

# 基于网页划分的网站应用程序测试新方法

濮方珺 卢炎生

(华中科技大学计算机学院 武汉 430074)

**摘要** 网站已经成为全球信息发布的重要渠道,确认网站应用程序的质量和可靠性成为软件测试的一项重要任务。现在有关网站应用程序的测试技术大多数是基于模型测试技术,依赖于从网站编码信息中产生模型,这些技术可统称为“白盒测试”。目前没有一种技术应用网页的划分进行测试。本文在“白盒测试”的基础之上,提出了基于网页划分的网站应用程序测试技术和基于网页划分的“白盒测试”技术,并通过实验证明基于网页划分的“白盒测试”技术是测试网站应用程序的较为有效的方法。

**关键词** 软件测试,网站应用程序,网页划分,实验分析

## A New Methodology for Web Application Testing Based on Web-page Partitions

PU Fang-Li LU Yan-Sheng

(Department of Computer Science, HUST, Wuhan 430074)

**Abstract** Web applications are important equipment for publicizing the global information. So it is important to validate their quality and dependability. Many techniques have been proposed for testing Web applications. But most of them are model-based testing and rely on information gathered from the Web application code to generate the models. They can be classified as “white-box” techniques. None have attempted to use Web-page partitions to support Web application testing. In this paper, based on white-box techniques, we present Web-page partitions techniques and white-box technique based on Web-page partitions. From the result of the experiment, white-box technique based on Web-page partitions is an efficient method for testing Web application.

**Keywords** Software testing, Web applications, Web-page partitions, Empirical studies

## 1 引言

随着 Internet 的广泛应用,各种各样的网站,如政府网站、商业网站、科学技术网站等正逐步走入人们的生活。这些网站实现了信息的共享和在线办事。网站应用程序已成为软件系统中发展最快的一类,但在实际应用中有超过一半的商业网站和政府网站的应用程序存在错误,因此网站应用程序的质量和可靠性的验证显得越来越重要。目前网站测试是确认网站应用程序质量和可靠性的重要手段。近年来,提出了一些测试网站应用程序功能要求的方法,这些方法可统称为“白盒测试”技术。它们的主要原理是从网站应用程序编码中建立系统模型并从这些模型中确定用户需求,再根据用户需求生成测试用例。这些“白盒测试”技术耗费较大,因为需要花费大量的精力去生成满足用户需求的测试用例。同时,这种“白盒测试”方法存在局限性:首先,网站应用程序的使用千变万化,一个网站每天可能被点击上万次,页面变化较快,因此依照部分用户需求生成的测试用例不能满足充分测试的要求;其次,网站运行过程中有许多细小增量,适应这些增量变化的花费太大,因此必须自动生成测试方法和相应的测试用例,而“白盒测试”主要是通过人工进行测试;最后,网站应用程序涉及复杂、多层次、成分混杂的结构,包括网站、应用和数据库等,网站应用程序测试技术必须能在这种复杂结构中处理各种不同的组成成分。

本文在“白盒测试”基础之上提出了基于网页划分的测试

方法,我们把网页分为:数据库驱动网页、用编程方法创建的网页和其它网页,同一类网页具有相同特性。这种基于网页划分测试方法不根据模型生成测试用例,克服了“白盒测试”方法中测试用例生成难及测试用例不充分的缺点,能对网站应用程序进行更有效的测试。本文提出了基于网页划分测试技术以及把“白盒测试”技术与网页划分相结合的基于网页划分的“白盒测试”技术,并分别运用“白盒测试”技术、基于网页划分技术和基于网页划分的“白盒测试”技术对一商业网站进行测试。实验结果发现基于网页划分的“白盒测试”技术不仅较易生成测试用例,而且与前两种方法相比,能发现更多的网站应用程序中的错误。基于网页划分的“白盒测试”方法是网站应用程序的较为有效的测试方法。

## 2 相关工作

### 2.1 网站应用介绍

网站应用可定义为用户对网站的使用。一个简单的网站应用如图 1 所示:用户通过网站浏览器发出申请,网站服务器收到用户申请后向应用服务器发出网页申请,网站服务器获得生成的网页后通过浏览器呈递给用户。实际生活中,用户对网站的申请是复杂和多样的,这就导致网站的应用是多种多样的和复杂多变的。如果我们仅根据网站的特定模型生成测试用例,具有很大的局限性且不能达到充分测试。

### 2.2 相关工作

近年来,提出了一些针对网站应用程序功能需求的测试

技术,这些技术均是基于模型测试技术,可归类为“白盒测试”。它们从网站应用程序编码中提取信息并在此基础上生成模型,再根据模型进行测试。Liu 等<sup>[10]</sup>提出了网站测试模型,他们把网站应用程序的组成部分视为对象,根据对象之间的数据流生成测试用例。Ricca 和 Tonella 提出了基于统一建模语言(UML)的模型对网站应用程序进行深入分析并生成测试用例。Di Lucca 等提出了与 Ricca 和 Tonella 提出的模型相同的模型,在此模型上对单个的网页进行单元测试,对互相作用的网页进行整合测试,并对客户端和服务器的网页测试提出了具体策略。总之,这些“白盒测试”方法扩展了对网站应用程序的传统的基于路径的测试,并对网站应用程序的控制流和数据流进行适当评价,以确定网站应用程序的质量和可靠性。这些“白盒测试”技术均是根据模型生成测试用例。根据模型生成测试用例存在较大缺点:一是较难生成,二是生成的测试用例不充分。本文在“白盒测试”技术基础上提出对网站的网页进行划分,在网页划分的基础之上对网站进行测试的方法,该方法能更有效地对网站应用程序进行测试。

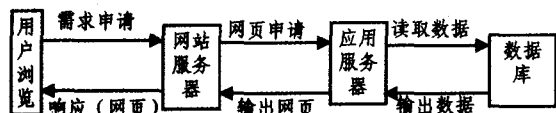


图1 网站应用示例

### 2.3 基于UML的 Ricca 和 Tonella 测试方法

Ricca 和 Tonella 提出的网站应用程序测试方法<sup>[20]</sup>首先创建一个模型,模型中节点代表网站对象(网页、结构等),边代表对象之间的联系(如提交、脱离等)。图2显示了一个汽车销售网站上用户购买汽车的例子。用户向一汽车销售网站提出需购买汽车的型号等(e1表示),网站收到申请后在服务器中查询是否有此类汽车,并将查询结果提交给用户(e2表示)。若有用户所需汽车,且用户现在就立即购买,则通过e1 e3实现。若用户不立即购买,而是预定后再购买,则通过e4e5实现。图2的整体路径需求表达式为

$$e1e3 * (e1e3 + e1e2 + e4e5)$$

“\*”表示“合”关系,“+”号表示“并”关系。根据路径需求表达式可确定用户需求,如 e1e3、e1e2、e4e5 等,根据用户需求选择测试用例进行测试。Ricca 和 Tonella 测试方法有两种运行模式,分别是:

1)RT1。根据路径需求表达式确定用户需求,根据用户需求选择测试用例进行测试,其前题条件是:仅测试线性相互独立路径;所测试的结构虽然包含在多重网页中,但仅从一个网页源头抽象出与上下文独立的相等功能;忽略环连接,单个路径均是在相同网页中。

2)RT2。减少 RT1 的前题条件,在输入值的选择上比 RT1 更精细。与 RT1 相比,RT2 把边界值(空串等)作为输入,采用“单个条件/所有条件”策略进行测试,即:根据整体的路径需求表达式确定用户需求,根据用户需求选择测试用例进行测试。在此基础上增加单个的边界值作为输入对用户需求进行测试,两种方法相比较,RT2 能发现较多的错误。

### 3 基于网页划分测试技术

网页是由文字、图片、声音、视频和链接等组成的文档,网站用户可以通过单击具有超级链接的文字和图片在网页间浏

览,搜索单词或短语,查看找到的信息。本文中网页被划分为两大类。

数据库驱动的网页:许多显示分类目录或货物清单的电子商务网页是数据库驱动的,HTML 提供 Web 内容的简单布局,而图片、文字说明、价格信息等从网站服务器上的数据中提取出来插入到网页中。

用编程方法创建的网页:许多网页,特别是包含动态内容的网页是用编程方法创建的,动态内容是根据当前条件而发生变化的文字和图片。在 HTML 中嵌入 JavaScript 之类的简单脚本语言可以对这些内容编程,这称为客户端编程。为了提高执行效率,大多数动态内容编程在网站服务器上进行,这称为服务器端编程。

其它网页:不属于以上两类的网页,如静态网页等。

基于网页划分的测试技术有:

#### 1)黑盒测试(WY1)

把同一类网页当做一个黑盒子,主要测试:文本。网页文本当做文档对待,对文本进行审查,测试每个网页是否都有正确标题;标题文本出现在浏览器的标题栏并且当把网页添加到收藏夹或书签时默认显示的内容就是标题文本。超级链接。链接可以与文字或图片拴在一起,每一个链接进行检查,确保它跳转到正确的目的地,并在正确的窗口中打开。图片。检测所有图片是否被正确载入和显示。表单。表单是指网页中用于输入和选择信息的文本框、列表框或其它域。测试表单如同测试常规软件程序的域,检测域的大小是否正确?是否接受正确数据,拒绝错误数据?在最后按 Enter 键时正确确认了吗?可选域是否真正可选并所提要求是否做到?如果输入较大数据是否提示错误信息?其他功能。网页可以包含其他特性,把这些特性按照常规程序的功能进行测试。

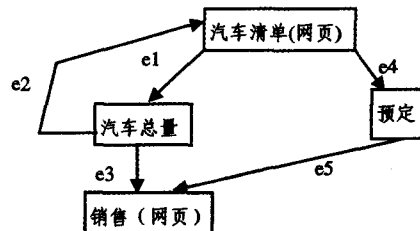


图2 简单的汽车销售网站

#### 2)灰盒测试(WY2)

灰盒测试结合黑盒测试和白盒测试,其工作原理是把软件当做黑盒来测试,但是通过简单查看(不是像白盒测试那样完整地查看)软件内部工作机制作为补充。网页的特点使其非常适合进行灰盒测试。大多数网页由 HTML(超文本标记语言)创建,HTML 与程序的区别在于 HTML 不能执行或者运行,只能确定文字和图片在屏幕上显示的方式。HTML 很容易查看,利用这点在黑盒测试基础上对 HTML 进行检查以实现网页的灰盒测试。

#### 3)白盒测试(WY3)

这种白盒测试是对网页的编码逐一进行检测,它不同于本文第2节所述的基于模型的“白盒测试”。如果网页数量较多,用这种方法测试工作量极大,因此不宜采用。一般采用白盒测试方法检测少量网页。

几点说明:

- 1)基于网页划分测试技术可采用自动测试。
- 2)基于网页划分测试技术中网页之间的链接是随意的、

非控制的,它不能测试如图2所示的网站工作流程,因此我们把“白盒测试”技术与基于网页划分测试技术结合起来,形成基于网页划分的“白盒测试”技术,以提高网站应用程序的测试效率。

#### 4 基于网页划分的“白盒测试”技术

基于网页划分的“白盒测试”技术(HH1)的测试用例生成过程为

- 选择一类网页  $P(p_1, p_2, \dots, p_n)$ ,  $p_i (1 \leq i \leq n)$  为网页地址;
- 从路径 R 中选择一用户需求  $r_i$ ;
- 将  $r_i$  转换成网站地址  $URL_{ri}$ ;
- 确认网页地址  $p_i$  中是否包含  $URL_{ri}$ , 若包含则跳转到第二步, 若不包含则将  $URL_{ri}$  作为测试用例添加进  $p_i$  中;
- 对  $r_i$  作标识为“已确认”, 从第二步开始重复执行上述步骤直到所有的用户需求都已确认;
- 对生成的新的网页组进行测试。

#### 5 实验数据

网站应用程序的主要错误有:

应用错误。包括与变量有关的错误, 如对值的定义、删除或修改中出现的错误。包括与控制流有关的错误, 如: 新模块的增加, 执行条件的重新定义, 模块的移出, 执行命令的改变, 功能的增加和移出中出现的错误。

形式错误。包括形式名的增加、删除、修改和定义中出现的错误。

数据库查询错误。包括数据查询修改中出现的错误, 这种修改会影响到操作类型、查询表格, 表格中的区域、搜索键和记录值。

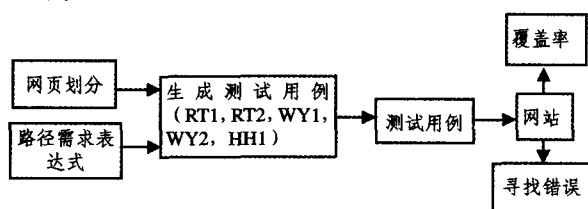


图3 实验图表

我们得到一汽车销售网站的大力支持, 并请一位资深的程序员在网站中插入错误, 我们对汽车销售网站进行测试, 测试过程见图3, 所得到的覆盖率和发现错误的数(以百分数计)见表1。从表1中我们得出基于网页划分的“白盒测试”技术发现的错误最多, 覆盖范围最大, 是网站应用程序测试技术中较为有效的测试方法。

表1 覆盖率与发现错误数(%)

测试方法	RT1	RT2	WY1	WY2	HH1
功能覆盖	96%	97%	95%	98%	99%
发现错误	67%	70%	73%	79%	81%

结论和今后工作 本文提出了对网站应用程序的基于网页划分测试技术以及把“白盒测试”技术与网页划分相结合的测试技术。基于网页划分测试技术弥补了“白盒测试”技术中测试用例生成难和生成不充分的问题。把两种技术结合起来形成的基于网页划分的“白盒测试”技术是一种更为有效的测

试方法。

网站应用程序类似于并行程序, 网站应用程序中存在不确定性。首先, 多用户同时点击相同网页, 用户需求之间存在竞争, 用户需求提出的次序影响网站响应用户需求的行为, 例如: 多名旅客同时在网上预定某旅馆的同一房间, 旅客之间存在竞争; 其次, 网站针对每个用户需求有一过程进行响应, 这些过程分享网站中的共享资源从而产生不确定性。对于具有不确定性的并行程序测试已有许多测试方法提出, 这些方法主要有: 不确定测试, 确定测试, 基于前缀测试。不确定测试是并行程序在同一输入条件下执行多次, 并检查每次测试结果, 这种方法的缺点是可能并行程序中的某些同步序列被测试多次, 而某些同步序列一次也没被测试。确定测试是从并行程序中挑选出一组同步序列进行测试, 这种方法能对所挑选的同步序列进行完全测试, 但对整个并行程序没能进行完全测试, 同时确定测试方法与不确定测试方法相比, 前者较难生成测试用例。基于前缀测试是从并行程序中挑选出一组同步序列进行确定测试, 然后进行不确定测试, 这种方法把不确定测试和确定测试两种方法结合起来, 动态地对并行程序进行测试, 这种方法比前两种方法更有效。本文提出的基于网页划分的测试方法类似于不确定测试, “白盒测试”方法类似于确定测试, 基于网页划分的“白盒测试”方法类似于基于前缀测试。随着网站建设的发展, 网站应用程序越来越复杂, 针对复杂的多同步执行的网站应用程序的更有效的测试方法和测试工具的研究是我们今后工作的主要方向。

#### 参考文献

- 1 Apache-Organization. Apache http server version 2.0 documentation, <http://httpd.apache.org/docs-2.0/>, 2004
- 2 Beizer B. Software Testing Techniques. New York: Van Nostrand Reinhold, 1990
- 3 Benedikt M, Freire J, Godefroid P. VeriWeb: Automatically Testing Dynamic Web Sites. In: Proc. 11th Int'l WWW Conf, May 2002
- 4 Binder R. Testing Object-oriented Systems. Addison Wesley, 2000
- 5 Carver R, Tai K. Deterministic Execution Testing of Concurrent Ada Programs. In: Proc. Conf Tri-Ada, Jan 1989. 528~544
- 6 Chays D, Dan S, Frankl P, et al. A Framework for Testing Database Applications. In: Proc. Int'l Symp Software Testing and Analysis, Aug 2000. 147~157
- 7 Conallen J. Building Web Applications with UML. Addison-Wesley, 2000
- 8 DiLucca G, Fasolino A, Faralli F, et al. Testing Web Applications. In: Proc. Int'l Conf Software Maintenance, Nov 2002. 310~319
- 9 Dickinson W, Leon D, Podgurski A. Finding Failures by Cluster Analysis of Execution Profiles. In: Proc. Int'l Conf Software Eng, May 2001. 339~348
- 10 Liu C, Kung D, Hsia P, et al. Structural Testing of Web Applications. In: Proc. 11th IEEE Int'l Symp Software Reliability Eng, Oct 2000. 84~96
- 11 Elbaum S, Malishevsky A G, Rothermel G. Test Case Prioritization: A Family of Empirical Studies. IEEE Trans Software Eng, Feb 2002, 28(2): 159~182
- 12 Empirix. Web Testing Solutions. <http://www.empirix.com/Empirix/Web+Test+Monitoring/Testing+Solutions/>, 2004
- 13 Harrold M, Gupta R, Soffa M. A Methodology for Controlling the Size of a Test Suite. ACM Trans Software Eng and Methodology, July 1993, 2(3): 270~285
- 14 Heatt E, Mee R. Going Faster: Testing the Web Application. IEEE Software, Mar 2002. 60~65
- 15 Hutchins M, Foster H, Goradia T, et al. Experiments on the Effectiveness of Dataflow and Controlflow-based Test Adequacy Criteria. In: Proc. Int'l Conf Software Eng, May 1994. 191~200
- 16 Software Research Inc. eValid. <http://www.soft.com/eValid/>, 2004
- 17 Kirda E, Jazayeri M, Kerer C, et al. Experiences in Engineering

- Flexible Web Services. IEEE MultiMedia, Jan 2001, 8(1): 58~65
- 18 Koppol P, Tai K. An Incremental Approach to Structural Testing of Concurrent Software. In: Proc. Int'l Symp Software Testing and Analysis, Jan 1996. 14~23
  - 19 Lee S, Offutt J. Generating Test Cases for XML-based Web Component Interactions Using Mutation Analysis. In: Proc. 12th IEEE Int'l Symp. Software Reliability Eng, Nov 2001. 200~209
  - 20 Lee T. World Wide Web Consortium. <http://www.w3.org/>, 2004
  - 21 Ricca F, Tonella P. Analysis and Testing of Web Applications. In: Proc. Int'l Conf Software Eng, May 2001. 25~34
  - 22 Macromedia Inc. Developing ColdFusion MX Applications. 2003. <http://www.macromedia.com/>
  - 23 Manley S, Seltzer M. Web Facts and Fantasy. In: Proc. 1997 Usenix Symp. Internet Technologies and Systems, 1997
  - 24 Nikora A, Munson J. Software Evolution and the Fault Process. In: Proc. 23rd Ann Software Eng Workshop, 1998
  - 25 Business Internet Group of San Francisco. The BIG-SF Report on Government Web Application Integrity. [http://www.tealeaf.com/downloads/news/analyst\\_report/BIG-SF\\_Report\\_Gov\\_2003-05.pdf](http://www.tealeaf.com/downloads/news/analyst_report/BIG-SF_Report_Gov_2003-05.pdf), 2004
  - 26 Business Internet Group of San Francisco. The Black Friday Report on Web Application Integrity. [http://www.tealeaf.com/downloads/news/analyst\\_report/BIG-SF\\_BlackFriday\\_Report.pdf](http://www.tealeaf.com/downloads/news/analyst_report/BIG-SF_BlackFriday_Report.pdf), 2004
  - 27 Parasoft. WebKing. <http://www.parasoft.com/jsp/products>, 2004
  - 28 Pressman R. Software Engineering: A Practitioner's Approach. Fifthed; McGraw-Hill, 2001
  - 29 Rational-Corporation. Rational Testing Robot. <http://www.rational.com/products/robot/>, 2004
  - 30 Testing a Website: Best practices. <http://www.reveregroup.com>, 2004
  - 31 Elbaum S, Karre S, Rothermel G. Improving Web Application Testing with User Session Data. In: Proc. Int'l Conf Software Eng, May 2003. 49~59
  - 32 Tai K C, Carver R H. A Specification-based Methodology for Testing Concurrent Programs. In: Proc. Fifth European Software Eng Conf, Sept 1995. 154~172
  - 33 Taylor R, Levine D, Kelly C. Structural Testing of Concurrent Programs. IEEE Trans Software Eng, 1992, 18(3): 206~215
  - 34 Tilley S, Shihong H. Evaluating the Reverse Engineering Capabilities of Web Tools for Understanding Site Content and Structure: A Case Study. In: Proc. Int'l Conf Software Eng, May 2001. 514~523
  - 35 Tzay J, Huang J, Wang F, et al. Constructing an Object-oriented Architecture for Web Application Testing. J Information Science and Eng, Jan 2002. 59~84

(上接第 40 页)

早期的 Internet 不具备拥塞控制机制, 从而产生“拥塞崩溃”。1989 年 10 月, Internet 发生了第一次拥塞崩溃, 导致从 LBL(Lawrence Berkeley Lab.) 到 UC Berkeley 的数据吞吐率从 32kbps 下降到 40bps。1988 年, Jacobson 基于网络包的守恒原理, 即“流入网络的包=流出网络的包”, 提出了端到端的 TCP 拥塞控制算法。然而仅有端到端的 TCP 拥塞控制算法, 还不能很好地解决 Internet 的拥塞问题。鉴于网络中间节点(路由器)具有及时检测拥塞是否发生的能力, 1993 年 Floyd 提出了一个用于网络中间节点的拥塞控制方法: RED 算法(Random Early Detection)。1998 年, IETF 在 RFC 2309 中建议在路由器中使用主动队列管理(Active Queue Management, 以下简称 AQM)算法, 并推荐 RED 算法为 AQM 的唯一候选算法。AQM 通过与 TCP 端到端拥塞控制算法相配合, 随机丢弃或标记欲进入路由器缓存队列的 IP 包, 从而实现拥塞控制。

Internet 业务流量由响应业务流量(即 TCP 长流 long lived TCP flows)和非响应业务流量(即 TCP 短流 short lived TCP flows 和 UDP 流)构成。由于非响应业务流量约占 Internet 业务流量的 70%~80%, 且不能响应 RED 算法的丢包信号, 因此本文提出的排队系统, 可用于评价 RED 算法在非响应业务流量下性能, 即分析 RED 算法的丢包率、链路利用率、队列长度的均值和方差等评价指标。详细的讨论请见我们已录用的论文(文[20])。

### 参 考 文 献

- 1 Kleinrock L. Queueing Systems. Volume I: Theory. New York: Wiley Interscience, 1975
- 2 Haight F A. Queueing with Balking. Biometrika, 1957, 44(3/4): 360~369
- 3 Haight F A. Queueing with Balking II. Biometrika, 1960, 47(3/4): 285~296
- 4 Ancker C J, Gafarian A V. Some Queueing Problems with Balking and Reneging, I. Operations Research, 1963, 11(1): 88~100
- 5 Ancker C J, Gafarian A V. Some Queueing Problems with Balking and Reneging, II. Operations Research, 1963, 11(6): 928~937
- 6 Rao S S. Queueing Models with Balking, Reneging, and Interruptions. Operations Research, 1965, 13(4): 596~608
- 7 Dreik A, Woolford D G. A Preemptive Priority Queue with Balking. European Journal of Operational Research, 2005, 164(2): 387~401
- 8 Younsuk K, Kiseon K. Loss Probability Behavior of Pareto/M/1/K Queue. IEEE Communication Letters, 2003, 7(1): 39~41
- 9 Cao J, Cleveland W S, Lin D, et al. The Effect of Statistical Multiplexing on the Long-Rang Dependence of Internet Packet Traffic; [Technical Report]. Bell Labs, 2001
- 10 Finch P D. Balking in the queueing system GI/M/1. Acta Math Acad Sci Hungar. 1959, 10: 241~247
- 11 Cox D R. The analysis of non-Markovian stochastic processes by the inclusion of supplementary variables. Proc. Cambridge Philos Soc., 1955, (51): 433~441
- 12 Hokstad P. The G/M/m Queue with Finite Waiting Room. Journal of Applied Probability, 1975, 12(4): 779~792
- 13 Vijaya P L, Gupta U C. Analysis of finite-buffer multi-server queues with group arrivals:  $G^X/M/c/N$ . Queueing System, 2000, 36(1/3): 125~140
- 14 Vladimir V K. Mathematical Methods in Queueing Theory. Dordrecht: Kluwer Academic, 1994
- 15 Cao J, Cleveland W S, Lin Dong, et al. On the nonstationarity of Internet traffic. In: Proc. SIGMETRICS'2001, ACM, 2001, 29(1): 102~112
- 16 Tudjarov A, Temkov D, Janevski T, et al. Empirical modeling of Internet traffic at middle-level burstiness. In: Proc. 12th IEEE Mediterranean, IEEE, 2004. 535~538
- 17 汪浩, 李晓明. 一个具有随机丢弃分组机制的 GI/M/1/N 排队系统. 通信学报, 2006, 27(1): 14~20
- 18 汪浩. 一个具有随机丢弃分组机制且分组成批到达的  $G^X/M/1/N$  排队系统. 数学的实践与认识, 2005, 35(9): 113~120
- 19 汪浩, 李晓明, 严伟. 一个具有阻行机制的成批到达排队系统  $G^X/M/1/N$ . 北京大学学报, 2006, 42(3): 406~411
- 20 汪浩, 严伟. 基于自相似聚合业务流量的 AQM 算法性能评价. 软件学报(已录用)