

# 多 Agent 系统的模型和形式语义<sup>\*</sup>

The Modeling and Formal Semantics of Multi-Agent Systems

张伟 徐晋晖 石纯一

(清华大学计算机系 北京100084)

**Abstract** Models and formal methods are important in the analysis and design of multi-Agent systems (MAS). In this paper, we give a detail summary and analysis about different formal methods used in the modeling, language and organization of MAS. Finally, further research topics of using formal semantics in the description and analysis of the social characteristics of MAS are proposed. The necessity as well as the possible approaches are also given.

**Keywords** Formal method, Formal semantics, Modeling, Multi-Agent system

## 1 引言

自90年代以来,关于 Agent 和多 Agent 系统逐渐引起重视并形成 AI 研究的热点,由于 Agent 表达能力强,市场求解机制以及把推理格局引伸到思维状态,因此适用于动态开放环境的问题求解<sup>[1]</sup>。Agent 和多 Agent 系统最初是作为一种分布式计算模型提出来的,旨在控制分布式计算的复杂性,克服人机界面的局限性,以及适应实际问题的开放性和分布性的要求。

随着对 Agent 理论和技术研究的逐步深入,已有一些理论结果和实际应用系统出现,从理论上讲,关于 Agent 和多 Agent 系统的形式化工作,已经可以对系统进行更准确描述和细致刻画,同时,深化了研究多 Agent 系统的特性和求解机制,使用的基本方法是模态逻辑,其形式系统的语义是基于可达关系的可能世界语义模型。模态逻辑的方法存在逻辑全知和副作用等问题,而且使用逻辑方法也难于解决理论和实践的分隔问题。

本文通过对常用的几个 Agent 系统形式化方法的分析,来讨论用形式语义学的方法对 Agent 和多 Agent 系统进行形式化描述的问题。形式语义学是描述计算机语言语义的一种形式化的方法,可作为对逻辑学方法的补充,有望向多 Agent 系统可计算性迈进一步,本文将分别讨论多 Agent 系统中常用的形式化方法及其特点,并给出使用形式语义学对多 Agent 系统进行形式化描述的研究步骤和有关问题,最后提出了用操作语义学对多 Agent 系统的社会性进行研究的

有关问题。

## 2 Agent 和多 Agent 系统的形式化

Agent 和多 Agent 系统的研究主要包括 Agent 和多 Agent 系统的模型、组织和语言。

### 2.1 Agent 模型

关于 Agent 的模型,最好的结果是思维状态模型,使用的工具是模态逻辑,其语义解释是基于 Hintikka 提出的可能世界语义模型<sup>[2]</sup>。

基于意识立场对 Agent 模型的研究主要是通过引入不同的模态算子对 Agent 的各种思维属性进行形式化的,这方面比较有影响的工作有基于正规模态逻辑的 Cohen 和 Levesque 的 Intention 理论<sup>[3]</sup>,提出了一个理性 Agent 的逻辑,列出了基本模态算子之间的关系和性质;在这个框架的基础上,引入了一些导出算子,Intention 是其中的一个,从而构成了关于理性动作的部分理论,Rao 和 Georgeff 的 BDI 模型结构<sup>[4]</sup>,给出基于三个模态词(beliefs, desires 和 intentions)的 Agent 逻辑,是建立在分枝时序逻辑模型之上的,其中 BDI 可达世界是分枝时序结构,他们使用的形式化系统是对计算树逻辑 CTL<sup>\*</sup> 的扩充。其它工作象 Konolige 和 Pollack—BDI 模型<sup>[5]</sup>, John Bell—BDI 模型<sup>[6]</sup>, Gaspar 和 Coelho—BDI 模型<sup>[7]</sup>等,都是针对已有的模型存在的问题而进行的改进,其框架大多仍然是基于模态逻辑的。

### 2.2 多 Agent 系统模型和多 Agent 系统程序设计语言

\* )本文受国家自然科学基金资助(69973023)。张伟 博士生,研究方向为分布式人工智能。石纯一 教授,博士生导师,研究方向为人工智能应用基础。

关于多 Agent 系统模型与程序设计语言,通常根据不同的需要和各自的观点选用不同的工具对系统建模并进行形式化。目前尚未有统一的模型出现,有的研究人员指出并不存在一个一致的多 Agent 系统的模型。dMARS 是一个基于 BDI 模型的多 Agent 分布式推理系统,系统使用了 Z 语言描述 Agent、思维状态的变化和规划的执行<sup>[1]</sup>,并用规划的执行和思维状态的变化作为系统的语义,但没有给出多 Agent 之间交互的形式描述和语义描述。Concurrent-METATEM 是基于可执行时序逻辑的多 Agent 系统编程语言,Agent 之间的通信是广播式的异步消息发送,一个 Agent 由一个接口和一个计算引擎组成,后者就是可执行时序逻辑的执行主体,该语言的特点是对 Agent 的定义用可执行时序逻辑公式表示。在对 METATEM 的形式化语义描述中曾经尝试过不同的方法<sup>[4]</sup>,如信念时序逻辑的方法和知识时序逻辑的方法。这两种方法都是在传统命题逻辑的基础上增加了时间模态连接词,不同的是前者是一种自顶向下的方法,可以通过公理给出系统的属性,而后者则可以给出公理系统和运行系统之间的关系,其中时序模态词的语义是基于普通模态逻辑可能世界模型给出,而知识模态算子的语义是用 Agent 内部的可能状态给出的。

CONGOLOG<sup>[10]</sup>和 Agent0<sup>[11]</sup>是两种较有影响的 Agent 设计语言。前者是在 GOLOG 的基础上扩展了并发执行功能,其理论依据是情景演算。由一系列动作作用于一个初始状态而形成的可能的历史状态称为一个情景,每个动作的执行必须满足一定的公理条件。CONGOLOG 的语义是由计算语义给出的,包括计算转换和终止条件,计算转换是指程序通过一步执行使一个情景变成另一个情景,而终止条件是指在一个情景下程序合法终止运行。Agent0 是基于 Shoham 的面向 Agent 的程序设计框架实现的,Agent 被定义成具有一定能力、一组初始的信念和承诺以及一组承诺规则。决定 Agent 如何动作的关键部分是其承诺规则集,每一条承诺规则包含消息条件,思维条件和动作。如果 Agent 收到的消息以及它当前的思维状态 (belief) 与一条承诺规则匹配,则该承诺规则被激活,Agent 变成对该规则动作的承诺。Agent0 没有给出明确的语义描述和定义。

文<sup>[12]</sup>提出用加标转换系统给出多 Agent 系统的交互和组织的形式语义,这种方法在其他文献中尚不多见,但该方法不适于描述复杂的系统,该文没有对这种方法的适用性和描述能力作出评价,从内容上讲也没有给出更进一步的组织表示,交互的机制也不完善。文<sup>[13]</sup>中尝试使用 Petri 网为多 Agent 系统建模,但提出的模型过于抽象,不易实现。Scott 等<sup>[4]</sup>基于着色

Petri 网对 Agent 之间的会话进行了建模尝试,通过对 Petri 网标记的彩色化,使标记可以包含数据和类型,而且可以对代表转换的弧赋予基本的布尔条件,作为对转换的约束。着色 Petri 网可以描述普通 Petri 网不能描述的 Agent 之间的交互问题,支持真并发操作而且直观,缺点是没有形式化的演算系统,而且当系统复杂到一定程度以后,系统描述变得十分复杂。

### 2.3 多 Agent 系统的组织

组织可以认为是多 Agent 的集合形式和多 Agent 系统的分布式问题求解机制,也是多 Agent 系统社会性的一个重要的方面,研究工作主要集中在组织的形成和演化等问题。

多 Agent 系统组织是多 Agent 系统研究的三个主要问题(模型、组织、交互语言)之一,组织的形成和演化问题是基于 Agent 的计算和 Agent 合作问题求解的关键<sup>[15]</sup>。目前关于组织的形成的方法有:(1)基于协商的合同网协议,当一个 Agent 发现自己面临的问题难以独立求解而合作求解更有效时,通过与其他 Agent 的协商,将其部分或全部任务委托其他 Agent 来完成,在它们之间形成顾客与服务员关系形式的组织;(2)基于依赖关系的社会推理,是通过 Agent 之间依赖关系的社会推理,发现与其目标有依赖关系的其他 Agent,与它们形成不同形式的合作组织;(3)基于对策论的联盟形成,通过联盟效用的计算与分配,依据 Agent 个体、联盟和群体理性原则,在 Agent 之间形成不同的联盟;(4)基于价格调控的市场机制,通过市场价格的调整,Agent 形成供求平衡下的买卖关系;(5)面向结构的组织形成<sup>[16]</sup>。

联盟形成方法是以对策论的 n 人合作为理论基础,将一群无联系的 Agent 分成多个有联系的 Agent 联盟,每一个联盟视为一个为了完成一组任务而建立的 Agent 组织。从 1993 年提出联盟方法以来,Shehory 和 Kraus 以及 Sandholm 和 Lesser 等人作了大量的工作,但是由于需要大量的计算,一直在实用方面存在困难。

多 Agent 在问题求解中表现的组织形态就是组织结构,现在对组织形态越来越采用显式的表示方式<sup>[16]</sup>,这方面工作的主要问题是缺少严格的形式化的描述和语义定义,这直接影响到基于组织的合作问题求解的可计算性。另外组织演化也仅有一些初步的工作。Ferber 等在组织的语义描述方面作了初步的工作,基于 Pi 演算和化学抽象机给出了组织形成的操作语义<sup>[17]</sup>,其工作是基于一个具体的组织模型<sup>[18]</sup>,使用一组 Pi 演算描述一个 Agent,组织的概念由化学抽象机的溶液表示,化学抽象机的运行可以动态准确地给出 Agent、角色和组的动态特性。

## 2.4 Agent 交互语言

Agent 交互语言是多 Agent 系统研究的一个重要方面,直接影响到多 Agent 系统能否推广应用,比较有影响的交互语言是 KQML 和 FIPA 的 ACL。

KQML 语言全称为知识查询与操作语言(Knowledge Query and Manipulation Language),是一种用于交换信息和知识的语言及协议,发布于1993年,是 DARPA 知识共享研究(Knowledge Sharing Initiative External Interfaces Working Group)的一部分,KQML 支持 Agent 间运行时知识共享的消息格式和消息处理协议,可作为多 Agent 系统为合作求解问题而进行知识共享的一种语言。

文[19]定义了 KQML 语言的设计和实现,通过定义 KQML 语句执行时的前件、后件和结束条件,给出 KQML 的语义描述<sup>[20]</sup>。前件是指发送和接收方在发送和接收一条 KQML 语句前应该具备的思维状态;后件是指发送和接收方接收一条 KQML 语句后应该达到的思维状态;结束条件是指完成一次会话后的结束条件,对前件、后件和结束条件的描述使用了元语言表达式,而表达式中操作符的语义是用自然语言给出的。尽管这种用类似公理语义学的方法给出 KQML 的语义比较直观易懂,但也存在一些明显的问题<sup>[21]</sup>:(1)由于 KQML 的语义是由有关的思维状态操作符给出的,而这些操作符本身的语义仍需要严格定义;(2)由于使用了思维状态操作符,因此存在句法结构和语义之间的断层问题;(3)用上述方式给出的 KQML 的语义对 Agent 接收一条 KQML 语句后应达到的状态给予了过强的限制,而事实上接受信息 Agent 的反应可能会因接收者的不同而异。

FIPA (Foundation for Intelligent Physical Agents)致力于制订有关 Agent 的标准,FIPA-ACL 是其标准中有关 Agent 通信语言的部分,FIPA 的 Agent 通信语言基于言语行为理论,定义了一组消息类型和这些消息对于发送者和接收者思维状态的影响,ACL 定义了五个通信原语并且可以组合使用,还包括了用户已经熟悉的 KQML 和几种高层交互协议,其语义是由发送者发送消息前应具备的思维状态和对于消息发送后的预期效果的描述给出的,这种类似 KQML 语义定义方法,存在和 KQML 同样的问题。

文[21]定义了一个多 Agent 并发操作的程序设计语言,Agent 之间的通信是基于集市方式的通信机制,每个 Agent 有信念和目标两个思维状态,强调了信息接收者在接收信息后思维状态的变化但不保证采取信息发送者要求的行动,该语言借鉴了面向对象语言中的类、对象和方法调用等概念,其语义使用基于转换规则的操作语义按照句法结构逐条给出,但当 Agent 的

数量超过一定的规模以后,Agent 之间的交互会变得十分复杂。

有了通信语言标准和通信协议并不等于多 Agent 系统之间就可以进行有效的通信、协调和协商。由于具体的系统是由不同机构研制的,并且可能基于不同的软件或硬件平台,因此需要进行语义验证。语义验证是指验证一个实现 Agent 通信的程序  $\pi$  是否遵守了 Agent 通信语言 Lc 的语义定义,而不管它是不是实际使用了 Lc 语言。不同 Agent 之间的通信程序,不管是由何种具体语言实现的,也不管是基于何种软件和硬件平台,只要是共同遵守了 Lc 的语义定义,就可以进行有效通信。Wooldridge 对 Agent 通信语言的语义验证问题进行了初步的研究<sup>[22,23]</sup>,文[22]给出了一个语义验证框架并分析了基于不同语义描述的 Lc 的可验证性问题和语义验证的计算复杂性,文[23]基于定量认知时序逻辑研究了 ACL 的语义验证问题。

## 3 引入形式化语义分析

任何一种形式化方法的使用都要考虑对被描述对象的适用性问题以及语义问题,用形式化方法描述系统依赖于形式化方法有严格的语义定义。对于多 Agent 系统来说,当我们关注系统实现的时候,从系统的定义到系统的设计实现,直至系统维护都需要严格的形式化描述和语义定义,而且所选择的形式系统的语义给出的方法也显得尤为重要,既要考虑计算的可实现性(即与实际系统的语义距离),还要考虑语义的可验证性等问题。

对多 Agent 系统形式化描述常用的方法有逻辑学方法、进程代数方法、Z 语言和情景演算方法<sup>[24]</sup>等。基于逻辑的形式语言,如 Z 语言,适于描述 Agent 的静态特性,但难于描述系统的交互以及在通信活动中表现出来的各种性质。可执行时序逻辑可以直接得到可运行的系统,但缺点是表达能力较弱。逻辑学方法的语义定义主要是基于 Kripke 的可能世界的语义模型,尽管数学上具有语义一致性和充分性,但难于与实际的可计算系统直接联系起来。如模态逻辑的方法可以用来定义 Agent 的情感、思维状态及其关系,具有较强的表达能力,但存在理论和实践的分离的问题。

情景演算虽然也可以描述并发系统但描述并发系统的交互尚有一定的困难。相对于逻辑学的方法,进程代数模型和形式化方法能够刻画更细致的系统行为。象 CCS 和 CSP,具有很好的代数性质,不仅具有刻画系统行为的模型,还有进行推理的形式演算系统。Pi 演算可以认为是 CCS 的扩充,特点是通过命名通道不仅可以传送消息而且可以传送通道名,适用于通信拓扑发生变化的并发系统,实际的 Pi 演算可以根据系统

描述需要进行剪裁和扩充。

多 Agent 系统的最终目的是研究分布问题的求解机制,在进行系统的形式化和语义解释时,要考虑系统的可计算性和可验证性等问题,从前面的分析可以看出,不同的形式化方法往往采用不同的语义模型进行语义定义,而引入形式语义学的方法,特别是操作语义的方法对系统的可计算性将是有益的。

#### 4 多 Agent 系统的形式语义描述

计算机语言的形式化语义有操作语义、指称语义、代数语义和公理语义等<sup>[14]</sup>。操作语义是用抽象的方法描述语言中的每一成分的执行效果,通过抽象机运行的状态转换给出语言的语义。指称语义是使语言的每一成分对应于一个数学对象,即该语言成分的指代,不考虑语言的执行,而只考虑语言执行的最终效果。代数语义是把描述语义的逻辑体系和满足这个逻辑体系的模型区分开来,而把模型看成一种代数结构。公理语义是在程序的正确性验证的基础上发展起来的,只是给出一种方法使人们能在给定的前提下,验证程序的某种性质是否成立。

考虑到多 Agent 系统是一个开发、分布、自治的系统,Agent 间的交互是多 Agent 系统的特点,在对多 Agent 系统进行描述和验证时,需要对传统的语义学方法进行扩展。

使用形式语义学对多 Agent 系统进行语义描述时,大致可分为几步来完成。

(1)建立一种形式描述语言(DL)可以是用户根据需要自己定义的,也可以是从已有的形式描述工具中选用的,或选择后针对多 Agent 系统的描述需求进行扩充或改造的。

在选择适用的描述工具时,要考虑候选语言的描述粒度。粒度太细的描述语言会使对象的描述陷入过多的细节中去,描述效率低;过粗的描述粒度,难于对系统进行细致的刻画,需要在二者之间进行权衡。

(2)给出描述语言 DL 的形式语义,一般说前面提到的形式语义学的方法都可以准确地给出 DL 的形式语义,但操作语义学的方法更接近语言的实现。

(3)建立被描述对象的模型(OM)。被描述对象的模型应该能够反映多 Agent 系统的静态和动态特性。

对被描述对象进行建模是建立在对被描述对象的分析和认识基础之上的,对被描述对象的认识越透彻,模型越完善,越容易作进一步的形式描述。

(4)用描述语言 DL 对 OM 进行描述。

用形式语义学的方法对多 Agent 系统进行描述时,应结合已有的逻辑学方法,同时考虑对多 Agent 系统中个体和群体的思维状态的描述。

对多 Agent 系统用形式语义学方法进行描述和验证的研究工作还是初步的,主要工作集中在对 Agent 通信语言的语义描述上,如前面提到的 Finin 和 Rouger 等的工作,Ferber 等关于组织形成的操作语义的工作仅描述了 Group 的创建一个 Agent 加入 Group 的过程,而没有考虑组织结构的设计以及组织的演化等问题,另外如何对组织规则进行抽象和形式化也是一个应考虑的问题<sup>[24]</sup>。

总体来说,用形式语义学的方法对多 Agent 系统进行研究的工作尚不多见,而且多集中在对 ACL 和 API 的形式描述问题上,除此以外,对多 Agent 系统的某个侧面和整体进行形式语义学描述的研究工作较少有文献报道,而 ACL 的语义验证也是一个尚未解决的问题。

对多 Agent 系统进行形式语义学研究可以从多方面展开,从整体上讲 Agent 的社会性及其形式语义是一个需要研究的重要问题,多 Agent 系统的组织、交互以及通信语言都可以看成是 Agent 社会性的反映,虽然在这方面已经有较多的工作,但尚缺乏统一的理论框架和形式语义学方面的工作,多 Agent 系统的社会性是否存在一个统一的语义模型框架,使得系统的各种属性及其规律可以由此框架一致性地给出,也是一个值得探讨的问题,另外,对多 Agent 系统的各个侧面及其属性进行形式语义学方面的分析和研究也是有益的工作。

对 Agent 社会性进行形式语义学的研究,存在自顶向下和自底向上的不同的研究路线,所谓的自顶向下是指先建立宏观的一致性的多 Agent 系统的社会模型然后在逐步向下分解进行语义学的研究;而自底向上的研究路线是指从对多 Agent 系统社会性的个别现象和侧面的语义研究做起,如组织、交互等,逐步建立起整个多 Agent 系统社会性的语义模型,在对多 Agent 系统社会性的基本运行机制和宏观规律没有完全把握以前,采用自底向上的方法是比较可行的,而对 Agent 的社会性及其形式语义研究的根本目的之一也是希望能最终把握多 Agent 系统社会性的基本规律和基本属性。

**结束语** 对于多 Agent 系统的描述需要进行形式化的处理,更需要严格的语义描述,通过分析可以看出,各种不同的形式化方法各有特点,而形式化语义的描述则需要考虑系统的可计算性和可验证性等问题,本文提出了用操作语义学的方法进行多 Agent 系统及其社会性的形式语义描述的问题,今后的工作将是选择合适的形式描述语言和抽象机,对社会 Agent 的诸方面进行描述研究,另外建立社会 Agent 的形式模型也是一个需要考虑的问题。

## 参考文献

- 1 石纯一,等. 基于 Agent 的计算. 知识科学与知识工程学术会议,1999.12
- 2 Hintikka J. Knowledge and belief. Cornell University Press, Ithaca, NY, 1962
- 3 Cohen P R, Levesque H J. Intention is Choice with Commitment. Artificial Intelligence, 1990, 42
- 4 Rao, Georgeff. Modeling rational Agents within a BDI-architecture. In: Proc of KR-91, San Mateo, CA, USA, 1991
- 5 Konolige K, Pollack M E. A Representationalist Theory of Intention. IJCAI-93, 390~395
- 6 Bell J. Changing Mental Attitudes, ECAI-94, Springer
- 7 Gaspar G, Coelho H. Where Do Intentions Come from? A Framework for Goals and Intentions Adoption, Derivation and Evolution. EPIA'95, Springer
- 8 d'Inverno M, Kinny D, Luck M, Wooldridge M. A Formal Specification of dMARS. In: M. P. Singh, A. S. Rao, and M. Wooldridge, eds. Intelligent Agents IV. Springer-Verlag Lecture Notes in AI Volume 1365, Feb. 1998
- 9 Wooldridge M. A Knowledge-Theoretic Semantics for Concurrent MetateM. In: J. Mueller, M. Wooldridge, and N. R. Jennings, eds. Intelligent Agents III, Springer-Verlag, 1997
- 10 Lesperance Y, Levesque H J, Reiter R. A Situation Calculus Approach to Modeling and Programming Agents. In: A. Rao and M. Wooldridge, eds. Foundations and Theories of Rational Agency. Kluwer, 1999
- 11 Shoham Y. Agent-oriented programming. Artificial Intelligence, 1993, 60, 51~92
- 12 Chanbr W. Proposition of formal semantics for multi-Agent systems. Computers and Industrial Engineering, 1999, 37, 453~456
- 13 Purvis M, Cranefield S. Agent modelling with Petri Nets. In: Proc. Of CESA '96 IMACS Multiconference, IEEE-SMC, Lille, 1996
- 14 Cost R S, et al. Modelling Agent conversations with Colored Petri Nets. Third Conference on Autonomous Agents (Agents-99), Workshop on Agent Conversation Policies, Seattle, WA, May 1999
- 15 Jennings N R. Agent-Based Computing, Promise and perils. In: IJCAI-99, 1429~1436
- 16 徐晋晖. 多 Agent 模型与联盟机制研究. [清华大学博士论文]. 2000
- 17 Ferber J, Gutknecht O. Operational semantics of multi-Agent organization, in ATAL'99
- 18 Ferber J, Gutknecht O. A meta-model for the analysis and design of organizations in multi-agent systems. ICMA-98 Proceedings. IEEE 1998
- 19 Finn T, et al. KQML: An Information and Knowledge Exchange Protocol. In: Kazuhiro Fuchi and Toshio Yokoi, eds. Knowledge Building and Knowledge Sharing, Ohmsha and IOS Press, 1994
- 20 Labrou Y, Finn T. Semantics for an Agent Communication Language. In: Intelligent Agents IV: Agent Theories, Architectures and Languages. Michael Wooldridge, Munindar Singh and Anand Rao (editors.). Springer-Verlag, Lecture Notes in AI, Volume 1365, 1998
- 21 Van Eijk Roger M, de Boer Frank S, van der Hoek, Wiebe, Meyer John-Jules. Operational semantics for Agent communication languages [Report Utrecht University UU-CS-1999-08]
- 22 Wooldridge M. Verifiable semantics for Agent communication languages. In: Y. Demazeau, ed. Proc. of the Third Int Conf on Multi-Agent Systems (IC 多 Agent 系统98). IEEE Press, July 1998
- 23 Wooldridge M. Verifying that Agent implement a communication language. In: Proc. of the Sixteenth National Conf. on Artificial Intelligence (AAAI-99), Orlando, FL, July 1999
- 24 Lesperance Y, Levesque H J, Reiter R. A situation calculus approach to modeling and programming Agents. In: A. Rao and M. Wooldridge, eds. Foundations and Theories of Rational Agency. Kluwer, 1999
- 25 陆汝铃. 计算机语言的形式语义. 科学出版社, 1992
- 26 Zambonelli F, Jennings N R, Wooldridge M. Organisational Abstractions for the Analysis and Design of Multi-Agent Systems. In: Proc. 1st Int. Workshop on Agent-Oriented Software Engineering, Limerick, Ireland, 127~141

(上接第63页)

## 参考文献

- 1 Allman M, Paxson V, Stevens W. TCP Congestion Control. Request for Comments 2581, April 1999
- 2 Rejaie R, Handley M, Estrin D. An End-to-End Rate-based Congestion Control Mechanism for Realtime Streams in the Internet. In: Proc of INFOCOMM'99, 1999
- 3 Padhye J, Kurose J, Towsley D, Koodli R. A Model Based TCP-Friendly Rate Control Protocol. In: Proc. of NOSS-DAV'99, 1999
- 4 翟明玉, 吴国新, 顾冠群. Internet 可靠多点投递拥塞控制研究进展. 计算机研究与发展, 2000, 37(1): 9~16
- 5 Mahdavi J, Floyd S. TCP-Friendly Unicast Rate-based Flow Control. Note sent to end 2end-interest mailing list, Jan 1997
- 6 Floyd S, Fall K. Promoting the Use of End-to-End Congestion Control in the Internet. IEEE/ACM Transactions on Networks, 1999, 7(4): 458~472
- 7 Padhye J, Firoiu V, Towsley D. Modeling TCP Throughput: A Simple Model and its Empirical Validation. SIGCOMM Symposium on Communications Architectures and Protocols, 1998(Aug): 303~304
- 8 Mogul J, Deering S. Path MTU Discovery. Request for Comments 1191 Nov. 1991
- 9 Jain R. The Art of Computer Systems Performance Analysis. John Wiley and Sons, 1991