

MANET 下自适应的服务发现系统研究

翁海斌^{1,2} 温元丰^{1,2} 金蓓弘¹ 张 彪^{1,2}

(中国科学院软件研究所 北京 100190)¹ (中国科学院研究生院 北京 100039)²

摘 要 MANET 环境下的服务发现系统需要面对节点可能频繁移动位置、节点新加入或离开网络,以及服务提供者可能主动撤销服务,因此应在上述情况下以合理的代价发现可用的服务。为了提高系统的自适应性,服务发现系统 SCN4M-DL 采用了无目录服务器结构。与底层地理由协议耦合,设计和实现了基于圆环的服务注册和发现机制。SCN4M-DL 能获得较高的发现请求的成功率,有效控制存储开销和网络负载。实验结果表明,SCN4M-DL 对于 MANET 特性以及服务状态的改变具有良好的自适应性。

关键词 服务发现, MANET, 自适应

Research on an Adaptive Service Discovery System for MANETs

WENG Hai-bin^{1,2} WEN Yuan-feng^{1,2} JIN Bei-hong¹ ZHANG Biao^{1,2}

(Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)¹

(Graduate University of Chinese Academy of Sciences, Beijing 100039, China)²

Abstract Service discovery systems over MANETs have to face node's frequently movement, random joining or leaving, and services revoked by their owners, therefore it is a big challenge to find available services at a reasonable cost. In order to meet above challenges and improve the adaptability of the system, we proposed a system SCN4M-DL, which adopts directory-less architecture. Coupling with underlying geographical routing algorithm, we designed and implemented a ring-based service register and discovery mechanism, which leads to high success rates of discovery request and also can efficiently control memory costs and network traffic. The experimental results show that SCN4M-DL can adapt well to MANETs' characteristics and the changes of service states.

Keywords Service discovery, MANET, Adaptive

1 引言

服务发现系统是 MANET 环境下进行资源共享和服务集成不可或缺的前提。MANET 环境下的现有服务发现系统都要面对以下问题: (1) MANET 节点的资源受限, 例如节点的存储空间有限, 电源能量有限; (2) MANET 节点间的通信其 QoS 往往是非常差而且是不稳定的, 例如通信延迟长以及会频繁掉线; (3) 节点可能频繁移动, 节点可以动态加入和离开。而且, 由于服务提供者可能主动注销服务, MANET 环境下服务的可用性并没有保证。总之, MANET 环境下的服务发现系统需要解决以下问题: (1) 如何在低开销(包括存储开销和网络消息量)条件下发现可用的服务; (2) 如何全面地支持节点移动性和动态性。

迄今为止, 已有的 MANET 下的服务发现系统从软件体系结构角度可以分为如下两类: 第一类采用了基于目录服务器的体系结构, 另一类则采用无目录服务器的体系结构。前者适用于 MANET 环境中存在着若干相对稳定节点的应用场景, 这些节点本身可以用作部署目录服务器, 而后者则更加适用于系统中的节点都是移动的应用场景。针对 MANET

环境下无稳定节点的应用场景, 我们设计并实现了一个基于无目录服务器的服务发现系统 SCN4M-DL。

考虑到大多数 MANET 应用都是在有限的地理范围内, 而且移动节点现在可以比较容易地获得自身的地理位置信息, 所以 SCN4M-DL 采用了基于地理位置的服务发现机制。在 SCN4M-DL 中服务以及对应的服务描述, 首先会被映射到一个具体的地理位置, 之后会把此服务和描述注册到此位置上的节点, 或者附近的节点上。随后, 服务发现请求同样也会映射到一个地理位置, 此服务请求将转发到此地理位置上的节点, 或者附近的节点上。这种方式可以避免基于无目录的服务发现系统通常由于请求的泛洪转发而导致的冗余消息传输问题。另外, SCN4M-DL 也考虑了 MANET 环境下节点的移动、节点的动态加入/离开以及主动服务注册/服务注销情况下的服务的可用性问题。在 SCN4M-DL 中, 服务绑定到一个确定的地理位置, 所以, 保证服务的可用性问题转变为确保在服务所对应的物理位置或者临近位置找到此服务的问题。SCN4M-DL 采用了可配置的服务重新策略来达到及时发现当前可用服务的目的。为了评估 SCN4M-DL 的服务发现机制, 我们还进行了大量的实验。

到稿日期: 2009-07-07 返修日期: 2009-09-23 本文受国家 863 项目(2006AA01Z231), 国家自然科学基金项目(60673123)资助。

翁海斌 男, 硕士生, 主要研究方向为分布式计算、软件工程, E-mail: wenghb07@otcaix.iscas.ac.cn; 温元丰 男, 硕士生, 主要研究方向为分布式计算、软件工程; 金蓓弘 女, 研究员, 博士生导师, 主要研究方向为分布式计算、软件工程; 张 彪 男, 硕士生, 主要研究方向为分布式计算、软件工程。

本文第2节介绍相关工作;第3节描述 SCN4M-DL 的系统模型;第4节详细刻画 SCN4M-DL 的服务发现机制;第5节通过实验来评估 SCN4M-DL 系统;最后是全文总结。

2 相关工作

针对 MANET 的动态拓扑和服务可用性问题,现有的解决方案包括:(1)通过广告策略以及事件驱动来及时更新服务;(2)为目录构建和维持一个覆盖层;(3)与底层路由算法紧耦合。

GSD^[1]是 MANET 环境下的一个早期服务发现系统,它采用了无目录架构;在 GSD 中,节点通过周期性的 n 跳广播来进行服务发布。节点接收到服务发布消息后,会缓存发布消息中附带的服务。GSD 还根据服务的语义类型把服务进行分组。各个节点会收集广告消息沿路所看到的组名,之后通过组名来对服务请求进行有选择的转发。SPIZ^[2]也是一个无目录的服务发现系统。SPIZ 扩展了路由消息包,在其中附带了服务广告和发现相关的信息。我们看到,为了适应 MANET 环境,GSD 采用了方案(1),而 SPIZ 结合了方案(1)和方案(3),不过,它们都忽视了服务的可用性问题。

有些服务发现系统依赖一组目录服务器来注册和定位可用的服务。SCN4M^[3]实现了内容可寻址的 P2P 覆盖层和相应的失效处理机制来处理系统的拓扑变化。所以,它采用了方案(2),而其他一些服务发现系统则采用了混合解决方案的方式。以 Ariadne^[4]为例,它选举一些节点作为目录服务器,如果某个节点不能收到目录服务器周期性的存活消息,此节点就会发起目录服务器的选举过程,这样每个节点都能在确切跳数内到达目录服务器。Ariadne 在目录服务器上以 bloom filter 方式构建服务索引,目录服务器通过两跳广播来发布各自存储的服务索引。Ariadne 还控制广播服务索引的频率来保证目录服务器上信息的即时性,从而缓解环境变化问题。简言之,Ariadne 结合了方案(1)和方案(2)。

而在 DSDM^[5]中,系统根据节点的资源能力来对节点进行评分,从中选择一些节点作为目录服务器,然后节点通过底层的 DSDV 协议来获得最新的路由信息,以及把发现请求转发到最近的目录服务器。DSDM 采用了方案(2)和方案(3)来适应环境变化。在 DSDM 中,目录服务器和服务提供者的离开信息可以通过底层的 DSDV 协议来获得。如果检测到一个目录服务器离开了,则通过评分选择出新的目录服务器来接管离开的目录服务器,同时服务提供者会重新注册他们的服务。如果检测到服务提供者的离开,目录服务器将会删除对应的服务。

另外需要提及的是 GHT 系统^[6],它提供传感器数据的网内存储和检索。GHT 采用 GPSR 作为底层地理路由算法,并且把数据的键值哈希成地理坐标,然后,转发(数据的键值、数据值)到对应的坐标位置上的传感器或者围绕此目的坐标位置的传感器集合上。同时,GHT 通过数据键值来查询获取数据。GHT 设计了基于周界(Perimeter)的刷新机制,以保证传感器数据总是存储在相对应的节点上,即使是在节点拓扑变化的情况下。

SCN4M-DL 采用了与 GHT 系统相似的方法,在 SCN4M-DL 中数据变成了服务,而数据的键值则变成了服务的键值。特别与 GHT 相似的是,SCN4M-DL 系统也结合方案(1)和方案(3)来处理环境的动态性,但 SCN4M-DL 较 GHT 不同的是:(1)SCN4M-DL 提供了更加丰富的表达能

力,(2)若 SCN4M-DL 提供与 GHT 相同的服务请求表达能力,那么它比 GHT 需要更少的存储开销。

3 系统模型

在 SCN4M-DL 中,把 MANET 环境下节点的通信距离表示为 1,那么就可以使用 UDG 图(Unit Disk Graph)来刻画底层网络,所谓 UDG 图是指任意两点之间的距离如果小于 1,则在这两点之间作一条边。为减少 UDG 图中边之间的不必要交叉,进一步将 UDG 图转化为 GG 图(Gabriel Graph),所谓 GG 图是指对于每条边而言,以该边为直径的圆内不存在其他节点。

在 GG 图中,随机选取一个位置作为目的位置,那么一定存在若干节点围绕此目标位置,称由这些节点形成的多边形区域为目标位置所在的面,而这个面的边界称之为周界(Perimeter)。GG 图的一个性质是离目标位置最近的节点一定在围绕此目标位置的周界上。

GG 图上的地理路由算法都是这样一类算法:仅仅借助本地和邻居的地理信息来实现从一个节点到另一个节点的数据包传输。这其中,路由协议用以下方法来获得所有邻居的位置信息:每个节点在一跳范围内周期性地广播自身的位置信息和 ID,这样每个节点都可以周期性地获得所有邻居的 ID 和位置信息。底层路由信息通过数据包本身来获得此数据包对应的目标位置,而且数据包中附带路由所需的信息至多为 $O(1)$ 。通常地理路由算法是通过当前节点和目标位置之间的相对位置关系来决定如何合理地数据包进行转发的。已有的此类路由算法中,有一些相对简单的算法,如贪心路由、面路由。除了以上算法外,还有一些相对复杂的算法,如 GPSR^[7],GOAFR^[8]算法,这些算法都混合使用贪心路由和面路由。

用以下接口来表示 SCN4M-DL 与路由的接口:

Node n GeoRouting (Packet p, Location (x, y))

该接口描述了数据包 p 转发到目的节点 n 的过程,此目的节点 n 的位置为 (x, y) 或者接近 (x, y) 。下面用 GeoRouting 表示该接口。

一个服务发现过程可以用 post-query 模型来进行建模。所谓 post-query 模型是一个时间相关协议,用来建模一个网络中的请求-应答交互,文献[9]首次提出用它描述分布式匹配问题。一个 post-query 模型包含一个发布函数 $P: s \rightarrow P(s)$ 以及一个查询函数 $Q: q \rightarrow Q(q)$,其中函数 P 的功能是把一个服务 s 映射到一组节点集合 $P(s)$,函数 Q 的功能是把一个查询 q 映射到一组节点集合 $Q(q)$ 。但是,post-query 模型并未给出如何进行这些映射的具体方法。文献[10]给出了若干种有用的 post-query 策略,如 greedy 策略、incremental 策略、uniform memory-less 策略和 with memory 策略。本文创建了地理 post-query 策略:将服务发布到一个地理位置附近的节点集合中,此位置通过 HashToPos(服务描述)来获得,同样查询也被发送到相同地理位置附近的节点集合中。在这其中,HashToPos()是一个一致性哈希函数,通过一个特定的服务描述返回一个地理坐标 (x, y) 。地理 post-query 策略和前面的 post-query 模型组成了我们的服务发现过程模型。并且,根据地理 post-query 策略,服务注册或者发现请求会被发送到某一目标位置,此目标位置由 HashToPos()函数计算而得,而距离此目标位置最近的节点一定在围绕此位置的面周界上。

4 服务发现机制

SCN4M-DL 中,服务发现机制是系统模型中地理 post-query 策略的一个实现。本节主要解决如何确定集合 $P(s)$ 和集合 $Q(q)$ 的问题。为方便起见,采用 GOAFR 作为底层的地理路由协议,GOAFR 综合了贪心路由和自适应面路由算法(OAFR)。GOAFR 本身有两个优点:(1)此路由的开销能达到在最差情况下最优和平均情况下最有效;(2)当路由结束时,最后终止的节点是所有节点中距离目标位置最近的节点。为简便起见,SCN4M-DL 中的节点用 SDN 表示。除了服务注册和服务发现两个基本操作外,SCN4M-DL 还提供两种额外操作,即服务注销和服务刷新。

4.1 服务注册

SCN4M-DL 中,服务用 XML 文件来进行描述。采用分布式后缀树来存储服务描述,大致有如下 3 个步骤:(1)分解服务描述为后缀集合;(2)计算 $P(s)$,把服务和后缀路由到 $P(s)$ 的节点中;(3)在这些节点上构造本地的后缀树。下面对这些步骤进行更为详细的说明。

当用户注册服务 s 时,描述 s 的 XML 文档会被分解为后缀集合,其中的每一个后缀都是该 XML 文档中从内节点到叶子节点的一条路径。随后,每一个后缀和对应的服务本身会被打包成数据包 p ,此数据包将被发送到地理位置 (x, y) ,此位置由 HashToPos (后缀的第一个标记)计算得到。

GeoRouting 协助系统把数据包 p 路由到节点 SDN_α ,此节点是距离位置 (x, y) 最近的节点。假设 SDN_α 到 (x, y) 的距离为 L ,以 (x, y) 为圆心, L 为内半径, $L+tw$ 为外半径构建一个圆环,其中, tw 是预定义的环境宽度,那么,把在 (x, y) 所在面周界上、同时在环内的节点确定为 $P(s)$ 集合。在这里,节点 SDN_α 称为主(Master)节点,其他节点称为镜像(Mirror)节点,服务描述和服务本身 s 均被存储在主节点和镜像节点中。通过此种方式确定的 $P(s)$ 集合保证了相比于 GHT 系统,SCN4M-DL 仅需更少的节点来存储服务,而且也能保证以较高的概率发现所需的服务。而在 GHT 系统中,服务需要在围绕目标位置的周界上的每一个节点进行存储。

图 1 描绘了一个典型的服务注册情形,星形表示的是目标位置, $\text{SDN}_1, \text{SDN}_2, \text{SDN}_3, \text{SDN}_6, \text{SDN}_9, \text{SDN}_8$ 组成围绕目标位置的周界。而根据以上定义的服务注册方式, SDN_1 为主注册节点, SDN_3 和 SDN_8 为镜像注册节点。

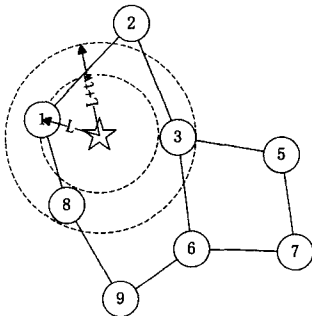


图 1 服务注册过程

在主节点和镜像节点上,服务描述后缀和服务本身均在本地进行注册。首先,给每一个服务 s 分配一个唯一的 ID 号,记为 SID 。其次,把描述后缀组成一棵后缀树,也就是说,对于 SDN 接收到的每一个后缀,SDN 从根节点开始,在树结构中增加此后缀 t 。为了记录属性和标记值,而且为了保存树中的记录在原有服务中的位置,每个记录都用如下三元组

$\langle \text{Attribute}, \text{SID}, \text{Position} \rangle$ 表示。其中 Position 用来保存此记录在原有 XML 文档中的分层位置信息,具体而言,它表示的是标记在服务描述中所处的位置。 Position 也有 3 个字段,为 $\langle \text{Level}, \text{Start}, \text{End} \rangle$,其中 Level 表示当前标记相对于根标记的层次信息,根标记对应的 Level 值为 1; Start 表示当前标记在 XML 文档中的起始偏移值; End 表示当前标记在 XML 文档中的结尾偏移值。因此,若两个节点为父子关系,父节点的 Level 值要比孩子结点的 Level 值少 1,而且,父节点的 $[\text{Start}, \text{End}]$ 范围覆盖孩子结点的 $[\text{Start}, \text{End}]$ 范围。我们称这种父子关系为 $p-c$ 关系。

由这些三元组形成的扩充后缀树大大加快了本地服务查询,图 2 给出了一个 SDN 的扩充后缀树的示例。

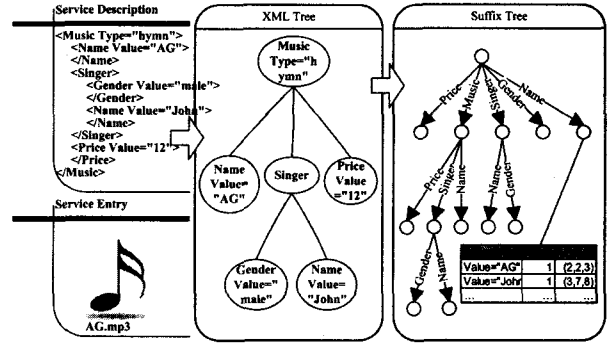


图 2 扩充后缀树

我们注意到,对于主节点而言,至少需要 tw/v (v 为节点的最大移动速率)时间离开环形区域,故在 tw/v 时间内,可以在环形区域内找到此服务。这条性质对 SCN4M-DL 支持 MANET 移动性方面起到了至关重要的作用。

图 3 用伪代码描述了服务注册的整个过程。

Procedure ServiceRegister (Service S)

1. Break Service's Description into a bunch of suffixes
 2. FOR each suffix, IN Service Description
 3. SET $(x, y) = \text{HashToPos}(\text{suffix's first TAG})$
 4. SET Packet p with SDS and S
 5. SET Node $n = \text{GeoRouting}(p, (x, y))$
 6. Mark n as MASTERNODE of S
 7. ServiceRegisterLocally(S, M)
 8. SET $P(S) = \text{Nodes on the ring}$
 9. FOR each node, M IN $P(S)$
 10. ServiceRegisterLocally(S, M)
 11. Mark M as MIRRORNODE of S
- End of procedure

图 3 服务注册过程

4.2 服务发现

SCN4M-DL 采用类 XQuery 语言描述服务发现请求。发现请求由路径约束和属性值约束组成,同时遵从 XQuery 语法规则。下面是一个发现请求的例子:

$\text{Music/Singer/Name}[\text{Value} = \text{"John"}] \text{ AND } \text{Music/Price}[\text{Value} = \text{"12"}]$

服务发现请求首先被分解成若干个原子请求,原子请求中不包含逻辑操作符“AND”和“OR”。服务请求者会在发现数据包中加入自身 ID ID_{req} 和地理信息 loc_m ,而这些信息在发现结果返回过程中非常有用。随后,发现请求通过底层的 GeoRouting 进行路由,而最终到达离目标位置最近的 SDN_α 节点。这里定义 SCN_α 与目标位置 (x, y) 的距离为 L' ,以 $(x,$

$y)$ 为圆心, L' 为内半径, $L'+tw$ 为外半径作一个圆环,所有在此圆环内的节点组成了集合 $Q(q)$ 。由于 MANET 环境中节点的移动,此时的 SCN_q 可能与在服务注册阶段给定的 SCN_s 不同。在这里,服务发现阶段会以一定概率成功匹配存在的服务,称这个概率为成功率,它是服务发现系统一个非常重要的指标。

由于采用了扩充后缀树,本地查询速率是非常快的,服务发现请求沿着后缀树的根节点往下进行匹配。在匹配的开始阶段,将匹配结果集合 $RetSet$ 置为空。随后,取出发现请求的第一个标记约束,与本地扩充后缀树中的边匹配,此边连接根节点及其孩子结点。如果此边的边名与标记名相同,那么继续检查此边所连的孩子结点的属性集合中是否有与第一个属性值约束匹配的元素,如匹配,则把该元素对应的三元组加入 $RetSet$ 集合。在后续的匹配过程中, $RetSet$ 中元素的三元组中的 SID 必须与后缀树节点的三元组中的 SID 相同, $Position$ 必须与后缀树节点的三元组中的 $Position$ 满足 $p-c$ 关系,这些后缀树节点本身也应该匹配发现请求中的标记名和属性值约束。通过逐步的匹配, $RetSet$ 集合会缩减到一个更小的集合。如果发现请求的所有标记都被匹配过,那么发现请求处理完毕,返回最终 $RetSet$ 集合中的 SID 对应的服务,这样就查找到了与发现请求匹配的服务。

当完成匹配过程后, SCN_q 需要返回最终匹配成功的服务给服务请求者。由于移动性,请求者可能从原先位置 loc_{res} 移动了一段距离,但是并不会离 loc_{res} 太远。因此,采用如下的两个阶段将结果返回给请求者。

第一步 *GeoRouting* 以 loc_{res} 作为目标位置,把结果数据包路由到离 loc_{res} 最近的节点 SCN_c 。 SCN_c 接收到结果数据包后,在其一跳传播范围内广播此数据包。

第二步 在上述一跳范围内的节点,如自身 ID 与请求者 ID ID_{req} 匹配,那么此节点就是服务请求者,从而请求结果正确返回给请求者,若不存在这样的节点,则发现请求阶段也同样结束。

4.3 服务注销

MANET 下的系统一般不会对长久地提供服务,这不仅仅由于节点的离开或失效,而且有可能服务提供者本身主动注销服务。因此,SCN4M-DL 提供了两种方式来注销服务。

第一种方式是使用软状态。当服务提供者注册服务时,对于每个服务本身会设定一个消亡时间。各个 SDN 会周期性地检查其上存储的服务,如有服务过期,则在本地把此服务进行删除,同时删除后缀树中的内容。

第二种方式是服务提供者主动注销服务。此过程与服务注册过程相似,服务提供者首先分解服务描述,得到后缀集合。通过后缀的第一个标记,来决定此后缀的目标位置,用 *GeoRouting* 路由到上述提到的环内节点。最后,在这些节点的本地后缀树中删除相应内容,同时删除对应的服务。

4.4 服务刷新

在 MANET 环境下,由于节点的移动性,会导致节点存储服务的不一致性,这分以下两种情形:(1)服务 s 注册的主节点的移动,使得在相应的目标位置 (x, y) 附近不能找到此节点;(2)主节点并未发生移动,但是由于其他节点的移动,使得这些节点更加接近于目标位置 (x, y) 。上述两种情形都有可能导致在 4.2 节提及的环区域内查找不到所需的服务。

为了解决上述问题,SCN4M-DL 采用了可配置的服务更新策略来保证服务的一致性。这里有两个配置参数,包括环宽度 tw ,以及周期刷新间隔 ri 。对于每个主节点而言,每隔一个刷新周期 ri ,便会对其上的服务进行重新注册,而 tw 的使用在 4.1 节已经提及。

5 实验与分析

在 Linux 操作系统(Ubuntu v7.10)下实现了 SCN4M-DL 系统,并成功部署在便携式移动 PC(Samsung Q1 和 Fujitsu U1010)上。

我们在 J-SIM^[11] 框架下评估 SCN4M-DL 在移动环境下的行为。为此,首先在 J-SIM 的应用层增加了 SCN4M-DL 的服务发现机制。其次,在 J-SIM 增加了地理路由算法 GOAFR。最后,调整 J-SIM 自有的节点移动模型的参数来模拟节点的移动性,并运行模拟实验。

在模拟实验中,部署了 100 个移动节点,这些节点均匀部署在 $1000 \times 1000m^2$ 区域内。每个节点的传输距离设定为 250m,节点的移动遵从 random waypoint 模型。

在实验的开始阶段,J-SIM 上层的服务发现应用获取 MANET 场景的地理信息,如长度、宽度等。在系统运行 30 秒后,每个节点会在后续的一分钟内随机选择一个时间点发起一个服务注册过程。这些服务的后缀均会被哈希成 MANET 中的一个地理位置,然后根据注册机制把这些服务注册到相应节点上。因此,模拟程序会一共生成 100 个服务。实验在运行 125s 后,每隔 1s 会随机选择 4 个移动节点,它们都会发起 1 个服务发现请求(请求生成规则保证一定存在与之匹配的服务),这样的请求生成过程会持续 125s。最终,系统将产生 500 个发现请求。我们在实验中设置不同的节点移动速率和系统参数(刷新间隔,环宽度),记录服务发现的成功率,以及存储开销和网络总负载。

图 4 给出了 tw 固定为 50m,刷新间隔和移动速率变化情况下系统的发现请求成功率。从图中可以看出,在刷新间隔从 60s 下降到 5s 的过程中,相应的成功率逐渐增大。同样,当刷新间隔小于 15s 时,相应的成功率大于 96%。

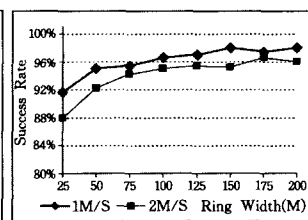
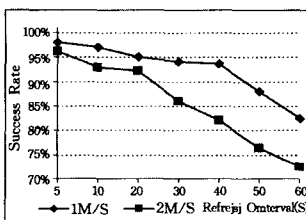


图 4 不同刷新间隔下的成功率 图 5 不同的环宽度下的成功率

我们也研究了不同的环宽度对于成功率的影响,图 5 显示了在两种移动速率下随着环宽度的增加,相应的成功率随之增大。当环宽度为 100m 时,对应的成功率均大于 95%。

在后续实验中,设定节点最大的移动速率为每秒 1m。图 6 给出了在不同数量节点离开/加入情形下发现请求成功率的变化,由此来查看系统对动态性的适应能力。这里 -10% 表示系统中有 10% 的节点离开系统,而 $+10\%$ 表示的是有 10% 数量的节点新进入系统中,其他与之相似。如图 6 所示,节点的离开相比于节点的加入对于系统的影响要小。这其中主要是因为主节点的离开,镜像节点很好地起到备份节点的作用。

现在,我们来度量基于环的注册过程对于存储开销的影响。图 7、图 8 中 Ring 线给出了 20s 内基于环的注册过程所需的存储开销(单位为 MB),而 Perimeter 线给出了 20s 内基于周界的注册过程所需的存储开销(单位为 MB),后者也是 GHT 的工作方式。如图 8 所示,随着环宽度的增大,Ring 线与 Perimeter 线之间的差距逐渐减少。平均而言,两者的比例为 48.67%,它反映了采用基于环的注册方式在存储上的减少程度,比值越小,意味着节省了更多的存储。

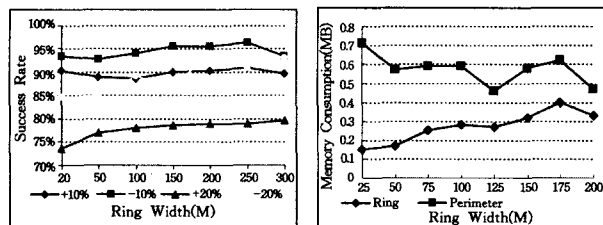


图 6 不同的动态情形下的成功率 图 7 不同的环宽度下的存储开销

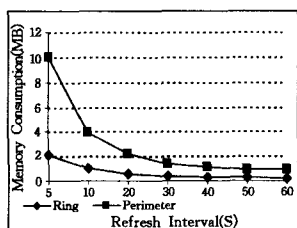


图 8 不同刷新间隔下的存储开销

图 8 表示在不同的刷新闻隔下,基于环以及基于周界的注册和发现机制下的系统存储开销。显而易见,不管哪种方式下,随着刷新闻隔的增大,对应的存储开销均随之减少。同时我们也注意到,相比于基于周界的注册机制,基于环的注册机制所需的存储开销更少。

最后查看在不同的刷新闻隔以及不同的环宽度下,系统的网络负载。如图 9 所示,随着刷新闻隔的增大,网络负载快速下降。另一方面,从图 10 中可看到,在不同的环宽度下,相应的网络开销虽有波动,但基本变化不大。

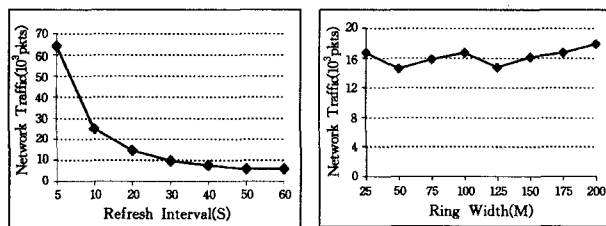


图 9 不同刷新闻隔下的网络负载 图 10 不同环宽度下的网络负载

结束语 本文设计并实现了无目录服务器 SCN4M-DL,

它能很好地适用于 MANET 环境。本文基于地理路由算法 GOAFR,给出了基于地理圆环的服务注册机制,以及相应的服务发现算法。而且,还给出了服务注销方式以及刷新策略,以更好地适应 MANET 环境的动态性和移动性。

从分布式系统的设计角度看,与底层路由协议的耦合,以及服务刷新策略,使得 SCN4M-DL 相对于运行在一个相对稳定的覆盖层上,而且不需要维护覆盖层所需要的那么多的开销。实验数据也表明,不管是何种运行环境,SCN4M-DL 能在可接受的开销内,获得满意的发现请求成功率。

参考文献

- [1] Chakraborty D, Joshi A, Yesha Y, et al. Toward Distributed Service Discovery in Pervasive Computing Environments[J]. IEEE Transactions on Mobile Computing, 2006, 5(2)
- [2] Noh D, Shin H. SPIZ: An Effective Service Discovery Protocol for Mobile Ad Hoc Networks[J]. EURASIP Journal on Wireless Communications and Networking, 2007
- [3] Wen Yuanfeng, Zhang Faen, Jin Beihong. A Novel Distributed Index Approach for Service Discovery in MANET[C]// IC-PADS'08, The 14th IEEE International Conference on Parallel and Distributed Systems. Melbourne, Victoria, Australia, December 2008
- [4] Sailhan F, Issarny V. Scalable Service Discovery for MANET [C]// PerCom'05, the 3rd IEEE International Conference on Pervasive Computing and Communications, 2005
- [5] Artail H, Merhad K W, Hamze H. DSDM: A Distributed Service Discovery Model for Manets[J]. IEEE Transactions on Parallel and Distributed Systems, 2008, 19(9)
- [6] Ratnasamy S, Karp B, Shenker S, et al. Data-Centric Storage in Sensor networks with GHT, a Geographic Hash Table[J]. Mobile Networks and Applications, 2003, 8(4)
- [7] Karp B, Kung H. GPSR: greedy perimeter stateless routing for wireless networks [Z]. Mobile Computing and Networking, 2000: 243-254
- [8] Kuhn F, Wattenhofer R, Zollinger A. Worst-case optimal and average-case efficient geometric ad hoc routing[C]// MobiHoc'03. June 2003
- [9] Mullender S J, Vitanyi P M B. Distributed Match-Making[J]. Algorithmic, 1988, 3: 367-391
- [10] Luo Honghui, Barbeau M. Performance Evaluation of Service Discovery Strategies in Ad Hoc Networks[C]// Second Annual Conference on Communication Networks and Services Research (CNSR). 2004
- [11] J-Sim. <http://www.j-sim.org>, 2009

(上接第 59 页)

- [3] Blake S, Black D, Carlson M, et al. An architecture for differentiated services [OL]. <http://www.ietf.org/rfc/rfc2475.txt>, Dec. 1998, IETF RFC2475
- [4] 冯杰, 夏尊铨. 基于多目标规划和业务区分的多 QoS 约束路由算法[J]. 大连理工大学学报, 2006, 46(4): 605-610
- [5] 王海涛, 郑少仁, 宋丽华. 移动 ad hoc 网络的体系结构和分簇算法[C]// 第八届全国青年通信学术会议. 绵阳, 2002
- [6] 徐泽水. 不确定多属性决策方法及应用[M]. 北京: 清华大学出版社, 2004

- [7] Karp B, Kung H. Greedy Perimeter Stateless Routing for Wireless Networks[C]// Proc. 6th Annual ACM/IEEE Int'l. Conf. Mobile Comp. Net. Boston, MA, Aug. 2000: 243-254
- [8] Lee S B, Ahn G S, Zhang X, et al. INSIGNIA: an Ip-based quality of service framework for mobile ad hoc networks[J]. J. Parallel and Distributed Comp, 2000, 60: 374-406
- [9] Chakareski J, Girod B. Rate-distortion optimized packet scheduling and routing for media streaming with path diversity[C]// Proc. Data Compression Conference (DCC). Snowbird, Utah, Mar. 2003: 203-212