

基于移动 Ad hoc 网络的服务管理研究^{*}

姜月秋^{1,2} 刘勇兵¹ 赵林亮¹ 王光兴¹

(东北大学信息科学与工程学院 沈阳 110004)¹ (沈阳理工大学通信与网络工程中心 沈阳 110168)²

摘要 本文基于移动 Ad hoc 网络的广播特性、分簇的普遍性等特点,在分析分布式服务发现协议 GSD 等的基础上,提出了新的服务发现机制,该机制考虑了各节点处理能力、存储能力以及所提供的服务种类等不可能完全相同的特性,克服了 GSD 同等对待所有节点的缺欠;另外,文中提出了基于优先级和两步资源预约的服务调度和 QoS 分配方式,依据服务所请求的 QoS 级别和资源情况来确定合适的端到端 QoS 级别,保证资源的充分利用,并给出了算法的一个实际应用的例子,该算法考虑了资源的抢占和两步方式的资源预约,比普通的确定 QoS 级别的算法具有更高的预约成功率,而比仅依据优先级的算法具有更高的资源利用率。

关键词 移动 Ad hoc 网络,网络管理,服务管理,服务调度,服务发现,服务描述

Research on Service Management Based on Mobile Ad hoc Network

JIANG Yue-Qiu^{1,2} LIU Yong-Bing¹ ZHAO Lin-Liang¹ WANG Guang-Xing¹

(School of Information Science & Engineering, Northeastern University, Shenyang 110004)¹

(Communication and Network Institute, Shenyang Institute of Technology, Shenyang 110168)²

Abstract Based on the broadcasting characteristic and generality of clustering in mobile Ad hoc network, this paper presents a new service discovery mechanism based on the analysis of the distributed service discovery protocol GSD. The mechanism considers the processing, storage ability and provides services difference of different nodes rather than considering each node entirely peer-to-peer as in GSD. Furthermore, this paper presents a new service schedule algorithm based on priority and two-phase resource reservation and a QoS assignment mechanism which determines the end-to-end QoS level according to the requested QoS level and resources state. An example using the algorithm is presented. The algorithm has higher reservation success rate and higher resource usage considering the resource preemption and two-phase reservation mechanism.

Keywords Mobile Ad hoc network, Network management, Service management, Service schedule, Service discovery, Service description

1 前言

近年来,随着网络技术和无线通信技术的飞速发展,移动 Ad hoc 网络——MANET(Mobile Ad hoc Network),这种具有广阔应用前景的新型网络正逐渐成为无线网络研究领域的热点。Ad hoc 一词来源于拉丁语,是“特别地,专门地为某一即将发生的特定目标、事件或局势而不为其他的”的意思。不同于有中心的集中式控制的蜂窝移动通信系统,Ad hoc 网络是一种有特殊用途的对等式网络,网络中各节点相互作为邻居节点的路由器,通过节点转发实现通信,这使其可以不受通信基础设施的约束和限制,在某些特殊环境或紧急情况下,能够被快速搭建并成为可以自由移动的局域网络,从而很好地适合如战场推进中的部队通信以及发生地震、水灾后的营救等应用场合。

移动 Ad hoc 网络是一种多跳的临时自治系统,由于拓扑结构具有动态、快速变化、任意移动等特性,被视为一种强壮而有效的无线移动网络。而在任何网络的建设中,控制网络,使网络具有最高效率和可靠工作的网络管理^[1]都是一个必需

内容,尤其是随着 Ad hoc 网络应用的不断扩展,其规模和应用范围急剧扩大,网络所承担的任务种类愈来愈多,如何成功地对 Ad hoc 网络进行管理和控制,提高网络的性能和服务质量,保证信息的安全和可靠性传输,已成为一项迫切的任务。

在网络管理领域,对服务管理的研究是一个崭新的内容,服务是指程序或者硬件设备提供给其它程序或设备使用的功能。通常用户只知道需要什么样的服务,而不知道如何去获取所需的服务。因此,服务管理包括服务发现模式、对服务的描述和服务调度等内容。

2 服务发现机制的设计

服务发现的主要目标就是提供一种机制:客户根据所需服务的类型和属性在网络中提交对服务的请求,网络返回满足其要求的服务的位置信息,然后客户根据位置信息动态绑定并访问所需服务;客户不必像以前那样事先静态地配置好各种与服务访问有关的参数,而是可以根据需要对服务进行动态的、及时的配置和访问^[2]。为达到这一目标,服务发现必须解决发现模式和服务描述两个基本问题,其中发现模式是

^{*}课题受国家 863 基金资助,资助编号 2003AA712032。姜月秋 博士后,副教授,主要研究方向:网络管理,Ad hoc 网络,移动代理。刘勇兵 硕士,主要研究方向:网络管理,无线通信。赵林亮 教授,主要研究方向:人工智能,Ad hoc 网络。王光兴 博士生导师,教授,主要研究方向:ATM 网络,卫星网络,电子商务等。

指各类服务怎样组织起来并声明自己以便被客户发现;服务描述则是指对服务的定义和表达,它是客户和服务提供者之间交互的共同语言。

2.1 服务发现模式

目前比较著名的几种服务发现协议是:(1)IETF 开发的服务定位协议 SLP^[3];(2)Sun 开发的基于 Java 的 Jini;(3)Salutation Consortium 的 Salutation^[4];(4)Microsoft 的 Universal Plug and Play(UpnP)用于服务发现的简单服务发现协议 SDP^[7];(5)加州大学 Berkeley 分校的一个研究小组设计的安全服务发现服务 SSDS^[5]。通常,可以将服务发现机制分为三类:

中心注册和查询——服务器将服务注册到中心注册器中,供客户查询;

服务通告——服务器向其它节点声明其服务;

服务查询——客户发送服务请求并从服务器得到应答。

而移动 Ad hoc 网络中的服务发现则出现了新的挑战。首先,移动 Ad hoc 网络没有固定的基础设施,无法事先配置中心注册器,从而必须通过服务通告或者查询方式来获得服务信息;其次,移动 Ad hoc 网络中节点的高度移动性造成网络拓扑和通信路径的动态改变,必须考虑服务器的选择和跳数问题;第三,移动 Ad hoc 网络中能源是一种非常重要的资源,受限的能源成了另一个挑战,因为服务的提供需要消耗节点的能源。

基于以上分析,本文基于分布式服务发现协议 GSD^[6]和分簇结构提出了新的服务发现机制,该机制考虑到各节点处理能力、存储能力以及所提供的服务种类等不可能完全相同的原因,有区别地对待节点,从而改进了 GSD 中同等对待所有节点的缺欠;另外,服务发现模式也考虑到移动 Ad hoc 网络中,分簇具有的普遍性,以及 Ad hoc 网络本质上的广播型特性等特征,进行了有针对性的设计。具体过程如下:

(1)簇内的非簇首节点只向簇首发送服务通告,而不是向其所有相邻节点发通告,当节点脱离某个簇而加入到新簇当中时,原簇首修改该节点所能提供的服务记录,该节点则向其当前所在簇的簇首发送服务通告;

(2)簇首缓存其成员成功使用过的其它簇提供的服务,当其成员提供的服务发生改变(包括取消服务),簇首向其它簇首发送服务更新通告;

(3)当其他簇首接收到更新通告时,如果缓存中没有相关表项,则保存下来,如果以前有相关表项,则更新;

(4)簇首发生变化时,如果是正常离开本簇,则将其服务信息移交给新选簇首;如果是发生意外(如簇首被毁坏),则该簇必须重选簇首并重新向簇首通告服务,当然,这样新簇首内就不具备原簇首中关于簇外所提供服务的信息。

簇内节点向簇首发服务通告时,主要包含如下内容:

PDUType	SourceNode	TimeToLive	SerialNumber	ServiceList
---------	------------	------------	--------------	-------------

其中的 SourceNode 是节点的地址,TimeToLive 表明了该通告有效的时间,每个节点都维护一个 SerialNumber,发送一次 PDU。值就加 1,可以用来排除重复的服务通告,保证通告的唯一性,同时为构造应答提供对应,而 ServiceList 则是具体的服务信息。

簇首存储簇内各节点的服务信息以及所知的簇外的服务信息(接收到其它簇首通告或者簇内节点使用过其它簇服务

后缓存的),缓存中表项的具体内容包括:

ClusterID	SourceNode	ServiceDescription	TimeToLive
-----------	------------	--------------------	------------

其中 ClusterID 是指簇首 ID(或者地址),SourceNode 是服务提供者节点的地址,ServiceDescription 是一个服务或者服务组的描述,而 TimeToLive 则是依据服务通告而设置的该表项可以存在的时间。

簇首之间发送服务更新通告时包含如下内容:

PDUType	ClusterID	SourceNode	Service Description	TimeToLive
---------	-----------	------------	---------------------	------------

收到该更新通告的簇首检查缓存中是否有相应的表项对应该服务,若有,则更新;若没有,则建立一个新的表项。

如果某节点的应用层要使用某项服务,则首先查询自身能否提供该服务,如果不能,则要向簇首发服务请求。簇首查询缓存,如果找到相应表项,则构造应答,若找不到,则向其它簇首转发请求。服务请求的内容如下:

PDUType	SourceNode	SerialNumber	ServiceRequest	LastNode	Hops
---------	------------	--------------	----------------	----------	------

其中 LastNode 表示当前节点是从何处接收到本服务请求的,而 Hops 则表示该服务请求最多可以经过转发的跳数。

找到匹配的服务后,要构造服务应答,应答中包含如下内容:

PDU-Type	Current-Node	Serial-Number	Destination-Node	Last-Node	Hops	Service-Description
----------	--------------	---------------	------------------	-----------	------	---------------------

SerialNumber 和 DestinationNode 表示该应答是对应哪个节点哪个服务请求的,而 Hops 则反应该应答所经过的跳数。

2.2 服务描述

表 1 特性比较

特性	RDF	OIL	DAML
语法	XML	RDF	RDF
形式化语义	不是	是	是
类层次	是	是	是
描述逻辑	否	是	是
谓词逻辑	否	否	否
类等价	否	否	是
谓词等价	否	否	是
实例等价	否	否	是
非集中 Ontology	是	是	是
Ontology 扩展	是	是	是
Ontology 修改	否	否	否

服务描述是服务发现的基础,已形成的不同协议在这方面采取了互不相同的方案。目前,一般是采用可扩展标记语言 XML 接口描述语言 IDL 以及相应的服务模板或接口标准来描述服务。服务描述用于通告服务的能力、接口、行为和服务质量。具体而言,服务接口描述发布了服务签名,而服务能力描述说明了服务的目的和可期望的结果。服务执行时的行为是通过其服务行为描述来描述的(如工作流程)。服务质量描述发布了关于该服务质量的属性,例如响应时间、安全属性、完整性等等。

服务描述是服务发现的基础,是服务管理的一个重要内容。DAML 提供的语义能力有效地描述了服务^[6],服务请求用 DAML 描述并通过服务匹配模块与服务描述进行匹配。这增加了发现服务的灵活性,适合解决 MANET 中服务的异构性。基于 DAML 提供的类-子类层次,节点中提供的服务被分成几个组。表 1 为对几种描述语言的比较^[4],本文选择了 DAML+OIL 作为服务的描述语言。

3 移动 Ad hoc 网络中的服务调度

3.1 服务调度策略的衡量指标

从不同角度衡量调度系统,有不同的衡量指标^[9]。从系统的角度来看,通常有吞吐率(Throughput)和利用率。吞吐率是单位时间内系统所提供的服务数,是服务管理系统性能的主要参数和重要指标;利用率是系统负载与其可承受负载的比率。调度策略在进行节点选择时要充分考虑到节点间的负载平衡,要尽量使它满负荷地运转,不能使节点过载,也不能使节点空闲。从用户角度看,有公平性和等待时间,公平性是指系统必须公平地对待每个用户,主要表现在每个用户对系统资源的平等享用上,即具有同一优先级的用户对于系统资源的占用机会(被调度的机会)是均等的,系统不能有任何偏见,但不同任务,其重要程度不同,因而在分配资源时所具有的优先级也不同;等待时间是指一个任务从提交给系统,到服务开始运行所经历的时间,对每个用户来说,都希望自己服务的等待时间短,而对系统管理员来说,则希望平均等待时间短。

在设计任务调度系统时,应该追求的是高吞吐率,高利用率,总运行时间短,平均等待时间小,具有较好的公平性和优先级策略。但是,同时达到这个目标是不太可能的。对于移动 Ad hoc 网络,本文在设计其服务调度时主要思想是保证关键服务的质量,尽量提高吞吐率。

3.2 移动 Ad hoc 网络服务调度策略

服务管理的目的就是保障关键任务的顺利执行,同时在有限的资源条件和任务执行时间的约束下执行尽可能多的任务,尽量提高资源的利用率。尽管对于服务的调度而言,

吞吐率和利用率是非常重要的两个指标。但实际上,不同的服务和用户拥有不同的重要性,例如一般而言语音业务就比数据业务优先级高一些。为了体现服务和用户在重要程度上的差异,以便在有限资源的前提下,尽量保障优先级高的服务的提供,本文提出了一种基于优先级的可抢占式服务调度策略。任务在重要程度上是有差异的,总体上可以分为三类:关键的(critical),必须保证其最小服务质量要求,不能被抢占;重要的(important),允许先占,优先级高的先执行;一般的(normal),允许被其他两类随意先占。

服务请求时要指定 QoS 参数,如图像解析度,数据采集率等。应用程序可以指定一个 QoS 范围,能保证该范围内的 QoS 都是该应用程序可以接受的。而有的应用程序也可以不指定 QoS 参数,表示对服务质量不做特殊要求。QoS 参数会成为资源评估时的一个重要依据。

从服务管理的角度来分析,分布式服务应用程序具有如下一些特点:

- 应用程序的执行需要经过异构的节点和(或)网络,即是分布的;
- 应用程序定义其所需的 QoS 参数的具体数值,在请求资源时提供这些数值,作为接入和调度的依据之一;
- 应用程序是分级别的。更重要的应用应该优先分到资源,应该根据应用的具体情况对应用分级;
- 资源分配应该能够动态调整。在一个应用程序的执行过程当中,其资源需求可能会发生改变,这时也应触发资源分配与调度。

从资源管理的角度,可将负责每种资源管理功能的实体抽象为资源代理(或者资源对象),为了完成服务调度,需要资源代理提供如下功能:提供评估自身资源的容量与可用性的功能;提供预留一定资源的功能;执行指定的任务;挂起或者终止任务的执行;释放资源。

服务管理需要使用资源协商功能,服务管理与资源代理间进行交互的基本框架如图 1 所示。

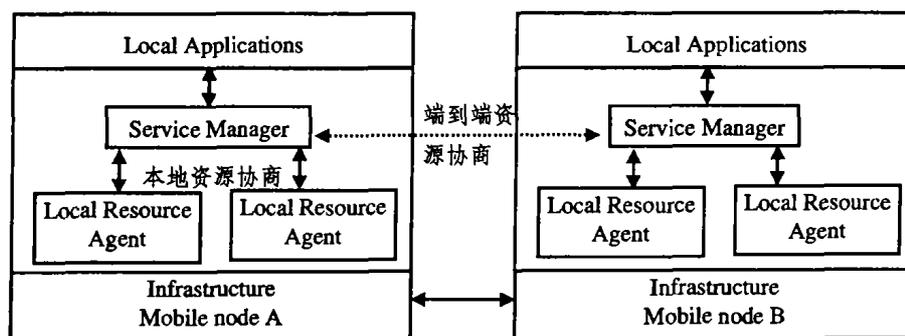


图 1 资源协商框架

为了描述服务管理过程中服务调度算法,先简单介绍伪代码中使用的各函数的主要功能:

- Evaluation: 对服务请求进行分析,分析本地资源使用情况,从而判断该请求在本地执行的可能性;
- LocalLookup: 进行本地查找,判断本地(当前节点)能否提供所请求的服务;
- ForwardRequest: 将收到的服务请求转发给其它节点;
- Evaluation: 评估当前资源能否满足所请求服务的需求;
- DetermineQos: 依据当前资源情况,确定服务执行时

的实际 QoS 参数;

- TestReleasedResource: 测试释放部分低优先级服务占用的资源后能否满足当前服务请求的资源需求;
- ResleasingResource: 实际释放资源;
- CancelPreemption: 取消对低优先级任务的抢占。

基本过程如下:当节点接收到服务请求后,首先判断当前节点能否提供该服务,如果不能,则转发该请求;如本地能提供服务,则从服务队列中选出最高级别的任务,判断当前资源能否满足其要求,若不能提供足够资源,则查看是否有可被占

(下转第 63 页)

程接口 API; (3) 服务可以会聚(到网关或设备); (4) 支持第三方服务开发。

其二: 移动性 为了支持移动性, 用户用相同的服务和环境表述而不考虑他们访问哪里。这一过程是通过把用户标识解析成存储有用户数据的资料库中的地址, 并绑定用户的通信控制配置记录。设备移动性可以建立在 MOBILE-IP^[49] 基础上, 另一种选择是第三代蜂窝网络, 设备可保持临时 IP, 访问类似拨号连接, 设备的移动性支持委托在链路层实现。

SIP 支持设备或用户的移动性。在 OSA 系统中, 设备通过注册被定位, 移动性并不是 OSA 系统主要关注点(在用户域网络网关可以定位 IA 设备)。在用户域中, SIP 被用于 IA 设备间的通信(设备代理或控制器实现注册功能以支持移动性)。

其三: 兼容和扩展 在 OSGI 平台上进行 SIP 通信时, 至少一个 SIP SERVER 必须在平台上提供, 考虑到 OSGI 框架的要求, 可以用 BUNDLE 形式实现 SIP SERVER。平台要安装提供给设备接口的服务并且 SIP SERVER BUNDLE 可以访问它, 如果代表某一设备的服务在框架上不可用, 则可认定网络上的设备不可用。系统的扩展性主要由 SIP 体现: 可以方便地开发新的请求方法和任意的 MIME 类型负载。新的技术会不断出现, SIP 的特性使得系统可以容易支持这些技术。

(上接第 30 页)

用的低优先级服务在执行, 测试释放这些资源后能否满足其需求, 若对各种所需资源的测试都能满足要求, 则实际释放被占用的低优先级服务的资源, 启动该服务, 只要有一种资源不够, 则取消对其它几种资源的占用决定, 不能执行所请求的服务, 而继续运行可被占用的服务。如果一个服务中包含多个子服务, 而且资源得不到满足的子服务可独立于其它子服务, 则将孩子服务作为一个新的独立服务, 为其发送服务请求。具体实现过程如下:

```

Evaluation(Si, QoS)
for task manager on current node k
  result = LocalLookup(Si);
  if(result == NO) {
    ForwardRequest(Si, QoS);
    Exit();}
  Else{
    LocalAdmission = yes;
    For each local resource needed by Si
      Evaluation(Si's subtasks, Qos);
    If(All local resource can support Si)
      DetermineQos(Si, QoS);
    Else LocalAdmission = NO;
    If(LocalAdmission == NO & there are lower-priority executing tasks Si)
      Status = TestRealsedResource(Qos, SI);
    If(Status == ok) LocalAdmission = YES;
    Else {LocalAdmission = NO;
      If(locally unsupported subtasks are asynchronous){
        ForwardRequest (locally unsupported subtasks, QoS);
        Remove forwarded subtasks from Si;
        LocalAdmission = YES; }
    }
    if(LocalAdmission == YES) Execute(Si, QoS);
    else Release(Si);
  }
}
Release(Si)
For current node k
  If(there are lower-priority executing tasks preempted by Si)
    CancelPreemption(Si);
  For all local resources reserved by Si's subtasks
    Release the reserved resources;
  Exit();
Execute(Si, QoS)
For current node k
  Admission = NO;
  If (there are lower-priority executing tasks SI releasing resource for Si)
    ResleasingResource(SI );
  
```

结束语 提出借助 SIP 方法和开发服务思想构筑一个新的 PVC 系统——OSA 普及计算系统, 该系统主要利用 SIP 的寻址机制及其扩展和 OSGI 的服务管理思想实现, 扩充了目前基于 APIS 和服务网关的 PVC 系统的功能和开放性。OSA 系统通过实验说明了在基于 OSGI 的普及计算系统中采用 SIP 方法对设备远程访问和控制的可行性——这和特定的网络类型无关, 而且命令的发送采用一般格式, 可以控制大多数设备, 因而是一种普遍适用的方法。

参考文献

- 1 谢伟凯, 徐光佑, 史元春. 普适计算——下一代计算模式. <http://www.ccw.com.cn/monitor/02/0208/b/0208b02-2.asp>
- 2 刘敏, 过晓冰, 伍卫国, 张然. 普及计算及其应用. 计算机工程, 2002(7): 8~9
- 3 OSGI service gateway specification, release, 1. 0. <http://www.osgi.org>
- 4 Honkanen T. OSGI-open services gateway initiative
- 5 Vicrh R, Malhotra V. JINI and PVC. <http://www.ibm.com/developerworks/cn/java/wi-jini/index-eng.shtml>
- 6 Supakkul S, Chung L. Virtual OSGI framework and telecommunications. <http://citeseer.nj.nec.com/supakkul01virtual.html>
- 7 魏春城. SIP 协议的特点和应用. <http://www.cnc-labs.com/cn/paper/default.asp>
- 8 王红漫, 等. SIP 协议栈的实现与应用. 北京邮电大学学报, 2000, 23(4)

```

For each needed local resource
  Start executing;
Admission = YES;
Exit();
  
```

结论 本文针对移动 Ad hoc 网络的特点对现有服务发现协议进行了改进, 提出了一种新的应用于移动 Ad hoc 网络的服务发现协议, 即一种基于 GSD 和分簇结构的服务发现机制, 该机制考虑到各节点处理能力、存储能力以及所提供的服务种类等不可能完全相同等原因, 不像 GSD 那样同等对待所有节点; 另外, 服务发现模式考虑到移动 Ad hoc 网络中, 分簇具有的普遍性, 以及 Ad hoc 网络本质上的广播型特性, 应该尽力避免广播风暴等特点, 选择 DAML+OIL 作为服务描述方式。考虑到移动 Ad hoc 网络中不同的应用有着不同的重要级别, 本文还提出了基于优先级和两步资源预约的服务调度和 QoS 分配方式, 依据服务所请求的 QoS 级别和资源情况来确定合适的端到端 QoS 级别, 保证资源的充分利用, 并给出了算法的一个实际应用的例子, 由于算法考虑了资源的抢占和两步方式的资源预约, 比普通的确定 QoS 级别的算法拥有更高的预约成功率, 而比仅依据优先级的算法具有更高的资源利用率。

参考文献

- 1 Stephen F B, Victor S F. Network Management of Predictive Mobile Networks [J]. Network and System Management, Feb. 1998
- 2 周晓, 常强林, 蒋序平, 陈鸣. 服务发现机制的研究[J]. 计算机工程, 2002, 28(10): 10~13
- 3 Guttman E. Service Location Protocol: Automatic Discovery of IP Network Service. IEEE Internet Computing, 1999(7): 71~80
- 4 The Salutation Consortium Salutation Architecture Specification (Version 2.0), 1999(6)
- 5 Czerwinski S E, Zhao B Y, Hodes T D, et al. An Architecture for a Secure Service Discovery Service. In: Fifth Annual Intl. Conf. on Mobile Computing and Networks, Seattle, WA, 1999(8)
- 6 Chakraborty D, Joshi A, Yesha Y, Finin T. GSD: A novel group-based service discovery protocol for manets [C]. In: 4th IEEE Conf. on Mobile and Wireless Communications Networks (MWCN2002), Stockholm, Sweden, Sept. 2002
- 7 房胜. 服务发现协议的设计和实现. 计算机工程与应用, 2003, 29: 173~176
- 8 张平, 郭金庚. 语义网描述语言分析. 电脑开发与应用, 2003, 16(4): 31~33
- 9 叶庆华, 梁毅, 孟丹. 一种简化的基于 First-fit 的 Backfilling 调度策略——RB-FIFT [J]. 计算机工程与应用, 2003, 2: 70~74