

# 基于散列值的广域网服务发现\*

周晓<sup>+</sup>, 陈鸣

(解放军理工大学 通信工程学院 计算机系, 江苏 南京 210007)

## Wide Area Service Discovery Using Hash-Value

ZHOU Xiao<sup>+</sup>, CHEN Ming

(Department of Computer, Institute of Communications Engineering, PLA University of Science and Technology, Nanjing 210007, China)

+ Corresponding author: Phn: +86-21-62511070 ext 8219, Fax: +86-21-62255529, E-mail: fancy\_space@263.net

Received 2003-06-07; Accepted 2003-09-09

**Zhou X, Chen M. Wide area service discovery using hash-value. *Journal of Software*, 2004,15(10): 1565-1573.**

<http://www.jos.org.cn/1000-9825/15/1565.htm>

**Abstract:** Service discovery is an important aspect of network self-configuration. Solutions to local area service discovery had been put forward and applied to practical systems for several years. In terms of wide area network, the scalability problem of service discovery system must be addressed because there are numberless dynamic services and clients distributed in wide area. To get good scalability, service discovery system should maintain right function and steady performance when the number of services and clients increases continuously. A hash-value based scheme for wide area service discovery is presented in this paper. Based on the model of service description, service query, and match condition, the sets of type name and attributes name in service description and service query are mapped into hash-value and hash-value area respectively. They are used to identify the service directory that will record the service description or satisfy the service query. Each service directory in wide area network is assigned a value area to indicate the range of service descriptions to be recorded in it. All of the service directories are organized into a content addressable tree according to the relationship of their value areas. In such systems, service description is forwarded to certain directory to be recorded in terms of its hash-value, and service query is forwarded to the directories that might satisfy it in terms of its hash-value, area. Simulation results, along with analysis and comparison, show that this scheme can avoid deficiency of the existing schemes and gain favourable scalability.

**Key words:** wide area service discovery; scalability; hash-value; hash-value area; CAT (content addressable tree)

**摘要:** 服务发现是网络自配置的重要内容. 局域网服务发现方案已日臻完善并且应用到实际系统之中. 对于广域网, 由于客户和服务的分布性、动态性及数目众多的特点, 必须解决包括可伸缩性在内的诸多问题. 良好的

\* Supported by the National High-Tech Research and Development Plan of China under Grant No.2001AA112090 (国家高技术研究发展计划(863)); the Natural Science Foundation of Jiangsu Province of China under Grant No.BK2001022 (江苏省自然科学基金)

**作者简介:** 周晓(1971—),男,湖南湘阴人,博士,主要研究领域为网络管理,分布式计算;陈鸣(1956—),男,博士,教授,博士生导师,主要研究领域为通信技术,计算机网络.

可伸缩性是指在客户和服务的数目不断增长的情况下,服务发现系统仍然能够保持正确的功能和稳定的性能.提出了一种基于散列值的广域网服务发现方案.其方法是在服务表示模型和匹配条件的基础上,根据类型名和属性名集合为服务信息生成散列值,为服务请求生成散列值域.散列值指示负责存储服务信息的服务目录,散列值域指示有可能满足服务请求的服务目录.为服务目录指定值域,并使得所有的服务目录能够按照值域间的关系组织成一棵内容编址树.根据散列值将服务信息转发并存储到特定值域的服务目录上,根据散列值域将服务请求转发到有可能使其得到满足的服务目录上,从而实现广域网范围内的服务发现.模拟系统以及分析和对比表明,这一方案可以有效克服已有方案的局限,达到良好的可伸缩性.

关键词: 广域网服务发现;可伸缩性;散列值;散列值域;CAT(内容编址树)

中图法分类号: TP393 文献标识码: A

服务发现是网络自配置的重要内容.工作于局域网范围的服务发现方案已日臻完善并且应用到了实际系统之中<sup>[1,2]</sup>.局域网服务发现的基本过程就是服务代理通过声明服务信息来表示其所提供服务的类型和包括位置在内的各种属性.客户代理通过声明服务请求查询其所需的服务,通过接收服务响应的方式获得相关服务的位置信息.根据是否存在用于存储服务信息的服务目录,发现模式可分为集中型和分散型两种基本类型.

具有良好的可伸缩性,且支持多指标查询的广域网服务发现机制是当前的重要研究方向.基本的可行途径就是在集中型发现模式的基础上,建立起各服务目录间的服务信息交换或服务请求转发机制,从而实现广域网环境下有效的服务发现.查询复杂性、控制复杂性以及因转发的需要而形成的信息表的大小是衡量系统可伸缩性的主要因素<sup>[3]</sup>.

结合已有服务发现方案和分布式文件检索系统的基本思想,本文提出了一种基于散列值的广域网服务发现方案.其方法是:为服务信息生成散列值;为服务请求生成散列值域,并保证满足它的服务信息的散列值属于该域中;为分布在广域网中的每个服务目录指定值域,并使得所有服务目录结点能够按照值域间的关系连接成一棵内容编址树(content addressable tree,简称 CAT).在此基础上,将服务信息根据其散列值转发到相关结点存储,将服务请求根据其散列值域转发到有可能满足它的结点进行匹配操作并确定是否存在所需服务.分析和模拟系统表明,该方案具有良好的可伸缩性,而且支持对服务信息的多指标查询.

本文第1节介绍服务的表示模型和匹配条件.第2节提出服务信息散列值和服务请求散列值域的基本生成方法及约束条件.第3节提出CAT的基本概念和模型.第4节提出将服务目录组织成CAT的方法,以及相应的对服务信息和服务请求进行转发的算法.第5节介绍我们实现的模拟系统,描绘了有关的性能曲线.第6节介绍相关工作并进行比较和分析.第7节是总结.

## 1 服务的表示模型和匹配条件

在服务发现系统中,任何一项服务信息和服务请求都包含相关的服务类型名、属性名及属性值等基本表征元素<sup>[4,8]</sup>.这些表征元素同时也是判断一项服务信息是否满足特定服务请求的基本依据.

服务信息  $p$  表示为

$$p = \langle t, AV(t) \rangle, \\ t \in T; AV(t) = \bigcup_{i \in I} \{ \langle pa_i(t), V(pa_i(t)) \rangle \}; V(pa_i(t)) \subseteq \Psi(pa_i(t)) \text{ 且 } V(pa_i(t)) \neq \emptyset; i = 1, 2, \dots$$

其中,  $T$  表示所有可能的服务类型的集合,  $t$  表示与  $p$  相关的服务类型名,  $pa_i(t)$  表示服务信息的第  $i$  项属性的属性名.  $\Psi(pa_i(t))$  表示属性名  $pa_i(t)$  所对应属性值的取值空间,它是相应属性所能够取得的所有可能值的集合,不同属性名对应值的类型不一样,因此其取值空间也不一样.  $V(pa_i(t))$  表示  $p$  中的名为  $pa_i(t)$  的属性的具体取值集合,它是  $\Psi(pa_i(t))$  的一个非空子集.以此为基础,  $AV(t)$  表示  $p$  所具有的属性及相应属性值的集合.

服务请求  $q$  表示为

$$q = \langle t, AR(t) \rangle,$$

$$t \in T, AR(t) = \bigcup_{j \in I} \{(qa_j(t), R(qa_j(t)))\}; R(qa_j(t)) \subseteq \Psi(qa_j(t)); j=1, 2, \dots$$

其中,  $t$  表示与  $q$  相关的服务类型名,  $qa_j(t)$  表示与服务请求所指定条件相关的第  $j$  项属性的属性名.  $\Psi(qa_j(t))$  的含义与服务信息  $p$  中  $\Psi(pa_i(t))$  项相同.  $R(qa_j(t))$  是  $\Psi(qa_j(t))$  的子集, 当  $R(qa_j(t))$  非空时, 表示  $q$  对名为  $qa_j(t)$  的属性的具体取值要求, 当  $R(qa_j(t))$  为空时, 表示  $q$  需要查询该项属性的所有取值. 以此为基础,  $AR(t)$  表示  $q$  对所查询的服务的属性和相应属性值方面的要求.

判断服务信息是否满足服务请求的过程称为匹配操作, 它由相关实体上的匹配引擎执行. 基于两者的表示模型, 可以定义相应的匹配条件. 一项服务信息  $p$  满足服务请求  $q$ , 当且仅当两者具备:

条件 1.  $p.t=q.t$ , 对任意一项  $(qa(q.t), R(qa(q.t))) \in AR(q.t)$ , 存在  $(pa(p.t), V(pa(p.t))) \in AV(p.t)$ , 使得  $qa(q.t)=pa(p.t)$  且  $R(qa(q.t)) \subseteq V(pa(p.t))$ .

$p$  和  $q$  所对应的服务类型名相同,  $p$  包含  $q$  中的所有相关属性名, 且  $p$  中属性名的取值集合包含  $q$  所要求的相关属性名的取值集合时,  $p$  满足  $q$ . 建立在这一条件基础上的查询是多指标查询. 当  $p$  和  $q$  满足:

条件 2.  $p.t=q.t$ , 对任意一项  $(qa(q.t), R(qa(q.t))) \in AR(q.t)$ , 存在  $(pa(p.t), V(pa(p.t))) \in AV(p.t)$ , 使得  $qa(q.t)=pa(p.t)$  时, 称  $p$  有可能满足  $q$ , 如果不具备条件 2, 则可以确定  $p$  不可能满足  $q$ .

因此, 可以利用类型名和属性名集合生成散列值和散列值域, 以建立服务请求和有可能满足它的服务信息之间的联系.

## 2 散列值和散列值域的生成

服务请求和服务信息是通过服务类型名、属性名及相应的属性值来表征的, 为了使散列值和散列值域能够体现服务信息和服务请求之间的相关性, 必须根据它们的表征元素来生成这两个参数. 在具体的系统实现中, 所认可的服务类型, 每一类型具有的属性名及相应属性值类型和取值空间, 以及各实体间交互的服务信息和服务请求的表达形式, 都需遵循相应的标准. 所以必须以这一标准为依据制定散列值和散列值域的生成规则和方法. 鉴于属性值类型各异且取值空间庞大, 而类型名和属性名集合足以建立服务请求和有可能满足它的服务信息之间的联系, 因此以类型名和属性名作为生成散列值(域)的依据.

定义 1.  $V$  为一个一维值域.

定义 2.  $T$  为所有服务类型名的集合.

定义 3. 服务发现系统的描述空间  $D$  为  $\cup\{t, S(t)\}$ ,  $t \in T$ . 其中,  $t$  为其中的某一特定的服务类型名,  $S(t)$  表示  $t$  类型的服务可能具有的所有属性名的集合.

定义 4. 服务发现系统的表达空间  $E$  为  $\cup\{e\}$ . 其中  $e=\langle t, A(t) \rangle$ ,  $t \in T, A(t) \in 2^{S(t)}$ ;

即每一项服务信息或服务请求都包含服务类型名及相关属性名集合. 相应地, 用  $e$  表示类型名和属性名集合.

定义 5.  $P$  为建立在  $E$  上的服务信息的集合,  $p \in P, Q$  为建立在  $E$  上的服务请求的集合,  $q \in Q. \Gamma$  为从  $P$  到  $E$  和从  $Q$  到  $E$  的一个映射, 即  $\Gamma(p)=e, p \in P, e \in E; \Gamma(q)=e', q \in Q, e' \in E$ . 其具体操作为, 将  $p$  或  $q$  中的类型名和属性名集合提取出来. 由于不同的服务信息或服务请求中可能含有相同的类型名和属性名集合, 因此  $\Gamma$  为多对一的映射.

定义 6. 对任何  $e_1, e_2 (e_1 \in E, e_2 \in E)$ , 如果  $e_1.t=e_2.t$  且  $e_1.A(t) \subseteq e_2.A(t)$ , 则记为  $e_1 \alpha e_2$ , 它表示服务类型名和属性名集合  $e_1$  包含于  $e_2$ , 或者称  $e_2$  包含  $e_1$ .

定义 7.  $H$  为一个从  $E$  到  $V$  的映射, 它使得  $H(e)=h, e \in E, h \in V$ , 而且对任何  $e_1 \alpha e_2 (e_1 \in E, e_2 \in E)$ , 有

$$H(e_1) \leq H(e_2).$$

针对一个具体的服务发现系统, 可以有多种  $H$  映射. 通过这种映射, 就可以将类型名及属性名的各种可能的组合形式映射为  $V$  中的具体值. 如果某一类型名和属性名集合与同一类型下的类型名和属性名集合之间是包含于(包含)关系, 那么, 通过  $H$  映射出的散列值便具有小于等于(大于)关系. 不同类型下的类型名和属性名集合不存在包含关系.

定义 8.  $p(p \in P)$  为一项服务信息,  $e=\Gamma(p) (e \in E)$  为其中所包含的类型名及属性名集合, 则取  $H(e)$  为该服务信息的散列值.

于是通过  $\Gamma$  映射和  $H$  映射的联合作用,将一项具体的服务信息  $p$  与值域  $V$  中的一个具体值关联起来,这个值  $h=H(\Gamma(p))(h \in V)$  就作为该服务信息的散列值.

定义 9.  $q(q \in Q)$  为一项具体的服务请求,  $e = \Gamma(q)(e \in E)$  为其中所包含的类型名及属性名集合且  $e = (t, A(t)), t \in T$ . 令  $\phi(e) = (t, S(t))$ , 显然有  $\phi(e) \in E$ , 则取  $[H(e), H(\phi(e))]$  为该服务请求的散列值域.

定理 1. 如果  $e_p$  为一项服务信息中包含的类型名及属性名集合,  $e_q$  为一项服务请求中包含的类型名及属性名集合, 如果  $e_p.t = e_q.t$  且  $e_q.A(t) \subseteq e_p.A(t)$ , 则  $H(e_q) \leq H(e_p) \leq H(\phi(e_q))$ .

证明: 由  $e_p.t = e_q.t$  且  $e_q.A(t) \subseteq e_p.A(t)$  及定义 6 和定义 7 可知  $H(e_q) \leq H(e_p)$ . 又由定义 9:  $e_p.t = e_q.t = \phi(e_q).t$  且  $e_p.A(e_p.t) \subseteq \phi(e_q).S(e_q.t)$ , 所以由定义 6:  $e_p \propto \phi(e_q)$ , 所以由定义 7:  $H(e_p) \leq H(\phi(e_q))$ , 则  $H(e_q) \leq H(e_p) \leq H(\phi(e_q))$ .  $\square$

由定理 1 可知, 如果一项服务请求  $q$  的类型名及属性名集合包含在一项同类型的服务信息  $p$  的类型名及属性名集合之中, 或者说根据条件 2, 服务信息  $p$  有可能满足一项服务请求  $q$ , 那么, 遵循定义 1~定义 9 而生成的服务信息  $p$  的散列值就必定属于服务请求  $q$  的散列值域. 通过这种方式, 就将服务信息和服务请求之间的可能满足关系转化成了散列值和散列值域之间的从属关系, 从而进一步将服务请求的转发目的和有可能满足它的服务信息的存储位置关联起来.

### 3 CAT 模型

为了能将服务信息根据其散列值存储到各个服务目录结点, 并以此为基础, 根据散列值域转发服务请求, 还需要适当的服务目录结点组织方式. 文献[5,6]提出了一种称为内容编址网(content addressable network, 简称 CAN)的分布式系统结点组织方式, 为系统中的每个结点指定一个值域, 各结点按照值域的相邻关系连接成网状结构, 数据项按照关键值被转发并存储在各个结点. 同样, 查询请求也按照关键值被转发到有可能满足它的结点进行匹配, 以提高查找效率并避免瓶颈结点. 但是, 这种方式不支持对数据项的多指标查询, 而且在值域的维数较低的情况下难以获得较高的查找效率, 在维数较高的情况下, 又增加了管理和维护的复杂性. 为此, 本文提出基于一维值域的 CAT 模型.

定义 10.  $R$  为实数的集合.

定义 11.  $R$  上的值域定义为  $A = \{x | a_1 \leq x < a_2, x \in R, a_1 \in R, a_2 \in R\}$ , 记为  $A = [a_1, a_2)$ ,  $A$  为一个实数的集合, 表示  $A$  由所有介于  $a_1$  (包含  $a_1$ ) 和  $a_2$  之间的实数组成.

所有服务信息的散列值包含于某个值域中, 因此下文中用  $G$  表示包含全体散列值的值域.

定义 12. 结点为四元组  $node = (Na, Ap, Ty, Ah)$ . 其中,  $Na$  表示此结点的名字,  $Ap$  表示结点的可访问地址及端口,  $Ty$  表示结点类型, 它只取 LEAF(叶结点)和 NONLEAF(非叶结点)两种值,  $Ah$  表示该结点所对应的值域  $G$ .

定义 13. 结点的转发信息为四元组  $F = (s, f, B, C)$ , 其中,  $s$  表示本结点, 类型为  $node$ ,  $f$  表示父结点, 类型为  $node$ ,  $B$  为兄弟结点集合,  $B = \cup \{b_i\}$ ,  $b_i$  的类型为  $node$ , 且  $f.Ah = s.Ah \cup b_1.Ah \cup b_2.Ah \dots \cup b_{|B|}.Ah$ ;  $s.Ah \cap b_i.Ah = \emptyset$ ;  $b_i.Ah \cap b_j.Ah = \emptyset$ ; ( $i, j = 1, 2, \dots, |B|; i \neq j$ ),  $C$  为子结点集合,  $C = \cup \{c_i\}$ ,  $c_i$  的类型为  $node$ , 且  $s.Ah = c_1.Ah \cup c_2.Ah \dots \cup c_{|C|}.Ah$ ;  $c_i.Ah \cap c_j.Ah = \emptyset$ ; ( $i, j = 1, 2, \dots, |C|; i \neq j$ ).

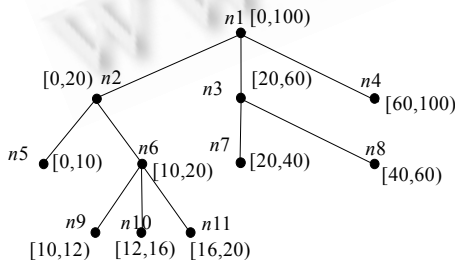


Fig.1 An example of CAT

图 1 CAT 的一个例子

结点的各兄弟结点及本结点的值域两两之间的交集为空, 而它们的并集则等于父结点的值域. 因此, 根结点对应的值域为  $G$ . 结点的各子结点值域的并集应该等于本结点的值域. 叶结点的  $C = \emptyset$ .

定义 10~定义 13 规定了一个 CAT 模型, 表示树中各结点的基本信息以及结点与结点之间的关系. CAT 的一个具体实例及其中结点和值域的分布情况如图 1 所示. 结点收到服务信息或服务请求报文时, 便根据这些信息及报文中的相关参数进行处理.

CAT 起始于一个原始结点, 分布在各地的结点通过动态加入的方式适应服务和客户数目的增长, 同时延伸

服务发现的范围.新结点加入时以叶结点的身份成为当前 CAT 树中某个结点(父结点)的子结点,并由该结点负责从其所对应的值域中划分出一个特定大小的子域赋予新结点.相应地,以父结点为根的子树中的部分结点所对应的值域也要进行相应调整.当所选择的父结点为叶结点时,还要从逻辑上分裂出一个子结点(新的叶结点).CAT 的生成与系统实现有关,父结点的选取、子域的指定、子树中结点值域的调整都需要根据实际情况采取相应的策略,前提是保证所生成的树符合 CAT 模型.

#### 4 结合散列值和 CAT 的广域网服务发现方案

基于散列值的广域网服务发现的主要思想就是按照 CAT 将各服务目录结点组织起来,并且服务目录结点均为 CAT 中的叶结点,它与 CAT 中的非叶结点在逻辑上是独立的,在物理上则为一体.除进行报文转发外,叶结点主要负责存储服务信息并据此对服务请求进行匹配操作,而非叶结点只起报文转发作用.

当叶结点收到从服务代理提交的服务信息报文  $PM$  时,从中生成散列值  $h$  并将其附加到报文中.当叶结点收到从客户代理提交的服务请求报文  $QM$  时,从中生成散列值域  $A_q$ ,并将其附加到报文中.各结点根据散列值、散列值域及本地的转发信息对相关报文进行转发.如果  $h$  属于某个结点的值域,则说明  $PM$  由该结点负责存储, $PM$  转发过程结束.如果  $A_q$  与某结点值域的交集不为空,则说明  $QM$  有可能在其上得到满足,必须要将  $QM$  转发到所有的这类结点上.

**定义 14.** 服务信息报文为  $PM=(A_p, pdestext, h)$ , 其中,  $A_p$  为发出该报文的代理地址,  $pdestext$  为服务信息的具体表达式,而  $h$  则为生成的散列值.

**算法 1.** 对  $PM$  的转发.

输入:  $F=(s, f, B, C), PM=(A_p, pdestext, h)$ .

输出:  $PM$  所要转发到的下一个结点  $n$ .

- (1) 如果  $s.T_j = \text{LEAF}$ , 则转(2), 否则转(5);
- (2) 如果  $PM.h \in s.A_h$ , 则  $n \leftarrow s$  并结束算法;
- (3) 如果  $PM.h \in f.A_h$ , 则从集合  $B$  中寻找满足  $PM.h \in b_i.A_h$  的结点  $b_i$  使得  $n \leftarrow b_i$  并结束算法;
- (4)  $n \leftarrow f$ ; 结束算法;
- (5) 如果  $PM.h \in s.A_h$ , 则从集合  $C$  中寻找满足  $PM.h \in c_i.A_h$  的结点  $c_i$  使得  $n \leftarrow c_i$  并结束算法;
- (6) 如果  $PM.h \in f.A_h$ , 则从集合  $B$  中寻找满足  $PM.h \in b_i.A_h$  的结点  $b_i$  使得  $n \leftarrow b_i$  并结束算法;
- (7)  $n \leftarrow f$ ; 结束算法.

在本结点是叶结点的情况下, (1)先判断  $PM.h$  是否属于本结点值域, 若是, 则返回本结点并结束(2). 否则, 判断  $PM.h$  是否属于父结点值域, 如果是, 则根据定义 13,  $PM.h$  必属于某个兄弟结点值域, 因此从  $B$  中返回相应结点并结束(3). 如果前两项均不成立, 则返回父结点并结束(4).

在本结点是非叶结点的情况下, 先判断  $PM.h$  是否属于本结点值域, 如果是, 则说明  $PM.h$  必然属于某个子结点值域, 该子结点应该是下一个要转发到的目的结点(5). 否则, 判断它是否属于父结点值域, 如果是, 则根据定义 13,  $PM.h$  必属于某个兄弟结点值域, 该兄弟结点是下一个要转发到的目的结点(6). 如果前两项均不成立, 则返回父结点并结束(7).

证明: 正确性和唯一性.

服务信息最终是存储在叶结点上的, 因此只有当报文到达某个满足  $PM.h \in s.A_h$  的叶结点  $s$  时, 整个转发过程才算结束. 算法执行结束后,  $n$  必然指向  $s, b, c, f$  中的一个. 1) 当  $n=s$  时(2), 结束整个转发过程. 2) 当  $n=b$  或  $n=c$  时, 对所要转发到的下一结点, 必然有  $PM.h \in n.A_h$ . 因此, 可进一步经后续结点执行算法的(5)、(2)而结束转发过程. 3) 当  $s=f$  时, 在后续结点转发算法执行的结果只能是  $n$  指向  $b$  或  $f$ (6)(7); 由于树的层次是有限的, 必然会在某一层结点使得算法执行结果是  $n$  指向  $b$ , 这就转化为 2) 的情况, 最终会将报文转发到一个叶结点而结束整个转发过程.

由 CAT 模型可知, 各个叶结点的值域之间是互不相交的, 因此最终只能有一个叶结点  $n$  使得  $PM.h \in n.A_h$ , 即转发到的目的结点是唯一的.  $\square$

转发  $QM$  时,必须将其转发到所有可能使其得到满足的服务目录结点.从 CAT 看,这些结点就是所有的值域与  $QM$  的散列值域相交的叶结点.为避免转发时的重复路径,在报文中增加一个指示转发方向的参数  $direction$ .当结点收到或转发  $QM$  报文时,都要对该参数进行处理. $direction$  的取值及含义如下所示:

FROM\_CLIENT:表示由客户提交给本域中的服务目录结点(叶结点)的报文;

FROM\_BROTHER:表示由兄弟结点转发过来的报文;

FROM\_FATHER:表示由父结点转发过来的报文;

FROM\_CHILD:表示由子结点转发过来的报文.

**定义 15.** 服务请求报文为  $QM=\langle Ap, qdestext, A_q, direction \rangle$ , 其中,  $Ap$  为发出该报文的客户代理地址,  $qdestext$  为服务请求的具体表达式,  $A_q$  为生成的散列值域, 而  $direction$  则表示它的转发方向.

**算法 2.** 对  $QM$  的转发.

输入:  $F=\langle s, f, B, C \rangle, QM=\langle Ap, qdestext, A_q, direction \rangle$ .

输出:  $QM$  所要转发到的后续结点的集合  $N$ .

- (1)  $N \leftarrow \emptyset$ ;
- (2) 如果  $s.Ty = \text{LEAF}$ , 转(3), 否则转(10);
- (3) 如果  $QM.direction = \text{FROM\_CLIENT}$ , 转(4), 否则转(9);
- (4) 如果  $QM.A_q \cap s.Ah \neq \emptyset$ , 则  $N \leftarrow N \cup \{s\}$ ;
- (5) 如果  $QM.A_q - s.Ah = \emptyset$ , 算法结束;
- (6) 如果  $QM.A_q - s.Ah \neq \emptyset$ , 则从  $B$  中选择满足  $QM.A_q \cap b_i.Ah \neq \emptyset$  的所有  $b_i$ , 使得  $N \leftarrow N \cup \{b_i\}$ ;
- (7) 如果  $QM.A_q - f.Ah \neq \emptyset$ , 则  $N \leftarrow N \cup \{f\}$ ;
- (8) 算法结束;
- (9) 如果  $QM.direction = \text{FROM\_FATHER}$  或  $QM.direction = \text{FROM\_BROTHER}$ , 则  $N \leftarrow N \cup \{s\}$  且算法结束;
- (10) 如果  $QM.direction = \text{FROM\_CHILD}$ , 则转(11), 否则转(14);
- (11) 如果  $QM.A_q - s.Ah \neq \emptyset$ , 则从  $B$  中选择满足  $QM.A_q \cap b_i.Ah \neq \emptyset$  的所有  $b_i$ ,  $N \leftarrow N \cup \{b_i\}$ ;
- (12) 如果  $QM.A_q - f.Ah \neq \emptyset$ , 则  $N \leftarrow N \cup \{f\}$ ;
- (13) 算法结束;
- (14) 如果  $QM.direction = \text{FROM\_FATHER}$  或  $QM.direction = \text{FROM\_BROTHER}$ , 则转(15);
- (15) 从  $C$  中选择满足  $QM.A_q \cap b_i.Ah \neq \emptyset$  的所有  $c_i$ ,  $N \leftarrow N \cup \{c_i\}$ ;
- (16) 算法结束.

以下是算法 2 的说明(括号内数字指算法 2 的步骤): 结点首先将  $N$  清空(1), 然后根据本身的类型及报文的来源对其进行处理.

在本结点是叶结点的情况下(2), 收到由客户代理直接发送过来的  $QM$  时(3), 如果  $QM.A_q \cap s.Ah \neq \emptyset$ , 说明本结点有可能满足  $QM$ , 因此将本结点置入  $N$  中(4); 如果  $QM.A_q - s.Ah = \emptyset$ , 说明不再有能够满足  $QM$  的其他结点, 算法结束(5). 如果还有  $QM.A_q - s.Ah \neq \emptyset$ , 则说明兄弟结点中还有可能满足  $QM$  的分支, 因此, 从兄弟结点中选取相应结点置入  $N$  中(6). 如果条件  $QM.A_q - f.Ah \neq \emptyset$  也满足, 说明以父结点为根的子树之外还有可能满足  $QM$  的分支, 需要将  $QM$  交给父结点由其继续转发, 将父结点也置入  $N$  中(7). 在这些情况处理完后算法结束(8).

在本结点是叶结点的情况下, 如果收到的是父结点或兄弟结点转发过来的报文, 则说明本结点就是目的结点之一, 将  $s$  置入  $N$  后, 算法结束(9).

在本结点是叶结点的情况下, 如果收到的报文来自子结点(10), 则首先判断, 如果  $QM.A_q - s.Ah \neq \emptyset$ , 则说明本结点的兄弟结点中可能有满足  $QM$  的分支, 因此, 从兄弟结点中选取相应的结点置入  $N$  中(11). 如果条件  $QM.A_q - f.Ah \neq \emptyset$  也满足, 说明以父结点为根的子树之外还有可能满足  $QM$  的分支, 需要将报文转发给父结点, 将父结点也置入  $N$  中(12). 在这些情况处理完后算法结束(13).

在本结点是叶结点的情况下, 如果收到的报文来自父结点或兄弟结点(14), 则说明以本结点为根的子树可能有满足  $QM$  的分支, 因此只需从子结点中选取满足要求的结点并置入  $N$  中(15), 算法结束(16).

算法结束后, $N$ 是一个结点集合.本结点要根据自身和集合 $N$ 中各个结点的关系向其转发包含不同 *direction* 标志的  $QM$ .具体而言,对于子结点,转发 *direction*=FROM\_FATHER 的报文;对于兄弟结点,转发 *direction*=FROM\_BROTHER 的报文;对于父结点,转发 *direction*=FROM\_CHILD 的报文.如果  $N$  中包含有结点  $s$ ,则说明本结点为  $QM$  的目的叶结点之一,应该在本地对  $QM$  进行匹配操作.

证明:完备性和正确性.

设收到 *direction*=FROM\_CHILD 的  $QM$  的叶结点为  $a_0$  且  $QM.A_q-a_0.Ah \neq \emptyset$  (如果  $QM.A_q-a_0.Ah = \emptyset$ , 则到  $a_0$  就为止了).假设树中任意一个散列值域与  $QM.A_q$  相交的叶结点为  $a(a \neq a_0)$ , 即  $QM.A_q \cap a.Ah \neq \emptyset$ . 根据 CAT 形式模型,从  $a$  到根结点的路径上,必然存在一个非叶结点  $a'$  (有可能是根结点),使得  $a_0.Ah \subseteq a'.Ah, a.Ah \subseteq a'.Ah$ . 而这个结点  $a'$  必有两个子结点  $a'_1, a'_2$ , 使得  $a_0.Ah \subseteq a'_1.Ah, a.Ah \subseteq a'_2.Ah$ . 由于  $QM.A_q-a_0.Ah \neq \emptyset$ , 所以报文经过(6)和(7)的转发必然会到达  $a_0$  的兄弟结点或父结点,再经由(10)(12)继续转发总会到达  $a'_1$ . 在  $a'_1$ , 必经由(10)(11)转发到  $a'_2$ , 然后再经由(14)(15)和(9)到达叶结点  $a$ . 因此,  $QM$  能够被转发到 CAT 中的任何一个有可能满足它的结点上. 同时,从(4)(6)(11)(15)保证  $QM$  不会被转发到不可能满足它的结点上.  $\square$

## 5 性能分析和模拟系统

在每个非叶结点有  $m$  个子结点,总的叶结点数目为  $N$  的条件下,一项服务信息或服务请求报文最多经过  $2 \times (\log_m N - 1)$  次转发便可以到达目的结点. 每一项服务信息报文的数目在转发过程中始终为 1. 每一项服务请求报文的数目在转发过程中则可能复制成多份,这主要取决于与其散列值域相交的叶结点的个数,而与  $N$  无关. 每个结点所保存的转发信息表的大小取决于  $m$ , 而与  $N$  无关. 根据文献[3]的定义,这样的系统具有良好的控制复杂性和查询复杂性. 报文在转发过程中不经过根结点,而最多到达根结点的次级结点,系统可以通过增加子结点数目的方式消除瓶颈,因此,这是一种可伸缩性好的方案. 而散列值和散列值域的生成规则保证系统支持对服务信息的多指标查询.

我们模拟了基于散列值的广域网服务发现系统<sup>[11]</sup>. 它由两部分组成. 第 1 部分是构造任意的服务信息和服务请求报文,并从中生成散列值和散列值域. 第 2 部分则是根据散列值和散列值域对相关报文进行转发的一个 CAT 实例. 其中的树结构根据定义中的基本规则随机生成,服务信息和服务请求用数据项代替,并为其随机地生成一个标志其存储位置或转发方向的散列值和散列值域,每个结点按照算法 1 和算法 2 对数据项进行转发. 散列值采用整数值. 模拟系统中所生成散列值和散列值域的相关性,以及两类报文的实际转发情况验证了内容编址树模型及相关转发算法的正确性.

在第 2 部分模拟系统的基础上,我们以服务信息散列值呈均匀分布为基本假设条件,分别对服务信息和服务请求的平均转发性能进行了计算. 系统结点数目从 100 开始并以 100 递增到 1000,以  $[0, 20000]$  作为整个系统的值域,每一结点数目下随机生成 200 个样本树,样本中每个结点的子结点数目不超过 16. 计算服务信息平均转发跳数时,起始服务目录(叶)结点和散列值都通过取随机数来指定. 计算服务请求平均转发跳数时,起始结点随机指定. 散列值域取 3 组数据,第 1 组为  $[\mu-200, \mu+200]$ , 第 2 组为  $[\mu-2000, \mu+2000]$ , 第 3 组为  $[\mu-8000, \mu+8000]$ , 在每一组中的  $\mu$  也随机生成. 针对每个样本树取 200 组随机散列值或散列值域,对每一组值计算转发跳数,报文从一个结点转发到另一个结点计为经过了一跳. 根据模拟系统运行结果的相关参数描绘了服务信息和服务请求的转发跳数与服务目录结点数目的关系曲线,其结果如图 2 和图 3 所示.

## 6 相关工作介绍

WASRV<sup>[7]</sup> 是美国贝尔实验室提出的直接建立在 SLP 上的方案. 其方法是在各 SLP 域中指定 DA 或若干 SA 作为广播代理 AA, 指定一个 SA 作为接收代理 BA, 各 SLP 域中的这两种代理组成一棵因特网中的多播树. AA 定时地在多播树上广播本域向外提供的特定服务信息, BA 则从多播树上接收其他域中的 AA 所发布的服务信息并存储起来. UA 只向本域 DA 请求所需服务,如得不到满足则转而向本域中的 BA 请求. 通过这种信息冗余方式, UA 就有可能在因特网范围内的其他 SLP 域中发现所需服务,其查询的时间复杂度为  $O(1)$ . 缺点是, AA 需要

记录所有 SLP 域中的服务信息,虽然查询效率很高,但控制复杂性较差,只适合 DA 及服务数目较少的情况。

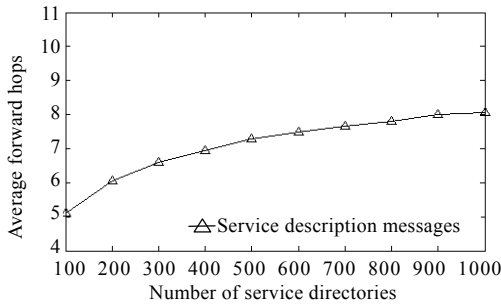


Fig.2 The forward performance of service descriptions  
图2 服务信息转发性能

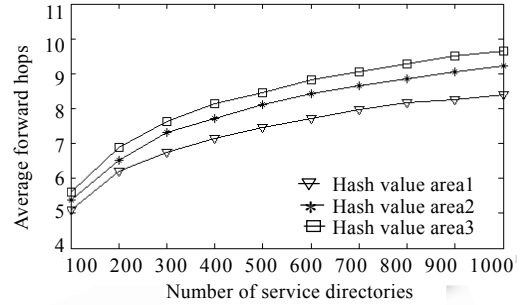


Fig.3 The forward performance of service queries  
图3 服务请求转发性能

SSDS 是美国加州大学伯克利分校为分布计算平台 Ninja 设计的一种服务发现系统<sup>[8]</sup>,通过在中心服务器之间转发服务请求来达到可伸缩性.系统由基于 SLP 机制、工作在局域网范围内的服务发现服务器 SDS 组成,并采取了以下措施:(1) 将分布在广域网中的各 SDS 组成一个树结构.(2) 各 SDS 除了存储本域内的服务信息外,还定期对服务信息中的属性名参数建立散列索引,并将这种索引以比特向量的方式上传到父结点,比特向量指示着以某结点为根的子树可能包含的服务属性.父结点对所有子结点传来的比特向量进行逻辑或运算,并以此作为对服务请求进行转发的依据.(3) SDS 收到服务请求时,先查找本域的服务信息,如果有则响应,否则查找所收集的各子结点的比特向量,从而决定将服务请求转发的方向.(4) 最终,服务请求被转发到有可能满足其要求的各个 SDS,并在其上最终进行匹配操作以确定是否存在所需服务.该方案基于比特向量和树结构,查询效率较高,其时间复杂度为  $O(\log_m N)$  ( $m$  含义同上,  $N$  为总结点数).但散列空间的碰撞和服务属性名的重复容易导致服务请求转发到错误结点,从而导致系统控制复杂性较差,同时,大部分服务请求都可能经由根结点转发而造成瓶颈.

Flexible Architecture 是卡耐基-梅隆大学提出的一种建立在 SLP 基础上的方案<sup>[9]</sup>.它规定了 4 种不同角色的 DA,其中,Local DA 与 SLP 中的 DA 的作用相同且只工作于局域网环境中.General DA 负责存储一般的服务信息.Specialized DA 负责存储特定类型或范围的服务信息,这 3 类 DA 可视具体应用的不同而有多.第 4 个是 DA Directory,它不存储服务信息,而只记录全网中 General DA 的位置信息以及 Specialized DA 的类型和位置信息,因此它相当于 DA 的目录代理.当客户请求某项服务时,只向本域的 Local DA 发出服务请求,如果 Local DA 上具备相应的服务信息则直接返回,否则直接或通过 DA Directory 间接地向 General DA 或 Specialized DA 转发该服务请求,从而实现广域网中的服务发现,且查询效率较高,其时间复杂度为  $O(1)$ .其缺点是当服务和客户的数目增加时,DA Directory 的负担也不断增加,从而形成访问瓶颈.

Carmen 是 HP 的研究小组提出的方案<sup>[10]</sup>.它通过多个树结构将分布在广域网环境下的各个服务目录组织起来,每个结点直接利用从子树各结点上的服务信息中提取出来的类型名和属性名作为转发信息,指示服务请求能够在某棵子树上得到满足的可能性.而每棵树的根结点又都与多个按类型来区分的若干个服务器相连,每个服务器负责存储特定类型服务信息的类型名和属性名集合,指示树与树之间对服务请求的转发.各个根结点将其存储的服务信息类型名和属性名按其类型存储在各个服务器上,以扩大它能够被发现的范围,查询的时间复杂度为  $O(\log_m N)$ .其缺点是服务属性名的重复容易导致服务请求转发到错误结点,而且,随着服务信息和客户数目的增加,各根结点的负担也增加,需要不断地对整个系统的结构进行调整,增加了系统的成本和不稳定性.

上述各种方案还不能同时在响应效率和所处理的服务数目方面达到令人满意的效果.它们有一个共同的特点:服务信息以无序的方式存储在分布于各地的服务目录结点上,增加了查找的复杂性和成本.本文提出的方案通过散列值和 CAT 对服务信息进行重新组织,并以此根据散列值域对服务请求进行转发.这样,类型和属性相近的服务信息被安排在相同或相邻的服务目录结点上,缩小了服务请求结点的转发范围,避免了将服务请求大量复制并转发到错误结点的问题,从而改善了控制复杂性.服务目录结点按照树结构连接在一起,可以达到时间



复杂度为  $O(\log_m N)$  的查询效率从而保持了良好的查询复杂性.转发信息表的大小不受结点数目的影响,而通过适当地增加子结点则可以避免瓶颈结点,因此系统具有良好的可伸缩性.

## 7 结语

分析和模拟系统的实验表明,采用基于散列值(域)的服务信息组织方式实现的广域网服务发现系统具有良好的可伸缩性,而且支持对服务信息的多指标查询.因此,它是实现广域网服务发现的一种有效途径.

### References:

- [1] Guttman E. Service location protocol: Automatic discovery of IP network services. IEEE Internet Computing, 1999,71~80.
- [2] Richard GG. Service advertisement and discovery: Enabling universal device cooperation. IEEE Internet Computing, 2000,18~26.
- [3] Gibbins N, Hall W. Scalability issues for query routing service discovery. 2001. [http://www.cs.cf.ac.uk/User/O.F.Rana/agents2001/papers/13\\_gibbins\\_hall.pdf](http://www.cs.cf.ac.uk/User/O.F.Rana/agents2001/papers/13_gibbins_hall.pdf)
- [4] Guttman E, Perkins C, Kempf J. Service templates and service: schemes. Technical Report, RFC 2609, 1999.
- [5] Ratnasamy S, Francis P, Handley M, Karp R, Shenker S. A Scalable content-addressable network. In: ACM Sigcomm 2001. 2001.
- [6] Ratnasamy S. A Scalable content-addressable network [Ph.D. Thesis]. Berkeley: University of California, 2002.
- [7] Rosenberg J, Schulzrinne H, Suter B. Wide area network service location. <http://www.belllabs.com/ mailing-lists/wasrv/>
- [8] Czerwinski SE, Zhao BY, Hodes TD, Joseph AD, Katz RH. An architecture for a secure service discovery service. In: Mobicom'99. 1999.
- [9] Huang A, Steenkiste P. A flexible architecture for wide-area service discovery. In: Proc. of the 3rd IEEE Conf. on Open Architectures and Network Programming. 2000.
- [10] Marti S, Krishnan V. Carmen: A dynamic service discovery architecture. Mobile and Media Systems Laboratory HP Laboratories, 2002.
- [11] Zhou, X. Research on the mechanism of wide area service discovery [Ph.D. Thesis]. Nanjing: Institute of Communications Engineering, PLA University of Science and Technology, 2003 (in Chinese with English abstract).

### 附中文参考文献:

- [11] 周晓.广域网服务发现机制的研究[博士学位论文].南京:解放军理工大学通信工程学院,2003.