

面向广域网的服务发现机制*

周晓** 陈鸣

(中国人民解放军理工大学通信工程学院 南京 210007)

【摘要】提出了一种新的用转发服务请求方法实现面向广域网的服务发现机制RFSD。分布在广域网中的服务器结点组成一个树结构，每个结点保存其父、兄弟和子等相邻结点的地址；服务请求传输报文中附加一定的转发信息。每个结点收到服务请求时，首先对其进行匹配操作，在满足的情况下响应服务请求，在不满足的情况下对附加信息进行处理，并将服务请求转发到某个相邻结点，这种方法可以有效地实现广域网中的服务发现。

关键词 服务发现；广域网；树结构；附加信息；请求转发
中图分类号 TP393.3

Wide Area Network Oriented Service Discovery Mechanism

Zhou Xiao Chen Ming

(Institute of Communications Engineering, PLAUST Nanjing 210007)

Abstract A wide area network oriented service discovery mechanism is described in this paper. Discovery servers in wide area networks are organized into a tree structure with each server node saves the address of its neighbor nodes, namely its father, brothers and children in the tree. The transmission message of service request is appended certain forwarding information. By matching the received service requests with service repository, each node response the service request or deals with the appended information in it and forward it to one of neighbor nodes. Thus, service discovery in wide area networks can be achieved effectively.

Key words service discovery; wide area network; tree structure; appended information; request forward

为适应网络的动态性和设备的移动性，需要建立一种称为服务发现的应用层自动配置机制，使各种设备和应用程序能够在网络中自动地声明自己所提供的服务，或者在权限范围内自动地发现所需服务的位置并访问它。这里，服务是指网络中设备和应用程序所提供的功能。目前已有多种服务发现解决方案，文献[1~4]是其中的代表，其基本工作原理是：作为服务使用者和请求者的客户根据所需服务的属性通过客户代理在网络中声明服务请求，相关服务的提供者通过服务代理回以包含服务位置信息的服务响应；或者是服务代理将服务信息存储在网络中的一个中心服务器上，由后者响应客户代理的服务请求。这些方案很好地解决了可管理的局域网范围内的服务发现问题。提高服务发现机制的可扩展性，使客户能够在广域网范围内更多的可用服务信息中发现所需服务是当前重要的研究课题。

2002年5月29日收稿

* 江苏省自然科学基金资助项目，编号：BK99128

** 男 32岁 博士生 工程师

1 相关研究

服务信息的组织和声明是服务发现机制的一个重要方面,它主要有分散型和集中型两种模式。分散型模式中参与服务的请求和发现的只有客户和服务提供者两类实体以及相应的客户代理和服务代理,它们以多播方式发布服务信息,声明或响应服务请求;而集中型模式则要求在网络管理域中设置中心服务器,用以存储服务信息和响应服务请求,要达到服务发现的可扩展性必须从发现模式着手。

1.1 SLP

SLP是IETF的SRVLOC工作组提出的建立在TCP/IP之上的服务发现协议,它定义了服务代理SA、用户代理UA和目录代理DA三类实体,SA代表程序或设备声明服务信息,UA代表客户声明服务请求和获取服务信息,DA则是一个中心服务器,它负责收集和存储服务信息并响应服务请求。SLP允许网络中可以没有DA,能够在分散型和集中型两种工作模式之间自动转换^[1]。

当服务和客户在数目上增加和在分布地域上扩大时,分散型模式不能工作,而单纯的集中型模式则效率极低。提高可扩展性的有效途径就是在集中型模式的基础上建立各中心服务器之间的协作关系,使客户的服务请求能够被本域之外的中心服务器响应。SLP由于其灵活性和基于TCP/IP的特点便成了各种可扩展性研究方案的基础,WASRV和SSDS就是这一思路的体现^[5,6]。

1.2 WASRV和SSDS

WASRV是由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所播送的服务信息并存储起来。UA只向本域的DA请求所需服务,当DA上没有合适的服务时转而向本域中的BA请求。这样,UA便有可能在因特网范围内的其他SLP域中发现所需服务。其缺点是当各DA上的服务数目呈线性增长时,AA上的服务数目呈其二次方急剧增长,因此这一方案只适合于DA及服务数目较少的情况。

SSDS是美国加州大学伯克利分校为分布计算平台设计的一种服务发现系统,通过在各中心服务器之间转发服务请求来达到可扩展性。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 (Service Discovery Server)组成,并采取了以下措施:1) 将分布在广域网环境中的各SDS组成一个树结构;2) 各SDS除了存储本域内的服务信息外,还定期按一定算法将本域中的服务信息摘要成聚集信息,并将其和从子结点收集到的聚集信息通过一定运算进行合并,从而形成新的聚集信息并上传给父结点。聚集信息相当于服务信息的索引,它指示着以某结点为根的子树可能包含的服务的属性;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聚集信息从而决定将服务请求转发给子结点还是父结点;4) 最终服务请求被转发到有可能满足其要求的各个SDS,并进行最终的匹配操作以确定是否存在所需服务。其缺点是由于信息聚集的需要对服务的表示方式有较大限制,且聚集信息的不断生成、合并、传输和更新产生了较多的系统和网络方面开销。

2 RFSD的工作原理

在SLP的基础上,本文提出了一种新的利用转发服务请求方法实现的服务发现机制RFSD (Request-Forward based Service Discovery),通过在各服务器间处理并转发附加了一定转发信息的服务请求传输报文的方式来实现广域网中的服务发现。

2.1 系统构成

系统由众多的分布在不同地域范围内,并且扩展了SLP中的DA的功能的RFSD服务器组成,这

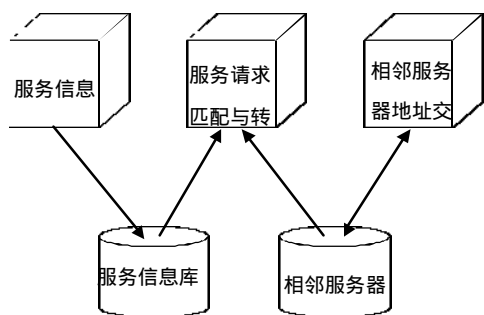


图1 RFSD服务器的组成

些服务器联结成一个树结构，这一工作由管理人员完成。树中的每个服务器结点具有自动配置功能，使得在与之相关的结点发生故障或通信中断的情况下，能够自动地建立起树中其他结点的联系，从而对树进行局部的重构以提高健壮性。服务的表示以及RFSD服务器与服务代理和客户代理之间的交互采用与SLP完全相同的方案。服务器由服务信息库、相邻服务器地址表、服务信息注册单元、服务请求匹配与转发单元以及相邻服务器地址交换单元组成，如图1所示。

2.2 工作原理

各服务器将其父结点、兄弟结点及子结点地址记录在本地，并形成子结点地址有序表和兄弟结点地址有序表，两者合称相邻服务器地址表；表中地址按升序排列，其中兄弟结点地址有序表中包含本结点地址。当子结点有序表发生变化时，必须将变化了的地址表及时通知各子结点(对各子结点，这些地址就是其兄弟结点地址)，这些工作由相邻服务器地址交换单元完成。

服务请求传输报文中附加的信息包括4个参数：1个是数组，表示服务请求所经过的树中各层结点的尚未被访问的兄弟结点的数目；然后是3个整数，分别表示下一个将对服务请求传输报文进行处理的结点在树中的相对层数，前一次转发的方向，以及对该服务请求生命期进行递减的计数器。附加信息的长度取决于第一个参数，其中，生命期参数的初始值由客户代理根据可以等待的时间来设置，而其余的3个参数则由服务代理进行设置。

在工作过程中，服务信息注册单元收集本域内服务代理所声明的可用服务信息并存储在服务信息库中。服务请求匹配与转发单元及其执行的转发算法是实现服务发现可扩展性的关键，该单元同时接收本域客户代理发送的服务请求和其他服务器转发过来的服务请求，并将其与本域的可用服务信息进行匹配操作，若满足则根据服务请求中的源地址响应服务请求，否则根据相邻地址表内容对服务请求传输报文中的参数进行处理，然后向某个邻结点转发。转发过程有可能一直进行下去，直到服务请求传输报文中的生命期计数器为0，或者服务请求被转发到根结点而终止。

3 RFSD中的服务请求转发算法

对服务请求进行转发是服务器之间的协作关系的核心。服务器收到一个服务请求后根据本地的可用服务信息对其进行匹配操作，在匹配满足的情况下响应服务请求，在不满足的情况下根据子、兄、父的顺序决定下一个结点并将服务请求转发给它。

3.1 基本假设

作以下基本假设：树结构的层数不超过 M ，每个结点的子结点数目不超过 X ；进行匹配操作时，只考虑精确匹配而不考虑模糊匹配的情况；由于服务信息和服务请求的表达方式不是本方案的重点，因此将这两者抽象为一段描述文本，并给出以下元类型：

DescriptionofService：按照一定语法规则定义的服务信息和服务请求的描述文本类型；

NetworkAddress：有关实体的网络地址类型；

ListofNetworkAddress：网络地址有序表类型。

3.2 有关定义

将服务请求传输报文定义为一个SRMType型对象，它没有方法，其属性有：

SRContent：服务请求的描述文本，类型为DescriptionofService；

SRAddress：发出该请求的客户的地址，类型为NetworkAddress；

remainder：大小为 $2M-1$ 的数组，数组中每一项是整型，表示服务请求所经过的各层结点的尚

未被访问的兄弟结点的数目；

pointer：正在处理本服务请求传输报文的结点在树中的相对层数，整型；

fromto：转发方向，0,1,2,3分别表示是本域客户，父结点、兄弟结点、子结点转发而来，整型；

counter：对服务请求生命期进行递减的计数器，整型；

将服务器定义为一个ServerType型对象，其属性有：

ASContent：服务器上存储的可用服务信息，DescriptionofService型；

FatherServer：父结点地址，NetworkAddress型；

NumofBrother,NumofChildren：分别为兄弟结点和子结点数目，整型；

BroServers,ChiServers：分别为兄弟结点地址有序表和子结点地址有序表，ListofNetworkAddress型；

服务器对象的方法有：

BOOL Matched(SR：SRMType)：对SR中的服务请求进行匹配操作，如满足则响应服务请求并返回TRUE，否则返回FALSE，每进行一次匹配操作时将SR中的counter减1；

BOOL IsLocalRequest(SR：SRMType)：判断SR是否来自本域客户，如果是则返回TRUE，否则返回FALSE；

NetworkAddress GetNextBrother()：取BroServers表中紧邻本结点的下一个兄弟结点地址，如果本结点地址在最后一项，则取表中的第一项；

NetworkAddress GetFirstChild()：取ChiServers表中第一个子结点地址；

ForwardRequest(SR：SRMType，NA：Network Address)：向地址为NA的结点转发SR；

DealWith(SR：SRMType)：处理所收到的服务请求传输报文SR，其描述见算法1。

算法1 服务请求转发算法

```
DealWith(SR: SRMType){
1  if((IsLocalRequest(SR)=TRUE){
2    if(Matched(SR)=TRUE) return;
3    for i = 0 until 2*M do SR.remainder[i] = X;
4    pointer = M; //设本域结点在树中的的相对层数为M;
5    SR.pointer = M;
6    SR.remainder[pointer] = NumofBrother;
7    SR.fromto = 0;
8  }else{//处理SR中的参数，在首次到达的结点进行匹配操作；
9    pointer = SR.pointer;
10   if((SR.fromto=1)OR(SR.fromto=2)){
11     if((Matched(SR)=TRUE)OR(SR.counter=0)) return;
12     SR.remainder[pointer] = SR.remainder[pointer]-1;
13   }
14   else if((SR.fromto=3)AND(SR.remainder[pointer]=M)){
15     if((Matched(SR)=TRUE)OR(SR.counter=0)) return;
16     SR.remainder[pointer] = NumofBrother;
17   }
18 }
19 //决定下一转发结点；
20 if(SR.fromto=3){
```

```

21  if((NumofBrother>0)AND(SR.remainer[pointer]>0)){
22      NextNode  GetNextBrother();
23      SR.fromto  2;
24  }
25  else{
26      if(FatherAddress=NULL) return;
27      NextNode  FatherAddress;
28      SR.pointer  SR.pointer-1;
29      SR.fromto  3;
30  }
31  }else if(NumofChildren>0){
32      NextNode  GetFirstChild();
33      SR.pointer  SR.pointer+1;
34      pointer  SR.pointer;
35      SR.remainer[pointer]  NumofChildren;
36      SR.fromto  1;
37  }else if((NumofBrother>0)AND(SR.remainer[pointer]>0)){
38      NextNode  GetNextBrother();
39      SR.fromto  2;
40  }else{
41      if(FatherAddress=NULL) return;
42      NextNode  FatherAddress;
43      SR.pointer  SR.pointer-1;
44      SR.fromto  3;
45  }
46  ForwardRequest(SR,NextNode);
48 }

```

3.3 算法说明

服务请求从本域服务器开始访问各个树结点，直至得到响应或超时而终止。在选择要转发到的下一结点时，需要综合考虑服务请求中4个参数和本结点的相邻结点。如果服务请求是由非子结点转发来的，则按照子结点、兄弟结点、父结点的优先顺序决定；如果服务请求是由子结点转发来的，则按兄弟结点和父结点的优先顺序决定。每次转发要对remainder, fromto和pointer的值进行判断和处理，

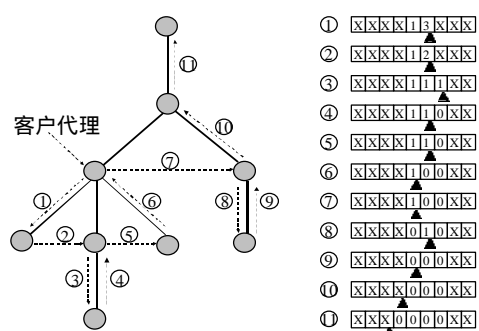


图2 服务请求在结点间转发的示意图

每次匹配操作中对counter的值进行递减。这样，服务请求便可以在每一个结点按先左子树，再右子树，后父结点的顺序转发到下一个结点直至得到响应或其生命周期结束。传输报文在各树结点间转发的一般情况如图2所示，其中 $M=5$ ， $X=4$ ， $counter=20$ ，虚线箭头表示转发方向，右边方框表示remainder数组中相应项的变化情况，小三角则是pointer的形象表示，fromto的变化情况可以从各结点的位置关系和虚线箭头方向体现出来。服务请求会重复到达一些非叶结点，但在其上不进行重复匹配操作。

4 结束语

本文实现了一个工作于集中模式的SLP系统。该系统采用<属性, 值>的集合及有关的逻辑连接符表示可用服务信息和服务请求; 采用IP地址表示三类实体的地址; 以TCP和UDP为通信手段实现了有关的交互报文^[1]: DAAdvert, ServiceRegister, ServiceRequest, ServiceReply, ServiceAcknowledgement。在此基础上进一步实现了由数台主机组成的RFSD模拟系统。每台主机充当RFSD服务器, 它和同一子网中的客户代理及服务代理组成一个RFSD域, 多个RFSD服务器可以通过手工配置的方式形成不同的树结构。实际运行表明: 该方案可以有效地实现服务请求在各服务器之间的转发, 从而使得客户可以在尽可能广的范围内查找所需的服务信息, 而其在网络和系统方面的开销很小。

RFSD方案中各服务器只存储本域中的服务信息, 因此避免了WASRV方案中各AA需要存储庞大数目的服务信息的缺点, 也省略了SSDS中由于周期性地对各子树的服务聚集信息进行生成、合并、传输和更新而产生的处理器和网络带宽方面的开销, 对服务的表达也没有特别的限制。其特殊之处在于为服务请求传输报文附加了一定长度的指示转发方向的信息。由于树结构可以进行手工配置和自动调整, 因此完全可以通过合理的配置机制使得树的层数较小从而使得附加信息也尽量少。分析和实验表明: 通过对服务请求传输报文附加一定的信息并对其进行转发是实现服务发现可扩展性的一种有效方法。

参 考 文 献

- 1 Guttman E. Service location protocol: automatic discovery of IP network service. IEEE Internet Computing, 1999, 3(4): 71-80
- 2 Keith W. Jini核心技术. 北京: 机械工业出版社, 2000
- 3 The Salutation Consortium. Salutation Architecture Specification. Version 2.0, June 2001, <http://www.salutation.org>
- 4 Microsoft. Universal plug and play device architecture. Version 1.0, June 2000, <http://www.microsoft.com/hwdev/tech/nonpc/UpnP/default.asp>
- 5 Rosenberg J, Suter B. Wide area service location. IETF Internet Draft, draft-ietf-srvloc-wasrv-00.txt.1997
- 6 Steven E, Ben Y, Todd H, *et al.* An architecture for a secure service discovery service. Fifth Annual International Conference on Mobile Computing and Networks August 1999 (MobiCom '99), 1999: 24-35