一种基于指向路径分次删剪的移动 Agent 通信算法*)

贾志勇 罗 杰 王德强 谢 立
(南京大学计算机系软件新技术国家重点实验室 南京210093)
(南京大学计算机科学与技术系 南京210093)

摘 要 在移动 agent 环境下,agent 的移动为应用程序的开发提供了更灵活的通信处理方式,但也同时对通信算法的设计提出了许多挑战,其中最突出的就是由于消息传输和 agent 移动之间的异步性而造成的通信不可靠问题。针对目前解决该问题的各种机制和算法的不足之处,该文提出了一种新的 agent 通信算法—S_COMP,它综合采用了 Home 寻址、途经节点转发、指向路径分次删剪和集中同步等手段,能够适应具有不同移动和通信特点的 agent 的通信需要,在确保消息传输可靠性的同时兼顾了效率和适应性。

关键词 移动 agent,agent 通信,可靠通信

A Highly Adaptive and Reliable Algorithm for Mobile Agent Communication

JIA Zhi-Yong LUO Jie WANG De-Qiang XIE Li
(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093)
(Department of Computer Science and Technology, Nanjing University, Nanjing 210093)

Abstract In mobile agent environment, the migration of agent provides a more flexible way to deal with communication in application development, while it also provides many challenges to the design of the communication algorithm. Among them, the most obvious one is the loss of messages caused by the asynchronous nature of message passing and agent migration. With the drawbacks of current mechanisms and algorithms to solve this problem considered, this article provides a new agent communication algorithm—S_COMP. The algorithm uses such methods as home registration, migration path forwarding, path shortening by stages and central synchronization, and can meet requirements of agents with different migration and communication properties. In addition to ensuring the reliability, it also gives considerations to efficiency and adaptivity.

Keywords Mobile agent, Agent communication, Reliable communication

移动 agent 技术是近年来较活跃的一个研究领域,在移动计算、分布信息处理、服务定制、电子商务、软件集成以及网络监控和管理等领域都有着良好的应用前景[1]。

在移动 agent 应用中,agent 除了进行大量本地交互外,还要通过与其它节点上 agent 的通信来交换一些状态、控制和协作信息。由于 agent 能自主移动,它就可能在消息到达之前离开当前节点,从而造成消息丢失。这种消息丢失与网络和节点故障无关,是纯粹由 agent 的移动造成的,而且即使采用了简单的广播机制和跟踪转发方法,也无法确保消息的可靠传输[2]。目前绝大多数的移动 agent 系统并没有对它给出相应的处理机制,只是简单假定 agent 在通信期间是静止的或是将这一问题丢给开发人员处理。已提出的一些算法[2~7]也有额外通信过多、过度限制 agent 移动、适应性差、算法实现困难等不足。针对这些不足,我们提出了一种具有良好可靠性、效率和适应性的算法—S_COMP,它在保证通信可靠性的同时支持位置透明的消息传输,并且能够适应具有不同移动和通信特点的 agent 的通信需要。

接下来的第1部分讨论相关研究工作,第2部分给出对 S_COMP 算法的描述,第3部分对算法的正确性加以证明,第4部分给出算法的性能分析,最后是对全文的总结和致谢。

1 相关研究工作

A. Murphy 等[2]提出了一个基于广播的可靠传输算法,它实质上是由分布式全局快照算法演化而来的。它对 agent 的迁移不做限制,且只要做很少的改动就可以提供针对 agent 群组的广播功能。但算法的广播特性所导致的大量额外通信使得算法不适合类似 Internet 这样拥有大量节点的网络。

Mogent 系统[3]通过引入"迁移状态"和"在途信件数目"两个信号量来对通信和移动所共享的位置信息进行同步,使得信件发送者只有在接收者处于静止状态时才能向其发送信件,并且接收方在所有尚在途中的信件到达之后才能迁移。这种算法在保证通信可靠性的同时对 agent 的迁移做了过多限制,且由于每发一封信都需要进行寻址,通信效率低。

X. Feng 等[4]给出一个基于邮箱的 agent 通信算法。每个 agent 被分配一个邮箱,其它 agent 将信件发往其邮箱,需要时 agent 从邮箱中将信件取走。邮箱可以和 agent 位于不同的节点,且可在 agent 的控制下移动,邮箱经过的每一节点都记录有邮箱当前所处位置并对发往邮箱的信件进行暂存和转

*)本文研究得到国家863项目基金(No. 2001AA113050)资助。贾志勇 博士生,主要研究领域为软件 Agent 技术、分布式计算。罗 杰 硕士生,主要研究领域为软件 Agent 技术、人工智能。王德强 博士研究生,主要研究领域为分布式并行计算与移动 Agent 技术。谢 立 教授,博士生导师,主要研究领域为分布式计算。

发。邮箱按照一定的同步规则向其所经过的节点发送注销和登记消息，来同步邮箱移动和向邮箱的发信操作，从而保证通信的可靠。该算法通过 agent 与邮箱的分离减少了对 agent 移动的限制，但由于要在每次移动后向邮箱路经的所有节点以及与邮箱曾经通信的所有 agent 发送地址更新信息，随着邮箱不断移动，通信量会急剧增加。

V. Belle 等[5]给出一种将消息路由和 agent 命名融合起来的通信算法。它首先要构造一个包含所有网络节点的树状结构，每个 agent 按照其出生地在树中的位置被赋以一个名字，agent 经过的每一节点上留下指向下一站节点的指针。消息首先根据 agent 的名字发往其出生地，再按照指针的指示沿 agent 的移动路径传送给 agent。算法的缺陷在于随着 agent 的移动，转发路径会不断增长，从而引起通信开销的增大，甚至会导致个别消息一直追击目标 agent 却始终无法到达；此外，树结构在 Internet 这样的环境中构造起来很困难。

M. Ranganathan 等[6]通过在 UDP 上构造一个类似 TCP 的协议来实现 agent 间消息的可靠传输。它利用一个位置管理器来跟踪和广播参与通信的各移动端点的位置，采用类似 TCP 中滑动窗口协议的办法对通信双方的收发操作进行协调，并通过重发来保证消息的送达。该算法的执行取决于位置管理器的可用性和稳定性，而在移动 agent 环境下建立一个稳定高效的位置管理器则具有相当的难度。

在 JATLite[7]中，有一个称为 AMR 的基本服务，它是一个具有消息缓存和转发功能的名字服务器。每个 agent 只记录有自身和 AMR 的地址，把要向其它 agent 发送的信件都发给 AMR，agent 需收取消息时就建立同 AMR 的连接并取回由 AMR 暂存的消息。算法的思想虽然简单，但与文[5]中所述的机制类似，在 Internet 环境下构建一个高效且有良好可用性的 AMR 非常困难。

以上各种机制除了已指出的不足外，还有一个共同缺陷，即未对不同 agent 的移动和通信需求加以区别，笼统以相同的手段统一处理，没有兼顾算法的效率、公平性和适应性。

2 S_COMP 算法介绍

2.1 系统模型假定与算法的需求分析

为了使算法更具通用性，它所基于的 agent 系统模型与已有的大多数 agent 系统和主要标准是相容的，如下所述：

(1)一个 agent 系统由多个分布于不同网络节点上的 agent 平台组成。

(2)每个 agent 平台都提供了 agent 的基本运行环境和对底层服务的抽象。

(3)agent 能够在各个节点平台之间自主移动，并能在节点上执行、访问相关服务以及与其它 agent 发生本地和异地交互。

另外，为了简化对算法的讨论，对通信状况做出如下一些假定：

(1)agent 间的通信是以异步消息传递的形式完成的(同步消息可以在异步传输的基础上进行构建)。

(2)通信线路和网络节点无故障。

(3)设 B 和 A 分别是消息的发送和接收者，它们可以是移动的，则从 B 发往 A 的消息不会发生丢失和破坏，且满足 FIFO 特性，即消息是按照发送顺序到达的。

由第1部分关于解决移动造成的通信不可靠问题的各种算法的讨论，以及上面对系统模型和通信状况的假定，可以得出对于一个良好的移动 agent 间通信算法的基本需求，如下所述：

(1)算法的可靠性高。在不发生线路故障和节点崩溃时，消息总能经有限次的转发到达目标 agent。

(2)算法应满足通信的位置透明(Location Transparency)特性。agent 的自主移动引起的物理地址变化对程序的开发者来说是不可见的。

(3)对 agent 的自主迁移影响小。

(4)通信代价低。这里通信代价是指包括定位、同步、消息的多次转发等各种因素在内的通信总开销。

(5)可扩展性好。算法应该能够面向 Internet 环境的应用。

(6)适应性好。算法能够适应具有不同移动和通信特性的各类 agent 的通信需求。

2.2 原型算法

作为 S_COMP 算法的原型，下面的 agent 间通信算法是对我们实验室所开发的 Mogent[3]系统原有通信机制加以改进得到的。在原算法中，即使是向同一 agent 发信，也需要每发一封信前重新向 Home 进行寻址，在这里，我们通过引入地址缓存来解决这一问题。由于本地通信的效果很容易保证，我们在算法中就不对其进行讨论，在描述算法前，我们先对通信模型和相关数据结构加以说明，如下所述：

(1)每个 agent 都对应一个称为 Home 的出生地，agent 的名字由 Home 节点的地址和在该节点具有唯一性的一个字符串组成。许多 agent 系统都采用这种命名方式，它能够直接支持通信的位置透明特性。

(2) 一个名为 A 的 agent 在其 Home 处有如下一些记录：一个称为“投递地址”的物理地址，在 A 创建时，该地址初始化为 Home 节点的地址，其它 agent 将按照这一地址向 A 发送消息，在这里，它用来记录 A 当前所处的节点位置；一个用来表示信件接收状态的信号量，它可以有“允许接收”和“拒绝接收”两种取值；一组表示“通信进行”的标志量，每个标志量对应一个正在与 A 进行通信的 agent。

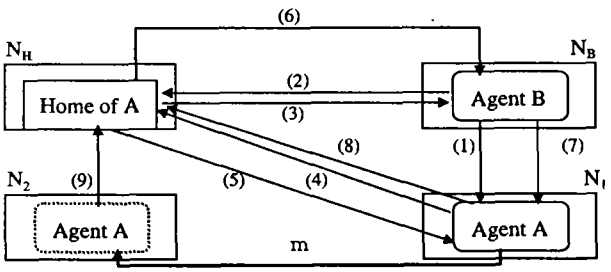

图1 原型算法中的消息发送

如图1所示，以 N_B 节点上的 agent B 向 N_1 节点上的 agent A 发送消息为例，将算法分为远程通信和迁移两部分进行说明：

a. 远程通信

(i)B 发信前先检查是否缓存有 A 的投递地址，如有，则将信件发往投递地址 N_1(A 当前所处节点位置)(图中(1))；否则，B 向 A 的 Home 发送寻址信件(图中(2))。

(ii)Home 收到寻址信件后，检查信件接收状态，若为“允许接收”，则向 B 返回 A 的投递地址 N_1(图中(3))，并设置对应 B 的通信进行标志量；若信件接收状态为“拒绝接收”，则将寻址信件放入阻塞队列，不对其标志量进行设置，直至

Home 释放这些阻塞信件后，再向信件发送者返回投递地址并设置标志量。

(iii)B 获得投递地址 N_1 后，将 N_1 缓存，并开始向 N_1 发送一系列的消息给 A。

b. 迁移

(i)A 向其 Home 发送一封请求迁移的信件(图中(4))。

(ii)Home 收到请求后，将 A 的信件接收状态置为"拒绝接收"，并检查是否有通信进行标志量被设置，如果没有，则向 A 发送"准许迁移"通知(图中(5))；若还有标志量被设置，Home 向这些标志量对应的 agent 发送"中断通信"通知(图中(6))，这些 agent 在收到通知后，停止向 A 发送消息，并向 A 发送一封"通信完成"信件(图中(7))，随后清除所缓存的 A 的投递地址。A 收到该信后将信件转发给 Home(图中(8))，Home 接到后则清除其对应的通信进行标志量。Home 在检查发现 A 的所有标志量均被清除后，向 A 发送准许迁移通知。

(iii)A 收到准许迁移通知后，开始迁移(图中(m))。当 A 到达新节点 N_2 后，向其 Home 发回注册信件(图中(9))，登记当前的投递地址 N_2。Home 收到注册信件后，将信件接收状态改为"允许接收"，释放所有阻塞信件，同时向这些被阻塞的寻址信件的发送者返回新的投递地址，并对通信进行标志量进行设置。

2.3 S_COMP 算法

S_COMP 在原型算法的基础上通过引入多个转发代理来减少对自主迁移的限制，并通过删剪指向路径来保证通信效率。在通信模型和相关数据结构方面，做出了如下增加：

(1)每个节点都能够暂存和转发信件，agent A 在从当前节点离开前会在该节点上留下一个称为转发代理(Forwarding Proxy)的结构，其中保存有 A 的下一站地址。在这里，Home 所记录的 A 的投递地址为第一个转发代理所处节点位置，该转发代理也称为基代理。

(2) 每个 agent 都携带有一个"途经节点数"变量，它在 agent 创建时为0，并在 agent 每次移动后增1。

此外，在具体讨论算法前，我们假定 A 在执行任务期间不会对同一节点进行重复访问，对于出现重复访问的情形，将在后面作为 S_COMP 的一种特殊情况来讨论。

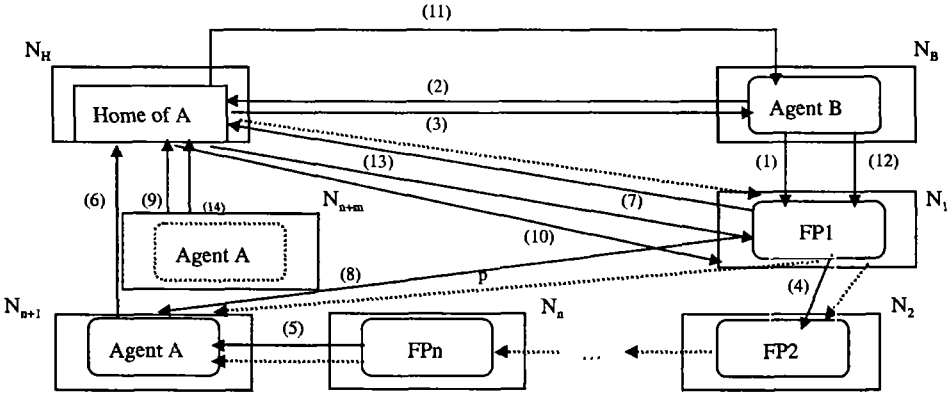

图2 S_COMP 算法中的消息发送

如图2所示，节点 $N_1, N_2, \cdots, N_n$ 上的 FP1，FP2，… ，FPn 是 A 在经过这些节点时留下的转发代理，这些转发代理共同组成 A 的指向路径，基代理 FP1位于指向路径的首部，位于指向路径末尾的 FPn 指向 A 当前所在节点 N_{n+1}。从 Home 经指向路径至 A 的指向关系如图中的虚线所示。下面仍以 B 向 A 发送消息为例，将算法分为远程通信、部分指向路径删剪和全部指向路径删剪三部分进行说明：

a. 远程通信　这一部分算法与原型算法中的远程通信基本相同，只是将原来由 A 完成的一些工作改由 FP1完成，并将其它 agent 对 A 的直接消息发送改为经由指向路径转发，如下所述：

(i)B 发信前先检查是否缓存有 A 的投递地址，如有，则将信件发往投递地址 N_1(FP1所处节点位置)(图中(1))；否则，B 向 A 的 Home 发送寻址信件(图中(2))。

(ii)Home 收到寻址信件后，检查信件接收状态，若为"允许接收"，则向 B 返回 FP1的地址 N_1(图中(3))，并同时设置对应 B 的通信进行标志量；若信件接收状态为"拒绝接收"，则将寻址信件放入阻塞队列，不对其标志量进行设置，直至 Home 释放这些阻塞信件后，再向信件发送者返回投递地址并设置标志量。

(iii)B 获得投递地址 N_1 后，将 N_1 缓存，并将要发往 A 的一系列消息发往 N_1。N_1 收到消息后，按照 FP1中保存的下一个转代理 FP2的地址 N_2 将消息转发出去(图中(4))。其它转发节点将重复这一操作，直至信件最终由 FPn 送达 A(图中(5))。

b. 部分指向路径删剪

(i)A 每到达一个新节点(设为图中的 N_{n+1})后，检查"途经节点数变量"的值，该变量的值每增加 hop1(hop1为一具体数值)，A 即向 Home 发送一封请求进行"部分指向路径删剪"操作的信件(图中(6))，信件中包含有 A 的当前位置 N_{n+1}，A 的自主移动不受发送该信件的影响。

(ii)Home 收到请求后，将请求信件转发给基代理 FP1所在节点 N_1(图中(7))，N_1 收到信件后，将 FP1中保存的 FP2的地址改为请求信件中携带的地址 N_{n+1}(如图中虚线指向 p 所示)，然后 N_1 将删剪请求信件发向原来 FP2所在的节点 N_2。此后，所有其它 agent 向 A 发来的信件都沿着新指向路径由 N_1 发往新的 FP2所在节点 N_{n+1}(图中(8))，并继续转发下去；同时，发往 N_2 的删剪请求信件会由 N_2 沿原来的指向路径转发下去，一直到信件到达 N_{n+1} 为止，在转发过程中，从 N_2 一直到 N_n 上的转发代理将被删除。

c. 全部指向路径删剪

(i)A 每到达一个新节点(设为图中的 N_{n+m})后，检查"途经节点数"变量的值，如果它等于某一数值 hop2，则停止移动，并向 Home 发送一封请求进行"全部指向路径删剪"操作

的信件(图中(9)),信件中包含有A的当前位置N_{n+m}。

(ii)Home收到请求后,将A的信件接收状态置为"拒绝接收",并检查是否有通信进行标志量被设置,如果没有,则向基代理FP1发送"准许迁移"通知(图中(10)),通知中附有删剪请求信件中所附的地址N_{n+m};如果还有标志量被设置,Home向这些标志量对应的agent发送"中断通信"通知(图中(11)),这些agent在收到通知后,停止向A发送消息,向FP1发送一封"通信完成"信件(图中(12)),并清除缓存的A的投递地址。FP1收到该信后将信件转发给Home(图中(13)),Home收到后则清除其对应的通信进行标志量。Home在检查发现A的所有标志量均被清除后,向FP1发送准许迁移通知。

(iii)FP1所在节点N_1在收到由Home发来的准许迁移通知后,将这一通知沿指向路径进行转发,一直到信件到达A当前所处节点N_{n+m}为止,在转发过程中,从FP1所在节点N_1起,一直到N_{n+m}前一节点上的转发代理都将被删除。A在收到准许迁移通知后,向Home发回确认信件(图中(14)),并置自身的途经节点数变量为0,开始新的迁移。

(iv)Home收到A发来的确认信件后,将A的投递地址改为N_{n+m},将信件接收状态置为"允许接收",释放所有阻塞信件,同时向这些寻址信件的发送者返回新的投递地址,并对通信进行标志量进行设置。

需要注意的是,上面的讨论假定了A在执行任务期间不会对同一节点进行重复访问,即A的移动路径不会形成环路。如果存在A对同一节点的重复访问,则要对上面的S_COMP算法加以改动,具体改变如下:(1)不再进行b中的部分指向路径删剪操作;(2)c.(i)中当A到达一个新节点后,若"途经节点数"变量值等于hop2或者该节点已经存在A的转发代理(出现环路),则A停止移动,并向Home发送一封请求进行"全部指向路径删剪"操作的信件,信件中包含有A的当前位置。(3)c.(iii)中,当准许迁移通知到达A当前所处节点时,检查该节点是否有A的转发代理,若有,则将准许迁移通知沿该代理指向继续转发出去,并在转发前删除该代理。同样,在准许迁移通知继续转发过程中各转发节点上的代理将被删除;若A所处节点上没有转发代理,则A向Home发回确认信件,并置自身的途经节点数变量为0,开始新的迁移。除了这三点外,算法的其余部分不变。

3 算法正确性证明

下面以B向A发送远程信件为例,给出不出现对同一节点的重复访问时S_COMP算法的正确性证明(即证明定理1),原型算法以及出现重复访问时的S_COMP算法的证明可以类似得出。

定理1 在2.1.1所述的假定条件下,发往一个agent的消息总能经过有限次的转发被目标agent可靠接收。

证明定理1之前,先证明下面的三个辅助定理:

辅助定理1 若B缓存有A的投递地址,则该地址一定真实反映了A的基代理FP1当前所处的位置。

证明:由算法的c.(ii)、c.(iii)和c.(iv)知,在FP1所处位置改变期间(在全部指向路径删剪中从FP1收到Home发来的准许迁移通知开始至Home收到A对该通知的确认信件为止),Home处的信件接收状态始终为"拒绝接收",也即在信件接收状态为"允许接收"时,FP1的位置不会改变。又由信件的FIFO特性知道,全部指向路径删剪请求总是按序处理的,且A每次都会停止等待直到自FP1起的所有转发代理都删除完毕后才会向Home发回对准许迁移通知的确认,因此,Home每次收到A对该通知的确认后对投递地址的更新一定是FP1的最新位置,此时将置信件接收状态为"允许接收",由前面的证明可知,在信号量重新被置为"拒绝接收"之前,FP1的位置不会发生变化。由此可知,若信件接收状态为"允许接收",则Home所记录的A的投递地址一定真实反映了FP1的当前位置。由于只有在Home处信件接收状态为"允许接收"时B才能取得并缓存A的投递地址,且由c.(ii)可知,在Home处的投递地址改变之前B所缓存的A的投递地址一定会被清除,因此,若B缓存有A的投递地址,它也一定真实反映了FP1的当前位置。故,辅助定理1得证。

辅助定理2 若尚有其它agent发往FP1的信件未被FP1接收到(信件处于传送途中),FP1在收到所有这些信件之前地址不会发生改变。

证明:由信件的FIFO特性和c.(ii)中的发信过程可知,B发往FP1的任一信件必然会在B的通信完成信件抵达FP1之前被FP1接收到;又由c.(ii)可知,在收到所有的通信完成信件之前FP1不会收到准许迁移通知,从而也不会发生FP1的删除和位置改变。因此,FP1不会在所有尚在途中的信件到达之前发生地址改变。故,辅助定理2得证。

辅助定理3 设M是B发往A的远程信件,它已被A的基代理FP1接收,则M在被FP1转发出后,经过有限次的转发一定会被A接收到。

证明:首先,由指向路径的生成过程可知,由FP1起始的指向路径必定最终能够指向A(在部分指向路径删剪完成之前,可能会同时存在两条从FP1指向新的FP2的路径),且由指向路径删剪操作可知,指向路径由有限个单次转发路径组成;另外,由c.(i)和c.(iii)可知,在每进行一次全部指向路径删剪时,A都会在某一个节点上等待,直到准许迁移通知沿指向路径经有限次转发到达;再由c.(ii)可知,准许迁移通知必定是在其它agent发给A的所有信件由FP1转出之后才会发出,即M会先于准许迁移通知发出,由信件的FIFO特性可知M必会先于准许迁移通知经有限次的转发被A接收到。故,辅助定理3得证。

下面对定理1进行证明:

证明:B要向A发送信件,会有以下两种可能:

Case1:B缓存有A的投递地址:由辅助定理1可知该缓存一定真实反映了FP1的当前位置,再由辅助定理2,在B向A发出信件后,FP1在收到信件前其位置不会发生改变,又由辅助定理3可知该信件一定会经过有限次的转发最终到达A。故,定理1得证。

Case2:B没有缓存A的投递地址:由a.(i),B会先向A的Home发去寻址信件,若此时信件接收状态为"允许接收",则Home会向B返回投递地址,B收到后会先对该投递地址进行缓存,之后的情况与Case1中的相同,已被证明;若信件接收状态为"拒绝接收",寻址信件会被阻塞,直到A发回对准许迁移通知的确认,此时,新的投递地址登记成功,释放寻址信件并向B返回投递地址,此后的情形已被证明。

综上,定理1得证。

4 性能分析

我们通过模拟实验从通信代价和对自主移动的限制两方面来分析和比较两种算法的性能。通信代价由通信总成本来

刻画，对 agent A 而言，它定义为在一次任务执行期间所有与 A 相关的消息传送开销。对移动的限制用移动总延时来刻画，它定义为 A 在执行某一任务时，由于算法对其移动的限制而导致的延时等待，在原型算法中，A 每次移动前都要等待，等待时间为从 A 发出迁移请求到它收到准许迁移通知所经过的时间；S_COMP 算法中，延时发生在执行全部指向路径删剪时，每次的等待时间为从 A 向 Home 发出删剪请求到它收到准许迁移通知所经过的时间。

为了方便陈述，在下面将 S_COMP 算法中的全部指向路径删剪和部分指向路径删剪分别简称为全删和部分删，并将 A 发出全删请求和部分删请求时它所处的节点称为全删点和部分删点。在模拟实验中，设单条消息在任意两个节点之间的传输时间和通信开销均为单位量1。需要注意的是，对于某一特定任务，通信开销会因 agent 的移动速率不同而不同，我们把 agent 根据移动特性分为弱移动 agent、中度移动 agent 和强移动 agent 三类。弱移动 agent 移动缓慢，在所经节点上停留时间较长，对已发出的信件而言，可以认为目标 agent 在信件被接收前是静止的，即信件在其发出时 agent 所处的位置被接收；强移动 agent 移动迅速，可以认为已发出的信件在目标 agent 于全删点停止等待之前始终处于对 agent 的追击状态而无法被接收，即信件全部在全删点被接收；中度移动 agent 的移动速率介于两者之间，可以看作是信件在以比 agent 更快的移动速率在追击目标 agent，信件的具体接收位置取决于两者各自的移动速率以及指向路径的长度。

我们设 A 具有上述三种移动特性之一且迁移速率均匀。另设在执行任务期间只有一个 agent B 向 A 发送远程信件，且信件的到达是均匀的，即在原型算法中，信件平均地到达各个节点（若信件总数小于迁移次数，则收信节点之间间隔均匀），在 S_COMP 算法中，信件平均地到达 A 迁移途中产生的各个 FP1所在节点，且同一 FP1节点上收到的多封信件之间的时间间隔是均匀的。令任务执行期间 A 的迁移次数为 n＝60，当 B 向 A 发送的信件总数 m 以及 hop2和 hop1取不同值时，得到的模拟结果如表1所示。

由表1可以看出，与原型算法相比，S_COMP 算法在保证 agent 的自主移动方面有较大的优势，且在通信量较少或 agent 移动缓慢的情况下通信总成本也明显低于原型算法；此外，对于 hop2＝1的特殊情形，S_COMP 算法与原型算法类似，其移动限制和通信代价与 agent 的移动特性无关，但在性能上比原型算法差一些。还可以看到，hop2的增加能够有效地减少移动延时；且当远程通信发生较多时，hop1的减少能够明显降低通信成本。需要注意的是，当 agent 为强移动 agent 且远程通信量较大时，通信成本会急剧上升，且当 hop2增大时通信成本的增加更为显著。由此可以看出，作为一种强适应性算法，S_COMP 算法可以通过调节 hop2和 hop1的值来满足具有不同移动特性的 agent 在移动延时和通信开销方面的要求。此外，由于 S_COMP 算法在 agent 的远程通信量较少时具有很好的性能，且在实际应用中，大多数情况下不同节点间的 agent 只发生很少量的通信，因此 S_COMP 算法非常适合在实际的 agent 系统中采用。与文[2 ～ 7]中的其它同类算法相比，除了具有强适应性这一突出优点外，它在自主移动和通信效率保证方面也不逊色，而且算法易于实现。我们在实现时，是以单位时间内 agent 移动所经过的节点数与其所接收的远程信件数之比作为度量，由程序自动按照给定的一组值对 hop2和 hop1进行动态设置。

表1　两种算法的性能比较

类别 / 比较项目		原型算法	S_COMP 算法												
			hop2＝1	hop2＝30 hop1＝10			hop2＝30 hop1＝5			hop2＝60 hop1＝10			hop2＝60 hop1＝5		
				弱	中	强	弱	中	强	弱	中	强	弱	中	强
总成本	m＝60	537	660	474	570	1344	336	380	1206	468	576	2248	330	381	2100
	m＝6	216	282	96	96	210	108	108	222	90	90	295	102	102	307
	m＝0	180	240	74			86			73			85		
总延时	m＝60	297	360	30			20			15			10		
	m＝6	138	198												
	m＝0	120	180	24			14			12			7		

总结　本文针对移动 agent 环境下由于 agent 的自主移动造成的通信不可靠问题，给出了一种具有强适应性的 agent 通信算法，它能够通过调节参数的值来适应不同类型 agent 在移动延时和通信开销方面的要求，在确保消息传输可靠性的同时兼顾了通信效率。

参考文献

1　David C, Colin H, Aaron K. Mobile Agents: Are They a Good Idea? IBM Research Division: [Technical Report: RC 19887]. 1995

2　Murphy A, Picco. Reliable Communication for Highly Mobile Agents. In: Proc. of the Agent Systems and Architectures/Mobile Agents (ASA/MA)'99, 1999. 141～150

3　Tao Xianping, Lü Jian, et al. Communication Mechanism in Mogent System. Journal of Software, 2000, 11(8): 1060～1065

4　Feng X, Cao J, et al. An Efficient Mailbox-Based Algorithm for Message Delivery in Mobile Agent Systems. In: Proc. of the 5th IEEE Intl. Conf. on Mobile Agents (MA2001), 2001

5　Van B, et al. Location transparent routing in mobile agent systems—merging name lookups with routing. In: Proc. of the Seventh IEEE Workshop on Future Trends of Distributed Computing Systems, 1999

6　Ranganathan M, Bednarek M, et al. A Reliable Message Delivery Protocol for Mobile Agents. In: Proc. of Agent Systems and Architectures/Mobile Agents (ASA/MA)'2000, 2000. 141～150

7　Stanford University. JATLite . http://cdr.stanford.edu/ProcessLink/papers/JATL.html