

# 基于时间约束的软件工程资源配置 Petri 网检测模型

黄 妮, 吴晓军

(陕西师范大学 计算机科学学院, 陕西 西安 710062)

**摘 要:**随着物联网、云计算等大规模分布式系统在国内的日益流行, 软件工程的应用环境已经不可避免的转向适用于分布式环境。而已有的基于时间约束的软件工程资源动态配置模型大都有所局限, 无法评价其在大规模分布式环境中的效果。文中提出了一种基于 Petri 网模型的基于时间约束的软件工程资源动态配置检测模型, 依据 petri 网理论, 模拟大规模分布式环境下, 检测异构软件工程资源动态配置模型间的协作与冲突。从而为软件工程资源动态配置模型更好的应用于实践, 提供坚实的理论基础。实验证明, 该方法可以及时的发现大规模分布式环境中, 软件工程资源动态配置模型的冲突, 为设计者及时反馈信息, 避免损失。

**关键词:**时间约束; 软件工程; 资源配置; Petri 网

中图分类号: TP31

文献标识码: A

文章编号: 1673-629X(2012)04-0021-04

## Petri Net Detecting Model of Software Engineering Resource Configuration Based on Time Constraint

HUANG Ni, WU xiao-jun

(School of Computer Science, Shanxi Normal University, Xi'an 710062, China)

**Abstract:** As the internet of things, cloud computing and other large-scale distributed systems become increasingly popular in the country, the environment of the application of software has inevitably turned to distributed environment. However, there are some limitations for the software engineering dynamic resource configuration model based on time constraint, and thus its effect in large-scale distributed environment can not be evaluated. In this paper, based on the Petri net model, propose a dynamic resource configuration detecting model of the software engineering based on time constraint. According to Petri net theory, with the simulation of large-scale distributed environment, the collaboration and conflict between the dynamic resource configuration models of heterogeneous software engineering has been detected, so that a solid theoretical basis will be provided and its application in practice will be bettered. Experiments show that this method, in the large-scale distributed environments, can timely find the conflicts in the dynamic configuration model of the software engineering resource and timely provide the feedback for the designers to avoid losses.

**Key words:** time constraint; software engineering; resource configuration; Petri net

### 0 引 言

软件在其自身的发展中出现了很多问题, 其中出现的软件危机阻碍了软件的开发与管理, 改进软件的开发过程对于缓解软件危机具有重要意义<sup>[1]</sup>。软件过程作为一个多元非线性系统, 是与活动、约束和资源相关的有顺序的集合<sup>[1~3]</sup>。

随着物联网、云计算等大规模分布式系统在国内的日益流行, 软件工程的应用环境已经不可避免的转

向跨地域大规模协作式合作开发的模式。然而, 在协作式开发过程中, 总会有一些共享资源需要在某一段时刻被某个开发小组独立占用, 当各个小组都需要该资源时, 不可避免的会出现冲突。针对该问题, 很多学者提出了解决方案, 大都集中在对软件过程中资源使用以及成本与工期的问题上。然而, 软件开发是一个协作式过程, 避免各个小组工作计划的冲突是软件工程建模的重点。文中结合 Petri 网理论, 提出一种检测冲突方法。

### 1 Petri 网基本知识

Petri 网是对离散并行系统的数学表示方法, 可以用于并发系统的建模和分析, 1960 年由 C. A. Petri 博

收稿日期: 2011-09-20; 修回日期: 2011-12-25

基金项目: 国家自然科学基金项目(11172342)

作者简介: 黄 妮(1981-), 女, 陕西蓝田人, 硕士研究生, 研究方向为数字信号处理与应用、计算机应用; 吴晓军, 硕士生导师, 副教授, 研究方向为计算机应用、模式识别。

士发明。基于 Petri 网的系统建模,是采用通用网论的思想,在系统建模设计时可按整体建模设计和局部建模设计等等进行,从而在最大限度上简化设计。通过形式语言等方法来分析 Petri 网模型的可达性、活性、公平性等性质,从而正确地产生系统性能评价,并提出相应的系统改进建议。假设读者对 Petri 网的基本概念、基本理论已有所了解,详细内容请参见文献[4]。

Petri 网进行实际系统建模时,其主要是通过系统模型以达到对实际系统的性能和功能进行分析的目的。该系统模型有直观的图形表示,还能给出形式化的表述。若某一 Petri 网模型能够准确地将一个实际系统的结构和运行描述出来,则 Petri 网模型就能将实际系统的性质体现出来。

Petri 网的动态性质包括<sup>[5-7]</sup>:

- 1) 可达性。在 Petri 网中,可达性作为最基本的动态性质,可以用来定义 Petri 网的其它性质;
- 2) 可逆性。可逆性反映系统的可回复性;
- 3) 可覆盖性;
- 4) 有界性。有界性是指被模拟的实际系统在运行中,其对有关资源容量的需求的体现;
- 5) 安全性。决定系统中正在执行的操作不会发出请求。若 Petri 网为有界,则称此 Petri 网是安全的。这种网的每一个库所要么有一个令牌,要么没有令牌。安全性是有界性的一种特殊情况;
- 6) 活性。一个 Petri 网,如果从  $M_0$  可达的任一标识出发,对于任一变迁,都可以通过一变迁序列而最终使该变迁发生,则这一 Petri 网具有活性;
- 7) 公平性。公平性是用来分析 Petri 网系统中两变迁发生时两者的关系,以便有效地反映在资源竞争中实际系统中各部的无饥饿性问题;
- 8) 持续性。

## 2 Petri 网的优点

与其他网系统模型相比,便于描述并发和冲突即是 Petri 网的突出优点之一。由于文中主要是应用 Petri 网的并发与协调。为此,以下将详细介绍 Petri 网是怎样描述事件之间的并发与冲突的。

### 2.1 并发

基本网系统  $\sum_1 = (B, E; F, c_0)$ , 其中  $c_0 = \{b_1, b_2\}$ 。在情态  $c_0$  下事件  $e_2$  和  $e_3$  都有权发生。这是因为

$${}^*e_2 = \{b_1\} \subseteq c_0 \text{ 且 } {}^*e_2 \cap c_0 = \{b_1\} \cap \{b_1, b_2\} = \emptyset \quad (1)$$

$${}^*e_3 = \{b_2\} \subseteq c_0 \text{ 且 } {}^*e_3 \cap c_0 = \{b_2\} \cap \{b_1, b_2\} = \emptyset \quad (2)$$

假设在情态  $c_0$  时  $e_2$  发生,则变为  $c_1 = \{b_2, b_3\}$  这个

新情态,容易知道在情态  $c_1$  下  $e_3$  仍有权发生。反过来也一样,若在情态  $c_0$  时  $e_3$  发生,则变为  $c_2 = \{b_1, b_4\}$  这个新情态,易知在情态  $c_2$  下  $e_2$  也仍有权发生。则称在情态  $c_0$  时,  $e_2$  和  $e_3$  为并发关系。

综上所述,在某种情态条件下,若两个事件都有发生权,且这两个事件中任意一个的发生与剩下的一个无关,也即两者都有发生权不会使另外一个失去发生权,即在该情态条件下这两个事件处于并发关系。

在某种情态下两个事件处于并发关系而并非为“同时发生”,并发是指两事件之间在因果上的无依赖性。按照网论的观点,事件发生只与事件本身的外延有关,不受全局情况的影响。

在基本网系统中,事件  $e_2$  的外延:  ${}^*e_2 \cup e_2^* = \{b_1, b_3\}$

事件  $e_3$  的外延:

$${}^*e_3 \cup e_3^* = \{b_2, b_4\}$$

从两者的外延公式可知,两者的外延无公共部分。因此,这两个事件是互相独立的事件。这是基本网系统中两个事件处于并发的必要条件。

在情态  $c_0 = \{b_1, b_2\}$  条件下,  $e_2$  有发生权,  $e_4$  没发生权,若  $e_2$  发生,则有新情态  $c_1 = \{b_2, b_3\}$  生成,易知  $e_4$  有发生权。即,在  $c_0$  情态下,  $e_4$  的发生权是通过  $e_2$  的发生而得到的。则  $e_2$  和  $e_4$  的发生是一种顺序关系。

### 2.2 冲突

基本网系统  $\sum_2 = (B, E; F, c_0)$ , 其中  $c_0 = \{b_2\}$ 。

在情态  $c_0$  条件下,  $e_1$ 、 $e_3$  这两事件都有发生的可能。若  $e_1$  发生,则产生  $c_1 = \{b_1\}$  这个新情态,同时在  $c_1$  情态下  $e_3$  失去发生权。反之亦然,在情态  $c_0$  下,若  $e_3$  发生,则产生这个新情态  $c_2 = \{b_3\}$ , 同时在情态  $c_2$  下  $e_1$  失去发生权。即为冲突关系。

事件间的冲突是对系统非确定性的描述:在某情态条件下,每个事件都有权发生(即都有发生权),然而系统的实际运行中,真正能发生的事件唯独只有一个。冲突的存在,使得外界环境正好能够对系统施加控制,做出选择。

在某个情态条件下,系统有时可能同时存在并发和冲突,但当并发事件中某些事件的发生,会使冲突事件中的冲突自动消失。也有另外一种情况:即在某个情态条件下,系统存在并发关系,但当并发事件中不同的事件发生,系统中冲突有可能出现,也有可能不出现。这两种现象称作混感现象。

若网系统中存在混感现象,则网系统在运行过程中,可能会出现冲突也可能不出现冲突(即冲突的出现与否无法确定),使得外部控制难以施加。因此,在实际系统 Petri 网模型建立过程中,应避免混感现象。

### 2.3 一般 Petri 网中的并发与冲突

一般 Petri 网中,并发和冲突也是存在的,且含义与基本网系统基本一致。但事件发生的规则在不同网系统中有所不同。具体区别分析如下:

设  $\Sigma$  为一个网系统,  $t_1$  和  $t_2$  是  $\Sigma$  中的两个变迁。如果  $\Sigma$  的一个标识  $M$  使得  $M[t_1 >]$  且  $M[t_2 >]$ , 那么

1) 若  $M[t_1 >] \rightarrow M_1[t_2 >]$  且  $M[t_2 >] \rightarrow M_2[t_1 >]$ , 则称  $t_1$  和  $t_2$  在  $M$  并发, 记为  $M[\{t_1, t_2\} >]$ ;

2) 若  $M[t_1 >] \rightarrow \neg M_1[t_2 >]$  且  $M[t_2 >] \rightarrow \neg M_2[t_1 >]$ , 则称  $t_1$  和  $t_2$  在  $M$  冲突。

在一般网系统和基本网系统中,并发和冲突两者的概念在定义上是一致的,但实际表现上还是有区别的,具体如下:一般 Petri 网系统中,不存在冲撞的概念。主要是因为 Petri 网系统的库所具有无穷大的容量,因此,在 Petri 网系统中,只需一个变迁(事件)的前集各库所具有的标志足够多,则该变迁(事件)就能发生。即使该变迁的后集中某些库所含有标志,也不影响变迁的发生。

对于有容量函数  $K$  的 P/T 系统,其库所容量有一定的限制,在某些情态下也会出现类似的冲撞现象,但又不完全相同。考察图 1 的 P/T 系统,把各库所的容量函数值标在表示该库所的小圆圈旁。在这个 P/T 系统中,变迁  $t_1$  的前集只有一个元素  $s_1$ 。在当前标识下,  $s_1$  中含有 3 个标志,等于弧  $(s_1, t_1)$  上的权。但由于  $t_1$  的后集库所  $s_2$  中已有 3 个标记,如果  $t_1$  发生将向  $s_2$  送入 2 个标记,标记总数(5)将超过  $s_2$  的容量( $K(s_2) = 4$ ),因此在当前标识下,  $t_1$  不能发生。然而,如果在当前状态下,  $t_2$  先发生,  $t_2$  发生后,  $s_2$  中只剩余一个标志。这时,虽然作为  $t_1$  的后集库所  $s_2$  中非空,但其容量足以容纳  $t_1$  发生后送到  $s_2$  的标志,所以在这种情况下  $t_1$  就可能发生。这种情况反映了实际系统中的阻塞现象。

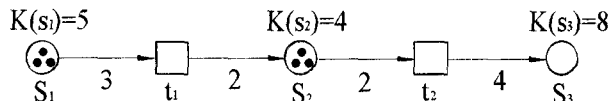


图 1 一个存在阻塞的 P/T 系统

### 3 UML 的传统顺序图

顺序图通过描述一些对象的消息交换。其重点关注对象传递消息的时序。消息既可以是同步的,也可以是异步的,通过对象间消息的传递顺序来表示对象间的交互行为。在本节中将对顺序图的相关内容进行简要的介绍<sup>[8-10]</sup>,具体如下:

(1) 对象 (Object): 顺序图中显示的对象为该消

息序列中所涉及的对象,对象作为类的实例,依次排列在水平轴上,用矩形来表示,矩形中的对象名都标有下划线。对象的命名方式为“对象名”:“对象所属类”,前者可省略。

(2) 生命线 (Lifeline): 对象的生命线即为对象下面那条垂直的虚线,其作用是用来显示对象在消息序列中的执行情况。生命线反映的是其相对应的对象在某个特定时间的存在现象,用从对象矩形向下至图底的一条虚线来表示。

(3) 激活 (Activation): 当对象接到一个消息,其活动就会启动,这一接收启动的过程即为激活。当显示控制点被激活后,就会显示对象在某一个时间点的执行。在图形上,一个被激活的对象或是执行它自身的代码,或等待另一个对象的返回,激活被绘制为对象生命线上一个瘦高矩形。

(4) 消息 (Message): 各对象间的通信(即消息)是通过水平消息线来表示的,这些水平消息线位于各对象的生命线间。消息的类型由水平消息线的箭头形式确定。从一个对象到另一个对象的通信都是消息,例如一个对象调用其他对象的操作,创建或销毁一个对象,向其他对象发送信号等。在顺序图中它由各个对象之间的一条水平消息线来表示。它包括三类:第一类,当一个对象调用操作后暂停执行,只有操作返回时,个体才会恢复执行,使用同步消息箭头。第二类,当一个对象调用操作后运行在其他线程或进程中,并且不等待调用的操作返回,这时候该使用异步消息箭头。第三类,用虚线箭头表示控制权从同步消息返回,以创建新实体。

(5) 返回 (Return): 当发送的消息返回同步消息等结果时,应用带箭头的虚线来表示消息返回,以便于与发送的消息区分,用带箭头的虚线来表示消息返回。但是要注意,消息的返回不能总被顺序图显示。

(6) 条件 (Condition): 消息能够有条件。消息的发送是由条件决定的,只有条件为真,消息才能被发送或被接收。条件在顺序图中是通过在消息前以方括号的形式表示。

为了能够更有效地应用大型异步系统中的并发、选择、同步等性质的建模,应当对 UML 的传统顺序图进行合理的扩展,以便能够更好地描述系统的行为。

### 4 基于时间约束的软件工程资源配置 Petri 网检测模型

通过绘制出各个协作个体在软件工程中使资源的 UML 顺序图,之后通过相应的转换方法,将 UML 顺序图转换成相应的 Petri 网模型<sup>[10-15]</sup>。

其步骤如下:

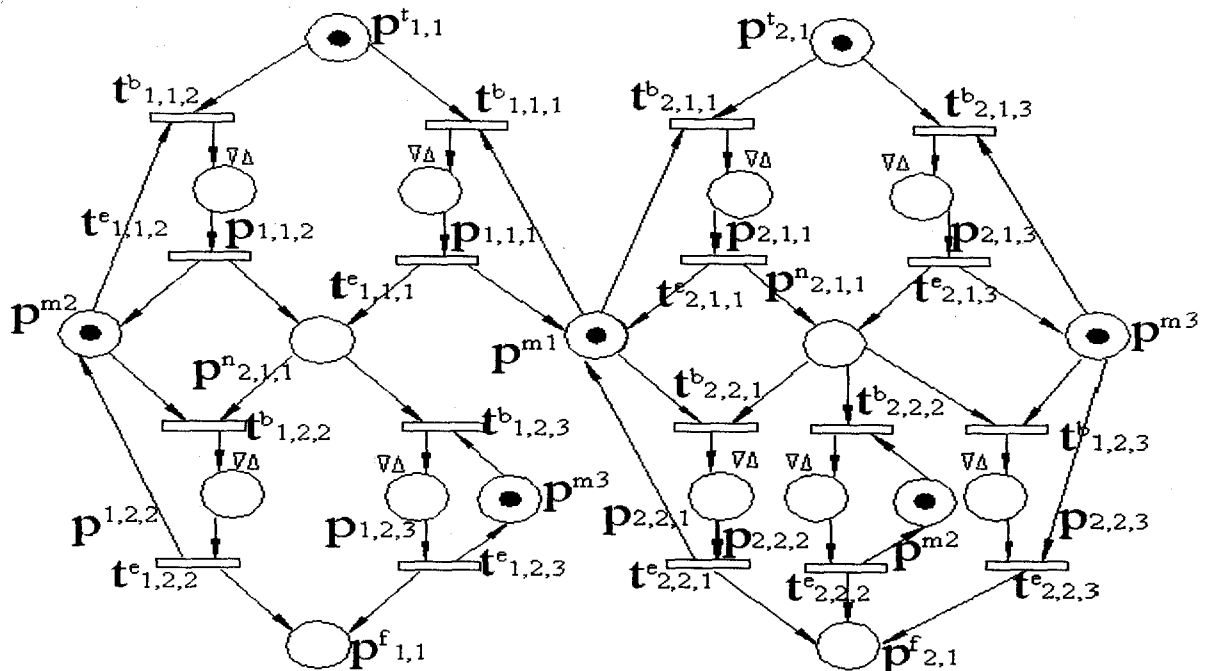


图 2 Petri 网检测模型实例图

Step1: 绘制协作个体的顺序图, 可用 UML 顺序图表示;

Step2: 将 UML 顺序图转换成相应的 Petri 网模型;

Step3: 分析 Petri 网模型冲突等特性;

Step4: 发现有色 Petri 网未发现的漏洞, 进行修补, 转入 Step3。否则, 则退出。

## 5 实验

其中实验主要使用有色 Petri 网分析工具 CPN Tools(2.2.0); 采用的程序设计语言为 CPN ML, CPN ML 是建立在 SML (Standard ML) 上的一种解释性语言, 对于模型中的变量定义、条件转移语句, 控制语句等都是通过 CPN ML 语法进行描述的, 对于可视化交互仿真、可视化查询、状态空间分析报告的产生也是通过 CPN ML 实现的, 对有色 Petri 网进行综合分析。

对通过有色 Petri 网分析工具分析出来的系统设计漏洞, 运用测试手段, 对系统设计漏洞进行实际检测, 以验证本项目提出方法的正确性; 对成功修正后的 UML 模型进行全面检测, 已验证不存在有色 Petri 网未发现的漏洞。使用实例<sup>[4]</sup>, 对应的基于时间约束的软件工程资源配置使用情况流程图已被描述。通过文中提出的 Petri 网验证方法, 可以得到通过图 2 生产的报表, 可知该模型没有冲突发生。

## 6 结束语

基于时间约束的软件工程资源配置 Petri 网检测模型成功地解决了协作式软件工程设计中资源合理分配中冲突检测的问题, 为软件过程的控制提供了科学

而又合理的依据, 从而实现了对其开发的工期、成本的要求, 仿真表明该方案是可行的。

### 参考文献:

- [1] 雒雪芳. 基于进化计算的软件过程优化及成本估算研究 [D]. 西安: 西北工业大学, 2005.
- [2] 刘 伟. 活动网络费用优化问题及其遗传算法 [J]. 系统工程学报, 1999, 14(2): 72-77.
- [3] 徐晶明, 杜宝珠. 基于网化简技术的工作流过程模型结构验证 [J]. 计算机技术与发展, 2009, 19(6): 51-54.
- [4] Park J, Reveliotis S A. Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings [J]. IEEE Trans. on Automat. Contr., 2001, 46(10): 1572-1583.
- [5] 吴哲辉. Petri 网导论 [M]. 北京: 机械工业出版社, 2006.
- [6] 刘林钢, 姜 浩. 基于时间约束 Petri 网的一致性验证算法 [J]. 计算机技术与发展, 2010, 20(1): 58-62.
- [7] 李金奎, 刘久富. 有色 petri 网在并行测试建模中的应用 [J]. 计算机技术与发展, 2011, 21(1): 9-11.
- [8] Lee D Y, Frank D. Scheduling flexible manufacturing systems using Petri nets and heuristic search [J]. IEEE Transactions on Robotics and Automation, 1994, 10(2): 123-132.
- [9] Kazuhiro S. Robust design of flexible manufacturing systems using colored Petri net and genetic algorithm [J]. Journal of Intelligent Manufacturing, 2002, 13(5): 339-351.
- [10] 刘广硕, 朱建冲. 基于广义随机 Petri 网的装备维修保障建模与分析 [J]. 计算机技术与发展, 2010, 20(12): 158-161.
- [11] Grolleau E, Choquet-Geniet A. Off-line computation of real-time schedules using Petri nets [J]. Discrete Event Dynamic

(下转第 28 页)

忙状态,吞吐量总体有所下降,4.0s后 n0 停止发送 ftp 数据,吞吐量又趋于稳定。

网络通过 NS2 仿真,使虚拟的网络环境具体化,能方便快捷地测试出网络性能参数,这对网络管理者管理网络具有较大地参考意义。

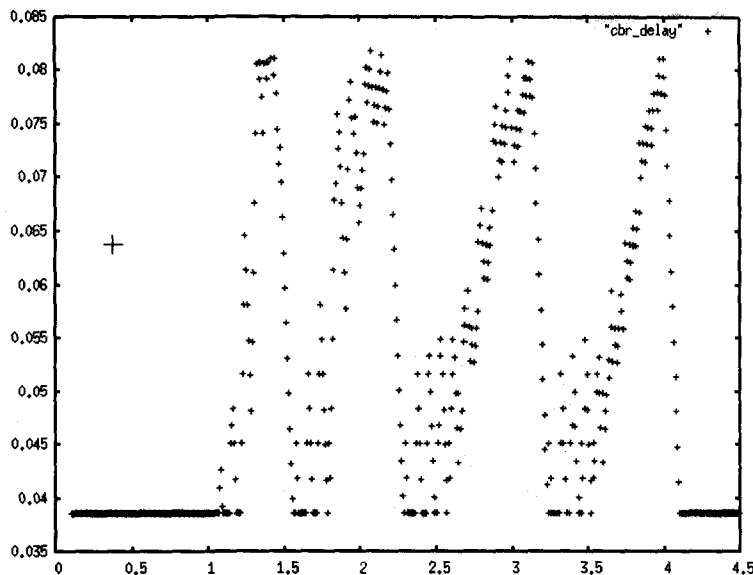


图 5 n2 发送的 CBR 封包端到端的延迟

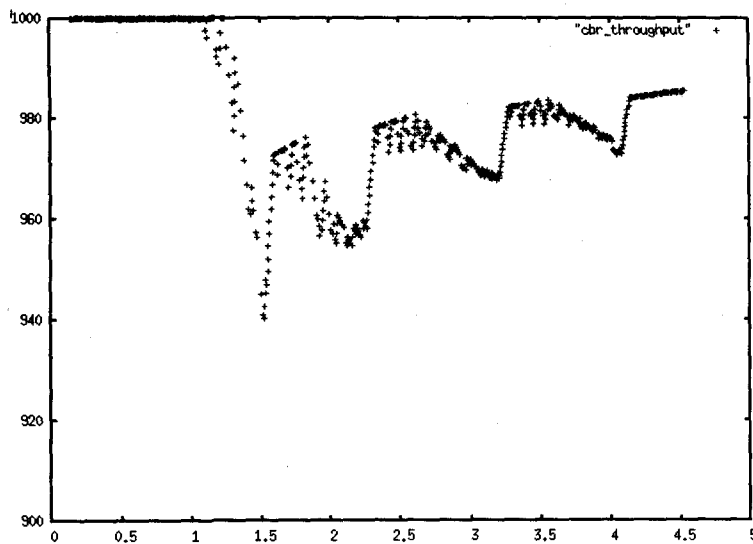


图 6 节点 n3 吞吐量

#### 参考文献:

- [1] 徐雷鸣, 庞博, 赵耀. NS 与网络模拟 [M]. 北京: 人民邮电出版社, 2003.
- [2] 吴仕浩, 林庆华, 胥布工. 网络仿真器 NS-2 及其一个应用实例 [J]. 计算机仿真, 2004, 21(7): 96-98.
- [3] 程涛. 基于 NS-2 的网络拓扑性能研究 [J]. 哈尔滨商业大学学报(自然科学版), 2010, 26(3): 577-582.
- [4] 蔡群英, 黄镇建. 搭建网络实验环境的探索与实践 [J]. 计算机与数字工程, 2009, 37(6): 195-197.
- [5] 柯志亨, 程荣祥, 邓德隽. NS2 仿真实验-多媒体和无线网络通信 [M]. 北京: 电子工业出版社, 2009.
- [6] Floyd S. Promoting the use of end-to-end congestion control in the internet [J]. IEEE/ACM Transactions on Networking, 1999, 7(4): 367-373.
- [7] 陈春梅. 基于 NS2 的网络仿真与性能分析 [J]. 通信技术, 2010, 43(8): 48-50.
- [8] 王明伟, 王奇, 杨洁, 等. 基于 NS-2 的视频流仿真方法研究 [J]. 计算机技术与发展, 2010, 20(2): 117-120.
- [9] 毛跟勇, 刘成安, 兰瑞莉. 料位控制测量的无线网络在 NS2 中的模拟 [J]. 计算机技术与发展, 2010, 20(3): 163-166.
- [10] Fall K, Varadhan K. The NS Manual [M]. [s.l.]: the VINT Project, 2001.
- [11] 李蓬, 黄河. 基于 NS2 的 MPLS 流量工程仿真研究 [J]. 计算机技术与发展, 2008, 18(9): 53-56.

## 5 结束语

网络仿真是网络安全管理和网络监测技术的重要手段。对于复杂的网络, 进行网络仿真时, 应抽取研究关注的部分进行模拟。从仿真的效果可知, 将真实的

- [12] Dagher J C, Marcellin M W, Neifeid M A. A theory for maximizing the life time of sensor networks [J]. IEEE Transactions on Communications, 2007, 55(2): 323-332.

(上接第 24 页)

Systems, 2002, 12(3): 311-333.

- [12] 杨玉梅, 刁永锋. 基于 UML 顺序图的 Petri 网建模 [J]. 计算机技术与发展, 2007, 17(10): 130-133.

- [13] 周长红. UML 图的 Petri 网建模 [D]. 青岛: 山东科技大学, 2004.

- [14] 陈志德, 曾凡平. UML 状态图和 Petri 网络在类测试用例生成的应用 [J]. 小型微型计算机系统, 2010, 31(3): 519-522.

- [15] 冯林, 姜浩. 基于时间约束 Petri 网的工作流可调度性分析 [J]. 计算机技术与发展, 2006, 16(11): 34-37.