

# 基于移动代理的主动网络\*

曾小平 孙永强 谢建华

(上海交通大学计算机科学与工程系 上海 200030)

**摘要** 在介绍主动网络的基础上,提出了一个在移动代理系统基础上建立的安全而高效的主动网络系统框架 MANet(mobile agents based active network),通过将程序代码和数据封装成主动数据包在主动结点上作为代理来运行,这个框架实现了集成化的主动网络,并且通过基于 CodeBroker 的代码装载和缓存技术改善了整个主动网络的性能和灵活性.

**关键词** 主动网络,移动代理,主动结点,主动数据包,代码装载.

**中图法分类号** TP393

在传统的计算机网络中,网络被动地将数据从一个系统传送到另一个系统.网络中的网络结点(如网关、网桥或路由器)的计算能力是非常有限的,通常它只能对数据包进行简单的处理,如包交网络中的报头处理、面向连接网络中的信号处理等.这种被动的网络结构存在许多问题:首先是很难将新技术和新标准集成到现有的共享网络基础设施中;其次,多层协议的协议层之间的冗余操作降低了网络系统的性能;并且在这种网络中增加新服务也是非常困难的<sup>[1]</sup>.因此,从 1994 年以来,在 DARPA(defense advanced research projects agency)的支持下,人们开始研究更加开放、性能更高的网络结构.主动网络(active network)就是以解决上述计算机网络所遇到的问题为目的的开放式网络结构.

本文首先介绍了主动网络及其结构和应用以及研究现状,在此基础上提出了一种利用移动代理(mobile agent)技术来建立主动网络的方法,进而提出利用 CodeBroker 来实现主动网络中程序代码的动态按需装载,实现高性能的主动网络.

## 1 主动网络(active network)

在主动网络中,网络结点对流经本地的消息能进行客户化的计算.例如,一个用户可以向某些路由器发送“trace”程序,以便在数据包经过这些结点时执行该“trace”程序,处理该数据包,以实现路由跟踪.也就是说,主动网络中的网络结点具有计算能力,它能对流经本地的数据进行计算和修改.这种计算可以是基于每个用户、每个应用或者是其他需求的.通过这种可编程的网络结构,新标准、新技术的采用变得迅速而简单,从而加快了计算机网络发展的步伐<sup>[1,2]</sup>.

### 1.1 主动网络

主动网络中的网络结点具有计算能力,它对流经本结点的数据包进行客户化的计算.如图 1 所示,具有计算能力的网络结点从网络设备接收数据包后执行相应的程序,对该数据包进行处理(如路由选择、数据合并、数据解包等).然后将数据包(可能已被修改)发送给其他网络结点.网络结点的计算能力使得主动网络结点本身具有可编程性,从而使得整个网络结构也是可编程的.

实现这种可编程的主动网络主要有以下两种方法<sup>[1,3]</sup>.

\* 本文研究得到国家自然科学基金(No. 69683001)资助.作者曾小平,1975 年生,博士,主要研究领域为并行/分布计算.孙永强,1931 年生,教授,博士生导师,主要研究领域为计算机语言,并行与分布处理.谢建华,1974 年生,硕士生,主要研究领域为移动代理,分布式计算.

本文通讯联系人:曾小平,上海 200030,上海交通大学计算机科学与工程系

本文 1998-06-30 收到原稿,1998-12-28 收到修改稿

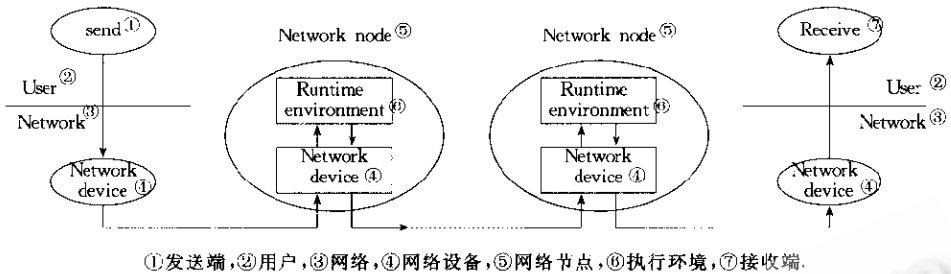


Fig. 1 Data transfer on active network  
图1 主动网络的数据传送

一种是对数据的处理同处理程序注入网络节点的过程分离的离散方法<sup>[1,3]</sup>. 用户需预先将数据包处理程序传送给数据包要经过的每个网络节点. 之后, 当相应的数据包经过该网络节点时, 网络节点通过检查该数据包的报头来决定调用相应的处理程序对它进行处理. 这种处理方式通过由报头决定报文内容的处理程序提供了一定的客户化计算能力. 对于需要严格控制所装载程序以及客户化程序比较大的场合, 这种离散的方法是非常有效的.

另一种更加灵活也更为“野心勃勃”的方法是集成化方法<sup>[1]</sup>, 即每个数据包都是由程序代码和数据组成的程序. 当数据包到达某个网络节点后, 数据包所携带的程序代码被执行, 以处理数据包所携带的数据. 这就像 PostScript 打印机对传送给它的 PostScript 文件的解释执行一样<sup>[3]</sup>. 这种方法具有更大的灵活性和可编程性, 但也更为复杂. 可移动代码和可编程节点等技术的发展为这种集成化的方法提供了坚实的基础.

### 1.2 主动网络的优点及其应用

Internet 的发展对计算机网络内部的计算提出了新的需求. 提供满足这些需求的服务要求网络结构能对自身进行调整, 但是传统的计算机网络结构不能很好地满足这类需求. 在传统的计算机网络中, 实现新技术从实验室原型到大规模应用的转变过程需要大约 10 年时间, 这个过程包括标准化、进入硬件生产商的硬件产品、相关程序的开发和用户的安装调试等. 在主动网络中, 网络节点对每个数据包的处理都是针对用户、应用或者其他需求的. 网络内部的计算能力得到了充分的体现, 多种网络协议可以共存. 通过增加新的协议处理程序, 网络协议的兼容性和更新能力都得到了很好的支持, 从而也提高了网络互操作性的抽象层次. 通过提供实现新服务的程序, 主动网络就能为用户提供这种新的服务. 这样, 网络服务的更新就取决于市场的接受能力, 而不再是网络设备生产厂家的产品更新能力. 它使得网络结构从传统的软件和硬件捆绑在一起的“主机”模式转变为软件和硬件分离的“虚拟”模式, 从而形成了功能强大的网络平台.

目前, Internet 对计算机网络内部的计算能力提出需求的应用领域主要有消息的合并与分布、用户化的网络保护、主动的网络管理、消息传送的自付费和动态的网络带宽划分等<sup>[1,3]</sup>. 主动网络节点的计算能力为这些领域的应用提供了非常有效的支持.

## 2 基于移动代理的主动网络系统 MANet

MANet(mobile agents based active network)是一个以移动代理系统为基础, 通过将主动数据包封装成移动代理来实现利用移动代理系统处理主动网络数据包的主动网络. 利用移动代理来实现主动数据包的封装和运行, 不仅可以充分利用移动代理的自主性和可移动性, 而且提供了更好的主动网络可编程性和灵活性. 将主动网络数据包所携带的程序和数据封装成移动代理, 使得数据包具有很好的主动性和处理能力. 例如, 主动数据包可以利用代理的克隆及子代理的生成和分发来实现多路分发等. 另一方面, 移动代理系统也提供了一些相对于主动网络来说多余的功能, 例如, 移动代理能在程序运行的任意点上挂起, 并在迁移到另一台主机后接着这一点继续往下执行. 为了提高整个主动网络系统的性能, 要针对主动网络的特性对移动代理系统进行修改, 以去除不必要的功能, 从而提高系统的性能. 事实上, 包括我们的系统在内的许多基于 Java 的移动代理系统, 如 Aglets, Odyssey, D'agent 等, 由于 Java 语言本身所存在的程序运行状态的获取问题, 都未能实现断点式的代理迁移. 此外, 支持主动数据包之间的通信对于主动网络来说, 开销也太大.

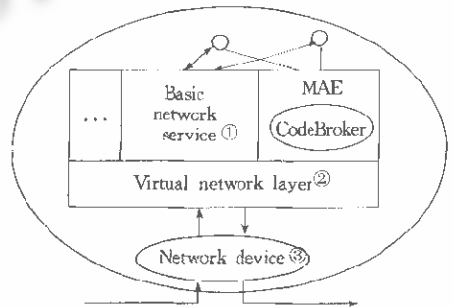
## 2.1 移动代理(mobile agent)

移动代理是能在异构网络主机之间自主地进行迁移的有名字的程序。程序能自主地决定在什么时候迁移到什么地方。它能在程序运行的任意点上挂起,然后迁移到另一台主机上,并接着这一点继续往下执行。移动代理也能克隆自己或者产生子代理,迁移到其他主机上以共同协作完成复杂的任务<sup>[4,5]</sup>。移动代理系统是指能创建、解释、执行、传送和终止代理的平台,它由名称和地址唯一标志。每个系统都可以运行多个代理。代理通过和主机服务进行交互来获得所需的服务。

目前我们正在实现的是一个以 Java 作为代理编程语言,并通过 CodeBroker 来动态地、按需地装载程序模块的移动代理系统<sup>[6]</sup>。该系统不仅能提供移动代理的基本运行环境,而且通过称为“CodeBroker”的类装载器实现了程序代码块的动态按需装载。在这种代码迁移方式下,代理在迁移时只需携带最小的代码集合。只有在执行尚未装载的代码时,移动代理执行环境才通过 CodeBroker 搜索并装载相应的程序代码块。这种代码迁移方式不仅提高了整个移动代理系统的性能,而且提供了更好的资源共享能力和容错能力。

## 2.2 MANet 主动结点的结构

在 MANet 中,每个内部网络结点都是一个具有计算能力的主动结点。每个主动结点不仅具有有相当功能的处理器,而且也具备由多种存储设备组成的多级存储系统。实际上它是一个由网络硬件设备、处理机以及相应的软件所组成的系统,如图 2 所示。虚拟网络层完成传统网络中数据链路层的功能,它通过网络设备接收数据包,对数据包的格式进行识别,并依据数据包格式对其进行分解,然后将数据包所携带的代码和数据等传送给移动代理运行环境 MAE,MAE 将程序代码作为代理装载运行。代理实现的是客户化的协议程序,它在运行过程中可能还要和基础网络服务,部分进行交互,调用最基本的网络服务,如获取本地以及相邻结点的信息、下一转发结点的地址以及本地结点状态的存取等。经过代理处理后的数据交由虚拟网络层重新封装,然后通过网络设备发送给下一个主动网络结点(或几个网络结点)。



①基础网络服务,②虚拟网络层,③网络设备。

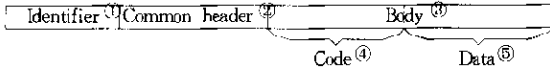
Fig. 2 Active node architecture  
图2 主动结点的结构

MAE 是一个基本的移动代理执行环境,它从虚拟网络层接收数据包并对其进行分析和分解,将其中的程序代码装载执行。同时,它还要监视代理的运行状态,在要执行的程序代码尚未装载时,通过 CodeBroker 搜索并装载相应的代码块。MAE 也要为主动网络系统的安全性提供诸如身份验证、授权以及代码的安全运行等有效的支持。基础网络服务主要是为主动数据包提供最基本的网络协议功能,如本地环境的存取、路由控制以及转发等。它有效地减少了数据包所携带代码的长度,提高了主动网络系统的性能。甚至较为上层的常用网络协议,如 TCP、UDP 和 IP 等,也可以作为基础网络服务的一部分以实现与传统网络的兼容性和提高主动网络的性能。

MANet 的主动数据包编程语言采用的是 Java 语言<sup>[7]</sup>,这有以下几方面原因。一方面,MAE 的基础是我们已经实现的移动代理系统。另一方面,Java 不仅通过虚拟机 JVM 实现了与系统平台的无关性,而且 Java 是一个通用型的程序设计语言,具有一般通用程序设计语言所具有的编程能力,为主动结点的编程提供了有力的支持。更重要的一个原因是,Java 所具有的安全性是基于网络编程的基础,它为主动网络的安全性提供了有效的保证。另外,通过 JIT 可以将 Java 程序编译为可执行代码,从而提供了比 Tcl、ML 等高层语言更好的性能。而且 Java 语言提供了丰富的类库,用户不仅可以方便地利用这些类库进行编程,而且这些专门类库有着很好的效率。在代码装载时,丰富的类库也使得大部分的程序代码只需从本地环境装载,有效地改善了 MANet 的性能。但是,Java 也包含了一些在建立主动网络结点时所不需要的语言成分,如 AWT、网络连接类等。对主动网络系统来说,这些成分不仅增加了主动网络编程的复杂性,而且影响了主动网络系统的性能。因此需要针对主动网络的特点开发针对主动网络的 Java 版本,如针对 WWW Browser 的 Java Applet,仅保留主动网络编程所必需的基本语言成分,使得数据包处理程序“轻量化”。

### 2.3 MANet 主动数据包的格式

主动数据包是在 MANet 中的虚拟网络层之间传送的数据实体,它封装了所要传输的数据以及数据的处理程序.图 3 所示的是 MANet 主动数据包的格式.标志符域标志了数据包的网络协议类型,它实现了主动网络和传统网络协议(TCP/IP,UDP 等)之间的兼容性.例如,当该标志符被设置为 TCP 时,该数据包包含的报头以及报文是遵循 TCP 协议的数据报格式.当标志符被置为主动数据包时,公共报头包含了源地址、目标地址、TTL 以及电子货币、数字签名等信息.网络结点利用数字签名来验证,并给予网络用户相应的授权以保证整个主动网络的安全性.利用电子货币则可以实现实时的网络自付费.报文部分则包含了程序代码以及传输数据等内容.



①标志符,②公共报头,③报文,④代码域,⑤数据域.

Fig. 3 Active package format  
图3 主动数据包格式

当一个主动数据包到达主动网络结点后,结点通过标志符检查数据包格式.若数据包是传统网络协议(如 TCP/IP),则向基础网络服务提出请求,调用相应的协议处理程序来对数据包进行处理;若数据包是主动数据包,则从报文的代码域抽取相应的程序代码,并将其装载执行,然后将经过处理的数据

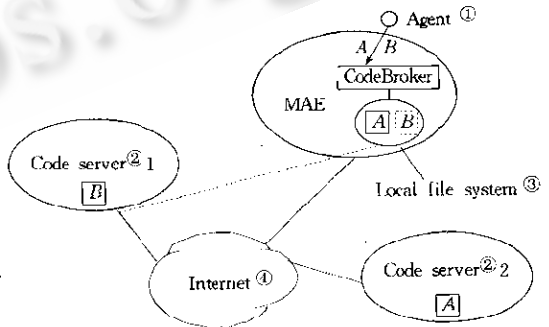
(可能已经改变)重新封装成主动数据包后发送给后面的主动结点.这样,用户通过主动数据包可以将网络协议程序注入到主动结点中,从而实现对主动网络的编程.

### 2.4 通过 CodeBroker 来实现代码的动态按需装载

从主动网络系统的性能出发,要求主动数据包中所包含的代码长度远小于数据的长度,否则,系统的性能损失将直接影响到主动网络的可行性.因此,除了精简数据包编程语言使得协议程序变得“轻量”之外,MANet 还将最基本的协议处理程序,如网络设备启动、路由选择、邻近主动网络结点信息的获取以及传统协议处理程序,如 TCP,UDP,IP 等作为主动结点的一部分,通过基础网络服务所提供的这些服务来减少数据包中处理程序的长度,减轻网络负载,提高系统性能.

但是由于主动网络是可编程的网络,它的许多服务程序是动态变化的.这些动态变化的协议处理程序不可能作为基础服务的一部分,但这些协议处理程序又是可以被用户共享的.这些协议处理程序可能很简单(如多路分发),也可能很复杂(如数据合并).如果能实现程序代码的动态按需装载,那么主动网络系统不仅可以共享那些由网络设备供应商、第三方软件开发商甚至由用户所开发的协议服务程序,而且可以进一步减轻数据包负载(即数据包中程序代码的长度),提高系统的性能以及容错能力.

MANet 充分利用原有移动代理系统中基于 CodeBroker 的代码迁移机制动态地、按需地装载代码块<sup>[4]</sup>,实现了服务程序的共享和系统性能的改善.当在移动代理执行环境中运行的代理执行到尚未装载的程序代码(或类)时,执行环境向 CodeBroker 提出该代码的装载请求,如图 4 所示. CodeBroker 接收到装载请求后,首先搜索该代码所在的位置,以确定从本地文件系统还是从远程的代码服务器装载代码.若代码存在于本地文件系统中,CodeBroker 将直接从本地装载该代码,随后将控制转移给移动代理执行环境.若代码不在本地文件系统中,则还要依据某种优化算法从多个包含该代码的代码服务器中选择一个代码服务器装载所需的代码.在图 4 中,代理在运行过程先是发现要执行的代码 A 尚未装载.它通过 MAE 向 CodeBroker 提出代码装载请求, CodeBroker 接收该请求后搜索该代码的位置信息,发现本地文件系统中代码 A 的一个副本.然后,CodeBroker 直接从本地文件系统装载代码 A,并交由 MAE 执行.一段时间后,代理又发现要执行的代码 B 尚未装载,向 CodeBroker 提出装载请求.这次 CodeBroker 查询



①代理,②代码服务器,③本地系统,④因特网.

Fig. 4 CodeBroker-based code loading  
图4 基于CodeBroker的代码装载

一段时间后,代理又发现要执行的代码 B 尚未装载,向 CodeBroker 提出装载请求.这次 CodeBroker 查询

到代码  $B$ , 位于代码服务器 1, 它将从代码服务器 1 装载代码  $B$ , 并将其缓存在本地文件系统中(如图中虚线所示)。

在计算机网络中, 数据包的传送存在着连续性, 即同一协议的数据包连续地到达某个网络结点, 或者说连续到达的数据包有着相同的处理程序。利用网络传输的这一特性, MANet 引入了代码的缓存技术。通过将已装载的代码缓存在本地文件系统中可以减少代码的传输, 缓解网络传输的拥挤现象, 改善整个主动网络系统的性能。由于装载的代码是动态变化的, 因此这些代码的缓存是由 CodeBroker 而不是由基础网络服务来管理的。CodeBroker 在装载代理所请求的代码后, 将该代码缓存在本地文件系统中, 并对该代码进行登记注册。CodeBroker 在搜索代码时, 总是先检查本地文件系统的代码注册信息, 在注册表中没有装载代码的信息时才向其他代码服务器查询该代码的信息。在一个时间或遇到存储空间不足时, CodeBroker 将采用具有先进先出原则的 FIFO 算法淘汰过时的代码, 为新的扫描腾出存储空间。

通过动态的按需装载和缓存技术, MANet 不仅减少了网络带宽的使用, 提高了系统的性能和容错能力, 而且实现了协议服务程序的共享。代码服务器的存在使得网络设备供应商、第三方网络协议软件开发商甚至终端用户都能独立地开发满足各种应用需求的客户化网络协议程序。只需将服务程序放在某个代码服务器中, 并通过类似于公告牌等的分布机制就可以实现这些程序的共享。这种迁移机制的灵活性加快了网络服务程序的更新速度, 有效地发挥了主动网络的潜力。

## 2.5 安全性

可移动代码在主动结点上的运行不可避免地引入了传统安全领域中所没有的安全性问题<sup>[1,3~5]</sup>。代理要在主动结点上执行就必须占用系统中的 CPU、内存等资源, 并对它们进行存取, 因此如何保护主动结点不受恶意代理的侵犯以及保护代理不受恶意主动网络结点的窃听或拒绝等都是主动网络系统所必须解决的问题。MANet 通过集成现有的安全技术, 从底层运行环境、编程语言以及应用等各个层次提供了整个主动网络系统的安全性。

迄今为止, 移动代理安全领域所做的研究作为主动结点运行环境的安全性提供了基础<sup>[4,6]</sup>。这些研究工作包括通过限制代理的运行空间和所能存取的资源保护运行环境不受恶意代理的侵犯以及通过跟踪检查保护代理等。另外, MANet 还通过公共报头域中的数字签名有效地实现了代码的身份验证并给予相应的授权。在语言层次上, Java 语言在设计之初就考虑了安全性问题。通过类装载器、验证器、安全管理器和严格定义的虚拟机字节码等机制, Java 语言提供了很好的语言级安全性。更为灵活的是, 主动数据包可以携带数据包自身的加密解密程序和安全协议程序。通过客户化的加密解密程序和安全协议程序, MANet 提供了灵活的可扩展安全机制, 为用户级的安全控制提供了基础。通过各个层次之间的密切合作, MANet 对数据窃取、服务否认和恶意攻击等都具有较强的防范能力。

## 3 和其他工作的比较

MIT 的 ANTS(active network transfer system)<sup>[8]</sup>是一个建立在链路层上、采用 Java 作为主动数据包(ANTS 中称为“capsule”)编程语言的主动网络。它采用集成化的方法, 每个“capsule”都是由可移动代码和数据组成的程序。ANTS 通过使用可移动代码技术能够在主动网络的中间结点和终端系统上自动配置新协议。它将网络看作分布式的可编程系统, 并提供了在结点上通过操作实现新协议的类似于编程语言的模型<sup>[3]</sup>。与 ANTS 相比, MANet 系统则是建立在移动代理系统上的主动网络系统。它比 ANTS 更为集成, 而且也具有更好的可编程性和灵活性。移动代理、按需装载和缓存等技术的使用也为 MANet 系统性能的进一步提高提供了可能。

宾夕法尼亚大学的 SwitchWare<sup>[9]</sup>是一个完全不依赖于其他 Internet 协议的主动网络系统。整个系统由主动数据包、Switchlet 和 SANE 三层组成<sup>[9]</sup>。主动数据包是用 PLAN 语言<sup>[2]</sup>编写的由代码和数据组成的可移动程序。PLAN 语言非常简单, 其目的是为了简化程序, 减少网络开销。更为复杂的功能是通过调用称为 Switchlet 的服务程序实现的。Switchlet 是用通用语言编写的服务程序, 这些程序是动态可装载的, 因此 Switchlet 具有很好的灵活性和可扩展能力<sup>[10]</sup>。安全主动网络环境层 SANE 的作用则是为上述两层提供安全的执行环境。Switch-

Ware 采用的是在集成和离散之间折衷的方法,它的系统实现和网络编程都比 MANet 更为复杂。

哥伦比亚大学开发的 NetScript<sup>[11]</sup>是一个用于开发主动网络服务协议的程序设计语言及其运行环境。NetScript 语言是一个专用于通信任务描述的小而简单的动态数据流语言<sup>[11]</sup>。它特别适合于编写完成路由选择、包分析、信号处理和管理等功能的程序。NetScript 本身并不是一个完整的程序设计语言,而只是一个控制描述语言。通过 NetScript 的控制结构可以将称为“box”的程序单元组合成数据包流处理引擎,并为其分配相关的资源。NetScript 对数据包的处理采用的是比前两种方法更为保守的离散方法——它根据数据包的报头来决定数据包的处理程序。

此外,Bell 和宾州大学合作的 Protocol Boosters、英国 BBN 的 Smart Packets、佐治亚理工学院的 CANES 和亚利桑那大学的 Liquid Software 等系统都在主动网络的体系结构、编程语言以及工作机制等方面开展了研究工作,并取得了许多研究成果,为主动网络的实用化开辟了道路。有关主动网络研究的综述见文献[1]。

#### 4 结束语

与现有的主动网络系统相比,MANet 以移动代理作为基础,将代码和数据封装成数据包,每个数据包都可以作为代理运行,使得主动网络具有很好的可编程性,它具有其他主动网络系统所没有的灵活性和编程能力,基于 CodeBroker 的代码装载以及缓存技术有效地改善了主动网络的性能和代码共享能力。但是,MANet 在与现有网络设备的兼容性、主动网络系统的安全性等方面还存在许多不足之处,在应用开发和系统性能测试工具方面的工作也尚未开始。

#### 参考文献

- 1 Tennenhouse D, Smith J, Sincoskie W *et al.* A survey of active network research. *IEEE Communications Magazine*, 1997, 35(1):80~86
- 2 Hicks M, Kakkar P, Moore J *et al.* PLAN: A programming language for active networks. In: Felleisen M ed. *Proceedings of the International Conference on Functional Programming'98*. New York: ACM Press, 1998
- 3 Tennenhouse D L, Wetherall D J. Towards an active network architecture. *Computer Communication Review*, 1996, 26(2):56~67
- 4 Gray R S. Agent Tcl: a transportable agent system. In: Finn T ed. *Proceedings of the CIKM Workshop on Intelligent Information Agents (CIKM'95)*. Baltimore: ACM Press, 1995
- 5 White J E. Telescript technology: mobile agents. In: Bradshaw J ed. *Software Agents*. Boston: AAAI Press/MIT Press, 1996
- 6 Zeng Xiao-ping, Sun Yong-qiang, Shi Rong *et al.* The code migration of mobile agents system. In: Chen Jian, Li Ming-shu, Minggins C eds. *Proceedings of TOOLS 27*. Meyer, CA: Los Alamitos, IEEE CS Press, 1998
- 7 Gosling J, Joy B, Steele G. *The Java language specification*. Reading, MA: Addison-Wesley Publishing Company, 1996
- 8 Wetherall D, Guttag J, Tennenhouse D. ANTS: a toolkit for building and dynamically deploying network protocols. In: *IEEE OPENARCH'98*, San Francisco, CA, 1998. [www.sds.lcs.edu/activeware/](http://www.sds.lcs.edu/activeware/)
- 9 Alexander D, Arbaugh W, Hicks M *et al.* The SwitchWare active network architecture. *IEEE Network Special Issue on Active and Controllable Networks*, 1997, 12(3):29~36
- 10 Alexander D, Shaw M, Nettles S *et al.* Active bridging. *Computer Communication Review*, 1997, 27(4):101~111
- 11 Yemini Y, Silva S da. Towards programmable networks. In: *IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*, 1996. <http://www.cs.columbia.edu/dasilva/>

### Active Network on Mobile Agents

ZENG Xiao-ping SUN Yong-qiang XIE Jian-hua

(Department of Computer Science and Engineering Shanghai Jiaotong University Shanghai 200030)

**Abstract** Based on the introduction of active network, the MANet (mobile agents based active network) is presented as a secure and efficient active network system constructed on the mobile agents system. It implements integrated active networks by capsule the program code and data into an active package which can run on active node as an agent. Also the performance and flexibility of active network are improved by the on-demand code loading and caching based on the CodeBroker.

**Key words** Active network, mobile agent, active node, active package, code loading.