4、7、8 等位置上发生了数据碰撞,且在第 4、7、8 位置上分别有 2、3、3 个包发生碰撞。

(4)使用 matplotlib 绘制可视化柱状图,用绿色表示无冲突,红色表示有冲突。

关键代码如下:

引入模块:

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

产生一组 $[0, N)$ 的随机序列数并从小到大进行排序, 序列长度等于 N :

```
time_seq = np.sort(np.random.randint(0, N, size=N))
```

对随机序列数进行冲突检查和记录,记录总冲突次数和单个时隙冲突次数:

```
while j < len(time_seq):
```

```
collide_count_per_slot = 0
```

while $j < \text{len}(\text{time_seq})$ and $\text{time_seq}[j] == \text{time_seq}[i]$: # 冲突检测

```
j += 1
```

```
collide_count += 1 # 总冲突次数增加
```

```
collide_count_per_slot += 1 # 本时隙冲突次数增加
```

```
flag = True
```

```
if flag: # 发现冲突,冲突数补1,登记冲突位置
```

```
collide_count += 1
```

```
collide_count_per_slot += 1
```

```
collide_seq.append(time_seq[i])
```

```
collide_count_seq.append(collide_count_per_slot)
```

```
collide_count_per_slot = 0
```

```
flag = False # 冲突标记重置为 False
```

```
i = j
```

```
j = i + 1
```

通过可视化,绘制柱状图:

```
plt.bar(time_seq, 1, width=unit_time)
```

```
plt.bar(collide_seq, collide_count_seq, color="red")
```

4.4 吞吐率曲线

功能目标:在时隙不变的情况下,分别显示纯 ALOHA 算法和时隙 ALOHA 算法工作时的实际吞吐率随标签数量变化而变化的曲线,并与标准曲线作比较。

效果:如图5所示,蓝色线条为纯 ALOHA 吞吐率变化曲线,黄色线条为时隙 ALOHA 吞吐率变化曲线。通过与图3标准函数曲线对比,其仿真模拟结果与理论一致。

算法说明:

(1)定义一个观察时间 $T=1.0$, 定义单位帧时隙 $t=0.001$, 因此总共有 $N=T/t$ 个时隙。

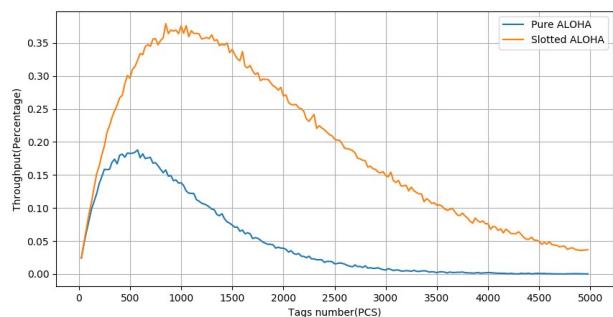


Fig.5 Numerical simulated throughput curve

图5 数值模拟吞吐率曲线

(2)标签数量从10依次变化递进到最大值 $M=5 \times N$,步进 $D=M/100$,设置 $n=\text{range}(D, M, D)$ 。

(3)对于每一个 n ,产生一个长度为 n 、范围为 $[0, N)$ 的随机序列数 time_seq ,对该序列从小到大进行排序,并统计该序列发生碰撞的标签数量 c 。然后计算吞吐率 $S=(n-c)/n$,多次执行计算平均值 Savg ,存入数组 S_arr 。

(4)最后以 n 为 x 轴序列、 S_arr 为 y 序列作曲线图。

关键代码如下:时隙ALOHA算法的冲突检测和记录方法参考第3.3节代码。纯ALOHA算法的冲突检查方法需在3.3节代码的基础上进行修改,将 $\text{while } j < \text{len}(\text{time_seq})$ and $\text{time_seq}[j] == \text{time_seq}[i]$ 修改为 $\text{while } j < \text{len}(\text{time_seq_all})$ and $\text{time_seq_all}[j] < \text{time_seq}[j-1] + t$ 。

5 结语

不同于传统的MATLAB方式,本文引入Python大数据分析工具和技术实现了ALOHA算法多标签碰撞现象的模拟和可视化分析。首先,实现了标签碰撞的可视化演示,通过柱形图对冲突位置进行标注,并且显示发生冲突的标签数量。其次,实现了ALOHA算法吞吐率的仿真计算,获得了信道吞吐率随标签数量增长(时隙数不变)而变化的曲线,该曲线与数学理论曲线比较结果一致,验证了数学结论的正确性,也反向验证了本文提供的仿真算法的正确性。本文提供的方法可用于物联网ALOHA算法的理论教学工作,也可用于进一步的算法改进研究(文中代码来源于<https://github.com/qwyang/AlohaSim>)。

参考文献:

- [1] ZHOU S K, DENG M L. Overview of ALOHA label anti-collision algorithm [J]. Computer Engineering and Application, 2017, 53 (14): 9-17.
周少珂,邓森磊. ALOHA 标签防碰撞算法综述[J]. 计算机工程与应用, 2017, 53(14):9-17.
- [2] Aloha. TAE [EB/OL]. <https://www.tutorialandexample.com/aloha>.
- [3] YANG L, LIU L A. An ALOHA based RFID tag multiple access algorithm [J]. Software Guide, 2015, 14 (2): 66-68.
杨灵,刘磊安. 一种基于ALOHA的RFID标签多址接入算法[J]. 软件导刊, 2015, 14(2):66-68.
- [4] YU Z J, DONG Q D. ALOHA protocol simulation design based on Matlab [J]. Journal of Guiyang University, 2019, 14 (4): 56-60.

- 于子甲,董全德. 基于Matlab的ALOHA协议仿真设计[J]. 贵阳学院学报, 2019, 14(4):56-60.
- [5] WANG D Y, ZHANG W P, WANG Z J. Research on anti-collision of RFID tags based on improved ALOHA algorithm [J]. Packaging Engineering, 2021, 42 (17): 244-248.
王冬云,张维平,汪志佳. 基于改进ALOHA算法的RFID标签防碰撞研究[J]. 包装工程, 2021, 42(17):244-248.
- [6] HE J D, BU Y L, SHI C C, et al. Frame length optimization for RFID dynamic frame time slot ALOHA protocol [J]. Computer Science and Exploration, 2021, 15 (7): 1227-1236.
何金栋,卜艳玲,石聪聪,等. 面向RFID动态帧时隙ALOHA协议的帧长优化[J]. 计算机科学与探索, 2021, 15(7):1227-1236.
- [7] WEI F L, LIU P. RFID anti-collision algorithm for dangerous goods transportation based on dynamic frame slot ALOHA [J]. Journal of Shenyang University of Technology, 2020, 42 (5): 540-544.
魏福禄,刘攀. 基于动态帧时隙ALOHA的危险品运输RFID防碰撞算法[J]. 沈阳工业大学学报, 2020, 42(5):540-544.
- [8] WANG W. Research on anti-collision algorithm of RFID tags based on multi branch tree [D]. Hangzhou: Hangzhou Dianzi University, 2021.
王威. 基于多叉树的RFID标签防碰撞算法研究[D]. 杭州:杭州电子科技大学, 2021.
- [9] WU F. Analysis and research on anti-collision analysis of algorithms of UHF RFID tags [D]. Nanjing: Nanjing University of Posts and Telecommunications, 2020.
吴凡. UHF RFID标签防碰撞算法分析与研究[D]. 南京:南京邮电大学, 2020.
- [10] LI X J. Research on adaptive label anti-collision algorithm with multi period response [D]. Dalian: Liaoning Normal University, 2022.
李雪娇. 多周期响应的自适应标签防碰撞算法研究[D]. 大连:辽宁师范大学, 2022.
- [11] WES M. Python for data analysis: data wrangling with Pandas, NumPy, and IPython[M]. Sebastopol: O'Reilly, 2018.
- [12] CHEN L, CHEN D X, LU W Z, et al. Design and implementation of score statistics based on Python+Matplotlib [J]. Intelligent Computer and Applications, 2020, 10 (8): 275-276, 280.
陈路,陈道喜,陆卫忠,等. 基于Python+Matplotlib的成绩统计设计与实现[J]. 智能计算机与应用, 2020, 10(8):275-276,280.
- [13] LIN Y C, ZHUO C Y, LIN Q. Visualization of antibacterial drug use intensity based on Python+Pandas+Matplotlib [J]. Straits Pharmacy, 2022, 34 (10): 126-129.
林郁苍,卓朝阳,林琦. 基于Python+Pandas+Matplotlib的抗菌药物使用强度可视化[J]. 海峡药学, 2022, 34(10):126-129.
- [14] GUAN C B. Exceeding economic order quantity based on data analysis method [J]. Logistics Technology, 2022, 41 (10): 83-86, 137.
关成斌. 基于数据分析法的超限经济订货批量[J]. 物流技术, 2022, 41(10): 83-86,137.
- [15] LIU X L, ZHANG Y Q, DONG A G. Design and implementation of a physics experimental data processing system based on Python [J]. Experimental Technology and Management, 2021, 38 (3): 74-78.
刘雪琳,章钰琪,董爱国. 基于Python的物理实验数据处理系统设计与实现[J]. 实验技术与管理, 2021, 38(3):74-78.
- [16] WU T, WANG S F. Exploration and reflection on online teaching of college physics using Jupyter-lab interactive platform [J]. Journal of Hubei Second Normal University, 2021, 38(8): 91-94, 103.
吴涛,王世芳. 利用Jupyter-lab交互式平台进行大学物理在线教学的探索与思考[J]. 湖北第二师范学院学报, 2021, 38(8):91-94,103.
- [17] LUO Z Y, YANG M M, LI Y F, et al. Principles and applications of Internet of Things radio frequency identification (RFID) [M]. Beijing: People's Posts and Telecommunications Press, 2019.
罗志勇,杨美美,李永福,等. 物联网射频识别(RFID)原理及应用[M]. 北京:人民邮电出版社, 2019.

(责任编辑:黄 健)