

基于标签估计的动态最优多分支搜索防碰撞算法^①

崔英花^② 赵玉萍

(北京大学信息科学技术学院 区域光纤通信网与新型光通信系统国家重点实验室 北京 100871)

摘 要 针对射频识别(RFID)系统的动态二分支搜索(DBS)算法,提出了一种改进的动态多分支搜索算法,然后通过对多分支算法系统效率的分析,给出了最优分支数目的计算方法,并结合标签估计方法,得到了动态最优多分支搜索(DOMBS)防碰撞算法。理论分析和实验仿真表明,与 DBS 算法相比,改进算法——DOMBS 防碰撞算法的查询次数较少,具有明显的性能优势。DOMBS 算法的系统识别效率达 58%,而 DBS 算法的效率仅为 50% 左右,如果标签估计方法能够进一步优化,DOMBS 算法的效率还能得到进一步提高。

关键词 射频识别(RFID),防碰撞协议,动态二分支搜索(DBS),多分支,标签估计

0 引 言

射频识别(radio frequency identification, RFID)是利用射频通信实现的非接触式自动识别技术,被广泛应用于制造业、物流管理、零售业和仓储管理等领域。RFID 系统主要由标签和阅读器构成,具有唯一标识号的标签附着在物体上,当阅读器与标签建立起通信后,阅读器通过读取标签内的相关信息,就可以识别该物体。由于被动式标签的造价低廉,其功能非常有限,标签之间无法通信,这就造成了阅读器发出询问指令时,会有读写区域内的多个标签同时响应的现象,即标签碰撞。标签碰撞引起信息的相互冲突,使得阅读器无法读取标签信息,造成信息识别的延迟和带宽的浪费^[1],因此防碰撞技术成为提高 RFID 系统识别效率的关键技术之一。

目前,RFID 防碰撞协议主要分两类,分别是基于 ALOHA 的类型和基于树搜索的类型。基于 ALOHA 的类型的协议包含 ISO180006A 和 ISO180006C 标准^[2]等,其特征为标签在随机选择的时隙中发送数据,多为不确定型。该类型协议易发生“标签荒”的现象,即有些标签可能会长时间得不到识别^[3]。对于基于 ALOHA 的类型的防碰撞算法的优化多采用估计标签数目,根据估计的标签数目来优选帧大小的方法^[4-11]。采用了这种优化方法后,系统的识别

效率大大提高。基于树搜索的类型的协议包含 ISO180006B^[1]标准和各种树查询算法^[12-15],多为确定型。但 ISO180006B 标准是一个特例,它属于基于树搜索算法的概率型协议。树搜索方法可以避免“标签荒”的现象,但由于要识别所有的 ID 比特,所以会有较长的时延^[3]。传统的基于树搜索的防碰撞算法大都采用动态二分支搜索(dynamic binary search, DBS)方法:当标签发生碰撞时,阅读器通过标签产生的随机数或唯一标识号 ID 将标签分为左右两个分支,对这两个分支上的标签依次进行识别,若某分支上有多于一个的标签,则对该分支继续进行分解,随着树深度的增加,分到各分支上的标签越来越少,当某分支只包含唯一一个标签时,就可以对其进行识别,如此反复,直至把所有标签都识别完毕。文献[16]指出,采用将标签分成多个分支($N > 2$)的方法可以大大提高系统的识别效率,例如文献[17]对 ISO180006B 标准做了多分支的改进,将其系统识别效率从约 34.8%^[18]提高到 43% 左右,文献[19]通过计算最优初始前缀长度,对树搜索算法进行多分支改进,得到的系统识别效果也有较大提高。与之相仿,本文针对 DBS 算法,给出了一种多分支搜索(multi-branch search, MBS)的改进算法,通过对系统识别效率的分析,给出了最优分支数目的计算方法,并结合标签估算方法,最终得到了动态最优多分支搜索(dynamic optimal MBS, DOMBS)算

^① 863 计划(2006AA01Z278)资助项目。

^② 女,1973 年生,博士生;研究方向:无线资源管理与优化,射频识别技术;联系人, E-mail: cui_ying_hua@sina.com (收稿日期:2009-04-03)

法。仿真结果表明,本文提出改进算法的系统识别效率大大优于 DBS 算法。

1 动态二分支搜索算法

动态二分支搜索(DBS)算法是基于树搜索的防碰撞算法^[12]。该算法把标签当成树结构上的叶子节点,标签的标识号 ID 决定其在树结构中的位置。每个标签都有唯一的标识符 ID 属于集合 $\{0,1\}^L$, 这里 L 为 ID 的长度。首先定义阅读器的前缀集合为 $Q = \{P_0, P_1, \dots, P_i\}$, 其中 $P_i = p_{i1} p_{i2} \dots p_{il}$, l 为前缀的长度, $l \leq L$ 。 Q 实际上是个堆栈,其元素遵从后进先出的原则,每当阅读器发送一个前缀后,从 Q 中删除该前缀。例如,当 $Q = \{P_i, P_{i+1}, \dots\}$ 时,阅读器发送长度为 X 的前缀 $P_i = p_{i1} p_{i2} \dots p_{iX}$, Q 更新为 $Q = \{P_{i+1}, \dots\}$ 。 DBS 算法由多轮询问和应答组成,在每一轮查询中,阅读器询问标签是否有相同的前缀。标签 ID 若与询问前缀相匹配,则发送 ID 的剩余比特。此时,若有多于一个的标签具有相同的前缀,它们同时响应就会产生碰撞,此时阅读器检测到碰撞,需要生成新的前缀。参看图 1,若接收的比特信息中第一个碰撞为第 j 位,则阅读器将形成两个新的前缀 $p_{i1} p_{i2}, \dots, p_{iX} p_{i(x+1)} p_{i(x+2)}, \dots, p_{i(x+j-1)} 0$ 和 $p_{i1} p_{i2}, \dots, p_{iX} p_{i(x+1)} p_{i(x+2)}, \dots, p_{i(x+j-1)} 1$ 放入到 Q 中,即更新后的 $Q = \{p_{i1} p_{i2}, \dots, p_{iX} p_{i(x+1)} p_{i(x+2)}, \dots, p_{i(x+j-1)} 0, p_{i1} p_{i2}, \dots, p_{iX} p_{i(x+1)} p_{i(x+2)}, \dots, p_{i(x+j-1)} 1, P_{i+1}, \dots\}$ 。



图1 阅读器发送 X 比特前缀,具有相同前缀的标签发送剩余 ID 比特

DBS 算法的具体工作过程可描述如下:

(1) 阅读器向标签发送前缀集合 Q 中前缀元素,集合 Q 的初始值为 $\{\&\}$;

(2) 若标签接收到的查询前缀与自己的标签 ID 相匹配时,则发送前缀之后的剩余 ID 比特;若接收询问信号为 $\&$,要求所有标签回复其完整的标签 ID;

(3) 此时,阅读器会检测到三种情况:(a)检测到只有一个标签回复,则识别该标签,回到步骤(1);

(b)检测到没有标签回复,则回到步骤(1);(c)检测到标签碰撞,则在第一个碰撞位置上加上符号‘0’,‘1’,将按照前面所述方法形成的新的前缀置于前缀集合 Q 中,等待下一轮查询时使用,回到步骤(1)。

阅读器持续查询前缀集合 Q 中的元素,当集合 Q 为空时,表示所有标签被识别完毕。

相应地,DBS 算法的流程图如图 2 所示。

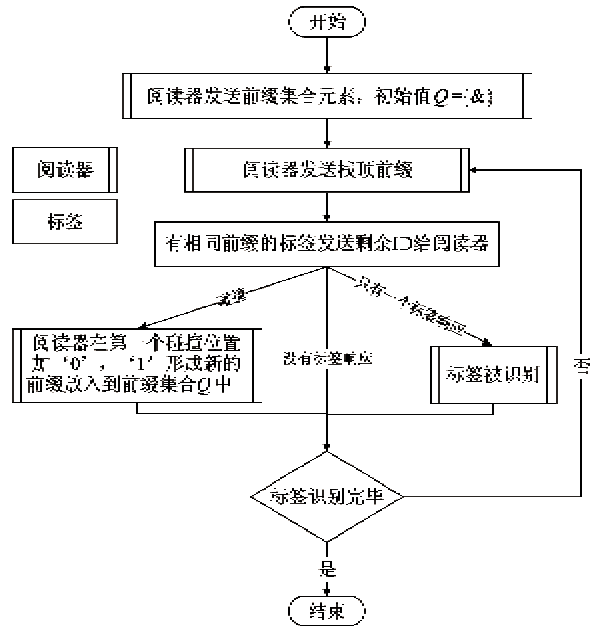


图2 查询树算法流程图

2 动态多分支搜索算法

传统的 DBS 算法都是通过二分支来进行标签识别的,本节对其进行改进,提出了一种动态多分支搜索(dynamic multi-branch search, DMBS)算法,其具体过程如下:首先选择初始分支数目为 $N = 2^M$, 并用 $\{0, \dots, N-1\}$ 来初始化阅读器的前缀集合 Q 。这里 M 取值为正整数,初始的前缀元素指的是采用 M 位编码的 N 种不同位指派方式。随后的操作就与 DBS 算法相同了。这相当于将待查询的标签分为 N 个子树,再对每棵子数分别应用 DBS 算法。该算法流程如图 3 所示。随后的章节将对动态二分支与多分支算法的效率进行分析,由此推导出最优多分支的数目,并在此基础上提出动态最优多分支搜索(DOMBS)算法。

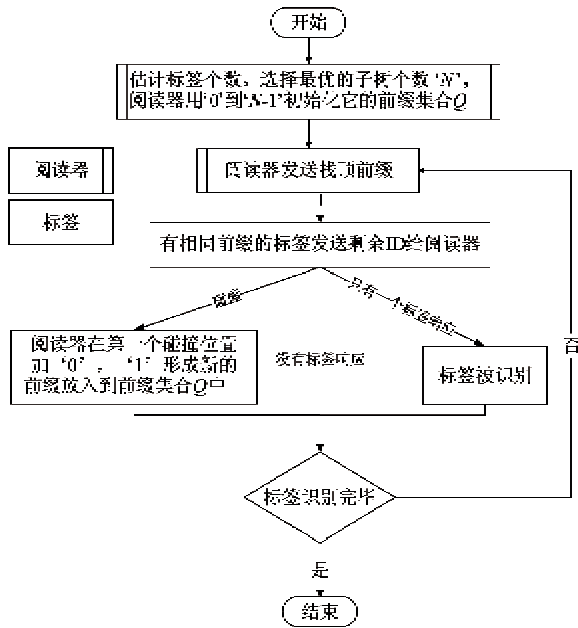


图3 最优多分支树算法

3 动态二分支与动态多分支搜索算法的效率

本节将对算法的识别效率进行分析。我们知道查询次数是决定系统效率的关键因素,系统效率通常定义为如下形式:

$$\text{效率 } e = \frac{\text{标签个数 } n}{\text{查询次数 } E_{\text{query_time}}} \quad (1)$$

3.1 动态二分支树搜索算法的识别效率

结论 1: 假设有 n 个标签, 若使用二分支查询树算法识别标签, 则查询次数的平均值为

$$E_{\text{query_time}}(N | N=2, n) = E_{\text{query_time}}(2, n) = 2n - 1 \quad (2)$$

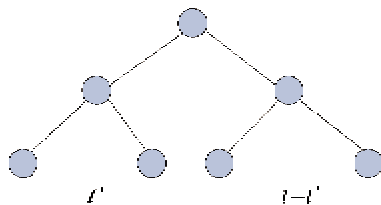
证明:

我们使用递归来证明。

(1) 当 $n = 1$ 时, $E_{\text{query_time}}(2, 1) = 2 \times 1 - 1 = 1$;

(2) 假设 $n < l$ 时命题成立;

(3) 当 $n = l$ 时, 经过一次查询后, 标签将产生碰撞, 分解为如图 4 所示的两棵子树。 l' 是左子树中的标签个数, $l' > 0$ 。

图4 l 个标签分布图

因此查询次数为 $E_{\text{query_time}}(2, l) = 1 + E(2, l') + E(2, l - l')$ 。

由于 $0 < l' < l, 0 < l - l' < l$, 我们可以得到 $E_{\text{query_time}}(2, l) = 1 + 2l' - 1 + 2(l - l') - 1 = 2l - 1$ 由 (1), (2), (3) 可得 $E_{\text{query_time}}(2, n) = 2n - 1$, 证毕。

由此, 可得到 DBS 算法的效率为

$$e_{\text{DBS}} = \frac{n}{E_{\text{query_time}}(2, n)} = \frac{n}{2n - 1} \quad (3)$$

当 n 比较大时, DBS 的效率接近 50%。

3.2 动态多分支搜索算法的识别效率

首先定义期望值 E_k : 假设有 n 个标签, 被随机分配到 N 个子树中, 则包含 k 个标签的子树个数的期望值 E_k 符合二项式分布^[20], 如下所示:

$$E_k = E_{N, n}(k) = NC_n^k \left(\frac{1}{N}\right)^k \left(1 - \frac{1}{N}\right)^{n-k} \quad (4)$$

进一步我们可以得到如下结论:

结论 2: 假设有 n 个标签, 选择 $N = 2^M$ 个子树 (M 是正整数), 则使用 N 分支查询这 n 个标签的查询次数为

$$E_{\text{query_time}}(N, n) = 2E_0 + 2n - N = 2N\left(1 - \frac{1}{N}\right)^n + 2n - N \quad (5)$$

证明: 由于 $E_k (k = 0, 1, 2, \dots, n)$ 是包含 k 个标签的子树个数的期望值, 有如下等式成立:

$$\begin{cases} 0 \times E_0 + 1 \times E_1 + \dots + n \times E_n = n \\ E_0 + E_1 + \dots + E_n = N \end{cases} \quad (6)$$

对于这 N 个子树, 将分别使用 DBS 算法作进一步查询, 由结论 1, 我们有

$$E_{\text{query_time}}(N, n) = E_0 + \sum_{k=1}^n E_{\text{query_time}}(2, k) \times E_k \quad (7)$$

把式(2)代入式(7)可以得到

$$E_{\text{query_time}}(N, n) = E_0 + \sum_{k=1}^n (2k - 1) \times E_k \quad (8)$$

整理可得

$$E_{\text{query_time}}(N, n) = E_0 + 2 \sum_{k=1}^n k \times E_k - \sum_{k=1}^n E_k \quad (9)$$

把式(4)、(6)代入式(9)可得

$$\begin{aligned} E_{\text{query_time}}(N, n) &= 2E_0 + 2n - N \\ &= 2N\left(1 - \frac{1}{N}\right)^n + 2n - N \end{aligned}$$

证毕。

由此, 就得到了多分支算法的效率为

$$e_{\text{DMBS}} = \frac{n}{E_{\text{query_time}}(N, n)}$$

$$= \frac{n}{2N(1 - \frac{1}{N})^n + 2n - N} \quad (10)$$

由式(5)可以看出,当 n 和 N 固定时, E_0 越小,查询次数越少。通常只有在标签 ID 均匀分布时,才能使空子树的数量最少。所以在查询标签时,要尽量保证标签的 ID 是均匀分布的。可以考虑采用将标签 ID 倒置的方法:假设标签 ID 为 $TagID = X_1 X_2 \cdots X_{64}$,若来自同一厂家的前 30 位都是相同的,而后 34 位将是均匀分布的,通过标签倒置为 $TagID' = X_{64} X_{63} \cdots X_1$,就可以达到使标签 ID 均匀的效果。

4 最优分支数目的计算及动态最优多分支搜索算法

由式(10)可知:在多分支算法中,分支数目 N 是影响其效率的重要因素。本节将证明最优多分支数目的存在性并给出其计算方法,如下面的结论 3 和结论 4 所示:

结论 3:假设有 n 个标签,其标签 ID 为均匀分布。则使查询次数 $E_{\text{query_time}}(N, n)$ ($N = 2^M$, M 为正整数)最少的最优分支数只能在 $N = 2^{T-1}, 2^T$ 或 2^{T+1} ($2^T \leq n < 2^{T+1}$) 中取得。

证明:由于式(5)是离散函数,不能采用求导方式来求最小值点。在这里,我们采用夹逼方法来证明。

(1) 对于 $n \geq 3N$, 由式(5), 我们可以得到

$$\begin{aligned} E_{\text{query_time}}(N, n) - E_{\text{query_time}}(2N, n) \\ = 2E_{N,n}(0) + 2n - N - (2E_{2N,n}(0) + 2n - 2N) \\ = 2E_{N,n}(0) - 2E_{2N,n}(0) + N \end{aligned} \quad (11)$$

把式(2)代入式(11),整理可得

$$\begin{aligned} E_{\text{query_time}}(N, n) - E_{\text{query_time}}(2N, n) \\ = 2N(1 - \frac{1}{N})^n - 4N(1 - \frac{1}{2N})^n + N \\ = N[1 + 2(1 - \frac{1}{N})^n - 4(1 - \frac{1}{2N})^n] \end{aligned} \quad (12)$$

两边除以 N , 可以得到

$$\begin{aligned} \frac{E_{\text{query_time}}(N, n) - E_{\text{query_time}}(2N, n)}{N} \\ = 2 \times (1 - \frac{1}{N})^n - 4 \times (1 - \frac{1}{2N})^n + 1 \end{aligned} \quad (13)$$

由于 $(1 - \frac{1}{N})^n > 0$, 所以

$$\frac{E_{\text{query_time}}(N, n) - E_{\text{query_time}}(2N, n)}{N}$$

$$> 1 - 4 \times (1 - \frac{1}{2N})^n \quad (14)$$

对于 $n \geq 3N$, 有

$$\begin{aligned} \frac{E_{\text{query_time}}(N, n) - E_{\text{query_time}}(2N, n)}{N} \\ \geq 1 - 4 \times (1 - \frac{1}{2N})^{3N} \\ = 1 - 4 \times [(1 - \frac{1}{2N})^{2N}]^{3/2} \end{aligned} \quad (15)$$

由于 $(1 - \frac{1}{N})^N$ 是单调递增的, $\lim_{N \rightarrow \infty} (1 - \frac{1}{N})^N = \frac{1}{e}$, 因此 $(1 - \frac{1}{N})^N < \frac{1}{e} < 0.369$, 则 $[(1 - \frac{1}{N})^N]^{3/2} < 0.369^{3/2}$, 所以

$$\begin{aligned} \frac{E_{\text{query_time}}(N, n) - E_{\text{query_time}}(2N, n)}{N} \\ > 1 - 4 \times (0.369)^{3/2} \\ \approx 1 - 0.8966 > 0 \end{aligned}$$

当 $n \geq 3N$ 时,可以得到

$$E_{\text{query_time}}(N, n) > E_{\text{query_time}}(2N, n) \quad (16)$$

(2) 同理,对于 $n \leq N$, 有

$$\begin{aligned} \frac{E_{\text{query_time}}(N, n) - E_{\text{query_time}}(2N, n)}{N} \\ = 2 \times (1 - \frac{1}{N})^n - 4 \times (1 - \frac{1}{2N})^n + 1 \end{aligned} \quad (17)$$

由于 $(1 - \frac{1}{N})^n < (1 - \frac{1}{2N})^n$, 所以

$$\begin{aligned} \frac{E_{\text{query_time}}(N, n) - E_{\text{query_time}}(2N, n)}{N} \\ < 1 + 2 \times (1 - \frac{1}{2N})^n - 4 \times (1 - \frac{1}{2N})^n \\ = 1 - 2 \times (1 - \frac{1}{2N})^n \end{aligned} \quad (18)$$

由于 $n \leq N$, 所以

$$\begin{aligned} \frac{E_{\text{query_time}}(N, n) - E_{\text{query_time}}(2N, n)}{N} \\ \leq 1 - 2 \times (1 - \frac{1}{2N})^N \\ = 1 - 2 \times [(1 - \frac{1}{2N})^{2N}]^{1/2} \end{aligned} \quad (19)$$

由于 $(1 - \frac{1}{N})^N$ 是单调递增的, 因此当 $N \geq 2$ 时,

$\sqrt{(1 - \frac{1}{N})^N} \geq 0.5$ 。由式(19)可得

$$\begin{aligned} \frac{E_{\text{query_time}}(N, n) - E_{\text{query_time}}(2N, n)}{N} \\ \leq 1 - 2 \times 0.5 = 0 \end{aligned}$$

可以得到

$$E_{\text{query_time}}(N, n) < E_{\text{query_time}}(2N, n) \quad (n \leq N) \quad (20)$$

由上述分析可知:

a) 对于 $N = 2^M > 2^{T+1} > n$, 根据式(20), 可得出查询次数是随着 N 的增大而增大的, 即

$$\begin{aligned} E_{\text{query_time}}(N, n) &> E_{\text{query_time}}\left(\frac{N}{2}, n\right) \\ &> \cdots > E_{\text{query_time}}(2^{T+1}, n) \end{aligned}$$

b) 对于任何 $N = 2^M < 2^{T-1} < 2^T \leq n$, 根据式(16), 可得出查询次数是随着 N 的增大而减小的, 即

$$\begin{aligned} E_{\text{query_time}}(N, n) &> E_{\text{query_time}}(2N, n) \\ &> \cdots > E_{\text{query_time}}(2^{T-1}, n) \end{aligned}$$

根据以上分析, 可知查询次数 $E_{\text{query_time}}(N, n)$ 最小值应存在于 $N = 2^{T-1}, 2^T$ 或 2^{T+1} ($2^T \leq n < 2^{T+1}$) 中, 即最优分支数存在于 $2^{T-1}, 2^T$ 或 2^{T+1} 中。

结论 4: 使 $E_{\text{query_time}}(N, n)$ 值最小的 N 值, 即最优分支数 N_{opt} 可以由下式得到:

$$\begin{aligned} N_{\text{opt}} = \arg \min_N \{ &E_{\text{query_time}}(2^{T-1}, n) \\ &E_{\text{query_time}}(2^T, n), E_{\text{query_time}}(2^{T+1}, n) \} \end{aligned} \quad (21)$$

证明: 由结论 3, 即可得到此结论。

由式(21), 当标签数量从 100 到 2000 时, 其最优分支数如图 5 所示。当标签数目等于 1000 时, 阅读器采用不同分支数目的查询次数如图 6 所示, 其最优分支数为 $N = 2^9 = 512$, 与图 5 显示的结果是一致的。

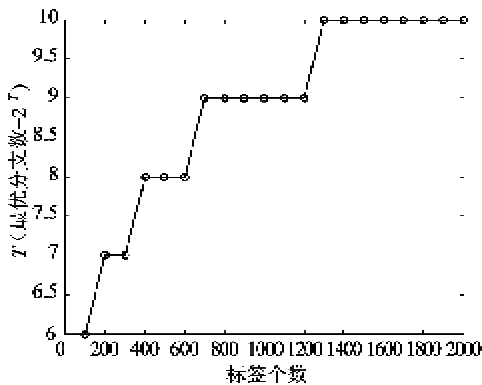


图 5 100 到 2000 标签的最优分支数

在此基础上, 我们提出了动态最优多分支搜索 (DOMBS) 算法, 其具体方法如下: 首先估计标签的数目, 再根据所得的标签数目得到最优分支数 $N = 2^M$, 并用 $\{0, \dots, N-1\}$ 来初始化阅读器的前缀集

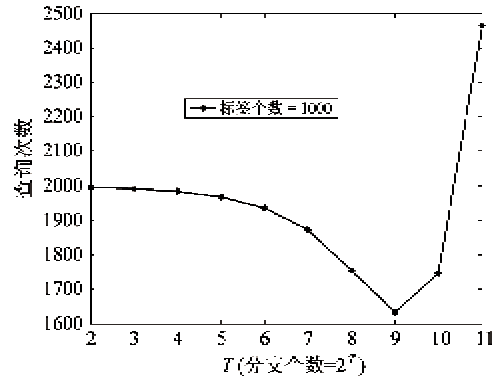


图 6 阅读器对 1000 个标签进行不同的分支后, 所用的查询次数

合 Q , 将标签分为 N 个分支。(这里 M 取值为正整数, 初始的前缀元素指的是采用 M 位编码的 N 种不同位指派方式), 然后再用 DBS 算法对每一个分支进行查询识别。由以上的理论分析可知, DOMBS 算法使用最少的查询次数, 从而可以得到很高的系统效率。

本文采用的标签估计方法参考文献[21]。首先使用 DBS 算法识别出前 m 个标签, 假设此时前缀集合为 $Q = \{P_0, P_1, \dots, P_k\}$, 其前缀长度分别为 $L_0, L_1, \dots, L_k$, 则识别这 m 个标签的概率可以由下式计算:

$$p = 1 - \sum_{i=1}^k \frac{1}{2^{L_i}} \quad (22)$$

由此可得到标签估计数为

$$n' = \frac{m}{p} \quad (23)$$

得到标签数目之后, 就可以选择最优分支数进行多分支搜索。

假设实际的标签个数为 n , 识别前 m 个标签所用的查询次数为 M , 则剩余的标签个数为 $n' - m$, 由此得到的分支个数为 N'_{opt} 。则这种情况下, 相应的系统效率为

$$\begin{aligned} e' &= \frac{n}{M + E_{\text{query_time}}(N'_{\text{opt}}, n - m)} \\ &= \frac{n}{M + 2N'_{\text{opt}} \left(1 - \frac{1}{N'_{\text{opt}}}\right)^{n-m} + 2(n - m) - N'_{\text{opt}}} \end{aligned} \quad (24)$$

5 仿真结果和性能比较

本节使用 MATLAB 仿真工具对动态最优多分支搜索算法 (DOMBS) 和动态二分支搜索算法 (DBS) 进

行了仿真比较。

假设标签 ID 长度等于 30 并且服从均匀分布, 将 2000 个标签分别分成 $2^2, 2^3, \sim 2^{10}$ 个分支, 利用公式(10)得到的系统效率与实际仿真结果比较如图 7 所示。仿真结果表明, 仿真值与理论值几乎重合。从图 7 中也可以看出, 当分支数为 $2^{10} = 1024$ 时, 系统效率可以达到最高, 这也与图 6 中的结论相同。

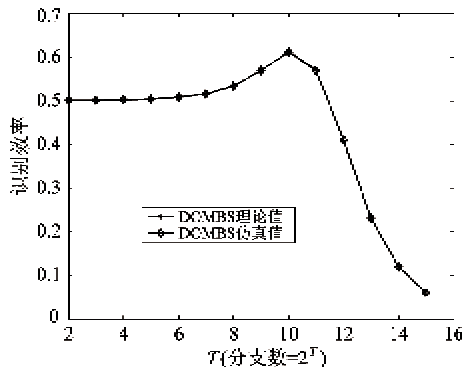


图 7 分别采用不同分支数,使用 DOMBS 算法理论值和仿真值比较图

图 8 针对标签数目从 100 到 2000 的情况, 比较了 DOMBS 算法与 DBS 算法系统效率的理论值和仿真值。其中在假设标签数目未知的场景下, 阅读器在识别完 8 个标签后估计标签数之后, 再根据估计出的标签数进行最优分支计算。每组标签仿真 100 次作平均, 仿真结果表明对于标签数目已知的 DOMBS 和 DMS 两种仿真值与理论值都很接近。DOMBS 的系统效率在 61% 左右, 而 DBS 的效率仅为 50% 左右。由于标签估计需要占用一些查询次数, 同时估计误差也会造成的最优分支数的误差, 因此会造成系统识别效率的少许损失, 但根据标签估计

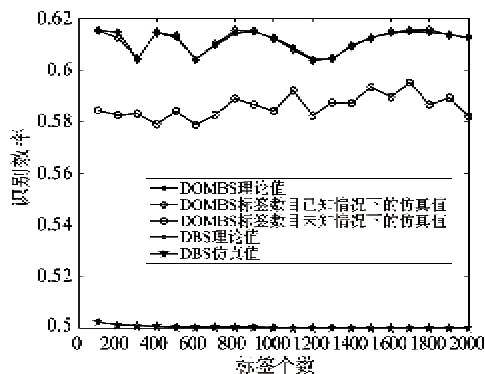


图 8 采用最优分支,分别使用 DOMBS 和 DBS 算法的理论值和仿真值以及有标签估计的 DOMBS 仿真比较图

来选择最优分支数的 DOMBS 算法识别效率仍在 58% 左右, 优于 DMS 算法, 若标签估计方法能够进一步优化, DOMBS 算法效率还可以得到提高。

图 9 显示了标签数目从 10 到 100 的情况, 可以看出, 当标签数目很小 (如 $n \leq 10$) 时, 系统效率与 DBS 算法基本相同, 这是由于我们使用前 8 个标签来估计标签数。当标签数目增大时, 系统效率将有较大提高。

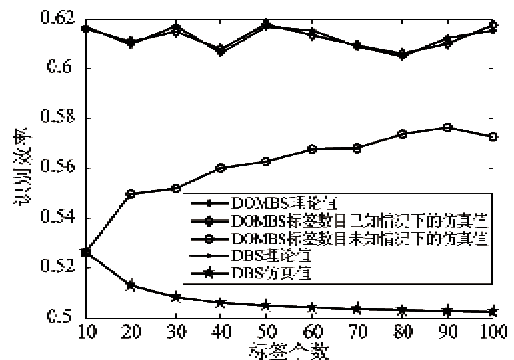


图 9 采用最优分支,分别使用 DOMBS 和 DBS 算法的理论值和仿真值以及有标签估计的 DOMBS 仿真比较图

6 结论

本文基于 DBS 算法, 给出了一种多分支改进的实现方法, 证明了最优分支数目的存在性, 给出了最优分支数目的计算方法, 并结合标签估计方法提出了一种动态最优多分支搜索 (DOMBS) 防碰撞算法。理论分析和仿真结果证明, 与动态二分支搜索算法相比较, 改进算法查询次数较少, 具有更优的性能。相比 DBS 算法 50% 的系统识别效率, DOMBS 的算法效率理论上可达 61% 左右。实际引入的标签估计会带来性能损失, 但其效率仍优于 DMS 算法, 大大提高了系统的标签识别效率。

参考文献

- [1] Shih D H, Sun P L, Yen D C, et al. Taxonomy and survey of RFID anti-collision protocols. *Computer communications*, 2006, 29:2150-2166
- [2] Information technology automatic identification and data capture techniques—Radio frequency identification for item management air interface—Part 6: Parameters for air interface communications at 860-960MHz, 2005
- [3] Myung J, Lee W, Srivastava J, et al. Tag-splitting: adaptive collision arbitration protocols for RFID tag identification. *IEEE Trans on Parallel and Distributed Systems*, 2007, 18 (6):763-775

- [4] Park J, Chung M Y, Lee T J. Identification of RFID tags in framed-slotted ALOHA with robust estimation and binary selection. *IEEE Communications Letters*, 2007, 11(5): 452-454
- [5] Tao C, Li J. Analysis and simulation of RFID anti-collision algorithms. In: Proceedings of the 9th International Conference on Advanced Communication Technology, Korea, 2007. 697-701
- [6] Huang X, Le S. Efficient dynamic framed slotted ALOHA for RFID passive tags. In: Proceedings of the 9th International Conference on Advanced Communication Technology, Korea, 2007. 94-97
- [7] Tong Q L, Zou X C, D S, et al. Modeling the anti-collision process of RFID system by markov chain. In: Proceedings of the International Conference on WiCom 2007, Shanghai, China, 2007. 2504-2507
- [8] Joe I, Lee J. A novel anti-collision algorithm with optimal frame size for RFID system. In: Proceedings of the 5th ACIS International Conference on Software Engineering Research, Management and Applications, Busan, Korea, 2007. 424-428
- [9] Peng Q S, Zhang M, Wu W M. Variant enhanced dynamic frame slotted ALOHA algorithm for fast object identification in RFID system. In: Proceedings of the IEEE International Workshop on Anti-counterfeiting, Security, Identification, Xiamen, China, 2007. 88-91
- [10] Woo J S, Jeong G K. Partitioning of tags for near-optimum RFID anti-collision performance. In: Proceedings of the IEEE Wireless Communications & Networking Conference, Hong Kong, China, 2007. 1673-1678
- [11] Bonuccelli M A, Lonetti F, Martelli F. Tree slotted aloha: a new protocol for tag identification in RFID networks. In: Proceedings of the 7th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, New York, USA, 2006. 6
- [12] Klaus F. RFID Handbook: fundamentals and applications in contactless smart cards and identification, 2nd ed. John Wiley & Sons, 2003
- [13] EPC radio-frequency identity protocols class-1 generation-2 UHF RFID protocol for communications at 860MHz-960MHz version 1.0.9, 2004
- [14] Draft protocol specification for a 900MHz Class 0 Radio Frequency Identification Tag, 23 Feb 2003
- [15] Law C, Lee K, Siu K Y. Efficient memoryLess protocol for tag identification. <http://www.autoidlabs.org/uploads/media/MIT-AUTOID-TR-003.pdf>; MIT, 2000
- [16] Hush D R, Wood C. Analysis of tree algorithms for RFID arbitration. In: Proceedings of the IEEE International Symposium on Information Theory, Cambridge, MA, USA, 1998. 107
- [17] Yeh M K, Jiang J R. Adaptive k-Way splitting and pre-signaling for RFID tag anti-collision. In: Proceedings of the 33rd Annual Conference of the IEEE on IECON 2007. Taipei, China, 2007. 40-45
- [18] Cui Y H, Zhao Y P. Mathematical analysis for binary tree algorithm in RFID. In: Proceedings of the Vehicular Technology Conference, Singapore, 2008. 2725-2729
- [19] Chiang K W, Hua C Q, Yum T-S P. Prefix-randomized query-tree protocol for RFID systems. In: Proceedings of the IEEE International Conference on ICC 2006 proceedings, Istanbul, Turkey, 2006. 1653-1657
- [20] 谢衷洁. 普通统计学, 北京大学出版社, 2004. 97
- [21] Popovski P, Fitzek F H P, Prasad R. Batch conflict resolution algorithm with progressively accurate multiplicity estimation. In: Proceedings of the Discrete Algorithms and Methods for Mobile Computing and Communications ACM Workshop on Principles of Mobile Computing, Philadelphia, PA, USA, 2004. 31-40

The tag estimation based dynamic optimal multi-branch search anti-collision algorithm

Cui Yinghua, Zhao Yuping

(State Key Laboratory of Advanced Optical Communication Systems & Networks,
School of electronics engineering and computer Science, Peking University, Beijing 100871)

Abstract

Based on the dynamic binary search (DBS) algorithm for radio frequency identification (RFID) systems, the paper presents an improved multi-branch algorithm, then gives the method for calculating the optimal branch number through the analysis of the system efficiency of the multi-branch algorithm, and finally proposes the dynamic optimal multi-branch search (DOMBS) algorithm with the combination of a tag number estimation method. Both the theoretical analysis and the simulation results show that the DOMBS algorithm has the prominent performance advantage compared with the DBS algorithm. The system efficiency of the proposed algorithm is about 58%, while that of the dynamic binary search algorithm is only 50%.

Key words: radio frequency identification (RFID), anti-collision protocols, dynamic binary search (DBS), multi-branch, tag estimation