

# 基于移动自组网的服务发现体系结构设计

张程 朱庆生 陈自郁 刘慧君  
(重庆大学计算机学院 重庆 400030)

**摘要** 移动自组网(MANET)是自治的无基础设施的网络,在此基础上进行的服务发现体系有别于传统的服务发现体系,需要从网络传输、服务注册以及服务发现等方面充分考虑移动无线网络和移动设备的特性。现有的基于有线网络或者单点自组网协议的相关服务技术标准及协议用于 MANET 存在很大的局限性和不足。充分考虑移动自组网中设备的移动性和网络环境的不稳定性后,提出了一个基于分簇的体系架构设计,并在该体系架构中就服务传输协议、面向容错的分簇策略、非集中式的服务注册和服务发现机制等方面提出了针对性的设计方案。仿真结果显示,在移动自组网的特殊网络和设备环境下,这一新的体系架构和相关服务发现策略从服务匹配准确度、反馈时间等方面都优于传统方式的服务描述模式和服务发现机制。

**关键词** 移动自组网,服务发现,容错服务匹配

**中图分类号** TP312, TN929.5 **文献标识码** A

## Novel Framework for Service Discovery in MANET

ZHANG Cheng ZHU Qing-sheng CHEN Zi-yu LIU Hui-jun

(Computer College, Chongqing University, Chongqing 400030, China)

**Abstract** The mobile ad hoc networks (MANET) are autonomous, infrastructureless network. Unlike the common service discovery used in wired applications, the service applications based on MANET have so many differences in data transmission, service registration and service discovery. All component of service discovery of MANET need to consider the features of mobile devices and mobile environment. Considering the characters and requirements, this paper presented a new framework for service discovery in MANET, which supports SOAP-over-UDP and adopts cluster architecture. And to make the service working smoothly in such framework, this paper prompted special service registration and discovery strategies. The experiment results demonstrate that the new framework with special service working strategies has great superiority both in response-time and efficiency in MANET environment.

**Keywords** MANET, Service discovery, Fault-tolerant service-matching

## 1 引言

自组网(Ad Hoc networking)最早由 Perkins<sup>[1]</sup>提出,它是移动计算机(或移动结点)使用无线通信设备与其他移动计算机通信的网络。移动自组网 MANET (Mobile Ad Hoc Network)是一种没有固定路由器的网络。它是一群移动节点通过分布式协议自组织起来的一种无线网络,具有临时性、无任何基础设施、无集中管理的网络等特点。每个结点既做主机又做路由器,通过无线介质把数据包转发到其他结点。服务发现协议是指网络中的节点自动获得其它节点所提供服务的有关信息的技术。只有在服务发现协议的支持下,节点才能快速有效地发现网络中可利用的资源和功能,并充分利用这些服务为用户提供更完善的功能,实现互连互通、移动计算的理想。

与其他网络的服务发现应用不同,在移动自组网中,由于节点硬件配置较低、内存小、计算能力不强,服务发现协议必须简单高效;由于能量及无线信道带宽的有限性及无线传输的低可靠性,服务发现协议必须具有很小的信息包开销;由于移动自组网的拓扑易变性,服务发现协议必须健壮、灵活,具有自适应性;由于移动自组网有多达上千个节点,服务发现协议须具有良好扩展性;所以简单高效、健壮灵活、具有良好扩展性的服务发现协议是自组网实用化的迫切要求。

当前大多数的移动自组网服务发现协议是在基于有线网络服务发现协议的基础上修改得到的,如文献[2-4]等将 SLP 移植到移动自组网中。由于基础协议的限制,这类协议无法令人满意。文献[5-7]等研究了将服务发现功能嵌入到移动自组网路由协议的效果。文献[8-13]等提出了几种平面型移动自组网服务发现协议。文献[14, 15]提出了多层服务发现

到稿日期:2010-01-20 返修日期:2010-04-01 本文受中央高校基本科研业务费(CDJRC10180006),科技部国家科技攻关计划重点项目(2007BAH08B04)资助。

张程(1977-),男,博士生,主要研究方向为 Web 智能、移动运算, E-mail: zc19999@sina.com;朱庆生(1956-),男,博士,教授,主要研究方向为多媒体信息技术、软件方法和重用技术、电子商务等;陈自郁(1977-),女,博士,讲师,主要研究方向为智能算法等;刘慧君(1975-),男,博士生,讲师,主要研究方向为 Web 挖掘等。

协议。但是,由于移动自组网自身的特点及其应用的典型性,使得服务类应用与现有的有线环境中的应用存在着很大的差异。基于传统的服务标准与协议进行的修改很难满足移动自组网中对于服务应用的实用化需要。本文则从服务发现协议的功能要素和工作过程入手,提出了可满足移动自组网特性需求的服务发现体系架构设计。

MWSCSP(Mobile Web Service Discovery Clustering & Semantic-based Platform)系统是在移动自组网研究中开发的一个实验平台,它主要通过移动自组网中的移动终端(例如 PDA、手机等)和蓝牙网络来共同搭建,主要应用于研究如何实现在无线自组网中以服务发现的方式进行相互信息沟通和应用互助的问题。本文 2.1 节介绍 SOAP-over-UDP 服务传输协议;2.2 节讨论了基于 MWSCSP 结构的服务注册及发现机制;2.3 节提出了容错服务匹配问题及相关策略。第 3 节介绍有关的仿真实验及实验结果对比。第 4 节则对本文研究进行总结并概述未来的主要工作目标。

## 2 服务发现体系架构设计

### 2.1 SOAP-over-UDP

作为服务的重要组件,SOAP 为异构环境提供了一个标准化机制。W3C 组织在定义 SOAP 标准时并没有限制 SOAP 消息绑定的传输协议<sup>[16]</sup>,当前常用的服务传输主要通过 SOAP-over-HTTP 方式进行。虽然 HTTP 广泛应用于当前的服务系统中,但对于移动自组网中的服务而言,HTTP 协议并不是一种有效的可用协议,其原因在于 HTTP 使用的底层传输协议 TCP 协议在无线移动环境中存在严重局限<sup>[17]</sup>,主要有:

1) TCP 协议的避免拥塞机制认为丢包主要是由网络拥塞造成的。而在无线网络中,连接中断或传输错误才是造成数据包丢失的最主要原因。

2) TCP 协议执行了 3 次“握手”过程。在发送数据前,首先要建立连接,然后传输数据,数据传输完成后,又需要再次通信来关闭连接。这些对于无线移动设备间网络传输而言,都是一些额外的任务和负担,白白消耗移动自组网中的有限带宽和电量。

相比 SOAP-over-HTTP,SOAP-over-UDP 是一种更优的选择。与 TCP 不同,UDP 是一个简单的、低负荷的传输协议,在无线环境中比 TCP 拥有更佳的优势:

1) UDP 无需建立连接即可传送数据,减少了创建连接消耗的时间和数据包;

2) UDP 数据包比 TCP 数据包小很多;

3) UDP 支持广播。

当然,UDP 传输协议可能导致重复发送、无序抵达,甚至丢包的情况。但由于其比 TCP 协议更适用于移动环境,本文选择了 UDP 协议作为消息传输的底层绑定协议,并在服务的匹配和发现策略上增加了一些补充处理机制(参见 2.3 节)。BEA, Lexmark, Ricoh 和 Microsoft 共同发布的 SOAP-over-UDP 规范<sup>[18]</sup>为本平台应用 SOAP-over-UDP 协议提供了参考依据。在具体实现中,本平台主要强调了 UDP 的以下 4 类消息特征:

1) 单向组播;

2) 单向广播;

3) 服务需求通过组播方式发送,并通过组播方式接收反馈;

4) 服务需求通过广播方式发送,但通过组播方式接收反馈。

### 2.2 服务注册及服务发现算法

在传统的服务规格中,UDDI 是一个重要的功能组件,其承担了服务注册和服务匹配等功能。但是,考虑到移动自组网的特殊性,尤其是移动终端设备可能由于电源、位置的变化而动态变化的情况,本文定义了专门的服务注册和发现机制。

#### 2.2.1 摘客

摘客,作为本平台分簇结构中簇内节点的管理者,不仅仅是路由和拓扑层面上的管理者,也是服务管理者和协作者。其具体特点如下:

1) 摘客是本簇的服务管理者

① 摘客本身也是一个相对计算能力和存储能力较强的移动终端。

② 摘客无法像 UDDI 服务器那样实现大容量存储和计算功能。因此,服务提供者们在摘客中注册的主要是该服务的类型和能力信息(例如计算能力、可持续电量等)。

③ 摘客不直接承担 UDDI 服务器的类似服务匹配计算功能。

2) 摘客是簇间沟通的协作者和路由决定者

① 摘客通过 Cache 保存其他簇的一些服务信息。

② 摘客中保存有最相邻簇的位置信息。

#### 2.2.2 服务注册

移动设备发布服务时,将服务元数据信息封装后,向本簇的摘客发布信息,并遵循算法 1 完成服务注册工作。

##### 算法 1 服务注册算法

```
Begin
初始化,服务提供者设置计数器清 0;
while(计数器≤5){
    服务提供者发出服务元数据描述;
    服务提供者设置计时器清 0;
    while(! 计时器超时){
        等待反馈;
        if(接收到反馈信息){
            注册成功;
            跳至“结束”;}
        计数器+1;}
本簇摘客出现故障,退出本簇;
服务提供者向其他簇发出加入申请;
结束;
End
```

摘客收到注册信息后,从中提取出服务基本分类信息、用户位置信息和设备动态信息(主要是设备类型、电源信息),然后增加当前时间并保存在本地 Cache。接着,摘客将反馈信息封装后发回给服务注册者。

#### 2.2.3 服务发现机制

簇内的无线设备需要服务时,将服务需求描述、本设备的用户信息描述和设备信息描述封装成为一个服务需求元数据后,发往本簇的摘客,并遵循算法 2 进行处理。

##### 算法 2 服务发现算法

Begin

设备向捐客发出服务需求描述信息;

捐客查看注册信息;

if(簇内有与该需求类型一致的服务注册){

簇内广播该服务需求;

设置计时器,等待反馈信息;}

else{

if(Cache中存在类型一致的服务注册信息){

读取这些服务一致的簇位置信息;

将本簇位置信息、时间等封装到该服务需求中

将服务需求转发给这些簇的捐客;

设置计时器,等待反馈信息;}

else {

将本簇位置信息、时间等封装到该服务需求中

将服务需求转发给相邻簇的捐客;

设置计时器,等待反馈信息;}

}

End

在簇间搜索过程中,捐客收到来自其他簇的服务匹配信息后,会将该簇的名称、位置和服务类型保存在自己的 Cache 中,从而形成了类似多个路由器之间路由表的方式过程。

### 2.3 容错服务匹配策略

#### 2.3.1 问题的提出

“虚假的”匹配不成功现象:当捐客在簇内广播服务匹配信息后,各 Web 服务提供者都接收到该信息并返回匹配结果。对于“匹配成功”的消息,由于移动自组网络自身的不稳定以及本文选择 UDP 协议传送数据,必然存在丢包的可能。如果“匹配成功”消息丢失,就使得原本成功的服务匹配由于网络原因而最终返回给捐客“不成功”的结果,这即是一个“虚假的”匹配不成功信息。

为了解决这类“虚假的”匹配不成功问题,本文借鉴了离散空间上的容错搜索模型并在此基础上展开研究,其具体情况如下。

#### 2.3.2 Ulam 问题

1976 年,Ulam<sup>[19]</sup>在他的自传中提出了有趣的双人搜索游戏问题,其具体内容如下:

A 和 B 人进行如下游戏:A 想了一件事,B 通过一系列提问设法猜出它。对于 B 所提的每一个问题,A 仅以“是”或“否”作答,并且允许 A 所给出的回答中固定比例是错误的,试确定 B 能够正确猜出 A 所想的事情的最少提问次数。

例如一个人在 1 到 1000000 之间想了一个数( $1000000 < 2^{20}$ ),另一个人最多允许提出 20 个问题。对于提问者的每个问题,假定回答者只以“是”或“否”作答。假如提问者第一次这样问:“这个数在 1 到 500000 之间吗?”,如果回答者回答“是”,那么提问者第二次这样问:“这个数在 1 到 250000 之间吗?”,否则提问“这个数在 250000 到 500000 之间吗?”,以后就按这样的方法问下去。显然,提问者至多用  $\log_2 1000000$  次提问就能确定出这个数。现在,如果回答者被允许撒谎一次或者两次,那么提问者至少需要多少次提问才能确定出这个数?

离散空间上的容错搜索 Ulam 问题也可以描述为下面的二人对策游戏。提问者和回答者在游戏开始以前,就以下事项达成协议:

条款 1 搜索目标所在的范围,即搜索空间;其中,有界空间通常用  $\{1, 2, \dots, M\}$  表示,  $M$  代表搜索空间的大小;无界空间,通常用  $\{1, 2, \dots\}$  表示。

条款 2 搜索目标的类型;一般取决于搜索目标的个数  $N$ 。  $N=1$  时称为单目标,  $N \geq 2$  时称为多目标。

条款 3 回答者的差错模式;主要类型有:

1) 固定差错总数:回答者在整个游戏过程中至多说谎  $e$  次,  $e$  为一个固定的正整数。

2) 全局有界的差错比例:如果整个游戏进行  $n$  次提问,回答者能够说谎至多  $pn$  次,  $p \in [0, 1]$ 。

3) 前缀有界的差错比例:在前  $m$  次提问中回答者能够说谎至多  $pm$  次,  $p \in [0, 1]$ 。

4) 随机差错:差错次数是随机的。即在每次回答之前,回答者掷一枚硬币,正面说谎,反面不说谎。

5) 半谎模式:回答者在整个游戏过程中至多说谎  $e$  次,但仅以回答“否”时可以说谎,所有回答“是”均保证是正确的。

条款 4 提问者的提问格式;主要类型有:

1)  $q$  维自由提问格式( $q \geq 2$  是任意一个固定的正整数):提问形如“未知目标  $x^*$  属于  $A_1, A_2, \dots, A_q$  的哪一个集合?”,这里  $A_1, A_2, \dots, A_q$  是搜索空间的任意一个部分。对于提问者  $Q$  的每次提问,回答者  $R$  以  $i$  作答,这里  $i \in \{1, 2, \dots, q\}$ 。

2) 比较型提问格式:提问形如“ $x < a$ ?”,这里  $a$  是搜索空间的任意一个元素。对于提问者  $Q$  的每次提问,回答者  $R$  只以“Yes”或“No”作答。

3) 区间型提问格式:提问形如“ $x \in [a, b]$ ?,  $a \leq b$ ”,对于提问者  $Q$  的每次提问,回答者  $R$  只以“Yes”或“No”作答。

4) 双区间型提问格式:提问形如“ $x \in [a, b] \cup [c, d]$ ?,  $a \leq b$  且  $c \leq d$ ”。对于提问者  $Q$  的每次提问,回答者  $R$  只以“Yes”或“No”作答。

5) 大小受限自由提问格式:提问形如“ $x \in A$ ?”,这里  $|A|$  不能超过一个给定的上界  $k$ 。对于提问者  $Q$  的每次提问,回答者  $R$  只以“Yes”或“No”作答。

6) 可变费用提问:提问者对不同的提问支付相应的费用,提问总费用有一个给定的限制。

7) 非重复提问:同一个问题不允许被重复提问。

条款 5 提问者和回答者的交互程度。主要类型有:

1) 适应的:提问者进行第  $j$  次提问之前已获得前  $j-1$  次提问的答案,即提出第 1 个问题,等待回答,提出第 2 个问题,等待回答,...

2)  $k$  批提问:提问分  $k$  批进行,这里  $k$  是一个固定的正整数,提第一批问题然后作答,再提第 2 批问题然后作答,...

3) 非适应的:即 1 批问题提问与作答,所有提问应该一次性提出,不等待任何作答;

4) 假定回答者在搜索空间  $S$  中选取了搜索目标(对提问者来说是秘密),接着提问者根据协议提出一系列问题而回答者根据协议作答。如果提问者能在  $k$  次提问和  $k$  次回答之后识别出所有搜索目标,则提问者取胜;否则,回答者取胜。称提问者有一个长度为  $k$  的取胜策略,即如果他在  $k$  次提问和  $k$  次回答之后总能取胜,不管回答者如何作答(只要不与协议冲突)。

#### 2.3.3 基于分簇的容错服务匹配策略

Ulam 问题是容错搜索的里程碑式的数学基础之一。其

模型为离散空间的容错搜索提供了数学工具。2.3.1节中提到的“服务成功匹配信息的丢失”问题虽然并不是一个单纯的离散空间容错搜索问题,但经过一定的转化后,仍然可以视为是一种典型的 Ulam 问题。其基本转换思路如下:

在移动自组网中,捐客向本簇内所有已注册的服务提供者广播服务匹配的需求信息,服务提供者接收到服务需求信息后,进行内部判断与识别,然后返回匹配成功与否的信息给捐客。如果返回值为匹配不成功,无论该返回值被捐客成功接收或者该返回信息由于 UDP 丢包而造成反馈超时,都可以认为是答案“否”,同时这是一个“正确”的答案;而如果返回值为“匹配成功”,但是由于 UDP 包丢失而造成反馈超时,结果捐客误认为答案是“否”,则这是一个“错误”的答复;当然,如果匹配成功,同时该反馈信息及时被捐客获取,则这是一个“正确”的答案,且答案为“是”。

基于上述映射推理,本文将分簇架构下的服务匹配搜索问题转换为 Ulam 问题,并做以下限定:

条款 1 在分簇架构的服务匹配过程中,搜索空间有界,本文用  $\{1, 2, \dots, M\}$  来表示。 $M$  是一个正整数,且  $M$  值为该簇中的服务提供者数量。

条款 2 由于在服务匹配过程中,搜索过程为根据一个服务需求在簇内的服务提供者中搜索其匹配度,然后返回是否匹配的结果,因此其搜索目标只有一个,而不是多目标的。

条款 3 对于服务匹配搜索而言,如果捐客收到了“匹配成功”的消息(即回答为“是”),这个答案一定是真实的,因为其意味着在簇内有移动终端提供所需的服务;而如果捐客收到为“否”的结果,该结果可能源自两种情况:簇内确实没有匹配成功的服务提供者或匹配成功但由于反馈超时而导致捐客收到的结果为“否”。这样的话,对于“否”这个结果而言,其可能是一个“虚假”的结果。结合 2.3.2 节中的条款 3 可以获知,本搜索匹配算法对应的 Ulam 问题的错误模式是“半谎”模式。即,只有答复“否”有可能是在撒谎,而答复“是”一定是诚实可靠的。且每轮搜索的最大撒谎次数为  $e$ 。此次,  $e$  是一个设定值,如果  $e$  取值过大,则簇内会反复发送服务需求并接收反馈,而  $e$  取值过小,则可能由于 UDP 网络传输异常而导致每次接收反馈都失败。考虑到服务匹配时应能够快速获得是否匹配成功的结论,同时移动无线网络的带宽有限,本次研究中设定  $e=4$ 。

条款 4 Ulam 问题格式是  $M$ \_问题:即如果搜索空间是  $S$ ,而搜索目标是  $x^*$ ,那么集合组  $T_1, T_2, \dots, T_m$  是集合  $S$  的细分( $\bigcup_{i=1}^m T_i = S; T_i \cap T_j = \Phi, i \neq j$ )。因此,本架构中的 Ulam 问题可理解为“服务提供者  $T_i$  是否匹配服务需求  $x^*$ ?”且  $i$  属于集合  $\{1, 2, \dots, M\}$ 。

条款 5 提问者与应答者之间的交互程度是自适应的,即当提问者提出第  $j$  个问题时,他已经得到了前面的  $j-1$  个答案。

有了上述约定后,本文的基于容错的分簇服务匹配搜索过程也就可以转换为满足上述 5 项条款的 Ulam 问题求解过程。而作为一类典型的离散空间容错搜索问题,Ulam 数学模型及其求解过程给我们提供了极大的帮助。

在文献[20]中,Vincenzo 针对允许撒谎 4 次的半谎模式进行了数学推算,以确定在搜索空间为  $[1, 100000]$  的条件下,取得 Ulam 问题求解的最小提问次数。其中,对于搜索 17 次

的 Ulam 问题,Auletta 通过数学推理和证明验证了如下命题:

Proposition 6<sup>[20]</sup>: Let  $(1, 2, C_{20}^0, C_{20}^1, C_{20}^2)$  be an Ulam set,  $U$  is 17-solvable

从命题中我们可以获知,在允许搜索 17 次,允许撒谎 4 次,且搜索范围为 4845 个元素的情况下,可以得到 Ulam 的求解。结合本文的容错服务匹配算法的需求,可以得到以下推导:在簇内服务提供者数量不超过 4845 个、允许对某个服务提供者的匹配需求反馈超时 4 次以及捐客发送不超过 17 次请求的前提下,如果得到“匹配成功”的反馈,则匹配成功,否则认为“匹配失败”为真实的结论。并可得到以下结论:

结论 1 如果本文把搜索范围局限在 4845 以内,则允许撒谎 4 次的 Ulam 问题成功搜索次数为 17。

根据结论 1,在面向服务的分簇策略中,对于每个簇中服务提供者的数量须做以下限制:每个簇中的服务提供者设备终端数量不超过 4845 个。

同时,对于服务请求和匹配,做以下约定:针对每个服务请求,捐客都会在本簇内多次发送服务匹配广播,直至收到成功匹配的消息,或广播次数达到 17 为止。如果捐客广播服务匹配消息次数达到 17 之后仍没有得到成功匹配的信息,则认为本簇内不存在满足该服务匹配的服务提供者。该服务匹配被发送到邻接簇中继续搜索,同时启动计时器,参见算法 3。

#### 算法 3 支持容错匹配的服务发现算法

Begin

设备向捐客发出服务需求描述信息;

捐客查看注册信息;

if(簇内有与该需求类型一致的服务注册){

    设置计数器  $n=17$

    While( $n>0$ ) {

        簇内广播该服务需求;

        设置计时器,等待反馈信息;

        If(收到匹配成功消息)

            将匹配成功的节点信息发给需求设备,服务发现算法结束

        If(计时满 and 未收到匹配成功消息)

$n$  减一操作}

}

if(Cache 中存在类型一致的服务注册信息){

    读取这些服务一致的簇位置信息;

    将本簇位置信息、时间等封装到该服务需求中

    将服务需求转发给这些簇的捐客;

    设置计时器,等待反馈信息;}

else {

    将本簇位置信息、时间等封装到该服务需求中

    将服务需求转发给相邻簇的捐客;

    设置计时器,等待反馈信息;}

End

同时,对于基于容错的服务注册机制也相应作出调整,详情参见算法 4。

#### 算法 4 基于容错匹配的捐客管理算法

Begin

分簇初始化,簇首自动成为捐客;

捐客将簇内服务提供者大小  $P$  设置为 4845;

while( $P>0$ ) {

    接收到服务提供者发出的服务元数据描述;

```

将服务元数据基本信息保存在 Cache 中;
计时器清 0,向服务提供者发送注册成功信息;
while(! 计时器超时){
    等待反馈;
    if(接收到反馈信息){
        注册信息发送成功;
        跳至“结束”;}
    计数器+1;}

```

```

发送“注册满”信息;
服务提供者退出本簇,向其他簇发出加入申请;
结束;
End

```

### 3 实验

#### 3.1 实验环境搭建

在仿真实验中采用了 Franson Bluetools<sup>[21]</sup> 和 Pocket-SOAP 工具包。Franson Bluetools 是开发 .net 环境下的蓝牙应用的快速开发工具,而 PocketSOAP 是一个基于 Windows 的 SOAP 客户端工具。此外,通过蓝牙技术来实现自组网的搭建,并使用了 5 个支持蓝牙传输的移动设备和 2 个带蓝牙适配器的电脑。整个实验环境的相关设备如表 1 所列。

表 1 实验设备列表

设备名称	操作系统	处理器	RAM
Mobile Device I	Windows Mobile 6.1	Qualcomm 528 MHz	100M
Mobile Device II	Windows Mobile 6.1	Qualcomm 528 MHz	70M
Mobile Device III	Windows Mobile 6.1	Qualcomm 528 MHz	107M
Mobile Device IV	Windows Mobile 6.1	Qualcomm 528 MHz	68M
Mobile Device V	Windows Mobile 6.1	Qualcomm 528 MHz	90M
PC I	Windows XP SP2	Intel Pentium4 1.8GHz	512M
PC II	Windows XP SP1	AMD Athlon 1.6G	256M

#### 3.2 基于分簇结构的服务发现模型与广播模式的比较

本实验主要验证 2.2 节提出的基于分簇架构的服务注册与匹配算法的性能效果。作为对比参照物,本实验模拟了没有分簇架构的服务广播模式进行比较。在分簇服务发现模型中,将 5 个移动设备终端作为同一个簇内成员,然后使用 Load Runner 软件来模拟其他簇和其他移动终端设备,从而架设了一个有 3 个簇,每个簇内有 150 个终端,共 450 个移动终端的分簇环境;广播模式的对比实验中保持同样数量的 450 个移动终端设备,但不分簇,而直接选择广播模式。注意:上述两种实验环境下的终端都可作为服务提供者工作。然后,通过电脑来发出服务请求。服务请求的发送频率从 0.1/秒到 1/秒递增,递增梯度为 0.1。在 0.1/秒的发送频率时,测试了 300 次,0.2/秒时测试 350 次,0.3/秒时测试 400 次,以此类推。最后记录在两种模式下每次服务的反馈时间并取平均值。图 1 是本实验结果的比较图。

当服务需求产生后,在本文的分簇服务发现模型中,服务需求快速在簇内匹配搜索,如果簇内不存在匹配,才在其他簇内进行匹配搜索。而广播模式下,任何一个服务需求都需要在 450 个终端范围内广播。从图 2 可以看出,本文的基于分簇服务发现模型与广播模型相比,其反馈延时最坏的时候也不到广播模型的 2/3,而在 1/秒的频率下,几乎不到广播模型的 1/4。

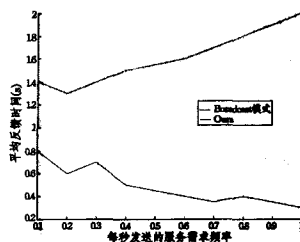


图 1 不同服务需求频率下的平均反馈时间

#### 3.3 服务发现机制的比较

作为对比参照物,本实验选择了在当前服务研究中很热门的基于语义 Web 服务发现的两个典型服务发现策略进行对比,它们分别是 OWL-S/UDDI 和 SSQD<sup>[22]</sup>。

在实验中,将移动终端移动到蓝牙网络覆盖范围的最边缘区域。检查发现,该位置的信号很不稳定,且数据包丢失率分别为 47% 和 79%。然后基于上述两种服务发现策略和本节提出的服务发现策略分别执行了 100 次服务匹配测试,记录下每次匹配的成功率、反馈时间和传输数据大小,并分别取平均值,如表 2 所列。

表 2 Ours, OWL-S/UDDI 和 SSQD 的不同表现情况

	Ours	OWL-S/UDDI	SSQD
成功率(%)	100	100	100
丢包率为 0 反馈时间(ms)	4	13	18
传输数据(kB)	1	1	1.3
成功率(%)	83.5	100	100
丢包率为 47% 反馈时间(ms)	62	164	211
传输数据(kB)	20	240	330
成功率(%)	48	22	100
丢包率为 79% 反馈时间(ms)	62	680	900
传输数据(kB)	20	1000	1400

从表 2 的统计结果来看,在一个良好的无线网络环境中,本文的方法(Ours)与其他两种方法取得相同的结果,但耗时和数据包都相对较小。但是在网络环境变坏的情况下,Ours 可以在很短的时间内结束服务发现过程,而 SSQD 虽然成功匹配率为 100%,但其耗时和网络传输数据量都太大,完全无法在真实的移动自组网应用中实现。

**结束语** 与传统的服务发现应用不同,移动自组网中的服务发现面对移动设备和移动环境的诸多限制和特点。因此,在无线自组网中的服务发现中,我们使用了 SOAP-over-UDP 传输协议,采用分簇结构和拥塞注册管理的方式来实现自主网中的服务注册和匹配发现等。对比实验表明, MWSCSP 平台设计在无线移动环境下的可用性和可靠性都高于传统的 Web 服务技术,而分簇结构设计解决了自主网络中集中管理、网络拥堵等问题。在接下来的研究中,我们将重点研究基于服务匹配策略的分簇路由算法和参考负载均衡因素的智能分簇算法。

#### 参考文献

- [1] Lei H, Perkins C. Ad hoc networking with mobile IP [C]// the Second European Personal Mobile Communications Conference (EPMCC'97). Bonn, Germany, 1997:197-202
- [2] Motegi S, Yoshihara K, Horiuchi H. Service Discovery for Wireless Ad Hoc Networks [C]// Proceedings of the 5th International Symposium on Wireless Personal Multimedia Communications (WPMC'02). Honolulu, Hawaii, USA, 2002(1): 232-236
- [3] Engelstad P E, Zheng Y, Jonvik T, et al. Service Discovery and

- Name Resolution Architectures for on-Demand MANETs [C]// Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW'03). Providence, USA, 2003; 736-742
- [4] Engelstad P E, Zheng Y. Evaluation of Service Discovery Architectures for Mobile Ad Hoc Networks [C]// Proceedings of the 2nd annual conference on Wireless On-demand Networks and Services (WONS'OS). St. Moritz, Switzerland, 2005; 2-15
- [5] Varshavsky A, Reid B, Lara E D. The Need for Cross-layer Service Discovery[P]. MANETs. Technical Report CSRG-492. Department of Computer Science, University of Toronto, 2004
- [6] Ververidis C N, Polyzos G C. Routing Layer Support for Service Discovery in Mobile Ad Hoc Networks [C]// Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications-Pervasive Wireless Networking Workshop (PerCom'OS). Kauai Island, Hawaii, USA, 2005; 258-262
- [7] Ververidis C N, Polyzos G C. Extended ZRP: Performance Evaluation of a Routing Layer based Service Discovery Protocol for Mobile Ad Hoc Networks [C]// Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'OS). San Diego, California, USA, 2005; 114-123
- [8] Azondekon V, Barbeau M, Liscano R. Service Selection in Networks based on Proximity Confirmation using Infrared [C]// Proceedings of the International Conference on Telecommunications (ICT'02). Beijing, China, 2002; 116-120
- [9] Meier R, Cahill V, Nedos A, et al. Proximity-based Service Discovery in Mobile Ad Hoc Networks [C]// Proceedings of the 5th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS'OS). Lecture Notes in Computer Science. v 3543, Athens, Greece, 2005; 115-129
- [10] Sen R, Handorean R, Roman G C, et al. Knowledge-driven Interactions with Services across Ad Hoc Networks [C]// Proceedings of the 2nd international conference on Service oriented computing (ICSOC'04). New York, USA, 2004; 222-231
- [11] Lenders V, May M, Plattner B. Service Discovery in Mobile Ad Hoc Networks; a Field Theoretic Approach [J]. Elsevier Journal on Pervasive and Mobile Computing, 2005, 1(3); 343-370
- [12] Bashir S, Li B C. KELOP: Distributed Key-Value Lookup in Wireless Ad Hoc Networks [C]// Proceedings of the 12th IEEE International Conference on Computer Communications and Networks (ICCCN'03). Dallas, Texas, USA, 2003; 471-476
- [13] Chakraborty D, Joshi A, Yesha Y, et al. GSD: a Novel Group-based Service Discovery Protocol for MANETs [C]// Proceedings of the 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN'02). Stockholm, Sweden, 2002; 140-144
- [14] Klein M, Ries B K, Obreiter P. Service rings; a Semantic Overlay for Service Discovery in Ad Hoc Networks [C]// Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA'03). Prague, Czech, 2003; 180-185
- [15] Klein M, Ries B K. Multi-layer Clusters in Ad-Hoc Networks-an Approach to Service Discovery [C]// Proceedings of the 1st International Workshop on Peer-to-Peer Computing (IWP2PC'02). Pisa, Italy, 2002; 187-201
- [16] Simple Object Access Protocol (SOAP) [EB/OL]. <http://www.w3.org/TR/soap>
- [17] Kwong Yuen Lai, Thi Khoi Anh Phan, Zahir Tari. Efficient SOAP Binding for Mobile Web Services [C]// the IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05); 987-1004
- [18] Harold C, Martin G. SOAP-Over-UDP[R]. Technique Specifications of Web service, Sep. 2004
- [19] Ulam S. Adventures of a Mathematician[M]. Scibner, 1976
- [20] Auletta V, Negro A. Solution of Ulam's Problem on Binary Search with Four Lies[J]. Combin. Theory, Ser. A, 1993(54); 1-19
- [21] Bluetooth [EB/OL]. <http://franson.com/bluetooth>
- [22] 吴芸. 基于语义的 Web 服务发现研究[D]. 西安: 西北大学计算机学院, 2006

(上接第 34 页)

周期里, 节点在侦听时间内侦听信道, 这样就造成了空闲侦听浪费。而 RT 业务比 BE 业务有更多的侦听时间, 因此比 BE 业务造成了更多的空闲侦听浪费。在成功发送的周期或者碰撞周期内, 节点在侦听到其他节点的 RT 时进入休眠, 从而避免了更多空闲侦听。

**结束语** 本文针对无线传感器网络不能提供数据业务优先级区分的问题, 提出了一种基于无线传感器网络 S-MAC 协议的改进机制 DQ-MAC。让高优先级数据有更多的侦听时间, 从而有更多发送数据的机会, 以达到减少高优先级数据的传输时延的目的, 从而确保高优先级数据业务的传输性能。相关分析结果表明, 此机制能够使高优先级数据业务拥有更高的吞吐量, 并且降低了高优先级业务的端到端平均时延。

我们下一步的工作将继续深入探讨在无线传感器网络中如何引入区分服务模型, 以及如何提高实时多媒体业务在无线网络中的传输性能。

### 参 考 文 献

- [1] Ye W, Heidemann J, Estrin D. Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks [J]. IEEE/ACM Transaction on Networking, 2004, 12(3); 493-506
- [2] Dam T V, Langendoen K. An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks [C]// Proc. of the 1st ACM Conference on Embedded Networked Sensor Systems (SenSys). Los Angeles, CA, USA, Nov. 2003; 171-180
- [3] Polastre J, Hill J, Culler D. Versatile low power media access for wireless sensor networks [C]// Proc. of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys). Baltimore, MD, USA, Nov. 2004; 95-107
- [4] Hoon K, Sung-Gi M. Priority-based QoS MAC protocol for wireless sensor networks [C]// Proc. of the 2009 IEEE International Symposium on Parallel & Distributed Processing. Rome, Italy, May 2009; 1-8
- [5] Zhang Ye, He Chen, Jiang Lingge. Modeling the S-MAC Protocol in Single-Hop Wireless Sensor Networks [C]// Proc. of IEEE ICC08. Beijing, China, 2008; 317-321
- [6] Bianchi G. Performance analysis of the IEEE802.11 distributed coordination function [J]. IEEE Journal on Selected Areas in Communications, 2000, 18(3); 535-547
- [7] Zhai H, Kwon Y, Fang Y. Performance analysis of IEEE802.11 MAC protocols in wireless Lans [J]. Journal of Wireless Communications and Mobile Computing, 2004, 4(8); 917-931
- [8] Bianchi G, Tinnirello I. Remark on IEEE 802.11 DCF Performance Analysis [J]. IEEE Communications Letters, 2005, 9(8); 765-767
- [9] Sung S, Kang H, Kim E. Energy Consumption Analysis of S-MAC Protocol in Single-Hop Wireless Sensor Networks [C]// Proc. of IEEE Asia-Pacific Conference on Communications (APCC). Busan, Korea, 2006; 1-5