

移动 Agent 系统通信失效问题研究^{*}

周竞扬 陈韬略 许 平 陆桑璐 陈道蓄

(南京大学软件新技术国家重点实验室 南京大学计算机科学与技术系 南京210093)

Research on Communication Failure in Mobile Agent Systems

ZHOU Jing-Yang CHEN Tao-lue XU Ping LU Sang-Lu CHEN Dao-Xu

(National Laboratory of Novel Software Technology, Computer Science and Technology Department, Nanjing University, 210093)

Email: Jingyang@nju.edu.cn

Abstract The Mobile Agent is thought to be the popular computing model for the next generation's distributed system. The mobile agent has the following key features: autonomy, collaboration, security and especially mobility. A good communication model for mobile agents cooperation is also necessary. In this article, after describing the main communication mechanisms, we analyze the methods to solve Agent Tracking and Communication Failure and make classification. We also propose a reliable and efficient mechanism named MEFS which uses Detecting methods and then analyze its performance by giving the experimental data at the end.

Keywords Mobile Agent, Agent communication, Communication failure, Message chasing

1. 引言

随着计算机与网络技术的不断进步,整个 Internet 发展成为一个巨大的分布式系统。上世纪末提出的移动 Agent (Mobile Agent) 技术作为运行于动态、开放网络环境中一种封装良好的运算对象,克服了以往分布式计算模式(如 Client/Server 模型)不能适应网络结构与功能需求变化的弱点,在分布式平台中自主地移动功能而不是移动数据,具有自主性、移动性、协作性和安全性等特征,能够充分发挥 Internet 潜力,因而被广泛认为是未来基于 Internet 的一种主流分布式计算模式。

移动 Agent 需要在运算平台上通过与其他 Agent 交互从而最终完成用户所指派的任务,协作性是移动 Agent 的必然要求。为了保证 Agent 间的协作,除了功能互通、协作联盟/模式以外,我们必须为移动 Agent 系统提供一种可靠并且有效的通信机制,满足移动 Agent 对周围合作伙伴的感知以及基本的通信需求。但是由于 Agent 的自主和移动特性,这种通信机制有别于一般分布式系统中的通信模式,需要进一步的讨论与研究。

Agent 之间的信息交流有多个层次:最底层的“数据传输”主要完成数据的转移,强调通信效率,却对数据的处理逻辑并不关心;高层的“知识交换”从语义层上支持用户进行知识交换和自动处理,便于进行复杂的协作,但以 KQML^[1]为代表的知识交换方案仅从较高的抽象层次上进行设计,不能解决 Agent 移动所带来的特殊通信问题,并且其实现仍然依赖于中间层次的消息传递。消息传递机制在传输数据的同时又能进行处理,具有较好的灵活性和较强的处理能力,我们将主要针对此领域展开研究(下文所说的“移动 Agent 通信”均指消息传递机制)。

目前针对移动 Agent 通信机制的研究大多以不考虑网络传输的故障为前提,这里我们也假设在物理网络上传输的数据包总是能够在有限时间内从信道的源端到达目的端,并

且这个信道是先进先出的通道(FIFO Channel)。另外,Agent 被认为总是联接到网络上的,不存在由于暂时与网络隔离而产生间歇性通信^[2]的情况。除此以外,我们不对网络条件和 Agent 的行为有任何预知假设或做任何限制。

2. 通信失效问题

2.1 通信失效

如果 Agent 总是静止不动的(没有网络位置的变化),那么根据前面的无故障假设,我们可以肯定 Agent a 向 Agent b 发出的消息总能在有限时间内被 b 收到;即使没有这个假设,只要使用 TCP/IP 等有连接的网络通信协议就可以保证 Agent 之间的通信可靠。但是在移动 Agent 的通信环境中,由于 Agent 的自主性和移动性,通信中的主体在任意时刻都可能发生网络位置的变迁,导致消息被发送到某一网络节点但接收者已经离开而无法收到消息;如图1,发送者 a 获知目标 Agent b 的地址 P 后在发送消息 M 的过程中 b 离开了 P 到达节点 P',但是 M 仍然被送到 P 使得 b 无法接收到 M。这样,基本的网络通信协议只能保证 Agent 接收到的消息是正确无误的,却无法确保 Agent 移动了位置后仍然能够收到消息。这种移动 Agent 系统中因通信主体物理位置变化而造成的通信不正常现象称为移动通信失效^[3]。通信失效是移动 A-

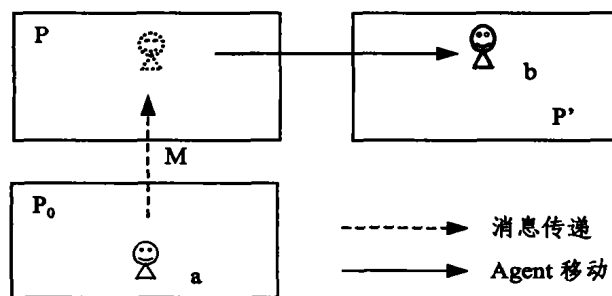


图1 通信失效示意

^{*} 课题得到国家863基金资助(项目编号 2001AA113050)。周竞扬 硕士研究生,主要研究方向为分布式并行计算与移动 Agent 技术。陈韬略 硕士研究生,主要研究方向为移动 Agent 技术与形式化方法。许 平 硕士研究生,主要研究方向为分布式并行计算。陆桑璐 博士,副教授,主要研究领域为分布式并行系统。陈道蓄 教授,博士生导师,主要研究领域为分布式并行系统。

gent 协作的致命缺陷。它使得协作中的 Agent 不能及时得到协同信息,从而导致协作的失败甚至造成整个系统的崩溃。

值得注意的是,解决通信失效的过程中可能出现消息追逐的问题,即在 Agent 移动非常快的情况下,消息追随目标 Agent 在各网络节点间游荡却永远无法送达的现象。这个问题虽然出现的概率较小,但造成的后果不但使得 Agent 间协作失败,而且会造成系统资源的浪费。

2.2 相关研究

虽然集中针对移动 Agent 通信的研究并不是很多,但一些移动 Agent 整体框架的设计方案或多或少涉及到了这方面的内容。

文[2]提出的 ICM (Inter-Agent Communication Model) 采取了“存储-发送”结构,并在每个节点上建立一个 CS (Communication Server) 来负责通信细节。消息发送者 Agent a 把消息发给 CS_a, 根据接收者 b 在 CS_a 的注册位置, CS_a 把消息发给 b 所在节点的 CS_b, CS_a 存储消息并与 b 建立连接等待 b 收取。所发送的消息中含有一个表,记录所有目标 Agent 有可能存在的节点地址,假如 b 已经移动, CS_a 查表并依次联系表中各节点,将消息发送给第一个可连接节点的 CS_b (b 又不在本节点且没有注册的话, CS_a 再重复此过程)。如果消息中的表为空, CS 将保存此消息一定时间然后丢弃或者立即丢弃此消息,并返回失败信息。

文[3]中介绍了一种称之为 Mogent 的系统,引入了 Agent “状态”的概念。Mogent 的每个节点除了有类似上文 CS 的 Communicator 以外,还设计了一个 Home 模块,用于记录在本节点上“出生”的 Agent 当前所在节点的地址和当前的状态。Communicator 则为本节点的所有 Agent 记录对应的“在途信件数”,用于表示已发送给某个 Agent 但仍在途中的消息个数。a 如发消息给 b, 首先把消息交给 C_a, C_a 向 b 的出生地 H_a 查询 b 的当前地址,如果 b 是“静止态”,直接返回 b 的地址,并向 C_a 发送控制信息通知其将 b 对应的“在途信件数”加1;如果 b 的状态为“移动态”则将该请求放入“寻址信件等待队列”,等到 b 的状态恢复为“静止态”再返回 b 的当前地址。C_a 得到 b 的地址后向 C_b 发送消息, C_b 收到后将“在途信件数”减1, 然后最终把消息交给 b。如果 b 需要移动,必须先向 H_a 发出迁移申请,待 H_b 将 b 的状态由“静止态”改为“移动态”并发出确认消息后, b 检查 C_a 上其对应的“在途信件数”,如果为零则可以移动否则等待。b 到达新的节点后立刻通知 H_b 修改其当前地址和状态。

文[4]提出了“Session-Oriented Communication”的想法,采取“请求-应答”方式,在一对互相知道对方地址的 Agent 之间通信。如果其中一方要求移动,必须等到应答收到会话结束以后。Mole 系统^[5]采纳了这种面向会话通信的思想,并对通信失效问题提出了几种可能的处理方法:将失效消息丢弃、丢弃的同时回送错误信息以及将失效消息就地保存,待目标 Agent 返回时再交付。

文[6]根据“网络快照”的思想,为每个节点的入度通道设计了多种状态。这样,虽然对 Agent 的移动不加任何限制,只要通过在整个系统内广播要发送的消息并配合以通道的状态控制,就可以确保目的 Agent 收到消息。然而此算法代价太高,不易实现。

文[7]为每一个 Agent 分配一个“邮箱”,所有消息都被送到对应的邮箱存储,等待 Agent 来主动收取。邮箱类通信方法关键在于设定收信频率:过低则消息不能及时到达目标 Agent, 过高就会引起过多“空收信”的情况。

DEMS^[8]在 RMI 的基础上使用了缓存机制。只有当前节

点和目标节点的 caching 都被确认开始工作后, Agent 才被允许移动。而一旦到达目的地, Agent 便使用同步的方法分别向两边的 caching 获取在移动期间发送给它而被缓存的消息。

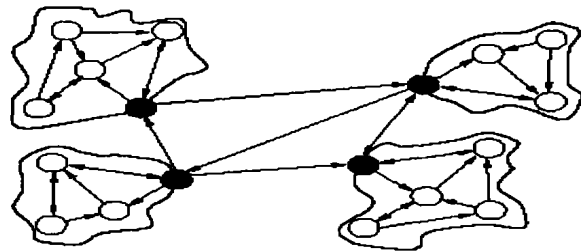


图2 有关键节点的 Agent 系统结构

还有研究提出对于预先知道网络结构的系统,可以把消息发送到目标 Agent 移动必经的所有“关键”节点(图2中的黑色实心点)并存储,直到 Agent 经过某一个关键节点后接收到此消息;另外,“发现”通信失败后建立良好的消息重发机制也被认为是一种可行的选择,这里不再一一赘述。

3. 解决方案分析

通信失效产生的直接原因是因为消息在传送过程中目标 Agent 发生了位置移动。由于消息不可能瞬时送达,因此此类问题普遍存在。但通信失效的根本原因在于目标 Agent 的当前地址是一个共享变量:发送消息过程中对 Agent 的寻址可以被认为是一种“读”操作, Agent 移动后修改当前地址则是一个“写”操作,由于读写冲突而导致了类似数据库系统的“脏读”现象的发生,使得消息被送到了 Agent 已经不在的节点。

通信目标地址的获知是通信过程的一个重要组成部分。一般来说, Agent 的寻址都要求实现透明寻址,即系统能够自动完成从 Agent 名到物理地址的映射。在此基础上,按照寻址服务的组织和部署的不同,移动 Agent 的寻址机制一般可以分为集中式寻址、指针寻址、广播和层次式寻址等四种^[7]。集中式寻址^[3]每次都向固定的地址服务器查询,方便但易形成系统瓶颈;指针寻址^[9]则从目标 Agent 移动后留下的地址指针获得其地址信息,如何提高指针的准确性还有待研究;广播方式^[6]通过消息发送者或者接收者向整个网络广播以获取或者通知当前地址,实现简单但容易造成网络拥塞;而树状的地址目录结构被用在层次型寻址机制中,效率较高但根节点负载很大,且在 Internet 环境下难于构造层次结构。

针对通信失效的处理方法,我们认为主要可以分为三类:鸵鸟法、预防法和检测法。鸵鸟法忽略问题的存在,不采取任何措施, Mole 系统提出的丢弃失效消息便属于此类。鸵鸟算法回避通信失效产生的后果,不能为移动 Agent 系统提供一种可靠的通信机制,故而很少被采用。

预防法采用同步通信的方法从结构上避免通信失效的发生。此时移动 Agent 的当前地址被视为一个信号量进行加锁,以控制读写冲突的发生;如果有发送给 Agent 消息在途中, Agent 就不能移动;如果 Agent 正在移动,其他 Agent 则不能向其发送消息。预防法使得通信失效问题不再存在,因此被广泛采纳, Mogent 便是预防法的一个典型。邮箱法的信息在邮箱移动时不可以随意发送, Agent 收信时需要同步也不能移动;文[6]的方案虽然不限制 Agent 的移动和消息的发送,但其要求对消息进行广播,并且从同一个节点(不是同一 Agent)发出的消息只能等待被系统依次处理,本质上也是一

种同步,故此这两种也都属于预防法。预防法虽然能彻底解决通信失效问题,但是代价十分高:如果不像“Session-Oriented Communication”那样提出额外的限制条件,就必须进行复杂的消息控制。以 Mogent 为例,不在同一节点的 Agent 之间每发送一条消息,节点之间的控制消息就有三条,而 Agent 移动也要发送三条消息。预防法往往需要优良的网络条件以减少通信延时,但即便如此,因为频繁同步而限制 Agent 移动或消息发送极大地损害了 Agent 的自主性。

检测法目的是通信失效发生后采取其他方法将消息最终正确地传递给目标 Agent。“异步是消息传递的天性”,检测法采用异步通信的方法,对 Agent 的移动几乎不加任何限制,通信开销也少,效率比较高,ICM 就属于此类。但是检测法设计难度比较大,而且会带来副作用一引起消息追逐问题。虽然没有得到具体的证明,但是我们相信单纯的异步方式(包括广播)不可能解决消息追逐问题。由于对消息追逐至今尚无彻底的解决方案,现今的移动 Agent 系统若非加强条件限制(比如假设网络的拓扑结构已知或预先知道 Agent 的活动范围),一般也较少采用检测法。

综合以上分析,我们发现采用预防法的移动 Agent 系统不会发生通信失效现象,但是控制复杂,效率低下;检测法尽管实现较困难,但它很好地符合了移动 Agent 的特性,具有相当的研究前景。结合前文所述的四种 Agent 寻址方法,下文提出的 MEFS 通信机制在 Agent 寻址上采取指针寻址和集中式寻址相结合的方式,对于通信失效的解决则采用了检测法作为移动 Agent 通信的主体,但在处理消息追逐现象时又采取了同步的方法:通常情况下使用地址指针获得目标 Agent 的地址并通过网络结点间转发以实现消息送达,而发生消息追逐时进行 Agent“限速同步”。这样既发挥了指针寻址和检测法的特长,又融入集中式寻址和预防法的优点以克服其不足,使得 MEFS 能够有效地应用于移动 Agent 系统。

4. MEFS:一种有效的通信模式

4.1 MEFS 结构与思想

消息的异步转发是 MEFS 机制的核心思想,而运用多种解决方法和参数动态可调是其主要特点。这里首先介绍 MEFS 的主要通信流程,然后着重对消息追逐现象的解决进行讨论。

4.1.1 主要通信流程 为了在 MEFS 系统中实现消息在节点之间自由转发且不限 Agent 的移动,任意 Agent 必须在移动前后向出生节点(称为 Home)、当前节点和移动目的节点注册。这样 Agent 既在移动前的节点留下了迁移目的地的信息,又能够地在出生地维护其当前所在节点的地址信息。值得注意的是,注册的过程是个异步的过程,即发送的登记消息并不需要等待应答,移动可以与发消息同时进行。另外,登记旅行目的地是在 Agent 自主做出移动决定之后,所以注册登记的动作既没有预知更没有限制 Agent 的移动。

如果 Agent a 和 b 处于同一个网络结点,消息可以在此节点内直接发送;如果不在同一节点,a 可以根据当前节点的指针信息或者向 b 的出生地查询得到 b 的地址后再发送。即使 Agent b 在消息传递的过程中发生了移动,由于 b 从一个节点移出的同时留下了其迁移目的地址指针,我们可以很方便地得知其去向,从而迅速地转发消息,并不需要每次都向出生地查询其地址而造成潜在的系统性能瓶颈。

然而,保留已离去 Agent 的指针过长时间会影响消息传递的效率;因为时间太长或 Agent 移动频繁,在一段时间内 Agent 已经移动了多个节点,但是消息仍追随 Agent 移动的

轨迹依次在各个节点间转发,所以,MEFS 定义了 Timeout (状态超时)参数 T_o 。如果 Agent a 留下的指针在某节点保留的时间达到了 T_o ,则将该指针删除。当本节点再需要发送目标为 a 的消息时,则需要采用类似集中式寻址的方法向 a 的 Home 查询,获得地址后再转发。对于如何设定状态超时参数,4.2 节将做讨论。

4.1.2 消息追逐的解决 虽然我们设置了 Timeout 参数,但是当 Agent 移动过于频繁时,消息每到达一个节点,Agent 都已经离开,这样消息不断追随目标 Agent 在各个网络节点间转发却永远无法送达,从而造成了消息追逐问题。

如前文所分析,纯粹的异步方式不可能解决消息追逐问题,所以当系统中出现或可能出现消息追逐时,必须采用同步通信方式,强迫 Agent 停止移动,等待追逐它的消息送达后才能继续自主移动。为此,我们从两方面入手实现同步:追逐消息登记和 Agent 超速等待。所谓“追逐消息登记”是指消息被转发的次数达到一个给定的值 C 。(称之为“追逐判定参数”)时,便认为该消息处于“追逐状态”而将此消息的编号提交给目标 Agent 出生节点保存。另一方面,Agent 每移动到一个节点都要计算自己的移动速率 V :曾经到达过的节点个数/(当前时间一出生时间)或者现在时刻以前的一段时间内到达的节点个数/时间跨度。当 V 超过给定的最高速率 V_o 时,Agent 必须与其出生地建立连接,获得保存在 Home 上的追逐消息编号列表,然后 Agent 被限制在当前节点,直到收到所有列表中包含的追逐消息后才能继续移动。这就是“Agent 超速等待”。

除此以外,我们又设定了 Agent 定期检测时间 T_s ,即 Agent 每隔 T_s 时间向 Home 获取追逐消息列表(由于已经有 Agent 超速等待, T_s 可以设得很大),然后停留等待这些消息的到达。这样,通过结合追逐消息登记以及 Agent 超速等待和定期检测,MEFS 解决了消息追逐问题,实现了 Agent 间的可靠通信,而无论移动 Agent 系统中的网络传输延时发生怎样的无规则变化。

4.2 MEFS 参数分析

对状态超时参数 T_o ,我们可以分析如下:

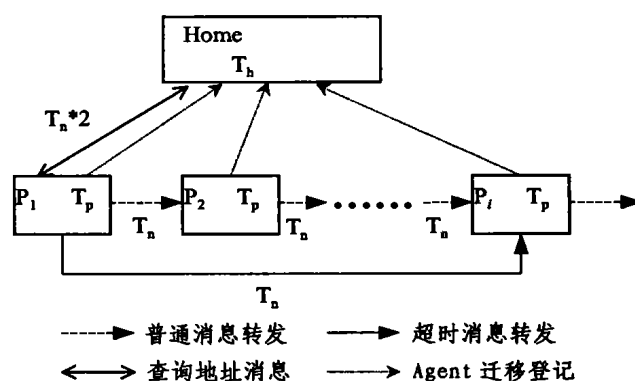


图3 状态超时参数分析

如图3,假设网络各节点相互连通,且通信延时为 T_n ,中间转发节点从接收到消息到转发消息的处理时间为 T_p ,Home 处理 Agent 地址查询请求的时间为 T_h ;Agent 的平均移动速率为 v ,并且已经从节点 P_1 循序移动到了 P_i (当前 Home 登记的地址为 P_i)。Agent 发生迁移后,会在节点 $P_1, P_2 \dots P_i$ 留下迁移状态信息分别指向 $P_2, P_3 \dots P_i$,如有没有状态超时的规定,这些信息会被一直保留(除非被新的信息覆盖)。

在状态信息存在的情况下,消息从 P_i 发送到 P_j 的时间代价为 $(i-1)T_s + iT_p$ 。

由于我们设定了 Timeout, 因此(从 Agent 移动开始计时的) T_0 时刻后, P_i 将先向 Agent 的 Home 查询当前地址, 然后直接转发消息到 P_j , 此时的时间代价为 $T_A + 2T_s + 2T_p$ 。联立两式得

$$(i-1)T_s + iT_p \geq T_A + 2T_s + 2T_p$$

代入 $i = \bar{v}T_0$ ($\bar{v}$ 为该 Agent 在这一段时间内的平均速度), 可得

$$T_0 \geq \frac{T_A + 2T_p + 3T_s}{(T_p + T_s)\bar{v}} \quad (1)$$

如果 $T_s \gg T_A + T_p$, 则有

$$T_0 \geq \frac{3}{\bar{v}} \quad (2)$$

或者

$$i \geq 3 \quad (3)$$

在 MEFS 机制中, 如果各节点处理消息的时间足够短, 那么只要 Agent 已经迁移超过两次, 引入状态超时就可以使得消息发送过程缩短。由式(2)还可以发现, 状态超时参数与 Agent 平均移动速率成反比, 并且其最小值与网络传输时间无关。

对于最高速率参数 V_0 、追逐判定参数 C_0 和定期检测参数 T_s 的设定, 我们首先要确保消息的送达, 然后在此基础上以提高系统性能为进一步目标, 并且所有参数都可以随着网络环境和机器负载作动态变化。这些都将在今后的工作中作深入的讨论。

4.3 性能比较

MEFS 相对于鸵鸟法和一般的检测法可以在网络容错的前提下绝对保证消息的送达, 是一个可靠系统。而在系统性能上, MEFS 则较预防法有较大的改善。系统性能指标可以是移动 Agent 系统中的消息平均送达时间, Agent 的被动扣留时间、系统负载等, 也可以是它们的综合。而在各个方面, MEFS 与预防法的代表 Mogent 相比都表现得较为优异。

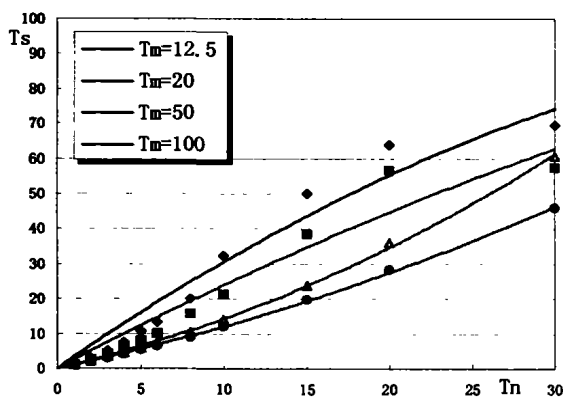


图4 MEFS 消息平均送达时间

由于 MEFS 采用的转发机制不会像 Mogent 系统那样, 每次都需要向 Home 查询目标 Agent 当前地址, 更不会出现寻址消息阻塞的情况, 因此总体上 MEFS 的消息平均送达时间要短。图4、图5分别是对于不同的 T_m 取值 (T_m 为移动 Agent 在系统各节点的平均停留时间), 由模拟实验得到的 MEFS 和 Mogent 的消息平均送达时间随 T_n 变化的二次回归曲线。两图比较可见, 相同条件下 Mogent 比 MEFS 的平均送达时间要多50%到200%不等, MEFS 在绝大多数情况下要优于 Mogent。

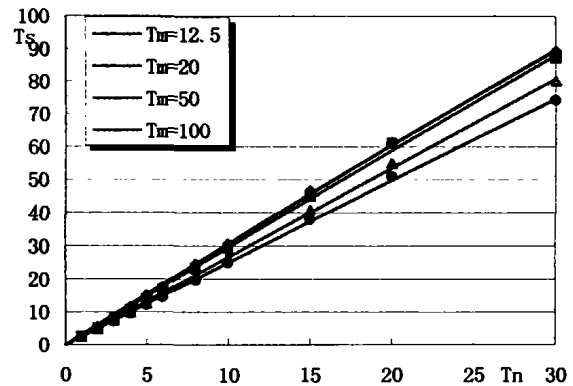


图5 Mogent 消息平均送达时间

MEFS 仅在 Agent 超速的情况下和定时检测的时候限制 Agent 的移动; 而 Mogent 系统中, Agent 每次移动都必须进行同步通信, 故此 MEFS 对 Agent 的扣留时间要远远小于 Mogent。

对于系统负载, MEFS 对 Agent 和 Agent 之间的消息都采取分布式的管理, 不可能产生系统瓶颈。而且 MEFS 控制消息较少, 消息送达前的转发次数也不会很多, 因此对网络负载也不会产生很大的影响。

总结与将来的工作 移动 Agent 的通信机制是移动 Agent 系统的重要组成部分, 本文列举了移动 Agent 系统中存在的问题, 分析了解决方案, 并提出了一种基于检测法的信息传递机制-MEFS (T_0 , C_0 , V_0 , T_s), 在保证通信可靠性的前提下, 有效地解决了通信失效等问题。MEFS 的性能要优于大多数通信机制, 并且对消息组播和消息保序都可以有较好地支持, 是一种多功能有效可靠的移动 Agent 间通信机制, 应用范围将会相当广泛。

MEFS 在参数的动态调整和路径压缩等方面, 还需要深入地加以研究。我们将在以后的工作中继续探讨, 以进一步提高其性能。

致谢 吕建教授对本文做出了悉心的指导, 乔浩硕士和汪孟余工程师对 MEFS 系统的实现和分析给予了很大的帮助, 在此表示真挚的感谢!

参考文献

- 1 Labrou Y, Finin T. a Proposal for a new KQML Specification, 1997. <http://www.cs.umbc.edu/kqml/papers>
- 2 McCabe F G. Interagent Communication Reference Model, V0.4 april 2000. <http://www.nar.fujitsulabs.com/documents/icm-manual.htm>
- 3 陶先平. 基于 Internet 的移动 agent 技术和应用研究:[南京大学博士学位论文]
- 4 baumann J, et al. Communication Concepts for Mobile Agent Systems In Mobile Agents. In: Proc. 1st Int. Workshop, Ma '97. Springer, 1997
- 5 baumann J, Hohl F, Straßer M, Rothmel K. Mole-Concepts of a Mobile Agent System In WWW Journal, Special issue on applications and Techniques of Web Agents, 1998
- 6 Murphy A L, Picco G P. Reliable Communication for Highly Mobile Agents In Agent Systems and architecture/Mobile Agents (aSa/Ma)'99, Oct. 1999
- 7 冯新宇. Design and analysis of Mobile Agent Communication Protocols:[南京大学硕士学位论文]
- 8 McCirmick J, Chacon D, McGrath S, Stoneking C. A Distributed Event Messaging System for Mobile Agent Communication
- 9 Desbiens J, Lavoie M, Renaud F. Communication and Tracking Infrastructure of a Mobile Agent System. In: Proc. 31st Annual Hawaii Intl. Conf. on System Science, 1998, 7:54~63