

## 物联网编码解析系统的体系结构和算法研究

于海宁<sup>1</sup>, 张宏莉<sup>1</sup>, 方滨兴<sup>1,2</sup>, 余翔湛<sup>1</sup>

(1. 哈尔滨工业大学 计算机网络与信息安全研究中心, 黑龙江 哈尔滨 150001; 2. 中科院计算技术研究所, 北京 100190)

**摘 要:** 在编码解析需求的基础上, 首先, 提出了一个应用在大规模动态场景下的物联网编码解析系统的体系结构, 它由多个相互独立且协作的编码解析服务组成。然后, 提出了一个基于改进的 SkipNet 的编码解析服务, 它不但继承了传统 DHT 覆盖网络的优点, 还支持区域自治。最后, 通过仿真实验表明, 该服务具有优良的性能。

**关键词:** 物联网; 产品编码; 系统体系结构; 编码解析服务; SkipNet

中图分类号: TP393

文献标识码: A

文章编号: 1000-436X(2012)Z2-0035-10

## Research on architecture and algorithm for code resolution system in the internet of things

YU Hai-ning<sup>1</sup>, ZHANG Hong-li<sup>1</sup>, FANG Bin-xing<sup>1,2</sup>, YU Xiang-zhan<sup>1</sup>

(1. Research Center of Computer Network and Information Security Technology, Harbin Institute of Technology, Harbin 150001, China;

2. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China)

**Abstract:** Based on a list of requirements for code resolution, a code resolution system architecture that is a distributed system of independent, collaborating code resolution services was presented. Meanwhile, an improved SkipNet based code resolution service named SkipNet-OCRS was proposed. It not only inherits DHT's traditional advantages, but also supports administrative control and autonomy. Simulation experiments verify the excellent performance of SkipNet-OCRS.

**Key words:** internet of things; product code; system architecture; code resolution service; SkipNet

### 1 引言

随着无线通信技术和嵌入式技术的发展, 物联网已经成为工业界和学术界共同关注的热点问题<sup>[1]</sup>。物理实体的产品编码是其在物联网中的主要标识之一, 然而, 产品编码承载的信息极少, 大量的与物理实体相关的信息资源往往被广泛的存储在分布式的异构的信息服务(IS, information service)中, 这些信息资源与编码相关联。用户需要一种发现机制来及时获取兴趣实体的相关信息资源, 即使他只持有兴趣实体的编码。编码解析系统旨在为用

户提供与兴趣实体相关的信息资源的发现服务, 实现兴趣实体的编码到相关信息资源解析过程<sup>[2]</sup>。在大规模动态的应用场景下, 为确保物理实体的相关信息资源都能被高效、准确和安全定位以及查询, 物联网需要完善的编码解析系统的支撑。物联网编码解析系统面临的挑战多数源于它需要管理维护的海量数据和运营的异构解析服务。海量数据主要指编码解析记录的数量巨大, 继而会在组织存储, 权限管理, 并发请求响应等方面引发一系列的急需解决的问题。独立的异构的解析服务是构成解析系统的主要部分, 它的异构性体现在运营商、参与公

收稿日期: 2012-06-28

基金项目: 国家自然科学基金资助项目(61073194, 61173144)

**Foundation Item:** The National Natural Science Foundation of China (61073194, 61173144)

司、体系结构、支持的编码标准等多个方面,异构性会导致解析服务互不相容、互不协作,最终形成一个信息孤岛。

本文在充分分析总结物联网编码解析需求的基础上,提出了一种应用在大规模动态场景下的分布式编码解析系统,该系统由独立的,且相互协作的编码解析服务以松耦合的形式组合而成。在该系统中,用户能够实现灵活动态的解析服务定位,解析服务之间互不冲突,在必要时可以相互协作为用户呈现更准确完整的解析结果。同时,本文还设计实现了一个基于改进的 SkipNet 的编码解析服务,该服务不但继承了传统 DHT 覆盖网络的可伸缩、高容错、自组织和负载均衡等优点,而且支持区域自治,提供内容局部性和路径局部性。这使得服务参与者能够完全控制自己的编码解析记录,保护了其解析记录的隐私。通过理论分析表明该服务能够满足编码解析的功能需求,通过实验分析证明了该服务能够满足编码解析的性能需求。

## 2 相关工作

EPCglobal 提出的 ONS<sup>[3]</sup>和 EPCDS<sup>[4]</sup>是目前最权威的物联网编码解析规范,ONS 利用 DNS 架构实现,完成编码到相关制造商的信息服务的寻址,ONS 只提供产品类型级别的解析,无法实现单一产品的解析。EPCDS 是对 ONS 的有效补充,其可以实现单一编码到多企业提供的信息服务的寻址,EPCDS 支持本地查询和全局查询,但具体的规范细节尚未公布。欧盟联合全球数据标准机构 GS1 提出的 BRIDGE 项目<sup>[5]</sup>在 EPCDS 规范的基础上,列举了编码解析应具备的接口,并将编码解析的实现分为配置、发现和响应 3 个阶段,最后,提出了 4 种可行的解析记录发布和查询模式,从较高的层面分析了编码解析的特性。Overlook<sup>[6]</sup>和 DDNS<sup>[7]</sup>都是基于 DHT 覆盖网络的域名解析系统,其中,DDNS 基于 Chord 实现,而 Overlook 基于 Pastry 实现。DDNS 和 Overlook 继承了 DHT 的性质,实现负载均衡,避免了单点失效,增强了系统的健壮性。但是,它们增加了记录查询延迟,忽视了编码具有分级结构的特性。文献[8]和文献[9]基于 EPCglobal 的 ONS 规范,提出了一个基于 DHT 的对象信息分布式体系机构 OIDA,并分析了基于 OIDA 实现的非集中式解析服务的访问方式,及其隐私保护机制。CoDoNS<sup>[10]</sup>基于 Pastry 实现的域名解析系统,

它利用 Beehive<sup>[11]</sup>机制将查询的平均跳数由  $O(\log N)$  降低到  $O(1)$ 。CoDoNS 意图一步步的代替 DNS,它介于客户端和 DNS 之间,复制缓存 DNS 记录为客户端提供解析服务,其对客户端是透明的。CoDoNS 依靠 DNS 获取记录,DNS 会成为系统的性能瓶颈,当 CoDoNS 脱离 DNS 后会退化为一个扁平的解析网络。文献[12]也提出一种 DNS 和 DHT 相混合的编码解析服务,其上层节点组织为树状结构,而下层节点组织为 P2P 结构,意图折中二者的优点。文献[13]提出一种基于分级 P2P 的 RFID 编码解析网络,节点被分为多个组,每个组构成一个的 P2P 网络,每个组中的超级节点组成上级 P2P 网络,编码自上而下逐层就解析。该模型破坏了原始编码的结构,实现复杂,组内的超级节点易成为系统的性能瓶颈。文献[14]将物联网编码解析系统视为互联网资源寻址系统的上层扩展,并提出物联网编码解析的通用层次迭代模型,但是它没有给任何具体的解析服务的设计和实现。上述模型虽然有些不是针对物联网编码解析提出的,但它们的设计思想可以被物联网编码解析所借鉴。目前,物联网编码解析的功能描述和基本组成模块已日渐清晰,但是,多数研究成果或者是一些部署在供应链闭环应用的集中式系统,又或者是一些单一应用场景下的仿真模型,它们与物联网的大规模动态应用需求之间的差距很大。

## 3 编码解析服务的基本需求

设计实现一个优良的编码解析系统的首要工作是分析总结物联网编码解析的主要功能需求和性能需求。物联网编码解析的相关功能需求描述如下。

RQ1: 多编码标准共存。各机构提出了各自的编码标准,例如,EPCglobal 的 EPC 编码标准,uID Center 提出了 uCode 编码标准。不同标准的命名空间可能相互重叠导致编码解析冲突。解析系统应该能够识别不同的编码标准,在需要的情况下可以实现不同标准之间的转换。

RQ2: 编码精确解析。客户端输入单一兴趣编码,解析系统应该返回与该编码相关的信息资源。

RQ3: 编码范围解析。客户端输入一个编码范围,解析系统应该返回与该范围内的编码相关的信息资源。

RQ4: 启动引导策略。编码解析系统应该为客户端提供一个启动引导过程,帮助其动态定位能够

解析兴趣编码的编码解析服务。

**RQ5:** 低客户端复杂度。在保证功能完整的基础上, 应该尽可能的降低客户端的实现复杂度。

**RQ6:** 高解析质量。编码解析系统应该根据客户端的访问权限, 在预定的时间内返回完整的, 准确的解析结果。

**RQ7:** 开放的体系结构。编码解析系统应该基于开放的体系结构实现, 兼容已有的解析服务, 易于添加新的解析服务, 并鼓励更多的公司参与编码解析服务, 发布编码解析记录。

**RQ8:** 多源共存。多个独立的授权公司可以为同一编码发布各自的解析记录。解析服务应该能够区分解析记录的来源。

**RQ9:** 区域自治。编码解析服务的参与公司应该能够完全的控制管理自己的编码解析记录, 包括记录的增删改操作, 访问权限的设置和访问日志的跟踪等, 此外, 公司内部的访问通信不会对外暴露。

**RQ10:** 避免对信息资源访问控制权限的管理。解析服务不应该负责管理信息资源的访问控制权限, 信息服务对客户端权限的变更不应该对解析服务产生任何影响。

物联网编码解析的相关性能需求描述如下。

**RQ11:** 负载均衡。编码解析记录应该大致均衡的分布在编码解析服务中。

**RQ12:** 强可伸缩性。编码解析服务应该具有强可伸缩性, 能够存储管理海量的编码解析记录, 并及时响应大量的并发解析请求。

**RQ13:** 快速更新。编码解析服务应该具有自组织, 自配置能力, 支持对解析记录的快速更新, 防止过期记录引发的解析错误。

**RQ14:** 强顽健性。编码解析服务应该具有强顽健性, 能够抵御 DDoS 攻击, 并具有自适应, 自愈合, 高容错能力。

**RQ15:** 高可用性和可靠性。编码解析服务应该具有较高的可用性和可靠性, 为客户端提供稳定可靠的解析服务。

## 4 编码结构

产品编码存储在与物理实体绑定的电子标签中, 编码遵循某种编码标准所定义层次结构, 并且在该标准的命名空间中具有唯一性, 这类编码被称为原始编码(OC, original code)。然而, 不同编码标准的命名空间可能存在重叠, 这导致同一编码可能

符合多个编码标准, 但其在不同的标准下的意义却截然不同。这种命名空间的重叠会引发编码的解析冲突, 为消除此冲突, 本文提出标准编码(SC, standard code)来分离不同编码标准的命名空间。SC 具有全局唯一性, 由权威机构为各类编码标准统一分配。在编码解析系统中, 多个独立的授权公司可能会对同一编码发布各自的编码解析记录, 这种解析记录多源共存的现象可能引起解析结果负载过度, 为此, 本文提出公司编码(CC, company code)来识别解析记录的来源。一般来说, CC 应该被广泛认可, 可以通过反转公司的域名获取。综上所述, 编码解析系统的输入编码由 SC、CC 和 OC 3 个字段组成, 表示为: IC=SC:CC:OC。不难发现, IC 具有全局唯一性, 其中, SC 和 CC 字段可根据编码解析服务的应用范围设置为缺省的默认值。

## 5 编码解析系统

### 5.1 体系结构

应用在大规模动态的应用场景下的物联网编码解析系统是一个松耦合的分布式系统, 它由许多个相互独立且相互协作的编码解析服务组成, 这些服务可能隶属于不同的机构, 由不同的公司参与, 基于不同的体系结构, 识别不同的编码标准。如图 1 所示, 物联网编码解析的过程包含 4 类角色: 标准编码解析服务(SCRS, standard code resolution service)、原始编码解析服务(OCRS, original code resolution service)、信息服务和客户端。编码解析系统由唯一的 SCRS 和多个相互独立的 OCRS 组成。OCRS 提供输入编码的 CC 和 OC 字段的解析服务, 存储编码解析记录, 并把包含其支持的编码标准、服务接口、访问地址等服务描述信息和其他一些相关元信息发布到 SCRS。SCRS 是由权威机构运营的全局统一的解析服务, 它提供输入编码的 SC 字段的解析服务, 实现 OCRS 的间接寻址, 帮助客户端动态定位合适的 OCRS。客户端通过 SCRS 间接的获得 OCRS 的服务描述, 按照描述中的接口和地址实现和 OCRS 的交互, 进而获取与兴趣编码相关的信息资源。信息服务存储着与编码相关联的信息资源, 并将持有的编码信息和自身访问地址发布到相关的 OCRS, 当这些信息更新时, 它也会通知 OCRS 进行编码解析记录的同步更新。此外, 信息服务为 OCRS 提供了查询接口, 负责处理 OCRS 转发的客户端的解析请求。

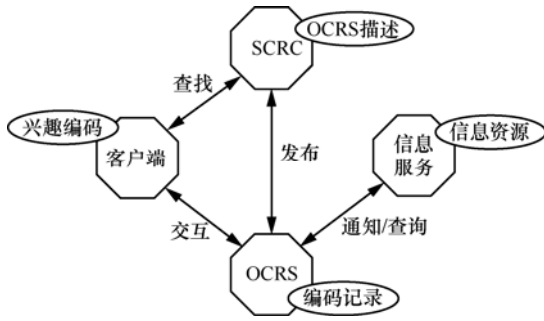


图 1 编码解析角色关系模型

### 5.2 工作流程

物联网编码解析执行过程如图 2 所示，为易于表达，图中只给出了一个 OCRS，即本文提出的基于改进的 SkipNet 的 OCRS，称为 SkipNet-OCRS，其他 OCRS 的查询执行流程与 SkipNet-OCRS 相似，例如，EPCglobal 的 ONS 和 EPCIS DS 等。编码解析执行的具体地过程描述如下：1) SkipNet-OCRS 向 SCRS 发布注册自身描述信息；2) 各个信息服务通知 SkipNet-OCRS 目前持有的编码及自身的访问地址；3) 客户端向 SCRS 发起对兴趣编码 SC 字段的解析请求；4) SCRS 将相应的 SkipNet-OCRS 的接口和访问地址应答给客户端；5) 客户端向该 SkipNet-OCRS 发起对兴趣编码 CC 和 OC 字段的解析请求；6) SkipNet-OCRS 查询存储的编码解析记录，发现存储与兴趣编码相关的信息资源的信息服务；

7) SkipNet-OCRS 向这些信息服务转发客户端的编码解析请求；8) 每个信息服务查询自身存储的与兴趣编码相关的信息资源，并做出应答；9) SkipNet-OCRS 将预设超时时长内达到的应答结果进行聚合；10) SkipNet-OCRS 将聚合结果应答给客户端。

SCRS 提供编码标准的解析服务(RQ1)，为客户端提供启动引导策略(RQ4)，它提供的间接寻址机制易于 OCRS 加入编码解析系统(RQ7)，由于编码标准的数量不大，SCRS 可以组织为树状结构，例如，组织为 DNS 的一个标识为“SCRS”的自治域。一般来说，现有 OCRS 都会提供编码精确解析服务(RQ2)，而第 2 节其他的需求不能完全满足。以这些需求作为指导，本文提出了 SkipNet-OCRS，下节中将对其进行详细的描述。

## 6 SkipNet-OCRS

SkipNet-OCRS 基于改进的 SkipNet 模型实现，SkipNet<sup>[15]</sup>是基于概率跳表实现的支持区域自治的 DHT 覆盖网络模型。本节将对 SkipNet-OCRS 的拓扑结构、编码解析记录存储、消息路由算法、编码解析过程，编码解析记录查询算法进行介绍。

### 6.1 拓扑结构和编码解析记录存储

SkipNet-OCRS 采用 2 套命名机制来命名节点和编码解析记录，即 NameID 和 NumericID，其中，节点的 NameID 由其隶属公司的公司编码和自定义的本地名构成，而节点的 NumericID 是通过分环得到的具有全局唯一性的定长随机二进制序列。SkipNet-OCRS 将概率跳表中的每一层组织成双向环，在每个环中节点按 NameID 排序。第  $i$  层中的每个环被拆分为  $k$  个互不相交的第  $i+1$  层的环，每一个环使用 RingID 标示，第  $i$  层环的 RingID 有  $i$  位。环拆分的过程表明了 RingID 产生过程，当拆分出只有一个节点的环且该环达到预设的最高层时，该环的 RingID 就是这个节点的 NumericID，最高层的环是按照 NumericID 排序的。图 3 展示了一个  $k=2$  的 5 层的 SkipNet-OCRS 模型，该模型由 8 个公司提供的 16 个节点构成。为直观表达，图中省略了节点 NameID 的本地名字段，仅保留了 NameID 的 CC 字段，例如，节点 E 的 NameID 的 CC 字段为 101，NumericID 为 1010，除根环外，它在各层分别位于 RingID 为 1、10、101、1010 的环中。

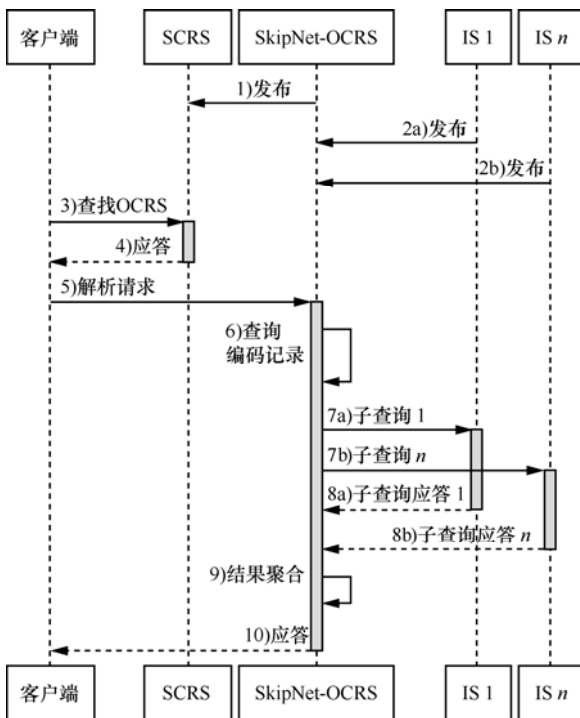


图 2 物联网编码解析执行过程

SkipNet-OCRS 具有传统 DHT 覆盖网络模型的可伸缩、高效率、高可用、高容错、自组织和自适应等特点，使其能够满足需求 RQ12、RQ13、RQ14 和 RQ15。此外，它在负载均衡和区域自治之间取了一个折中，采用受限的负载均衡(CLB, constrained load balancing)，也就是说编码解析记录在某个范围内是负载均衡的，但从网络整体上看未必成立。SkipNet-OCRS 存储的编码解析记录是编码与相关信息服务之间的映射信息，表示为<CC:OC, IS\_addresses>。与节点的命名方式相似，编码解析记录同样使用 NameID 和 NumericID 命名，因此，编码解析记录的标识由 2 部分构成：1) NameID 等于输入编码的 CC 字段，它指明负载均衡节点的范围，也称为 CLB 域；2) NumericID 为输入编码的 OC 字段的散列值，它指明解析记录的存储节点，也称为 CLB 后缀。SkipNet-OCRS 如此命名解析记录，既通过 CLB 域保证了解析记录的区域自治，并区分了解析记录的来源(RQ8)，又通过 CLB 后缀在 CLB 域内实现了解析记录的负载均衡(RQ11)。

区域自治保证了 SkipNet-OCRS 可以提供的 2 种局部性：1) 内容局部性指显式的指定编码记录放置位置或放置范围。2) 路径局部性指从一个节点到另一个节点的消息路由限制于 2 节点的最小公共域内。内容局部性保证了参与 SkipNet-OCRS 的公司完全控制自己的编码解析记录，路径局部性保证了公司内部解析请求不会被外部节点截获(RQ9)。

### 6.2 SkipNet-OCRS 的路由算法

SkipNet-OCRS 继承了 SkipNet 的 2 种路由机制：NameID 路由和 NumericID 路由。其中，NameID 路由算法采用邻居路由的机制，选择路由表中不超过目的节点 NameID 的最高层节点作为下一跳，直到目标节点 NameID 是当前节点 NameID 的前缀为止。NameID 路由的平均跳数通常为  $O(\log N)$ 。NumericID 路由算法采用前缀匹配机制，首先在第 0 层的根环查找第 1 位 NumericID 与目标 NumericID 前 1 位匹配的节点，然后在第 1 层环查找 NumericID 与目标 NumericID 前 2 位匹配的节点，也就是说，在第  $h$  层环查找 NumericID 与目标

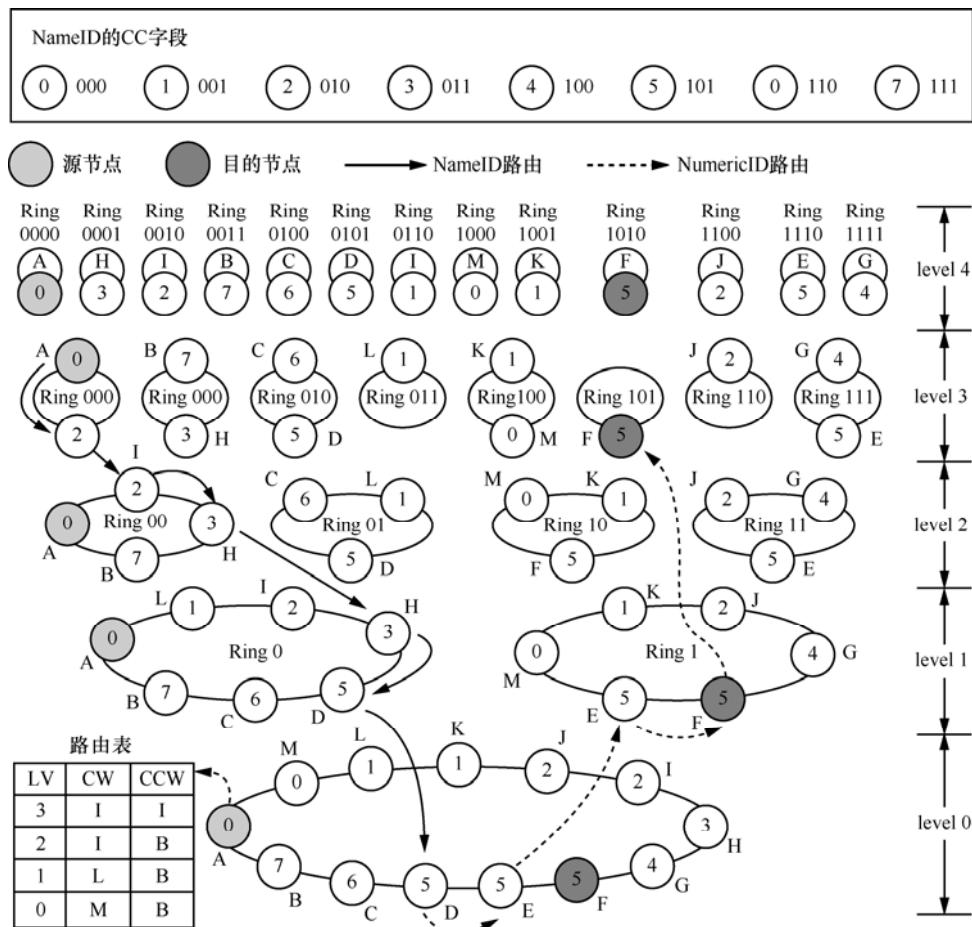


图3 SkipNet-OCRS 拓扑结构示意图

NumericID 前  $h+1$  位匹配的节点, 依此类推, 直到到达一个 NumericID 与目标 NumericID 最接近的节点为止, 此时到达的节点就是目的节点。NumericID 路由的平均跳数通常为  $O(\log N)$ 。

### 6.3 SkipNet-OCRS 的编码解析过程

SkipNet-OCRS 的编码解析过程可分为 5 个步骤: 1) 根据客户端的解析请求中的兴趣编码, 构造以目标解析记录标识为目的地的请求消息, 消息的目标 NameID 为兴趣编码的 CC 字段(目标 CLB 域), 目标 NumericID 为兴趣编码的 OC 字段的散列值(目标 CLB 后缀)。2) 通过 NameID 路由将消息发送到目标 CLB 域中的任一节点。3) 通过 NumericID 路由在目标 CLB 域中查找 NumericID 最接近目标 CLB 后缀的节点, 该节点是存储目标编码解析记录的目的节点。4) 向目标编码解析记录中的信息服务转发客户端的编码解析请求, 并启动超时计时器。5) 聚合在计时器超时之前收到的信息服务的响应结果, 为客户端提供信息资源的视图, 并将信息资源的合集发送给客户端。其中, 1)、2)、3) 是 SkipNet-OCRS 编码解析记录的查询过程, 4) 和 5) 是 SkipNet-OCRS 转发请求和聚合应答的过程。SkipNet-OCRS 只负责向相关信息服务转发客户端的解析请求, 而将客户端对信息资源的访问权限控制完全交给相关的信息服务处理, 信息服务与客户端之间的权限变更不会对 SkipNet-OCRS 产生任何影响。这不但降低了客户端实现的复杂度(RQ5), 还避免了 SkipNet-OCRS 对信息资源访问权限的管理所引发的信息不同步、冗余和泄露问题(RQ10)。SkipNet-OCRS 应答聚合机制可以根据网络状况和相关信息服务性能设置合理的超时时长, 根据信息服务的响应确定是否重发请求, 这不但降低了客户端的实现复杂度(RQ5), 还保证了高质量的解析结果(RQ6), 更具可行性和合理性。

编码解析记录的查询算法如图 4 所示, 在函数 *Query* 中, 首先构造了请求消息, 它的目标 NameID 赋值为参数 CC, 目标 NumericID 赋值为参数 OC 的散列值(如图 4 中(A)所示)。而后, 根据源节点的 NameID 和目标 NameID 之间的关系确定消息的转发方向(如图 4 中(B)所示)。调用函数 *RouteByNameID*, 对消息进行 NameID 路由, 找到目标 CLB 域内的任一节点。如果无法找到这个节点, 那么, 源节点将收到查询失败应答(如图 4 中(C)所示)。否则, 由该目标 CLB 域内节点发起消息的 NumericID 路由

```

// 初始化: msg.ringLvl = -1; msg.startNode = null;
msg.finalDestination = false
Query(CC, OC, msg) {
    msg.NameID = CC; msg.NumericID = hash(OC); ... (A)
    if(LongestPrefix(msg.NameID, localNode.NameID) == 0) ... (B)
        msg.dir = RandomDirection();
    else if(msg.NameID < localNode.NameID)
        msg.dir = CCW; // 逆时针方向
    else
        msg.dir = CW; // 顺时针方向
    RouteByNameID(msg);
}

RouteByNameID(msg) {
    h = localNode.maxHeight; // 获取路由表的最高层
    while(h >= 0) {
        nbr = localNode.RoutingTable[h][msg.dir];
        if(LiesBetween(msg.dir, localNode.NameID, nbr.NameID,
            msg.NameID)) {
            if(!CheckIfAlreadyVisited(msg, nbr)) {
                msg.AlreadyVisited(localNode);
                SendtoNode(nbr, msg); return;
            }
        }
        h--;
    }
    if(!msg.NameID.isPrefixof(localNode.NameID)) { ... (C)
        NegativeAck(msg); return;
    }
    msg.dir = CW; // NumericID 路由的初始方向为顺时针方向
    RouteByNumericID(msg); ... (D)
}

RouteByNumericID(msg) {
    if(msg.finalDestination == true ... (L)
        || msg.NumericID == localNode.NumericID) { ... (M)
        DeliverMessag(msg); ... (N)
        return;
    }
    if(msg.StartNode != null && localNode == msg.startNode) { ... (J)
        msg.FinalDestination = true;
        SendtoNode(msg.bestNode);
        return;
    }
    h = CommonPrefixLen(msg.NumericID, localNode.NumericID); ... (E)
    if(h > msg.ringLvl) { ... (F)
        msg.ringLvl = h; // 消息 NumericID 路由由 h 层起始
        msg.startNode = msg.bestNode = localNode; // 记录 h 层的起始节点
    }
    if(abs(localNode.NumericID - msg.NumericID)
        < abs(msg.bestNode.NumericID - msg.NumericID)) { ... (K)
        msg.bestNode = localNode;
    }
    if(localNode.RoutingTable[h][msg.dir].NameID == msg.NameID) { (G)
        SendtoNode(msg, localNode.RoutingTable[h][msg.dir]);
    }
    else if(msg.dir == CW) { ... (H)
        msg.finalDestination = false;
        msg.dir = CCW;
        SendtoNode(msg, msg.startNode);
    }
    else if(msg.dir == CCW) { ... (I)
        msg.finalDestination = true;
        SendtoNode(msg, msg.bestNode);
    }
}

```

图 4 编码解析记录查询算法

(如图 4 中(D)所示)。在函数 *RouteByNumericID* 中, 首先, 本地节点通过计算自身 *NumericID* 和消息的目标 *NumericID* 的公共前缀的长度, 确定 *NumericID* 路由的起始层(如图 4 中(E)所示), 如果公共前缀长度  $h$  大于消息中记录的层次, 那么, 消息将攀升到  $h$  层转发, 同时, 该本地节点做为  $h$  层的起始节点和初始最优节点(最优节点是 *NumericID* 最接近目标 *NumericID* 的节点)记入消息(如图 4 中(F)所示)。然后, 本地节点在  $h$  层的 CLB 域内沿着选定方向转发消息(如图 4 中(G)所示)。事实上, 消息 *NumericID* 路由的节点结构可分为双向列表和双向环两类, 在多数层次上消息沿列表转发(如图 5(b)所示), 而在少数较高层次上消息沿环转发(如图 5(c)所示)。在列表结构中, 消息首先顺时针方向沿着列表转发, 当遇到 CLB 域的一个边界节点时, 边界节点反转消息路由的方向, 并将消息直接发送给消息中记录的起始节点(如图 4 中(H)所示), 然后, 沿逆时针方向继续转发消息, 直至消息到达另一个 CLB 域的边界节点为止(如图 4 中(I)所示)。在环结构中, 消息顺时针方向沿着环转发, 直至消息又一次到达消息中记录的起始节点为止(如图 4 中(J)所示)。在上述转发过程中, 如果本地节点发现其邻居节点的 *NumericID* 比它的更接近目标 *NumericID*, 则更新消息中的最优节点为它的邻居节点(如图 4 中(K)所示)。当消息完成本层的列表或环的遍历, 并且消息无法到达更高的层次时, 消息中记录的最优节点就是消息路由的目的节点(如图 4 中(L)所示)。至此, 请求消息路由结束。需要注意的是, 上述消息 *NumericID* 路由过程的前提是在 CLB 域内无法找到 *NumericID* 与目标 *NumericID* 相同的节点。无论在任何层次和任何节点结构中, 一旦发现 *NumericID* 与目标 *NumericID* 相同的节点, 那么, 消息的路由终止, 该节点就是消息路由的目的节点(如图 4 中(M)所示)。当消息路由完成后, 目的节点首先修改并缓存消息, 然后, 根据客户端的权限决定是否向相应的信息服务转发该消息(如图 4 中(N)所示)。

图 3 中列举了一次编码记录查询的消息转发路径, 消息 *NameID* 路由的路径由实线表示, 而消息 *NumericID* 路由的路径由虚线表示。源节点 A(*NameID* 为 000)初始化了一个请求消息, 消息的目标 *NameID* 为 101, 目标 *NumericID* 为 1011。首先, 消息通过 *NameID* 路由经过节点 I(*NameID* 为

010)和 H(*NameID* 为 011)到达 CLB 域内的一个节点 D(*NameID* 为 010), 然后, 节点 D 以根环作为起始环开始消息的 *NumericID* 路由。节点 D (*NumericID* 为 0101)将消息发送到节点 E (*NumericID* 为 1110), 节点 E 的 *NumericID* 匹配目标 *NumericID* 的第 1 位, 因而它在第 1 层的环 1 中转发消息给 CLB 域内的节点 F (*NumericID* 为 1110)。节点 F 的 *NumericID* 匹配目标 *NumericID* 的前 3 位, 因此, 节点 F 在第 3 层的环 101 中继续转发消息。节点 F 是环 101 中的唯一节点, 也是 *NumericID* 最接近目标 *NumericID* 的节点, 所以它就是消息路由的目的节点。

*SkipNet-OCRS* 的解析记录查询算法的性能主要由 *NameID* 路由和 *NumericID* 路由确定, 其平均跳数通常为  $O(\log N)$ 。一般来说, 消息 *NumericID* 路由的跳数远远多于 *NameID* 路由, 而原始 *SkipNet* 的 *NumericID* 路由在列表结构中存在冗余的消息转发(如图 5(a)), *SkipNet-OCRS* 改进了 *NumericID* 路由, 从而有效的降低了消息路由的平均跳数。此外, 为了提高 *SkipNet-OCRS* 的健壮性, 避免节点失效引起编码解析记录的缺失, 每个节点可将其编码解析记录冗余的存储在与其 *NameID* 的 CC 字段相同的最高层环的邻居节点中。

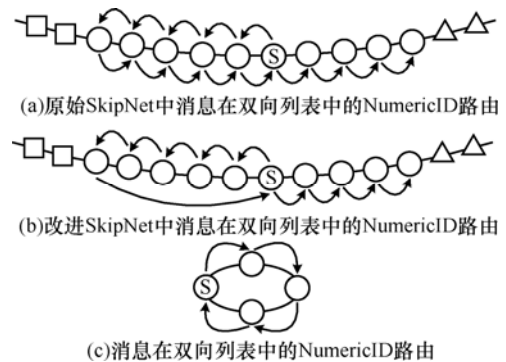


图 5 消息 *NumericID* 路由的节点结构和转发路径

通过 *SkipNet-OCRS* 的 *NameID* 路由很容易实现关于入编码的 CC 字段(CLB 域)的范围查询(RQ3), 而对 OC 字段的范围查询可以通过其他附加机制, 以时间或空间开销为代价实现, 也可通过将部分 OC 前缀字段的前缀扩展到 CLB 域的方法实现。

## 7 仿真实验

本文建立了一个简单的消息离散事件模拟器,

在 Mercator 物理拓扑<sup>[16]</sup>之上构造 SkipNet-OCRS, Mercator 中每个节点都属于一个自治系统(AS), 每个物理链路记为一物理跳。实验所用机器配置如表 1 所示。假设 SkipNet-OCRS 中节点数量为  $N$ , 参与的公司数量为 50, 公司编码由十进制数字标识, 各公司拥有的节点数量服从分布  $x^{-1.25}+0.5$ , 经过标准化可获得各个公司节点数。隶属于同一公司的节点位于同一 AS 内, 具体布置方法如下: 随机在某 AS 中选取一个起始节点, 而后通过分布  $x^{-1.25}+0.5$ , 计算其他节点到起始节点的物理跳数, 根据该跳数确定其他节点的放置位置。在实验前, 预先生成  $100 \times N$  条解析记录, 格式统一为  $\langle CC:OC, IS\_addresses \rangle$ , 并将解析记录发布并存储在 SkipNet-OCRS 中。编码解析记录的查询分为 2 类: 本地查询和域间查询。本地查询是指源节点和目标解析记录位于相同的 CLB 域的查询, 域间查询与此相反。

表 1 实验所用的机器配置

CPU	内存	操作系统	开发平台	开发语言
Intel 双核 E4500 2.2GHz	DDR2 667 4G	Windows XP SP3	Visual Studio 2008	C#

在仿真实验中主要考察以下 3 个参数。

1) 查询命中率(QHR): 在预设的时长内, 成功的查询次数与总查询次数之间的比值。

2) 平均查询跳数(AQH): 完成一次成功查询在覆盖网络上消息的平均跳数。AQH=成功查询总跳数/查询成功次数。

3) 平均查询时间(AQT): 完成一次成功查询的平均时延。AQT=成功查询总时间/查询成功次数。

由于 Mercator 拓扑以链路计数作为消息传输延迟的衡量, 所以在仿真模拟在不同的硬件平台下所得到的 AQT 互不相同, 实验中主要通过计算 AQT 确定某硬件平台下的查询超时时长, 本实验的超时时长是 0.5s。本文构造了 2 个实验, 通过 QHR 和 AQH 来评测 SkipNet-OCRS 的性能。

第一个实验主要测量 AQH, 本地查询比例和节点数量之间的关系。图 6 给出了在不同本地查询比例下, AQH 与节点数量之间的关系, 图 7 显示了当本地查询比例为 60% 时, AQH 与节点数量的关系。图 8 显示当节点数量为 20 000 时, AQH 与本地查询比例的关系。如图 6 和图 7 所示, AQH 随着节点数量的增加而缓慢提升, 节点数量由 500 增加至

20 000, AQH 的增幅小于 16 跳, 特别地, 当节点数量由 8 000 增加至 20 000 时, AQH 的增幅小于 2 跳。这证明了 SkipNet-OCRS 拥有优良的可伸缩性、可用性, 并确保负载均衡。如图 6 和图 8 所示, AQH 与本地查询比例成线性递减关系, 这证明了 SkipNet-OCRS 的区域自治能够有效的提升服务的性能。

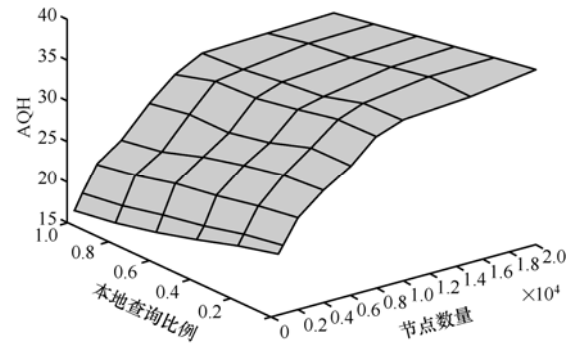


图 6 不同本地查询比例下, AQH 与节点数量的关系

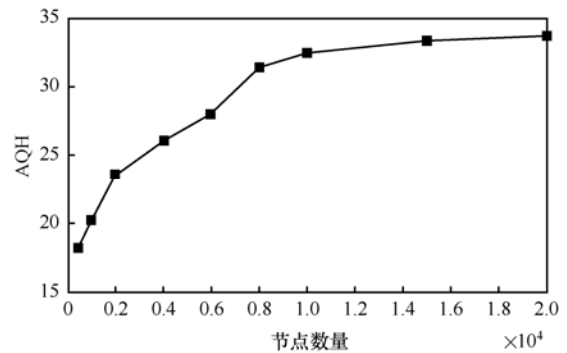


图 7 当本地查询比例为 60%, AQH 与节点数量的关系

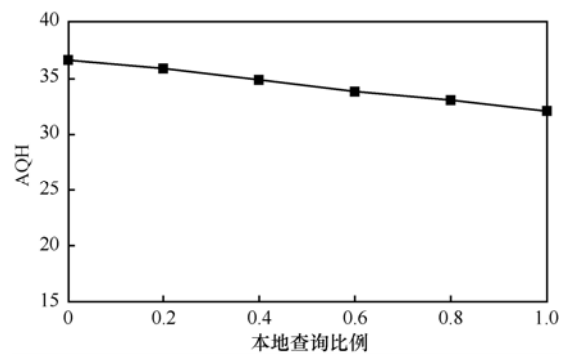


图 8 当节点数量为 20 000 时, AQH 与本地查询比例的关系

第二个实验主要测量 QHR, 本地查询比例和随机节点失效比例之间的关系。图 9 给出了当节点数量为 20 000 时, 在不同本地查询比例下, QHR 与随机节点失效比例之间的关系, 其中, X 轴表示局域查询的比例, Y 轴表示随机节点失效比例, Z 轴表示 QHR。图 10 显示了当本地查询比例为 60% 时,

QHR 与随机节点失效比例之间的关系。在实验中, 随机失效节点的比例由 0% 递增到 60%, 每次增加 10%。如图 9 和图 10 所示, 随着随机失效节点比例的增加, QHR 缓慢递减, 当比例小于 40% 时, QHR 高于 90%, 当比例升至 60% 时, QHR 仍高于 75%, 特别地, 本地查询的 QHR 更高于 80%。这证明了 SkipNet-OCRS 具有强顽健性, 高可靠性和高容错能力。

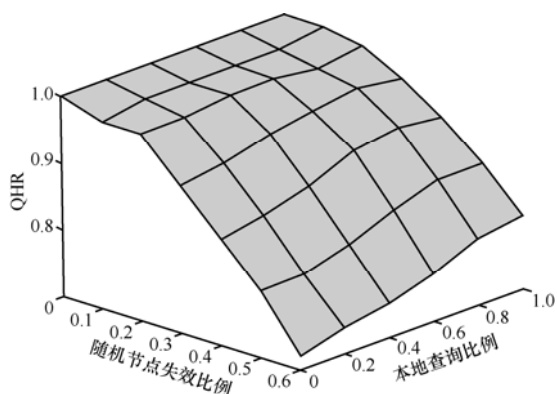


图 9 不同的本地查询比例下, QHR 与随机节点失效比例的关系

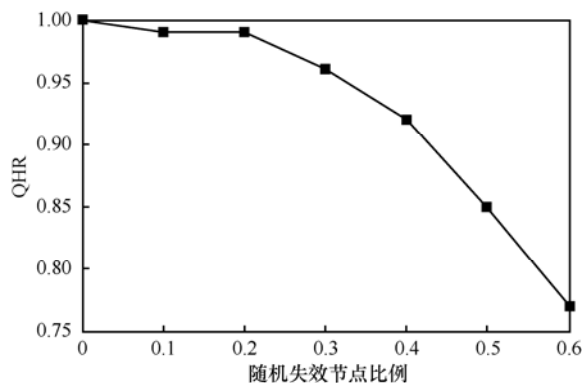


图 10 当本地查询比例为 60%, QHR 与随机节点失效比例的关系

## 8 结束语

本文在总结物联网编码解析的功能和性能需求的基础上, 首先, 提出了一个应用在大规模动态场景下的分布式编码解析系统, 该系统由多个独立且相互协作的编码解析服务组成。该系统的编码解析过程涉及 SCRS, OCRS, 客户端和信息服务 4 类角色, 而编码解析系统由一个 SCRS 和多个 OCRS 组成。然后, 提出了一个基于改进 SkipNet 的 OCRS, 即 SkipNet-OCRS, 该服务不但继承传统 DHT 覆盖网络的特性, 还支持区域自治, 满足内容局部性和路径局部性。经过理论分析和仿真实验, 可以证明如果将 SkipNet-OCRS

运用在本文提出的编码解析系统中, 那么, 编码解析系统就能够基本满足本文提出的物联网编码解析的需求。SkipNet-OCRS 仍存在一些尚未解决的问题, 首先是安全问题, 虽然 SkipNet-OCRS 具有较强的可用性、可靠性和顽健性, 同时, 区域自治为权限控制和隐私保护提供了基础支持, 然而, 在机密性、完整性、可信性方面 SkipNet-OCRS 还需要进行更加广泛和深入的研究, 例如 PKI 等授权和身份验证机制的引入, 可信等级的划分等。其次, SkipNet-OCRS 的复杂查询功能还需要进一步完善和拓展。最后, SkipNet-OCRS 需要缓存机制进一步提高编码解析的效率, 有效的缓存机制也是未来需要突破的方向。

## 参考文献:

- [1] ATZORI L, IERA A, MORABITO G. The internet of things: a survey[J]. *Computer Networks*, 2010, 54(15): 2787-2805.
- [2] EVDOKIMOV S, FABIAN B, KUNZ S, *et al.* Comparison of discovery service architectures for the internet of things[A]. *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*[C]. Newport Beach, CA, USA, 2010. 237-244.
- [3] EPCglobal. EPCglobal object naming service (ONS) 1.0.1[EB/OL]. <http://www.epcglobalinc.org/standards/ons>.
- [4] EPCglobal. The EPCglobal architecture framework – version 1.2[EB/OL]. [http://www.gs1.org/gsm/kc/epcglobal/architecture/architecture\\_1\\_2-framework-20070910.pdf](http://www.gs1.org/gsm/kc/epcglobal/architecture/architecture_1_2-framework-20070910.pdf).
- [5] BRIDGE. BRIDGE WP02 – Requirements Document of Serial Level Lookup Service for Various Industries[R]. 2007.
- [6] STOICA I, MORRIS R, LIBEN-NOWELL D, *et al.* Chord: a scalable Peer-to-Peer Lookup protocol for internet applications[J]. *Transactions on Networking*. 2003, 11(1): 17-32.
- [7] COX R, MUTHITACHAROEN A, MORRIS R T. Serving DNS using a Peer-to-Peer lookup service[A]. *Proceedings of International workshop on Peer-To-Peer Systems*[C]. Cambridge, MA, USA, 2002. 155-165.
- [8] FABIAN B, GUNTHER O. Distributed ONS and its impact on privacy[A]. *Proceedings of the IEEE International Conference*[C]. Glasgow, Scotland, 2007. 1223-1228.
- [9] FABIAN B. Implementing secure P2P-ONS[A]. *Proceedings of the IEEE International Conference*[C]. Dresden, Germany, 2009. 1-5.
- [10] RAMASUBRAMANIAN V, SIRER E G. The design and

implementation of a next generation name service for the Internet[A].  
 Proceedings of ACM SIGCOMM[C]. Portland, Oregon, USA,  
 2004,34(4): 331-342.

- [11] RAMASUBRAMANIAN V, SIRER E G. Beehive: exploiting power law query distributions for  $O(1)$  lookup performance in peer to peer overlays[A]. Proceedings of Symposium on Networked Systems Design and Implementation[C]. San Francisco CA, USA, 2004. 1-14.
- [12] DOI Y, WAKAYAMA S, ISHIYAMA M, OZAKI S, *et al.* On scalability of DHT-DNS hybrid naming system[J]. Lecture Notes in Computer Science, 2006, 4311: 16-30.
- [13] 刘学洋, 赵文, 张世琨等. 基于 P2P 的 RFID 编码解析网络结构与算法研究[J]. 电子学报, 2008, 36(12): 102-108.  
 LIU X Y, ZHAO W, ZHANG S K, *et al.* Research on P2P-based RFID code resolution network architecture and algorithm[J]. Acta Electronica Sinica. 2008, 36(12): 102-108.
- [14] 孔宁, 李晓东, 罗万明等. 物联网资源寻址模型[J]. 软件学报, 2010, 21(7):1657-1666.  
 KONG N, LI X D, LUO W M, *et al.* Model of the resource addressing in the Internet of things[J]. Journal of Software, 2010, 21(7): 1657-1666.
- [15] HARVEY N J A, JONES M B, SAROIU S, *et al.* SkipNet: a scalable overlay network with practical locality properties[A]. Proceedings of the 4th conference on USENIX Symposium on Internet Technologies and Systems[C]. Seattle, WA, USA, 2003.
- [16] TANGMUNARUNKIT H, GOVINDAN R, SHENKER S, *et al.* The impact of routing policy on internet paths[A]. Proceedings of the IEEE International Conference on Computer Communications[C]. Anchorage, Alaska, USA, 2001. 736-742.

作者简介:



于海宁(1983-), 男, 黑龙江通河人, 哈尔滨工业大学博士生, 主要研究方向为物联网和车载路由协议。



张宏莉(1973-), 女, 吉林榆树人, 哈尔滨工业大学教授、博士生导师, 主要研究方向为网络与信息安全、网络测量与建模、网络计算、并行处理等。



方滨兴(1960-), 男, 江西万年人, 哈尔滨工业大学教授、博士生导师, 主要研究方向为计算机网络与信息安全理论与技术、并行计算等。



余翔湛(1973-), 男, 黑龙江哈尔滨人, 哈尔滨工业大学副教授, 主要研究方向为网络容灾和信息安全。