

ISO18000-6C 标准的防冲突改进算法*

An Improved Anti-collision Algorithm Based on ISO18000-6C Protocol

余 智, 蒋 泰, 朱廷生

SHE Zhi, JIANG Tai, ZHU Yan-sheng

(桂林电子科技大学计算机科学与工程学院, 广西桂林 541004)

(School of Computer Science & Engineering, Guilin University of Electronic Technology, Guilin, Guangxi, 541004, China)

摘要:对超高频 RFID 系统中 ISO18000-6C 标准使用的时隙 Aloha 算法进行改进,并用仿真试验来检验系统的性能。改进的算法运用切比雪夫不等式估计现场未识别的标签数目,根据最优帧长原理动态地调整帧时隙数和分群数。在标签数目相同情况下,改进算法后系统的吞吐率与信道利用率都有所增高。

关键词:防冲突算法 射频识别 帧时隙数

中图分类号: TP391.45 **文献标识码:** A **文章编号:** 1002-7378(2012)01-0011-05

Abstract: This research paper improves the slotted ALOHA anti-collision algorithm that is used in ISO18000-6C protocol, and the simulation experiment is used to test the performance of the algorithm. The number of the tags in the identification field of reader are estimated by using Chebyshev-Inequality. According to the optimal frame length principle, the number of slots and groups in every frame are dynamically adjusted. In the condition that the tags are the same amount, the throughput rate and the channel utilization rate of system increase after improving the algorithm.

Key words: anti-collision algorithm, radio frequency identification (RFID), number of slots

射频识别技术(RFID: Radio Frequency Identification)是一种非接触式自动识别技术^[1]。一个基本的 RFID 系统由读写器、标签和天线三部分组成。在同一个 RFID 系统中所有的标签都工作在相同的频率下,所以实际应用中,如果读写器的作用范围内存在多个标签,很容易出现多个标签同时向读写器发送数据的现象,即发生冲突。为了防止这种因冲突导致的信息丢失必须设计一定的防冲突算法。

在超高频 RFID 系统中,ISO18000-6C(以下简称 Type C)标准采用的基于概率的时隙随机防冲突算法,由于其具有优越的性能,正被逐步推广。但是该算法还存在一些不足,当标签的个数远远大于它

的帧时隙数时,将大大增加标签的冲突概率。为提高 RFID 系统识别能力,本文提出一种基于检测和分群思想的防冲突算法,并进行仿真试验。结果表明,该算法能有效提高系统的吞吐率。

1 Type C 标准中防冲突算法存在的问题

Type C 标准中的防冲突算法本质上是帧时隙算法。当读写器开始对一组标签进行盘存周期时,读写器发出 Query 命令(含有一个参数 Q,其取值范围为 1~15),参与的标签收到 Query 命令后,向其时隙计数器内载入一个随机数(取值范围 $0 \sim 2^Q - 1$)。选到零值的标签转移到应答状态,并立即应答;选到非零的标签转移到仲裁状态,并等待 QueryAdjust 或 QueryRep 命令。处于仲裁状态的标签接收到 QueryRep 命令时,其时隙计数器值减 1。当标签内时隙计数器值减为 0 时,标签转移到应答状态,并回答一个 RN16(16 位随机数)。当有两

收稿日期:2011-11-12

修回日期:2011-12-25

作者简介:余 智(1985-),男,硕士研究生,主要从事智能信息处理与嵌入式应用的研究。

* 广西科学研究与技术开发计划项目(桂科合 0992032-9)资助。

个或两个以上标签的时隙计数器值同时为0时,这些标签会同时对读写器进行应答并造成冲突。读写器检测到冲突发生后,发出相关命令,让冲突标签的时隙计数器值从0变到0x7FFF,并继续留在这轮盘存周期内,之后读写器再通过 QueryAdjust 命令设置新的 Q 参数来分散发生冲突的标签。当时隙计数器值不为0时,标签不对读写器进行应答,而是根据读写器的下一条 QueryAdjust 或 QueryRep 命令,执行时隙计数器值继续减1操作,或者根据新的 Q 参数值来再次载入另一随机数($0 \sim 2^Q - 1$)。已经被成功识别的标签,退出这轮标签盘存。处于仲裁或应答状态的标签接收 QueryAdjust 命令后,需要对读写器命令参数 Q 的进行调整(增加、减少或不变),然后在($0 \sim 2^Q - 1$)范围内选出一个随机数,置入它们的时隙计数器。这个识别过程一直继续下去,直到完成这轮盘存周期。 Q 值的调整采用了图1中的算法。图中 Q_{fp} 是参数 Q 的浮点表示,读写器对 Q_{fp} 取整得到 Q , 标签用 Q 作参数,在($0 \sim 2^Q - 1$)取值范围内,随机载入时隙计数器的值,以实现标签的高效率读取。图1中 $0.1 < C < 0.5$ 。当 Q 较大时, C 取较小值;当 Q 较小时, C 取较大值^[2]。

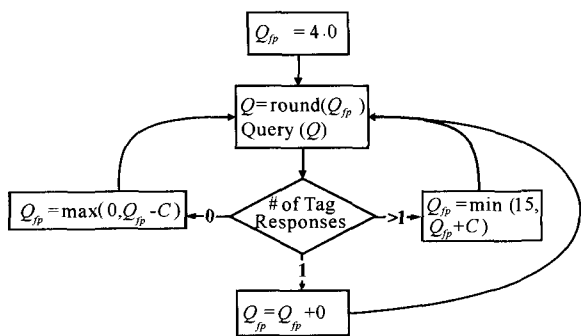


图1 Q 值的调整算法

上述 Type C 中的防冲突采用增减参数 C 值方法处理冲突问题,使得它在前帧未结束时需重新启动一个新帧来识别剩下的标签,在一定程度上能使系统自动地处于较佳的状态。但是,Type C 标准中的防冲突算法仍有以下几个问题:

(1) 这种机制采用相同的方法修改 Q 值而没有考虑“冲突应答”和“无应答”这两种情况,这样会降低标签的识别速度^[3]。

(2) 采用的 Q 算法可能会出现不当的 Q 值调整,这会导致过多的开销。

(3) Q 值不能无限增加,当标签的个数远远大于它的帧时隙数时,将大大增加标签的冲突概率。

2 防冲突算法改进的理论依据

2.1 最优帧长证明

在 Type C 标准采用的动态帧时隙 Aloha 防冲突算法中,当条件“一帧里的时隙数与待识别的标签数基本相同”满足时,系统将具有最高的效率。我们从基于识别时间和系统吞吐率这两个方面分别对其进行证明。

2.1.1 识别时间

帧时隙 Aloha 防冲突算法是基于伯努利(Bernoulli)试验的概率算法理论。在该算法的实现设计中,将帧分为 N 个时隙,假设读写器的有效范围内有 n 个待识别的标签。当这些标签收到由读写器发送过来的 Query 命令时,标签将会在这 N 个时隙中随机选择一个时隙,作为它的应答时间。根据伯努利试验,每个标签出现在任意一个时隙的概率为 $1/N$,不出现的概率则为 $(1 - 1/N)^{[4]}$ 。

同时由于各个待识别的标签相互独立,因此在某一时隙里同时出现 r 个标签的概率是为

$$P_n(r) = C_n^r \left(\frac{1}{N}\right)^r \left(1 - \frac{1}{N}\right)^{n-r} \quad (1)$$

当 $r = 0$ 时,即没有标签选中该时隙,此时的概率为

$$P_n(0) = C_n^0 \left(\frac{1}{N}\right)^0 \left(1 - \frac{1}{N}\right)^{n-0} = \left(1 - \frac{1}{N}\right)^n \quad (2)$$

当 $r = 1$ 时,即某一个时隙恰好出现一个标签的概率为

$$P_n(1) = C_n^1 \left(\frac{1}{N}\right)^1 \left(1 - \frac{1}{N}\right)^{n-1} \quad (3)$$

此时读写器能够正确识别标签。

当 $r \geq 2$ 时,即有多个(两个或两个以上)的标签同时选择某一时隙,发生该事件的概率为

$$P_n(r \geq 2) = \sum_{i=2}^n P_n(i) = 1 - P_n(0) - P_n(1) = 1 - \left(1 - \frac{1}{N}\right)^n - C_n^1 \left(\frac{1}{N}\right)^1 \left(1 - \frac{1}{N}\right)^{n-1} \quad (4)$$

因此,在一轮循环以后,能够正确读出标签的数目期望值为

$$E_s = N * P_n(1) = N * C_n^1 \left(\frac{1}{N}\right)^1 \left(1 - \frac{1}{N}\right)^{n-1} = N * n * \left(\frac{1}{N}\right) * \left(1 - \frac{1}{N}\right)^{n-1} = n * \left(1 - \frac{1}{N}\right)^{n-1} \quad (5)$$

系统识别效率 a 等于平均一轮循环中能够对标签进行识别的时隙数除目前一帧中的时隙总数,因此可以得到

$$a = \frac{E_s}{N} = \frac{n}{N} * \left(1 - \frac{1}{N}\right)^{n-1}. \quad (6)$$

为求效率最大值,对 a 求导,其中 n 为变量。可以得到

$$\begin{aligned} \frac{d(a)}{dn} &= \frac{d\left(\frac{n}{N} * \left(1 - \frac{1}{N}\right)^{n-1}\right)}{dn} = \left(1 - \frac{1}{N}\right)^{n-1} + \\ &n\left(1 - \frac{1}{N}\right)^{n-1} * \ln\left(1 - \frac{1}{N}\right) = \left(1 - \frac{1}{N}\right)^{n-1} \{1 + n * \\ &\ln\left(1 - \frac{1}{N}\right)\}, \end{aligned} \quad (7)$$

将其导数设为 0,得到

$$N = \frac{e^{\frac{1}{n}}}{e^{\frac{1}{n}} - 1}, \quad (8)$$

利用泰勒(Taylor)级数将其展开,可以得到

$$N \approx \frac{1 + \frac{1}{n}}{1 + \frac{1}{n} - 1} = n + 1. \quad (9)$$

这就从理论上证明:在一帧中,当时隙个数与待识别的标签数大致相等时,该系统的效率最大。

2.1.2 系统吞吐率最大的理论证明

系统吞吐率 S 是指在某一个时隙中,标签能被正确识别的概率和总概率的比值,表达式如下:

$$S = \frac{P_n(1)}{P_n(0) + P_n(1) + P_n(\geq 2)}. \quad (10)$$

由于 $P_n(0) + P_n(1) + P_n(\geq 2) = 1$,所以

$$S = P_n(1) = C_n^1 \left(\frac{1}{N}\right)^1 \left(1 - \frac{1}{N}\right)^{n-1}. \quad (11)$$

为了求得 S 的最大值,我们对 S 求 N 的导数,并将其导数设置为 0,由(7)式,(8)式和(9)式可知,当 $N \approx n + 1$ 时,吞吐率 S 将达到最大值,

$$S_{\max} = \left(1 - \frac{1}{N}\right)^{N-1}, \quad (12)$$

$$\lim S_{\max} = \left(1 - \frac{1}{N}\right)^{N-1} = \frac{1}{e} \approx 0.368. \quad (13)$$

因此,通过以上理论推导可以证明,当帧时隙个数与标签数目大致相同时,系统的吞吐率 S 将接近理论最大值 0.368。

2.2 改进算法中分群调整

动态帧时隙 Aloha 虽然能够根据标签的多少来动态地改变帧的大小,提高时隙的利用率,但是当周围有大量的标签时,帧的长度也变得很大,这对标签识别将会造成局部的延迟,并消耗更多的存储空

间。因此当有大量的标签同时出现时,需要将标签进行分群,缩小识别的范围。例如:当 Q 的值等于 7 时(即 $2^7 = 128$),标签的响应概率为 $2^{-7} \approx 0.0078$ 。当标签数远大于 128 时,所需时隙数将呈指数增长,标签的响应概率将大大降低。为了在标签数很大时仍能有效识别标签,通过分群方式限制每次响应的标签数来能效提高标签的响应概率。

在基于 Aloha 的算法中,影响系统性能最重要的参数是每一识别周期的帧长和响应标签数。特别是在动态帧时隙 Aloha 算法中,选择合适的帧长和分群将大幅改善算法的性能。改进的算法根据不同分组数(表 1)对应的吞吐率仿真图(图 2)来确定系统的最佳分群数,使系统的吞吐率达到最佳。分析可知将标签分成 m 组和 $m + 1$ 组时系统性能曲线交点处的标签数可由等式 $P_{n/m} = P_{n/(m+1)}$ 计算确定^[5],即

$$\begin{aligned} \frac{n}{m} * \left(\frac{1}{L}\right) * \left(1 - \frac{1}{L}\right)^{n/m-1} &= \frac{n}{m+1} * \\ \left(\frac{1}{L}\right) * \left(1 - \frac{1}{L}\right)^{n/(m+1)-1}. \end{aligned} \quad (14)$$

由(14)式可知,在需要调整分群数时对应的标签数为

$$n = m * (m + 1) * \frac{\ln(m + 1) - \ln(m)}{\ln(L) - \ln(L - 1)}. \quad (15)$$

即当标签数达到 n 时需要重新调整系统分群数。由此可以得到不同数量的标签对应的最优帧长及分群数的列表,如表 1 所示(初始帧长 $L_0 = 4$,系统允许的最大帧长 $L_{\max} = 128$)。

表 1 最优帧长和分群数

未识别的 标签数量 n	最优帧 长 L	对应分 群数 m	对应的 Q 值
1-4	4	1	2
5-11	8	1	3
12-22	16	1	4
23-44	32	1	5
45-88	64	1	6
89-177	128	1	7
178-310	128	2	7
311-440	128	3	7
441-569	128	4	7
570-697	128	5	7
.....

根据图 2 中不同帧长系统性能的变化曲线,确定相邻分群性能曲线交点处的标签数,该标签数即为调整分群数的临界点,若待识别标签数达到该临

界点时就调整相应分群数大小。该方案可以使系统达到理论上的最优性能。

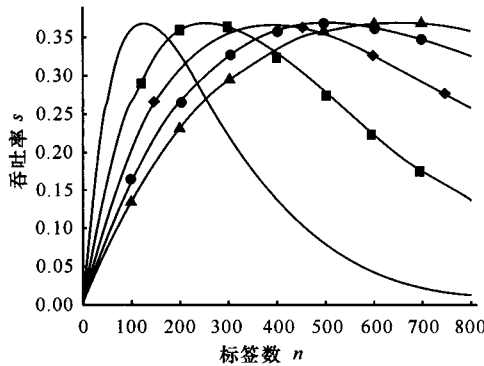


图2 不同分群数对应的吞吐量

—: $m = 1$; \blacksquare : $m = 2$; \bullet : $m = 3$; \blacklozenge : $m = 4$; \blacktriangle : $m = 5$ 。

2.3 改进算法标签数目估计

动态帧时隙 Aloha 算法只在一帧结束后对帧的长度进行调整,动态地调整帧的尺寸。为了合理的设定帧的长度,必须估算总时隙的值,并以此来估计最佳帧的长度。在估算时隙值之前,需要估计标签的数目。而算法中对下一帧长度的估计可以转换为对剩余标签数目的估计。对标签的估计主要有最小值估计算法,泊松估计算法,碰撞概率估计算法,切比雪夫(Chebyshev)不等式算法等。

由文献[6]可知,切比雪夫不等式无论在估计误差,还是消耗时隙总数上都是最优的。改进算法对标签数目的估计采用切比雪夫不等式算法。用数学符号表示即是,能使下面这个函数值最小的 n 即为标签数目。

$$\epsilon(N, c_0, c_1, c_2) = \min \left| \begin{pmatrix} a_0^{N,n} \\ a_1^{N,n} \\ a_{\geq 2}^{N,n} \end{pmatrix} - \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} \right| \quad (16)$$

其中, $a_0^{N,n}, a_1^{N,n}, a_{\geq 2}^{N,n}$ 分别是 $N * P_n(0), N * P_n(1), N * P_n(\geq 2)$ 代表的数学期望值,而 c_0, c_1, c_2 分别代表上一次读写器对标签进行识别后的空闲时隙数目,仅有一个标签的时隙数目和发生冲突的时隙数目。

3 改进的防冲突算法

基于以上分析,提出一种基于检测和分群思想的防冲突的改进算法,具体流程见图3和图4。改进算法在 Q 算法的基础上增加了检测机制,利用切比雪夫(Chebyshev)不等式对标签数目进行估计,进而实现对 Q 值的合理调整。同时该算法采用一种分群的思想,实现群间的随机避让、在群内进行冲

突检测和标签的仲裁。

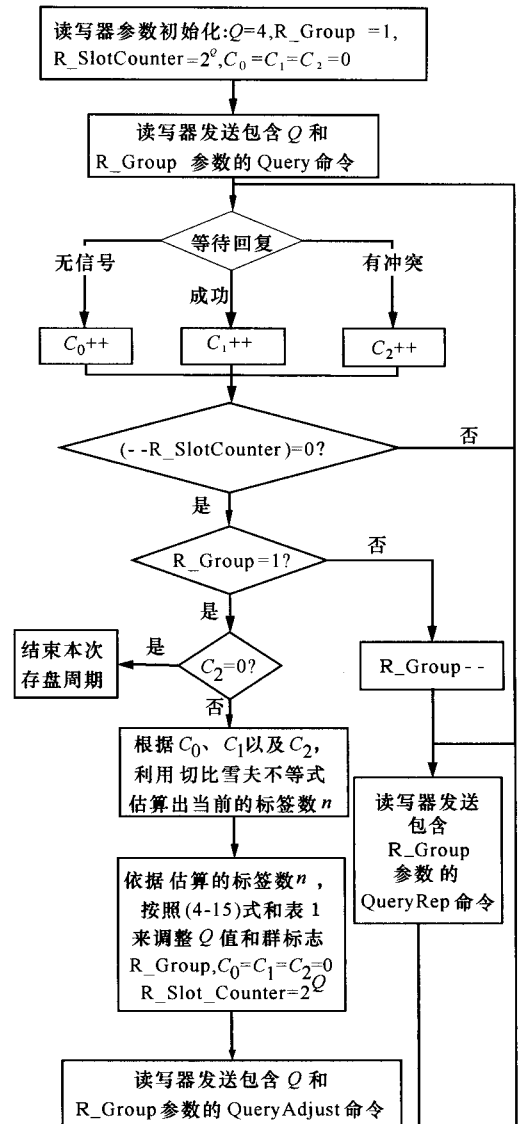


图3 读写器的程序流程

具体的描述如下:

(1)初始化参数: $Q = Q_{fp}, Q_{fp}$ 为 Q 的初始值,一般根据具体的应用环境进行设定,由于帧长太小不符合实际应用,故一般设 Q_{fp} 初始值为 4。读写器中的群标志 $R_Group = 1$,时隙计数器 $R_SlotCounter = 2^Q, C_0 = C_1 = C_2 = 0$,其中 C_0 代表空闲时隙数, C_1 代表成功时隙数, C_2 代表冲突时隙数。

(2)读写器发送包含 Q 和 R_Group 参数的 Query 命令,等待标签回复。

(3)参与的标签收到 Query 命令后进入仲裁态,并向其时隙计数器 $T_SlotCounter$ 内载入一个随机数(取值范围 $0 \sim 2^Q - 1$),同时其群标志 T_Group 载入一个随机数(取值范围 $1 \sim R_Group$)。当且仅当群标志 $T_Group = R_Group$ 时, T_Slot-

Counter 为零值的标签转移到应答状态,并立即应答;T_SlotCounter 为非零值的标签转移到仲裁状态,并等待下一条 QueryAdjust 或 QueryRep 命令,当群标志 T_Group! =R_Group 时,标签保持不变并等待下一条 QueryAdjust 或 QueryRep 命令。

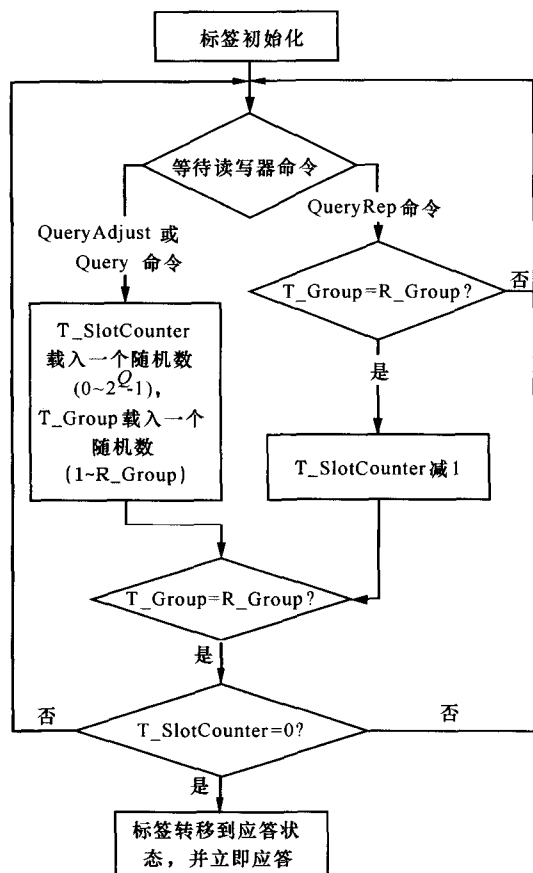


图 4 标签的程序流程

(4)读写器接到的回复可能有以下 3 种情形:

- (A) 冲突:冲突时隙数加一,即 $C_2 ++$;
- (B) 成功接收:成功时隙数加一,即 $C_1 ++$;
- (C) 无信号:空闲时隙数加一,即 $C_0 ++$ 。

任意情况下,读写器时隙计数减一,即 $R_SlotCounter--$ 。当 $R_SlotCounter! =0$ 时,读写器发送包含 R_Group 参数 QueryRep 命令,而当 $R_SlotCounter=0$ 时可能有以下 3 种情况:

- (D) $R_Group=1$,同时 $C_2=0$,则结束本次盘存周期;
- (E) $R_Group=1$,同时 $C_2 > 0$,则转到步骤(5);
- (F) $R_Group>1$,则 $R_Group--$,同时读写器发送包含 R_Group 参数 QueryRep 命令。

(5)读写器根据上一帧的识别情况,得出的空闲时隙数 C_0 、成功识别的时隙数 C_1 和产生冲突的时

隙数 C_2 ,通过切比雪夫(Chebyshev)定律(16)式,估算出当前的待识别的标签数目 n 。然后根据估算出标签个数 n ,应用(15)式和表 1 来调整 Q 值和群标志 R_Group 。同时 $C_0 = C_1 = C_2 = 0$, $R_SlotCounter=2^Q$ 。系统的最大帧时隙数是 128 ($2^7 = 128$,即 Q 值最大为 7)。读写器发送包含调整后的 Q 和 R_Group 参数的 QueryAdjust 命令,等待标签回复。

(6)处于仲裁状态的标签接收到 QueryRep 命令后,当且仅当群标志 $T_Group=R_Group$ 时,其时隙计数器 $T_SlotCounter$ 减 1, $T_SlotCounter$ 为零值的标签转移到应答状态,并立即应答; $T_SlotCounter$ 为非零值的标签转移到仲裁状态,并等待下一条 QueryAdjust 或 QueryRep 命令,当群标志 $T_Group! =R_Group$ 时,标签保持不变并等待下一条 QueryAdjust 或 QueryRep 命令。

(7)处于仲裁或应答状态的标签接收 QueryAdjust 命令后,在 $(0 \sim 2^Q - 1)$ 范围内选出一个随机数,置入它们的时隙计数器 $T_SlotCounter$,同时其群标志 T_Group 载入一个随机数(取值范围 $1 \sim R_Group$)。当且仅当群标志 $T_Group=R_Group$ 时, $T_SlotCounter$ 为零值的标签转移到应答状态,并立即应答; $T_SlotCounter$ 为非零的标签转移到仲裁状态,并等待下一条 QueryAdjust 或 QueryRep 命令,当群标志 $T_Group! =R_Group$ 时,标签保持不变并等待下一条 QueryAdjust 或 QueryRep 命令。这个识别过程一直继续下去,直到完成这轮盘存周期。

在算法中,假设 R_Group 的最大值为 1024,则该算法最多能识别 1024 个群,而每个群最大能仲裁的标签数目为 $2^7 = 128$,则本算法能仲裁的标签数理论上限是 $1024 \times 128 = 2^{17} = 131072$ 。

4 仿真试验

在 MATLAB R2007a 下对改进后的算法进行了仿真(图 5)。由图 5 可以看出,标签的数量从 1 到 1000 变化,当标签数小于 100 时,原 Type C 中的算法的吞吐率明显比改进算法的吞吐率要低,随着标签数量的增加,算法吞吐率会有小幅的波动,而改进算法的吞吐率趋于一个稳定值。因为改进算法采用了检测机制,能较好地预测未识别标签的数量,从而能更好地实时分配 Q 值,即较好的时隙数。同时

(下转第 27 页)

匹配,还能进行一对多的服务匹配,把语义关联的多个服务组合起来满足一个服务请求,与传统的一对多匹配不同的是我们的方法无需创建过程模型,通过匹配度计算对匹配结果进行优化,在查准率、查全率和查询效率之间得到有效平衡。在后继的工作中,我们将通过更丰富的实验对该方法进行进一步验证,并对匹配度计算方法做深入探讨,考虑更丰富的服务语义关联优化。

参考文献:

- [1] Mcilraith S A, Son T C, Zeng H L. Semantic web services[J]. IEEE Intelligent Systems Special Issue on the Semantic Web, Mar-Apr, 2001, 16(2):46-53.
- [2] Fernandez A, Polleres A, Ossowski S. Towards fine-grained service matchmaking by using concept similarity[C]//Proc of the 1st Int Joint Workshop SMR2 on Service Matchmaking and Resource Retrieval, Berlin: Springer, 2007:31-45.
- [3] Gunay A, Yolum P. Structural and semantic similarity metrics for web service matchmaking[C]//Proc of EC-Web, Berlin: Springer, 2007:129-138.
- [4] Sirin E. Combining descriptin logic reasoning with AI matching for composition of Web service[D]. College

- Park; University of Maryland, 2006.
- [5] Benatallah B, Hacid M S, Leger A, et al. On automating Web service discovery [J]. The International Journal on Very Lange Data Bases, 2005, 14(1):84-96.
- [6] 李瑞,蒋翠清. 基于接口匹配的语义 Web 服务自动组合研究[J]. 情报理论与实践, 2009, 32(8):109-112.
- [7] 骆琴,张永胜,聂学武,等. 一种基于语义 Web 服务的匹配算法研究 [J]. 计算机技术与发展, 2011, 21(2):136-139.
- [8] Zeng L, Benatallah B, Ngu A, et al. QoS-aware middleware for web services composition[J]. IEEE Transactions on Software Engineering, 2004, 30(5):311 - 327.
- [9] Alrifai M, Risse T. Combining global optimization with local selection for efficient QoS-aware service composition [C]. WWW Track: Web Engineering / Session: Service Oriented Development, 2009:881-890.
- [10] Caragea D, Tanveer S M. Semantic API matching for automatic service composition may[C]. Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters, 2004:436-437.

(责任编辑:尹 闯)

(上第 15 页)

改进算法应用了分群思想,当标签的个数远远大于时隙数时,通过分群来大幅减小冲突的概率。所以改进的算法较好地解决了原 Type C 标准算法中分配的时隙数太多导致的响应概率呈指数降低的问题。

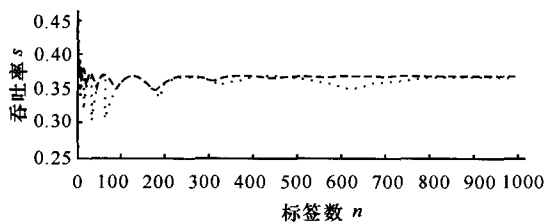


图 5 两个算法的吞吐量比较
·····:原算法;----:改进后的算法。

参考文献:

- [1] ISO/IEC. Information technology-radio frequency identification for item mangement-part 6: parameters for air interface communications at 860 MHz to 960MHz, amendment 1: extension with type C and update of

types A and B[S]. ISO/IEC 180002-6:2004/A md 1: (E), 2006.

- [2] EPCglobal Inc. EPCTM radio-frequency identity protocols class 1 generation-2 UHF RFID protocol for communications at 860-960MHz version 109[S]. EPCglobal IncTM, 2005.
- [3] 蒋泰,高广尚. ISO18000-6 Type C 标准中防冲突技术的改进[J]. 计算机工程与应用, 2010(20):42-04.
- [4] 陈家鼎,郑忠国. 概率与统计[M]. 北京:北京大学出版社, 2007.
- [5] 禹士朋, UHF 射频识别系统的防冲突算法研究[D]. 郑州:郑州大学, 2010.
- [6] XiaoFan, InChanSong, Kyung Hi Chang. Gen2-based hybrid tag anti-collision Q algorithm using chebyshev's inequality for passive RFID systems[EB/OL]. IEEE press, 2010. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4699508.

(责任编辑:尹 闯)