

移动 Agent 系统的主动通信机制*

杨博⁺, 刘大有, 杨鲲, 张朝辉

(吉林大学 计算机科学与技术学院, 吉林 长春 130012)

(吉林大学 符号计算与知识工程国家教育部开放实验室, 吉林 长春 130012)

Active Communication Mechanism of Mobile Agent System

YANG Bo⁺, LIU Da-You, YANG Kun, ZHANG Zhao-Hui

(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

(Open Symbol Computation and Knowledge Engineering Laboratory of State Education Department, Jilin University, Changchun 130012, China)

+ Corresponding author: Phn: 86-431-5166479, Fax: 86-431-5166-063, E-mail: yangbo@email.jlu.edu.cn; yangbo@hotmail.com

<http://www.jlu.edu.cn>

Received 2002-10-22; Accepted 2003-03-04

Yang B, Liu DY, Yang K, Zhang ZH. Active communication mechanism of mobile Agent system. *Journal of Software*, 2003,14(7):1338~1344.

<http://www.jos.org.cn/1000-9825/14/1338.htm>

Abstract: The issue of reliable communication resulting from Agent moving is studied in the paper. Based on analyzing current solutions to the issue, a mechanism called Active Communication Mechanism (ACM) is proposed, which can provide reliable and efficient communication for mobile Agents. The detailed analysis regarding the reliability and efficiency of ACM is also presented. ACM can reliably and efficiently deliver the sender's message to receiver allowing the free migration of both them and guaranteeing the exactly-once semantics for message delivery. ACM can serve as the basis of high-level communication mechanisms such as Agent communication language.

Key words: Agent; mobile Agent; message passing; communication mechanism; reliable communication

摘要: 解决由 Agent 移动产生的可靠性通信问题, 在分析已有方法的基础上, 提出一种保证移动 Agent 之间可靠、高效通信的“主动通信”机制, 并给出它的可靠性分析和通信效率分析。它能够在通信双方自由移动的情况下, 将消息可靠、高效地从发送方提交给接收方, 并承诺消息传递的 exactly-once 语义, 为 Agent 通信语言等高层通信方式奠定了可靠的基础。

关键词: 主体; 移动主体; 消息传递; 通信机制; 可靠通信

中图法分类号: TP393

文献标识码: A

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2001AA115160 (国家高技术研究发展计划(863))

第一作者简介: 杨博(1974—), 男, 河南新乡人, 博士, 讲师, 主要研究领域为移动 Agent 技术, 多 Agent 系统, 知识工程。

通信是移动 Agent 系统的关键技术之一.通信的可靠性和效率直接影响到整个移动 Agent 系统的性能.如何保证移动 Agent 之间能够可靠、高效地进行通信仍是移动 Agent 系统的一个研究重点.

本文所讨论的通信可靠性问题不是针对网络底层的通信设施所引起的数据包丢失,而是指由于 Agent 自主的移动行为导致发送给它的信息无法被正确接收.例如,Agent A 向 Host1 上的 Agent B 发送信息,在信息的传输过程中,B 从 Host1 移动到了 Host2,因此当信息到达 Host1 时,无法提交给信息的接收者.这种在信息传输过程中由于目标 Agent 发生物理位置的变化,从而导致信息无法正确提交的现象称为通信失效(communication failure).

尽管 A 可以移动到 B 所在的节点,与 B 进行本地通信,从而避免通信失效问题.但是,在某些情况下,Agent 之间的远程通信是必要的.比如在分布式信息查询中,多个 Agent 移动到不同节点,并行地搜索信息,在搜索过程中,Agent 之间经常需要交换信息,协同工作.再如,为了保证系统的安全,有些节点(如公司中重要的服务器)不允许 Agent 移动到其上.进行本地访问,Agent 只能移动到附近节点与其进行远程通信.

1 研究现状与问题

移动 Agent 系统的建造者为了解决通信失效问题,给出了各种方法.在仔细分析与评价这些方法之后,我们将它们分为两大类,一类是“移动受限的通信方式”,另一类是“移动自由的通信方式”.

发生通信失效的根本原因在于通信过程中 Agent 发生了移动.因此,解决该问题的最简单方法是,通信双方遵守共同的约定,即通信未结束前不能发生移动.这种方法称为“移动受限的通信方式”.该方法可以避免由于移动导致的信息丢失,实现也相对简单.早期的移动 Agent 系统(如 Agent TCL^[1])大多采用这种通信方式.但如果双方通信量大,持续时间长,则会长时间延迟 Agent 的移动.另外,当 Agent 同时和多个 Agent 通信时,必须等待所有通信过程全部结束后才能响应移动请求.在某些情况下,移动请求必须及时被处理.例如,Agent 收到当前主机可能崩溃的警告以后,必须立即移动到其他主机.

以上方法的一种改进是,当通信的一方需要移动时,向对方发送通知,等待对方的应答;对方收到通知以后,停止发送信息,发送应答;需要移动的一方在收到对方的应答后发生移动,当到达新位置以后,重新建立连接,从中断处继续通信.由于在移动过程中双方停止通信,因此不会发生信息丢失.该方法提高了移动请求的优先级,可以及时地响应移动请求.但该方法仍有较大的局限性.例如,由于底层网络不可靠,通信一方发送给请求移动一方的应答信息丢失,或故意不发送应答信息,都会导致 Agent 无法移动.当多个 Agent 相互通信时,请求移动的一方必须收到所有其他 Agent 的应答信息后才能移动.

Mogent^[2]作了进一步的改进.Agent 在移动前无须征求对方同意,而是检查“途中信件数”是否为 0.若为 0,说明没有信件在发送途中,可以立即移动;否则,等待所有途中信件到达后才能移动.

通过以上的分析不难看出,“移动受限的通信方式”将“通信”和“移动”看作是互斥的两个过程,保证它们被“串行”处理,即在通信的过程中不发生移动,在移动的过程中不发生通信,从而避免了通信失效.这种通信方式损害了 Agent 移动的自主性.

在移动自由的通信方式中,通信双方可以自由移动,不必等待整个通信过程结束或征求对方的同意.该方式将“通信”和“移动”看作是独立的两个过程,并行处理,保证 Agent 移动的自主性.移动自由的通信方式必须保证在 Agent 移动行为不可预测的情况下消息被可靠地提交.实现该通信方式的方法可以归纳为两类:广播方式和消息转发方式.

第一,广播方式.最初,研究者借用了 Emerald 系统^[3]的广播方式,将网络看作是以消息发送者为根节点的树型结构,广播使消息按照某种策略(深度优先,广度优先)在网络中传播,到达所有叶节点.这种方法有时不能保证消息的可靠传递.如当消息从节点 1 向节点 2 传递时,Agent 正好以相反的方向移动,两者在网络中交错,消息不能被正确提交.Murphy 等人在文献[4,5]中改进了广播算法,在每个节点上保留消息的副本,直到该消息被正确地提交为止.广播方式的缺点是显然的,消息副本过多会增加网络流量,每个节点可能会无限期、无止境地存储消息副本;另外,广播方式不能保证每个消息只被接收一次,即不能承诺消息传递的 exactly-once 语义.

第二,消息转发方式.根据定位机制,消息转发方式又可细分为路径转发和 Home 转发.Mole^[6]采用路径转发

方式.Agent 途经的每个节点都保留有后继节点的位置信息,它们共同维护 Agent 的移动路径.发送给 Agent 的消息按这条路径传递,直到消息被正确接收为止.路径转发方式不能保证消息能否或何时被正确提交给 Agent.如 Agent 总是在消息到达前移动到新位置,可能导致消息永远无法被 Agent 接收.Aglets^[7]和 Voyager^[8]采用 Home 转发方式.Home 是创建 Agent 的主机,用于管理 Agent 的位置信息.Agent 每移动到一个新位置都向其 Home 报告新地址.所有发送给 Agent 的消息,首先发送给其 Home,然后 Home 根据记录的地址信息转发给 Agent.Home 转发方式有时也会发生通信失效,如当 Home 按它记录的地址向 Agent 发送信息时,Agent 正好移动到其他节点.文献[9]给出了 Home 转发的改进方法,可以保证信息的可靠提交.

Home 转发方式最大的缺点在于通信效率低.当从 Home 派生的 Agent 数量较多时,所有发送给这些 Agent 的消息都需要通过 Home 转发,大大增加了 Home 的负担.当信息量大到一定程度时,Home 的性能成为通信效率的瓶颈.当 Home 接收的信息量超过它的处理能力时,会导致 Home 崩溃.此外,由于信息需要转发一次,因此通信过程的网络总流量将是通信量的两倍.

2 主动通信机制

从以上分析可知,目前用于解决“通信失效”的方法或者存在缺陷,或者效率不高,不能满足实际系统的要求.我们在这些工作的基础上提出了一种保证移动 Agent 可靠、高效通信的新方法,称为“主动通信”机制.

主动通信机制与其他方法最主要的区别是,消息不发生移动,接收者主动获取发给它的消息,从而避免了通信失效问题.而在其他方法中,消息向接收者移动,接收者被动地等待消息到达,在等待过程中,若接收者发生移动,则会发生通信失效.此外,与 Home 转发机制不同,主动通信机制把发送消息的任务分散给网络中的多个节点,避免某一个节点负载过重.主动通信机制的体系结构如图 1 所示.Place 是移动 Agent 的运行环境.Home 和 Communicator 是运行在 Place 中的两个服务 Agent.Home 负责移动 Agent 的定位.Home 记录 Place 创建的移动 Agent 的信息,移动 Agent 每到达一个新 Place,向其 Home 报告新 Place 的地址信息.Communicator 负责 Agent 之间的通信.Home-Communicator 合作完成消息的可靠传送.

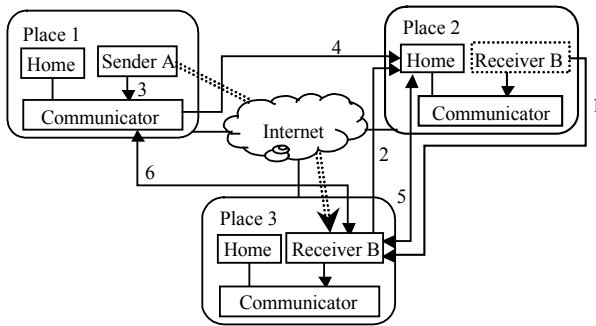


Fig.1 The workflow of active communication mechanism
图 1 主动通信机制的工作流

下面我们来说明主动通信机制的主要思想.Agent B 从“出生地”Place 2 移动到 Place3(图 1 中的 1);B 到达 Place 3 以后,向其 Home 报告新位置(图 1 中的 2);发送方 A 将要发送的消息发送给当前 Place 的 Communicator(图 1 中的 3); Communicator 通知接收方 B 的 Home(图 1 中的 4);Home 通知 B 到 Place1 取信息(图 1 中的 5);B 主动联系 Place1 的 Communicator 并取回信息(图 1 中的 6).接收方的地址对于发送方是透明的,发送方在发送信息的时候只需指定接收者的名字,而不必关心接收方在哪里或是

否正在移动.发送方将消息发送给本地的 Communicator 以后,即认为消息已经正确地发送到接收方,可以继续执行其他操作.主动通信机制由 Home-Communicator-Mobile Agent 三方协作完成.以下给出了三者与通信有关的模块描述.为了便于描述,假定 Agent A 向 Agent B 发送信息.

Communicator 包括 4 个与通信有关的操作.Communicator 为每个发送消息的 Agent 创建一个消息队列,Queue_A 表示 Agent A 对应的消息队列,存放 A 发送的数据消息.Queue_A 有两个状态变量 hassend 和 interrupt. Hassend 避免 Communicator 频繁地向接收方的 Home 发送通知.当 Communicator 向接收方的 Home 发送过一次通知以后,hassend 被置为 true,当接收方取空 Queue_A 以后,hassend 被置为 false.Interrupt 表示 Communicator 和接收方的交互状态,在交互过程中,接收方发生移动,interrupt 被置为 true,触发中断处理操作 Interrupt.

```
ReceiveMessage(msg){ /* Communicator 收到 A 发送的消息时,该操作被触发 */
    将 msg 存入 QueueA;
```

```

    IF (QueueA.hassend=false){向 B 的 Home 发送“新消息”通知; QueueA.hassend=true;}}
QueueNull(){ /* 消息队列 QueueA 被取空时触发该操作,Communicator 不必无限期为 A 保留通信资源 */
    IF (A 已经离开本地) 释放消息队列 QueueA 和其他相关资源;}
CreateQueue(){ /* 接收到 Agent A 请求创建消息队列的请求时,该操作被触发 */
    创建 QueueA; QueueA.hassend=false; QueueA.interrupt=false;}
Interrupt(){ /* 接收到接收方的中断请求时(Queue.interrupt=true),该操作被触发 */
    IF (QueueA 不空) {向 B 的 Home 发送“新消息”通知; QueueA.hassend=true;} /* 中断时消息没有取完 */
    ELSE QueueA.hassend=false; /* 中断时消息已取完 */
    QueueA.interrupt=false;}

```

Home 包括两个与通信有关的操作.Queue 为 Home 的消息队列,存放来自 Communicator 的“新消息”通知.HaveMessage 表示消息队列的状态.Loc 记录 Agent 的当前位置.isReading 记录 Agent 与 Home 之间的交互状态,如果 Agent 正在读取 Queue 中的消息则为 true,否则为 false.

```

ReceiveMessage(msg){ /* Home 收到 Communicator 发送的通知时,该操作被触发 */
    将“新消息”通知 msg 存入消息队列 Queue;
    IF (isReading=false) 发送“消息到达”通知给 Loc 位置的 Agent;}
Updata(in Location, out haveMessage){ /* Agent 到达新位置时触发该操作 */
    Loc=Location;
    IF (Queue 不空) haveMessage=true;
    ELSE haveMessage=false;}

```

移动 Agent 包括 6 个与通信有关的操作.Queue 为 Agent 的消息队列,存放来自 Communicator 的“新消息”通知.CurrentLoc 为 Agent 当前所在的位置.LocofCom 记录 Communicator 位置.state 记录 Agent 与 Communicator 之间的交互状态,如果 Agent 正在读取 Communicator 的消息队列,则为 true,否则为 false.

```

OnArrive(){ /* 每当移动 Agent 到达新位置时,该操作被触发 */
    ProcessingMessage(); /* 启动线程 ProcessMessage,处理 Agent 的消息队列 Queue */
    Home->Updata(CurrentLoc,haveMessage);
    IF (haveMessage=true) 发送“消息到达”通知给自己;}
GetMessagefromHome (){ /* Agent 接收到“消息到达”通知时,触发该操作 */
    Home->isReading = true; /* 表示 Agent 正在读取 Home */
    While (Home->Queue 不空) {
        从 Home->Queue 读取一个通知 msg; /* 该操作被定义为原子操作 */
        将 msg 存入自身的消息队列 Queue;};
    Home->isReading=false; /* 读取完成 */
    Wake ProcessingMessage; /* 唤醒线程 ProcessingMessage,处理消息队列 Queue */}
ProcessMessage(){ /* 消息处理线程,可与主线程并行执行,处理消息队列 Queue 中的通知消息 */
    While(1) {
        While(Queue 不空) {
            从 Queue 取出通知 msg; /* msg 中含有 Communicator 的地址和发送方的名字 A */
            LocofCom=Communicator 的地址;
            state=true;
            While(LocofCom->QueueA 不空) {
                从 LocofCom->QueueA 中读取一个数据消息; /* 该操作被定义为原子操作 */};
            LocofCom->QueueA.hassend =false; /* 此时已取空 LocofCom->QueueA */
            state=false;}
    }
}

```

```

    Sleep; /* 消息队列处理完后挂起 */}
InitCommunication(){ /* 如果 Agent 需要向其他 Agent 发送消息,则首先执行该操作 */
    向本地的 Communicator 请求创建消息队列;}
SendMessage(msg){ /* Agent 发送消息 */
    向本地的 Communicator 发送 msg;}
OnLeave(){ /* Agent 离开当前位置时,该操作被触发 */
    通知本地的 Communicator;
    IF (state=true){LocofCom->interrupt=true; state=false }}

```

3 可靠性分析

性质 1. 如果底层网络可靠通信,则主动通信机制能保证消息的可靠提交。

需要说明的是,在主动通信机制的各个阶段(图 1 中的 3~6)都不会发生信息丢失。阶段 3:发送方向 Communicator 发送消息。由于该阶段是本地通信,因此不会丢失信息。阶段 4:Communicator 向 Home 发送“新消息”通知。由于 Home 不发生移动,并且底层网络可靠通信,因此该阶段不会丢失信息。Home 将“新消息”通知保存在其消息队列中。阶段 5:Home 向接收方发送“消息到达”通知,接收方从 Home 取“新消息”通知。接收方与 Home 的交互状态有两种。第一,接收方正在从 Home 读取消息,即状态变量 Home->isReading=true。此时,新到达 Home 的通知一定可以被接收方获取;第二,接收方没有从 Home 读取消息,即状态变量 Home->isReading=false。此时,若有新通知到达 Home,Home 会向接收方发送“消息到达”通知,此时可能有两种情况。(1) 接收方收到该通知,则执行 GetMessagefromHome,从 Home->Queue 读取所有新到达的通知。(2) 接收方没有收到该通知。由于底层通信网络不存在数据包丢失的情况,则可判定其原因是接收方发生移动,“消息到达”通知在接收方离开原位置以后到达。当接收方到达新位置以后,执行 OnArrive 中的 Home->Updata(CurrentLoc,haveMessage)主动查询其 Home 是否有新通知到达,如果有,则执行 GetMessagefromHome,从 Home->Queue 读取所有新到达的通知。阶段 6:接收方从 Communicator 读取数据消息。在此过程中,接收方可以一次取走 Communicator 的所有消息,也可以移动到不同的位置,分多次读取所有消息。在 Agent 移动过程中,剩余的消息始终保存在 Communicator 的消息队列中,没有发生传输,等待下次被读取。底层网络的可靠通信保证每次读取消息的过程中不会丢失信息。因此,该阶段不会发生信息丢失。综上所述,如果底层的网络不存在数据包丢失,则本通信机制能够保证所有消息的可靠提交。

性质 2. 主动通信机制属于“自由移动通信”方式。

需要说明的是,发送方和接收方在通信过程中都可以自由移动,且不会影响通信的可靠性。第一,在通信过程中,发送方移动。发送方每到达一个新位置,就向本地的 Communicator 申请一个消息队列,将要发送的消息存入该消息队列。Communicator 负责将消息队列中的所有消息可靠地送到接收方。在发送过程中,发送方可以随时移动到新位置,向新位置的 Communicator 继续发送剩余消息。第二,在通信过程中,接收方移动,又可以分为以下 5 种情况讨论。(1) 接收方在发送方发送消息的过程中移动(图 1 中的 3)。发送方并不是将消息直接发给接收方,而是发送给本地的 Communicator。性质 1 保证 Communicator 能将这些信息可靠地提交给接收方。(2) 接收方在 Communicator 和 Home 交互的过程中移动(图 1 中的 4)。接收方到达新位置时,主动查询 Home 状态,与 Home 交互,获取 Communicator 发送给 Home 的通知。(3) 接收方在和 Home 交互的过程中移动(图 1 中的 5)。接收方从 Home 的消息队列中读取消息,在此过程中发生移动。当接收方到达新位置时,主动查询 Home 的状态,如果 Home 的消息队列中还有消息未被读取,则执行 GetMessagefromHome 操作,读取剩余消息。(4) 接收方在和 Communicator 交互的过程中移动(图 1 中的 6)。接收方从 Communicator 的消息队列 Queue_A 中读取消息,在此过程中发生移动。接收方在离开原位置前触发 OnLeave 操作,执行 LocofCom->interrupt=true 语句,该语句触发 Communicator 的 Interrupt 操作。此时有两种情况。第一,Queue_A 中仍有消息未被读取,Communicator 向接收方的 Home 发送“新消息”通知,当接收方到达新位置时,主动从其 Home 读取该通知,放入自己的消息队列,当 ProcessMessage 线程处理到该通知时,继续从 Communicator 的 Queue_A 中读取剩余消息;第二,Queue_A 中所有消

息均被读取,执行 $Queue_A.hassend=false$,保证在有新消息进入 $Queue_A$ 时,Communicator 向接收方的 Home 发送通知。(5) 接收方在自己的消息队列未处理完时移动.当接收方到达新位置时,触发 OnArrive 操作,该操作启动消息处理线程 ProcessMessage,继续处理消息队列中尚未处理的消息.通过分析以上 5 种情况可知,接收方在接收和处理消息的各个阶段都可以自由的移动,而不会造成信息的丢失.

综上所述,本通信机制属于“自由移动”通信方式.

性质 3. 如果底层网络可靠通信,主动通信机制承诺消息提交的 exactly-once 语义.

底层通信网络可靠,性质 1 和性质 2 保证每个消息至少能被接收方接收 1 次.每个消息只被发送 1 次,在网络中只有 1 个副本,因此每个消息最多被接收方接收 1 次.所以每个消息能被且只能被接收方接收 1 次.

4 效率分析

本节分析主动通信机制的通信效率,并与 Home 转发机制(文献[9]提出的改进算法)、广播方式(文献[4]提出的改进算法)进行比较.影响通信效率的主要因素是:Home 的负载、通信流量和通信时间.假定网络中有 m 个节点,移动 Agent A 向移动 Agent B 发送信息,每次发送的信息量为 X ,共发送 n 次. L 表示 Home 的负载量, B 表示通信过程的网络流量, T 表示通信时间. C 表示通知消息的大小. δ 表示网络的平均带宽.

在 Home 转发方式中,发送一次消息的过程如图 2(a)所示.① A 向 B 的 Home 发送 X 大小的消息;② Home 向 B 发送“消息到达”通知;③ B 从 Home 取回 X 大小的消息.发送一次消息,Home 的负载量 $L=X$,网络流量 $B=2X+C$.因此,整个通信过程的通信开销表示为式(1).

$$\begin{cases} L_{Home} = n \cdot X \\ B_{Home} = n(2X + C) \\ T_{Home} = B_{Home} / \delta \end{cases} \quad (1)$$

在主动通信机制中,发送一次消息的过程如图 2(b)所示.① A 向本地的 Communicator 发送 X 大小的消息;② Communicator 向 B 的 Home 发送“新消息”通知;③ Home 向 B 发送“消息到达”通知;④ B 从 Home 取回“新消息”通知.⑤ B 从 Communicator 取回 X 大小的消息.发送一次消息,Home 的负载量 $L=C$,网络流量 $B=X+3C$.因此,整个通信过程的通信开销表示为式(2).

$$\begin{cases} L_{主动} = n \cdot C \\ B_{主动} = n(X + 3C) \\ T_{主动} = B_{主动} / \delta \end{cases} \quad (2)$$

在广播方式中,每个节点收到消息后都向其他节点转发.每发送一次消息,所有节点都会收到 $m-1$ 个副本,因此 Home 的负载量 $L=(m-1)X$,网络流量 $B=P_m^2 \cdot X$.因此,整个通信过程的通信开销表示为式(3).

$$\begin{cases} L_{广播} = n \cdot (m-1) \cdot X \\ B_{广播} = n \cdot P_m^2 \cdot X \\ T_{广播} = B_{广播} / \delta \end{cases} \quad (3)$$

在主动通信机制和 Home 转发方式中,Home 负载和网络流量与网络的节点数无关.但在广播方式中, $L_{广播}=O(m)$, $B_{广播}=O(m^2)$,网络节点个数的增加会导致 Home 负载和网络流量迅速增加,因此该通信方式的效率最低.下面比较主动通信机制和 Home 转发方式的效率.(1) 当 $C < X$ 时, $L_{主动} < L_{Home}$.当 $C \ll X$ 时, $L_{主动} \ll L_{Home}$.(2) 当 $C < 0.5X$ 时, $B_{主动} < B_{Home}$;当 $C < X$ 时, $B_{主动} \approx 0.5B_{Home}$.(3) 通信时间主要由信息在网络中的传输时间来决定,假定网络带宽为平均值,则通信时间与网络流量成正比.

在实际系统中,Communicator 发送给 Home 的通知只需包含该 Communicator 的地址和发送方的名字,Home 发送给接收方的通知不需要包含任何信息.因此,通知消息的字节数很少.为了进一步提高主动通信机制的效

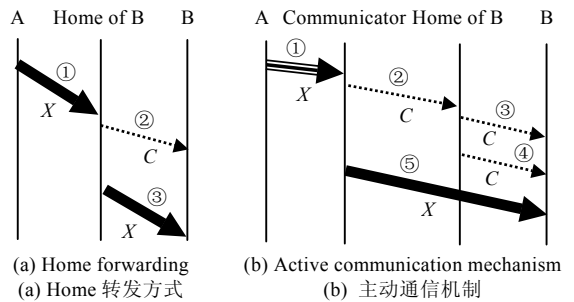


Fig.2 The process of sending one message in two mechanisms

图 2 两种通信机制发送一次消息的过程

率,发送方可以减少发送的次数,使 $X \gg C$.

5 实验

我们在移动 Agent 原型系统 JamAgent^[10]上实现了 3 种通信机制:主动通信、Home 转发(文献[9]提出的改进算法)和广播方式(文献[4]提出的改进算法),并对 3 种通信机制的效率进行了实验.在实验中,每个 JamAgent 系统代表一个节点,共有 3 个节点.每个消息的大小为 2k,“通知”消息的大小为 0.1k.图 3(a)表示 Home 负载量随消息数目的变化趋势,图 3(b)表示网络流量随消息数目的变化趋势.

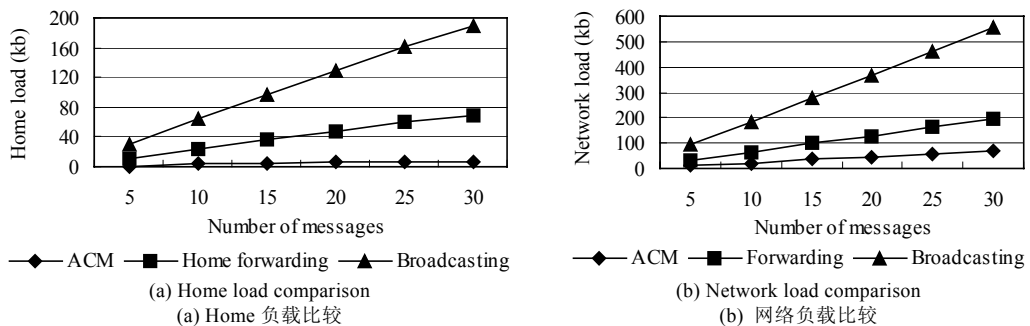


Fig.3 Communication efficiency comparison of three communication mechanisms

图 3 3 种通信机制通信效率的比较

6 结 语

本文研究移动 Agent 的可靠性通信问题,分析了该问题的研究现状及已有方法的不足,提出了一个能够保证移动 Agent 之间可靠、高效地进行通信的主动通信机制,并深入分析了该机制的可靠性和效率.与已有方法相比,主动通信机制具有以下优点:(1) 保证在 Agent 自由移动的情况下将消息可靠地提交给接收方;(2) 承诺消息传递的 exactly-once 语义;(3) 具有较高的通信效率.基于消息传递的主动通信机制为 Agent 通信语言(Agent communication language)等高层通信方式的研究奠定了可靠的基础.

References:

- [1] Gray R. Agent Tcl: A flexible and secure mobile Agent system [Ph.D. Thesis]. Hanover: Dartmouth College, 1997.
- [2] Tao XP, Feng XY, Li X, Zhang GQ, Lü J. Communication mechanism in Mogent system. Journal of Software, 2000,11(8): 1060~1065 (in Chinese with English abstract).
- [3] Jul E, Levy H, Hutchinson N, Blad A. Fine-Grained mobility in the emerald system. ACM Transactions on Computer Systems, 1988,6(2):109~133.
- [4] Murphy AL, Picco GP. Reliable communication for highly mobile Agents. In: Spring P, Milojevic D, eds. Proceedings of the 1st International Symposium on Agent Systems and Applications. New York: IEEE Computer Society, 1999. 141~150.
- [5] Murphy AL, Roman GC, Varghese G. An exercise in formal reasoning about mobile communications. In: Proceedings of the 9th International Workshop on Software Specification and Design. New York: IEEE Computer Society, 1998. 25~33.
- [6] Baumann J, Hohl F, Straber M, Rothermel K. Mole—Concepts of a mobile Agent system. World Wide Web Journal, 1998,1(3): 123~137.
- [7] Lange D, Oshima M. Programming and deploying Java™ mobile Agents with Aglets™. Reading: Addison-Wesley, 1998. 91~117.
- [8] ObjectSpace, Inc. ObjectSpace voyager core package technical overview. 1997. <http://www.objectspace.com/voyager/whitepapers>.
- [9] Wu G, Wang HM, Wu QY. An algorithm on location management and reliable communication for mobile Agents. Journal of Software, 2002,13(2):269~273 (in Chinese with English abstract).
- [10] Yang K, Liu DY, Guo X. A template architecture for mobile Agent system of high security. Journal of Software, 2002,13(1): 130~135 (in Chinese with English abstract).

附中文参考文献:

- [2] 陶先平,冯新宇,李新,张冠群,吕建. Mogent 系统的通信机制. 软件学报, 2000,11(8):1060~1065.
- [9] 吴刚,王怀民,吴泉源. 一个移动智能体位置管理与可靠通信的算法. 软件学报, 2002,13(2):269~273.
- [10] 杨鲲,刘大有,郭欣. 一个具有高安全性的移动 Agent 系统模板结构. 软件学报 2002,13(1):130~135.