

本科毕业设计（论文）外文翻译

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓名** | 吴斌 | **学号** | 18408020129 | **专业及班级** | 网络工程1801 |
| 外文译文内容 | | | | | |
| **大量电子标签ID信息的高速读出方法** 摘要 RFID 等电子标签有望创造出传统条码无法实现的新服务。具体而言，在分销系统中，需要同时读出嵌入在产品中的大量电子标签的方法，以降低成本和时间。在本文中，我们提出了称为响应概率控制 (RPC) 的新方法来实现这一要求。在 RPC 中，阅读器首先向其访问区域内的电子标签发送 ID 请求。只有当其他标签没有响应时，它才能成功读取标签上的信息。为了提高读出效率，阅读器根据标签的数量适当控制响应概率。但是，这种方法不能完全避免多个响应的冲突。发生冲突时，ID 信息会丢失。为了减少丢失的数据量，我们将 ID 注册过程分为两个步骤。读者首先按照上述方法收集原始ID的前一部分，称为时间ID。得到时间ID后，根据时间ID依次收集ID的后半部分，称为剩余ID。请注意，我们根据访问区域中标签的数量来确定时间ID的位数，以便可以区分每个标签。通过模拟实验，我们从读出效率方面评估 RPC。仿真结果表明，RPC比传统方法读取1000个ID为128位的电子标签的方法提高了1.17倍。  **关键词**：RFID，接入区域，时间ID，剩余ID 1.引言 条形码一直是区分多个对象的最流行的技术。但是，它不能同时读取多个对象。射频识别 (RFID) 系统 1 有望创造条码系统无法实现的新服务。 RFID系统由一个阅读器和多个电子标签组成，每个电子标签都嵌入一个物体中。 RFID系统的一个重要特点是为电子标签供电。无源标签没有自己的电源，因此无源标签运行所需的所有电力都必须用于读取器的电场/磁场。相反，有源标签包含一个电池，可为操作提供全部或部分电力。  与条码系统相比，RFID系统具有两大关键优势。首先，电子标签具有能够存储比条形码更多的数据的存储器。接下来，即使它们之间有障碍物，阅读器也可以通过无线通信从电子标签中收集信息。尽管具有吸引人的优势，但众多电子标签的同时读出方法一直是一个悬而未决的问题。  多路访问众多电子标签可以在各个工业领域实现新的服务：制造领域的库存和物流控制，金融领域的票据或证券的防伪等。但是，一个阅读器不能同时读取多个标签的信息。一次，因为它们的响应之间发生了冲突。  为了防止这种碰撞，需要一种防碰撞机制。在上述服务中，将需要大量的电子标签。由于无源标签的成本相对低于有源标签，因此考虑引入成本而采用无源标签。由于无源标签的处理能力和天线性能有限，无源标签通过相互感应来避免碰撞是不切实际的。因此，阅读器必须完成一种防碰撞机制，以便它可以从多个标签中一个一个地读取信息。  动态框架开槽 ALOHA (DFSA)4、5 是防碰撞机制之一。 DFSA 基于 Slotted ALOHA.6, 7 在 DFSA 中，阅读器构造一个由多个时隙组成的帧。在一个帧中，每个标签在随机选择的插槽中将自己的信息发送给阅读器。如果一个时隙只有一个标签响应，则阅读器可以读取信息。否则，会发生冲突或超时。为了减少后一种情况，帧的大小在 DFSA 中是最重要的。为了解决这个问题，阅读器通过估计其访问区域中的标签数量来确定帧的适当大小。但是，DFSA 存在一些问题。在 DFSA 中，所有时隙必须具有相同的长度。如果每个电子标签的数据大小不同，则每个槽的长度变大，以便阅读器可以从访问区域中数据大小最大的电子标签读取信息。此外，即使没有标签在该时隙响应，阅读器也不能跳过时隙来保持标签之间的同步。最后，当发生碰撞时，所有信息都会丢失。当时隙大小变大时，这会降低系统性能。  在本文中，我们提出了 RPC（响应概率控制）方法，该方法能够有效地从多个电子标签中收集信息，而与电子标签的数量无关。在 RPC 中，阅读器首先向其访问区域内的电子标签发送 ID 请求。如果其中一个只响应请求，它会成功读取标签上的信息。为了避免多个标签响应同一个ID请求，阅读器根据标签的数量适当地控制电子标签响应ID请求的概率（即响应概率）。此外，为了减少因碰撞而丢失的 ID 数量，我们将 ID 注册过程分为两个步骤。读者首先根据使用响应概率的方法收集原始 ID 的前一部分，称为时间 ID。然后，它从获得的时间 ID 指定的节点收集原始 ID 的后半部分，称为剩余 ID。我们首先通过数学分析对 RPC 进行定量评估。然后，我们通过几个模拟实验比较了 RPC 和 DFSA。请注意，为了简单起见，我们排除了耦合效应和捕获效应，就像在 DFSA 中一样。  第 2 节介绍了 RPC 的细节。接下来，我们进行模拟实验来评估第 3 节中的 RPC。  最后，我们在第 4 节中描述结论和未来的工作。 **2 建议方法**概述 我们首先解释一下我们的防冲突协议 RPC 的概述。在 RFID 系统中，电子标签维护其 ID 和有关嵌入标签的产品的数据。阅读器首先从其访问区域中的所有标签中收集 ID，然后从中收集数据。由于它事先并不知道该区域中存在哪个ID，所以它应该以概率的方式一一收集它们的ID。虽然 DFSA 采用 Slotted ALOHA，但 RPC 控制电子标签响应 ID 请求的概率，即响应概率。阅读器向该区域内的所有标签发送包括响应概率的 ID 请求。每个标签以响应概率响应 ID 请求。因此，会出现三种情况：成功、超时和冲突。成功是只有一个标签响应 ID 请求的情况。如果没有标签响应 ID 请求，则阅读器检测到超时。当多个标签响应同一个 ID 请求时，阅读器会检测到冲突。我们假设阅读器可以使用错误检测代码检测碰撞。为了抑制发生超时或冲突情况，必须考虑访问区域中的标签数量来适当地确定响应概率。我们将在 2.3 小节中描述如何估计该区域中的标签数量，以及如何在 2.4 小节中确定和控制响应概率。  我们进一步将ID的注册过程分为两个步骤。阅读器首先使用上述概率方式从标签中收集时间 ID。时间ID由原始ID生成，以便阅读器可以区分其访问区域中的每个标签。在访问区域中区分标签所需的比特数往往小于原始ID的比特数。例如，如果时间 ID 分布良好，我们可以仅用 10 位区分一千个标签。如果阅读器可以获得一个时间ID，则直接向该时间ID指定的标签请求剩余ID。这种方法有两个优点。首先，我们可以减少由于冲突而丢失的比特数。接下来，标签可以通过接收对其剩余 ID 的请求来了解其临时 ID 是否已注册到阅读器。阅读器可以隐式放弃已经注册的标签将响应时间 ID 的后续请求。 ID注册过程的划分细节在2.5小节中给出。  最后，我们在 2.6 小节推导出 ID 注册的完成时间。 阅读器和电子标签之间交换的命令 在解释RPC的细节之前，我们首先介绍阅读器和电子标签之间交换的命令如下。  • 从阅读器发送到电子标签的命令  – 临时 ID 请求  该命令用于从电子标签中获取时间 ID。 它包括每个标签响应此命令的响应概率。  – 剩余 ID 请求  该命令用于从响应最后一个时间 ID 请求的电子标签中收集剩余 ID。  • 从电子标签发送到阅读器的命令  – 时间 ID 响应  此命令用于向阅读器发送时间 ID。 它包括一个错误检测代码，以便阅读器可以检测到碰撞。  – 剩余 ID 响应  ∗ 此命令用于向阅读器发送剩余 ID 估计标签数量 一般而言，当阅读器最初开始从它们收集信息时，阅读器无法知道访问区域中未注册到它的标签的数量。如前所述，读者应适当控制响应概率，以提高ID注册的成功率。在本小节中，我们提出了一种基于 ID 注册过程的先前结果来估计访问区域中未注册标签数量的方法。  ID注册过程的结果分为三种情况：成功、超时和冲突。成功表明阅读器可以适当估计该区域内未注册标签的数量。如果发生超时，我们怀疑估计值低于未注册标签的实际数量。相反，冲突表明估计超过了未注册标签的实际数量。估计算法的细节如下。  1.当阅读器最初开始信息收集时，它使用以下方法之一确定未注册标签的估计数量m。  阅读器将随机值设置为 m。  如果有监控访问区域的摄像头可用，则阅读器根据从摄像头获得的标签数量确定 m。  通过使用以下自适应机制，m 的初始值对系统性能不是那么重要。  2. reader发送temporal ID request时，根据最后一次ID注册的结果调整m  过程如下：  只有一个未注册标签响应最后一个临时ID 请求的成功案例。阅读器期望 m 大约等于标签的实际数量 n，并将 m 减 1。没有未注册标签响应最后一个临时ID 请求的超时情况。  读者怀疑 m 与 n 相比被低估了，然后设置 m ← m ∗ Cd。光盘是范围为 (0,1) 的控制参数。 Cd 代表精度之间的权衡估计和对 n 变化的适应性。  两个或多个未注册标签同时响应最后一个临时 ID 的冲突情况要求。  读者认为 m 比 n 被高估，然后调整如下：m ← m ∗ Ci。 Ci 是大于一的控制参数。与 Cd 一样，Ci 表示准确性和适应性之间的权衡。阅读器继续执行步骤 2，直到完成从访问区域中的所有标签收集 ID。 响应概率的决定 在本小节中，我们将描述读者如何根据 m 确定响应概率 Prsp 以提高 ID 注册的效率。 我们首先推导出当阅读器发送时间 ID 请求时每种情况（即成功、超时或冲突）发生的概率。 我们应该注意到，概率不是 m 而是 n 的函数，因为未注册标签的实际数量是 n。 没有标签不响应时间 ID 请求的概率 P0(n)，即超时，为  *P*0(*n*) = (1 *− Prsp*)*n.*  (1)  只有一个标签响应时间 ID 请求，即成功的概率 P1(n) 变为  *P*1(*n*) =*n C*1*Prsp*(1 *− Prsp*)*n−*1*.*  (2)  最后，两个或多个标签同时响应时间ID请求的概率P2+(n)如下。  *P*2+(*n*) = 1 *−* (1 *− Prsp*)*n −n C*1*Prsp*(1 *− Prsp*)*n−*1  (3)   * + 1. **P1(n)下限分析**    ID注册流程的划分 我们将原始ID的注册过程分为两个步骤：时间ID的注册和对应的剩余ID的注册。首先，阅读器向访问区域中的所有标签发送包含 Prsp 的临时 ID 请求。如果阅读器只收到一个标签的时间ID响应，则直接向该时间ID指定的标签发送剩余ID请求。通过尽可能减小时间 ID 的大小，我们可以减少超时和冲突所浪费的时间。  组成原始 ID 的位数通常为 64 或 128 位，而区分访问区域中的多个标签所需的位数似乎要少得多。例如，如果时间 ID 分布良好，我们可以仅用 10 位区分一千个标签。但是，在许多情况下，原始 ID 由分层位模式组成。这个特性使得原始 ID 的前一部分很难用作时间 ID，因为它可能会增加时间 ID 的大小。解决此问题的一种可能方法是预先加密原始ID，因为加密可能会降低原始ID的位模式的规律性。  接下来，我们讨论ID注册过程的划分在多大程度上提高了有效性。为简单起见，我们在下面的讨论中忽略了除法引起的开销。我们将 α 定义为时间 ID 的大小与原始 ID 的大小之比。从方程式。如图 8 所示，阅读器接收到的原始 ID Erd 的预期比率如下所示。 3 模拟实验 在本节中，我们通过与传统方法 DFSA 的比较，进行了几个模拟实验来评估 RPC 的性能。 系统性能通过两种标准进行评估。 一是阅读器从其访问区域中的所有标签收集 ID 所花费的读出时间。 另一个是 2.3 节提出的估计方法中对初始值的敏感性。 模拟设置 我们在以下模拟环境中评估数据速率为 26 Kbps 的 RFID 系统 (ISO15693)。我们先解释一下RPC的参数设置。命令的传输时间：临时 ID 请求和剩余 ID 请求，设置为 1 ms。时间 ID 和剩余 ID 的大小分别设置为 20 位和 180 位。结果，时间ID和剩余ID的传输时间分别为1毫秒和5毫秒。请注意，它包括划分原始 ID 所需的开销。超时设置为 1 毫秒，这意味着阅读器在发送时间 ID 请求后等待时间 ID 响应 1 毫秒。根据我们初步模拟实验的结果，我们将 Cd 和 Ci 设置为 0.9462 和 1.08。  接下来，我们描述DFSA的参数设置。与 RPC 一样，用于请求 ID 的命令的传输时间设置为 1 ms。时隙大小设置为 5 毫秒，比 RPC 中 ID 的总传输时间短 1 毫秒。 DFSA 有一个额外的命令来通知一个确认（ACK）给一个标签，这样标签就不会响应后续的 ID 请求。通过假设理想和现实情况，我们将附加命令的传输时间分别设置为 0 和 1 ms。在以下结果中，我们将理想情况称为 DFSA1，将另一种情况称为 DFSA2。 读出时间的评估   图 2 展示了当访问区域中标签的初始数量从 50 到 1000 时，RPC、DFSA1 和 DFSA2 的读出时间的变化。在这种情况下，我们假设一个理想的情况，阅读器可以精确估计初始标签数量。系统启动时访问区域中的标签。  如图 2 所示，无论采用何种方法，读出时间都会线性增加。然而，RPC 不断地克服 DFSA1 和 DFSA2。具体来说，RPC 的读取时间分别比 DFSA1 和 DFSA2 快 1.17 倍和 1.26 倍。在 DFSA1 和 DFSA2 中，阅读器每次超时都会浪费 5 ms。另一方面，RPC 通过引入对原始 ID 的划分，将时间浪费减少到 1 ms。此外，作为 DFSA 的现实版本的 DFSA2 需要额外的 1 毫秒来向响应最后一个 ID 请求的标签发送 ACK。在 RPC 中，剩余 ID 请求可以在收集剩余 ID 的同时发挥作用。  正如本文未显示的那样，我们还期望在访问区域中共存不同大小的 ID（例如 64 位和 128 位）的环境中，RPC 的读取时间会变得更短。相反，DFSA 不能提高读出时间，因为它必须设置时隙大小，以便在时隙中传输最大大小的 ID。  **表1.** RPC 中分析的读出时间与实验读出时间   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 标签的初始数量 | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000 | | 分析读出时间 (*tf* ) | 1.144 | 2.287 | 3.431 | 4.574 | 5.718 | 6.862 | 8.005 | 9.149 | 10.292 | 11.436 | | 实验读出时间 | 1.159 | 2.271 | 3.482 | 4.561 | 5.792 | 6.866 | 8.069 | 9.241 | 10.354 | 11.503 |   **表 2.** 读出时间与标签的初始估计数量   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 初始估计标签数量 | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000 | | DFSA1读出时间(s) | 14.55 | 14.57 | 14.21 | 14.46 | 13.63 | 14.12 | 13.81 | 13.53 | 13.36 | 13.05 | | RPC 读取时间(s) | 11.54 | 11.45 | 11.56 | 11.57 | 11.57 | 11.46 | 11.46 | 11.61 | 11.51 | 11.45 |   接下来，我们评估 2.6 小节中 tf 分析的有效性。考虑到第 3.1 小节中的模拟设置，将 T1 和 T2 设置为 2 和 6 ms。表 1 显示了当标签的初始数量以 100 为增量从 100 到 1000 变化时分析的读出时间 Tf 和实验读出时间。我们发现无论标签的初始数量如何，它们之间几乎没有差异。因此可精确计算读出时间。 估计方法中对参数设置的敏感性评估 在实际情况下，阅读器不一定知道访问区域中标签的初始数量。在本小节中，我们评估初始标签数量的估计误差在多大程度上增加了读出时间。我们将标签的初始数量设置为 1000。表 2 显示了当初始估计的标签数量以 100 为增量从 100 到 1000 变化时 RPC 和 DFSA1 的读出时间。  如表中所示。如图 2 所示，DFSA1 的读取时间最多增加 1.52 秒，而 RPC 的读取时间在最坏情况下仅增加 0.16 秒。由于 DFSA 是基于 Slotted ALOHA 的，它估计每帧开头的标签数量。标签的初始数量越大，估计误差对读出时间的影响越大。另一方面，RPC 对每个时间 ID 请求进行估计，因此与 DFSA 相比，它可以提高估计的准确性。我们还发现 RPC 对初始估计误差的程度不敏感，因为 RPC 的读出时间几乎没有变化。实际上，RPC 的读出时间方差为 0.0032，远小于 DFSA1，即 0.257。 4 结论 在本文中，我们提出了 RPC，它是一种高速读取大量电子标签 ID 信息的方法。 RPC由三种方法组成。首先，我们讨论了阅读器如何正确估计其访问区域中的标签数量。然后，我们以分析的方式在估计的基础上推导出响应概率的最优值。最后，我们介绍了将 ID 注册过程分为两个步骤，以缩短读出时间并减少因碰撞而丢失的数据量。通过多次模拟实验，我们通过与DFSA的比较来评估 RPC 的有效性。具体来说，当初始标签数为 1000 时，RPC 的读取时间比DFSA的读取时间快 1.17 倍。  作为未来的研究工作，我们在嵌入电子标签的对象以一定速率进入和离开访问区域的环境中进一步评估 RPC 和 DFSA。在物流系统中使用的带式输送机系统就是一个这样的例子。在这种情况下，阅读器应该经常估计访问区域中的标签数量，以适应系统条件的变化。在这种情况下，我们预计 RPC 比 DFSA 更合适。 参考文献 **[1]** K. Frinkenzeller，RFID 手册，John wiley and Sons，2003 年（第二版）。  **[2]** R. Want，“RFID 技术简介”，IEEE Pervasive Computing 5，第 25-33 页，2006 年。  **[3]** A. Juels，“Rfid 安全性和隐私：一项研究调查”,IEEE 通讯选定领域期刊，2006 年。  **[4]** J.-R. Cha 和J.-H. Kim,，“用于快速识别 rfid 系统的新型防碰撞算法”，载于 ICPADS 2005 年，第 2 期，第 63-67 页，2005 年。  **[5]** H.-S. Choi, J.-R. Cha, 和 J.-H. Kim, “无处不在的身份识别系统中的快速无线防碰撞算法”，IEEE 车辆技术会议 6，第 4589-4592 页，2004 年。  **[6]** T. N. SAADAWI 和 A. EPHREMIDES，“具有大量缓冲用户的时隙 aloha 的分析、稳定性和优化”，IEEE自动控制汇刊，第 680–689 页，1981 年。  **[7]** Y. Yang and T.-S. P. Yu, “开槽 aloha 和 csma 的延迟分布”，IEEE Transactions on Communications 51，第 1846-1857 页，2003。 | | | | | |
| 原文内容 | | | | | |
| High-speed readout method of ID information on a large amount of electronic tags  Wataru Nagate *a* and Masahiro Sasabe *b* and Hirotaka Nakano *b*  *a*Graduate School of Information Science and Technology, Osaka University 1-32 Machikaneyama-cho, Toyonaka-shi Osaka 560-0043, Japan;  *b*Cybermedia Center, Osaka University  1-32 Machikaneyama-cho, Toyonaka-shi Osaka 560-0043, Japan ABSTRACT An electronic tag such as RFID is expected to create new services that cannot be achieved by the traditional bar code. Specifically, in a distribution system, simultaneous readout method of a large amount of electronic tags embedded in products is required to reduce costs and time. In this paper, we propose novel methods, called Response Probability Control (RPC), to accomplish this requirement. In RPC, a reader firstly sends an ID request to electronic tags in its access area. It succeeds reading information on a tag only if other tags do not respond. To improve the readout efficiency, the reader appropriately controls the response probability in accordance with the number of tags. However, this approach cannot entirely avoid a collision of multiple responses. When a collision occurs, ID information is lost. To reduce the amount of lost data, we divide the ID registration process into two steps. The reader first gathers the former part of the original ID, called temporal ID, according to the above method. After obtaining the temporal ID, it sequentially collects the latter part of ID, called remaining ID, based on the temporal ID. Note that we determine the number of bits of a temporal ID in accordance with the number of tags in the access area so that each tag can be distinguishable. Through simulation experiments, we evaluate RPC in terms of the readout efficiency. Simulation results show that RPC can accomplish the readout efficiency 1.17 times higher than the traditional method where there are a thousand of electronic tags whose IDs are 128 bits.  **Keywords:** RFID, access area, temporal ID, remaining ID 1. INTRODUCTION Bar code has been the most popular technique to distinguish multiple objects. However, it cannot accomplish to read multiple objects simultaneously. Radio Frequency Identification (RFID) system 1 is expected to create new services that cannot be achieved by the bar code system. The RFID system consists of a reader and multiple electronic tags each of which is embedded in an object. One important feature of RFID system is the power supply to the electronic tag. Passive tags do not their own power supply, and therefore all power required for the operation of a passive tag must be drawn for the electrical/magnetic field of the reader. Conversely, active tags incorporate a battery that supplies all or part of the power for the operation.  Compared with the bar code system, the RFID system has two kinds of crucial advantages[2][3]. First, an electronic tag has a memory which enables to store much data than a bar code. Next, a reader can collect information from an electronic tag via a wireless communication even if there are obstacles between them. Despite of the attractive advantages, simultaneous readout method of numerous electronic tags has been an open issue.  Multi-access to numerous electronic tags can achieve new services in various industrial fields: stock and physical distribution controls in manufacturing fields, anti-counterfeit of bills or securities in financial fields, etc. However, a reader cannot simultaneously read information from multiple tags at once because a collision occurs among responses from them.  To prevent such collisions, an anti-collision mechanism is required. In the above mentioned services, numerous electronic tags will be needed. Since the cost of a passive tag is relatively smaller than that of an active tag, the passive tags are employed taking into account the introduction cost. Because the processing power and the antenna performance of the passive tag are limited, it is impractical for passive tags to avoid the collisions by sensing career each other. Consequently, the reader must accomplish an anti-collision mechanism such that it can read information from multiple tags one by one.  Dynamic Framed Slotted ALOHA (DFSA)4, 5 is one of the anti-collision mechanisms. DFSA is based on Slotted ALOHA.6, 7 In DFSA, the reader constructs a frame which consists of multiple slots. In a frame, each tag sends its own information to the reader at a slot randomly selected. If only one tag responds at one slot, the reader can read the information. Otherwise, a collision or timeout occurs. To reduce the latter case, the size of the frame is the most important in DFSA. To tackle this problem, the reader determines an appropriate size of a frame by estimating the number of tags in its access area. However, there are some problems in DFSA. In DFSA, all slots must have the same length. If the data size of each electronic tag is different, the length of each slot becomes large so that the reader can read information from an electronic tag whose data size is maximum in the access area. Furthermore, the reader cannot skip a slot to maintain the synchronization among tags even if none of the tags respond at the slot. Finally, all information is lost when a collision occurs. This deteriorates the system performance when the slot size becomes large.  In this paper, we propose RPC (Response Probability Control) method that enables to effectively gather information from multiple electronic tags independent of the number of electronic tags. In RPC, a reader firstly sends an ID request to electronic tags in its access area. It successes reading information on a tag if one of them only responds to the request. To avoid that multiple tags respond to the same ID request, the reader appropriately controls a probability that an electronic tag responds to an ID request (i.e., response probability) in accordance with the number of tags. In addition, to reduce the amount of ID lost by a collision, we divide the ID registration process into two steps. The reader first collects the former part of an original ID, called temporal ID, based on the approach using the response probability. Then, it gathers the latter part of the original ID, called remaining ID, from the node designated by the obtained temporal ID. We first make quantitative evaluations of RPC by mathematical analyses. Then, we compare RPC with DFSA through several simulation experiments. Note that we exclude coupling effect and capture effect for simplicity as in DFSA.  Section 2 presents the details of RPC. Next, we conduct simulation experiments to evaluate RPC in section3.  Finally, we describe conclusion and future works in section 4. 2 PROPOSED METHODOverview We first explain the overview of our anti-collision protocol, RPC. In a RFID system, an electronic tag maintains its ID and data about the product in which the tag is embedded. A reader first collects the IDs from all tags in its access area, and then gathers data from them. Since it does not know which ID exists in the area in advance, it should collect the IDs of them one by one with a probabilistic way. Although DFSA employs Slotted ALOHA, RPC controls a probability that an electronic tag responds to an ID request, i.e., response probability. The reader sends an ID request including a response probability to all tags in the area. Each tag responds to the ID request with the response probability. As a result, three kinds of cases occur: success, timeout, and collision. Success is the case that only one tag responds to the ID request. If no tag responds to the ID request, the reader detects timeout. The reader detects a collision when more than one tag responds to the same ID request. We assume that the reader can detect the collision using error detecting code. To suppress that timeout or collision case occurs, the response probability must be appropriately determined taking into account the number of tags in the access area. We describe how to estimate the number of tags in the area in subsection 2.3 and how to determine and control the response probability in subsection 2.4.  We further divide the registration process of an ID into two steps. The reader first collects a temporal ID from a tag with the above mentioned probabilistic way. The temporal ID is generated by the original ID so that the reader can distinguish each tag in its access area. The number of bits required to distinguish tags in the access area is often smaller than that of the original ID. For example, we can distinguish a thousand tags with only 10 bits if the temporal IDs are well distributed. If the reader can obtain a temporal ID, then it directly requests a remaining ID to the tag designated by the temporal ID. This approach has two advantages. First, we can reduce the number of bits lost due to a collision. Next, a tag can know whether its temporal ID is registered to the reader by receiving a request to its remaining ID. The reader can implicitly abandon that a tag already registered will respond to succeeding requests of temporal IDs. The details of the division of the ID registration process are given in subsection 2.5.  Finally, we derive the completion time for the ID registration in subsection 2.6. Commands exchanged between a reader and electronic tags Before explaining the details of RPC, we first introduce commands exchanged between a reader and electronic tags as follows.   * Commands sent from a reader to electronic tags   + Temporal ID request   This command is used to obtain a temporal ID from an electronic tag. It includes a response probability with which each tag responds to this command.  *·*   * + Remaining ID request   This command is used to collect the remaining ID from the electronic tag that responded the last temporal ID request.   * Commands sent from an electronic tag to a reader   + Temporal ID response   This command is used to send a temporal ID to a reader. It includes an error detecting code so that the reader can detect a collision.   * + Remaining ID response   ∗ This command is used to send a remaining ID to a reader. Estimation of the number of tags In general, a reader cannot know the number of tags in the access area, which are not registered to it, when it initially starts to gather information from them. As mentioned before, the reader should appropriately control the response probability to improve the success ratio of ID registration. In this subsection, we propose a method to estimate the number of unregistered tags in the access area based on the previous result of the ID registration process.  The result of the ID registration process is categorized in three cases: success, timeout, and collision. Success indicates that the reader can appropriately estimate the number of unregistered tag in the area. If timeout occurs, we suspect that the estimation is lower than the actual number of unregistered tags. On the contrary, a collision indicates that the estimation exceeds the actual number of unregistered tags. The details of the estimation algorithm are following.  1.When a reader initially starts the information gathering, it determines the estimated number of unregistered tags, *m*, by using one of the following methods.  The reader sets a random value to *m*.  If a camera monitoring the access area is available, the reader determines *m* based on the number of tags that obtained from the camera.  By using the following adaptive mechanism, the initial value of *m* is not so critical to the system perfor- mance.  2. When a reader sent a temporal ID request, it adjusts *m* based on the result of the last ID registration process as follows. *•* Success case in which only one unregistered tag responded to the last temporal ID request. **–** The reader expects that *m* was approximately equal to the actual number of tags, *n*, and reduces *m* by one. *•* Timeout case in which no unregistered tag responded to the last temporal ID request. **–** The reader suspects that *m* is underestimated compared with *n*, then it sets *m ← m ∗ Cd*. *Cd* is a control parameter that ranges (0,1). *Cd* represents the trade-off between accuracy of the estimation and adaptability to changes of *n*. *•* Collision case in which two or more unregistered tags simultaneously responded the last temporal ID request. **–** The reader imagines that *m* is overestimated than *n*, then it adjusts as follows: *m ← m ∗ Ci*. *Ci* is a control parameter that is larger than one. As in *Cd*, *Ci* indicates the trade-off between the accuracy and the adaptability.  The reader continues Step 2 until it finishes collecting IDs from all tags in the access area.   * 1. **Decision of response probability**   In this subsection, we describe how the reader determines a response probability *Prsp* based on *m* to improve the efficiency of ID registration. We first derive a probability that each case, i.e., success, timeout, or collision, occurs when the reader sends a temporal ID request. We should here note that the probability is a function of not *m* but rather than *n* because the actual number of unregistered tags is *n*. The probability *P*0(*n*) that no tag does not respond to the temporal ID request, i.e, timeout, is  *P*0(*n*) = (1 *− Prsp*)*n.*  (1)  The probability *P*1(*n*) that only one tag responds to the temporal ID request, i.e., success, becomes  *P*1(*n*) =*n C*1*Prsp*(1 *− Prsp*)*n−*1*.*  (2)  Finally, the probability *P*2+(*n*) that two ore more tags simultaneously respond to the temporal ID request is as follows.  *P*2+(*n*) = 1 *−* (1 *− Prsp*)*n −n C*1*Prsp*(1 *− Prsp*)*n−*1  (3)   * + 1. **Analysis of lower bound of** *P*1(*n*)    Division of ID registration process We divide the registration process of an original ID into two steps: registrations of a temporal ID and the corresponding remaining ID. At first, a reader sends a temporal ID request which includes *Prsp* to all tags in the access area. If the reader receives a temporal ID response from only one tag, it directly sends a remaining ID request to the tag designated by the temporal ID. By reducing the size of a temporal ID as possible, we can alleviate the time wasted by timeout and collision.  The number of bits consisting an original ID is typically 64 or 128 bits while that required to distinguish multiple tags in the access area seems to be much smaller. For example, we can distinguish a thousand tags with only 10 bits if the temporal IDs are well distributed. However, an original ID consists of a hierarchical bit pattern in many cases. This feature makes it difficult to use the former part of the original ID as a temporal ID because it may increase the size of a temporal ID. One possible way to solve this problem is to encrypt an original ID in advance because the encryption may reduce the regularity of the bit pattern of the original ID.  Next, we discuss what extent the effectiveness is improved by the division of the ID registration process. For simplicity, we ignore the overhead caused by the division in the following discussion. We define *α* as the ratio of the size of a temporal ID to that of an original ID. From Eq. 8, the expected ratio of the original ID received at the reader, *Erd*, becomes as follows. 3 SIMULATION EXPERIMENTS In this section, we conduct several simulation experiments to evaluate the performance of RPC by comparing with the traditional method DFSA. The system performance is evaluated by two kinds of criteria. One is readout time that the reader spends collecting IDs from all tags in its access area. Another is the sensitivity to the initial value in the estimation method proposed in section 2.3. Simulation settings We evaluate in the following simulation environment taking into account the RFID system (ISO15693) in which data rate is 26 Kbps. We first explain the parameter settings of RPC. The transmission time of the commands: temporal ID request and remaining ID request, is set to 1 ms. The size of temporal ID and remaining ID is set to 20 and 180 bits, respectively. As a result, the transmission time of temporal ID and remaining ID become 1 and 5 ms. Note that it includes the overhead required for dividing the original ID. Timeout is set to 1 ms, which means that the reader waits for a temporal ID response for 1 ms after it sent a temporal ID request. We set *Cd* and *Ci* to 0.9462 and 1.08 in accordance with the results in our preliminary simulation experiments.  Next, we describe the parameter settings of DFSA. The transmission time of commands used to request an ID is set to 1 ms as in RPC. The slot size is set to 5 ms which is 1 ms shorter than the total transmission time of an ID in RPC. DFSA has an additional command to notify an acknowledgement (ACK) to a tag so that the tag will not respond the succeeding ID requests. By assuming the ideal and realistic situations, we set the transmission time of the additional command to 0 and 1 ms, respectively. We call the ideal case as DFSA1 and another as DFSA2 in the following results. 3.2 Evaluation of readout time Figure 2 illustrates the transitions of readout time of RPC, DFSA1, and DFSA2 when the initial number of tags in the access area varies from 50 to 1000. In this scenario, we assume an ideal situation where the reader can precisely estimate the initial number of tags in the access area when the system starts.  As shown in Fig. 2, the readout time linearly increases regardless of the methods. However, RPC constantly overcomes both DFSA1 and DFSA2. Specifically, the readout time of RPC is 1.17 and 1.26 times faster than that of DFSA1 and DFSA2, respectively. In DFSA1 and DFSA2, the reader wastes 5 ms every time timeout occurs. On the other hand, RPC reduce the waste of time to 1 ms by introducing the dividing of the original ID. Furthermore, DFSA2 that is a realistic version of DFSA requires extra 1 ms to send ACK to the tag that responded to the last ID request. In RPC, the remaining ID request can play the role while gathering the remaining ID.  As not shown in this paper, we also expect that the readout time of RPC becomes shorter in an environment where IDs of different size, e.g., 64 and 128 bits, coexists in the access area. On the contrary, DFSA cannot improve the readout time because it must set the slot size so that an ID of the maximum size can be transmit in the slot.  **Table 1.** analyzed readout time vs. experimental readouttime in RPC   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | initial number of tags | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000 | | analyzed readout time (*tf* ) | 1.144 | 2.287 | 3.431 | 4.574 | 5.718 | 6.862 | 8.005 | 9.149 | 10.292 | 11.436 | | experimental readout time | 1.159 | 2.271 | 3.482 | 4.561 | 5.792 | 6.866 | 8.069 | 9.241 | 10.354 | 11.503 |   **Table 2.** readout time vs. initial estimated number of tags   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | initial estimated number of tags | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000 | | readout time of DFSA1 (s) | 14.55 | 14.57 | 14.21 | 14.46 | 13.63 | 14.12 | 13.81 | 13.53 | 13.36 | 13.05 | | readout time of RPC (s) | 11.54 | 11.45 | 11.56 | 11.57 | 11.57 | 11.46 | 11.46 | 11.61 | 11.51 | 11.45 |   Next, we evaluate the validity of the analysis of tf in subsection 2.6. T1 and T2 are set to 2 and 6 ms by taking into account the simulation settings in subsection 3.1. Table 1 presents the analyzed readout time, Tf , and experimental readout time when the initial number of tags varies from 100 to 1000 in increments of 100. We find that there are almost no differences between them regardless of the initial number of tags. Thus, Eq. 14 can precisely calculate the readout time. 3.3 Evaluation of sensitivity to parameter setting in estimation method In an actual situation, the reader does not necessarily know the initial number of tags in the access area. In this subsection, we evaluate what extent the estimation error of the initial number of tags increases the readout time. We set the initial number of tags to 1000. Table 2 presents the readout time of RPC and DFSA1 when the initial estimated number of tags varies from 100 to 1000 in increments of 100.  As shown in Tab. 2, the readout time of DFSA1 increases 1.52 sec at the maximum while that of RPC increases only 0.16 sec in the worst case. Since DFSA is based on Slotted ALOHA, it estimates the number of tags at the beginning of each frame. The larger the initial number of tags is, the more the estimation error affects the readout time. On the other hand, RPC conducts the estimation for each temporal ID request, thus it can improve the accuracy of the estimation compared with DFSA. We also find that RPC is not sensitive to the degree of the initial estimation error because the readout time of RPC does not almost change. Actually, the variance of the readout time of RPC is 0.0032 that is much smaller than that of DFSA1, i.e., 0.257. 4 CONCLUSION In this paper, we proposed RPC that is a high-speed readout method of ID information on a large amount of electronic tags. RPC is composed of three kinds of methods. First, we discussed how the reader appropriately estimates the number of tags in its access area. Then, we derived the optimal value of the response probability based on the estimation in an analytical way. Finally, we introduced the division of the ID registration process into two steps to shorten the readout time and reduce the amount of data lost by a collision. Through several simulation experiments, we evaluated the effectiveness of RPC by comparing with that of DFSA. Specifically, the readout time of RPC becomes 1.17 times faster than that of DFSA when the initial number of tags is 1000.  As future research works, we further evaluate RPC and DFSA in an environment where objects embedded electronic tags enter and leave the access area at a certain rate. A belt conveyor system used in a physical distribution system is one such example. In such a case, the reader should frequently estimate the number of tags in the access area to adapt the changes of system conditions. We expect that RPC is more suitable than DFSA in that case. REFERENCES **[1]** K. Frinkenzeller, *RFID Handbook*, John wiley and Sons, 2003 (second edition).  **[2]** R. Want, “An introduction to rfid technology,” *IEEE Pervasive Computing* **5**, pp. 25–33, 2006.  **[3]** A. Juels, “Rfid security and privacy: A research survey,” *IEEE journal on selected areas in communica- tions* **24**, 2006.  **[4]** J.-R. Cha and J.-H. Kim, “Novel anti-collision algorithms for fast object identification in rfid system,” in *ICPADS 2005*, **2**, pp. 63–67, 2005.  **[5]** H.-S. Choi, J.-R. Cha, and J.-H. Kim, “Fast wireless anti-collision algorithm in ubiquitous id system,” *IEEE Vehicular Technology Conference* **6**, pp. 4589–4592, 2004.  **[6]** T. N. SAADAWI and A. EPHREMIDES, “Analysis, stability and optimization of slotted aloha with a fine number of buffered users,” *IEEE Transactions on AUTOMATIC CONTROL* , pp. 680–689, 1981.  **[7]** Y. Yang and T.-S. P. Yu, “Delay distributions of slotted aloha and csma,” *IEEE Transactions on Commu- nications* **51**, pp. 1846–1857, 2003. | | | | | |
| 审核意见： 无  **指导老师（签名）：**  2022年3月24日 | | | | | |