

# 一种基于 SRLG 分离的双路由选择策略

孙 晨,白显毅

(江南计算技术研究所,江苏 无锡 214083)

**摘 要:**在计算机通信网中,链路故障往往会导致大量的连接中断,造成严重的损失。因此,为业务的连接请求预先分配工作与备用双路由是提高网络抗毁性的重要途径,而保证工作路由与备用路由 SRLG(共享风险链路组)分离又是选路过程中的一个关键问题。文中列出了几种传统的 SRLG 分离双路由选择策略,针对它们的不足,提出了一种新的策略,并构建拓扑模型进行了仿真验证,最后,基于结果的理论分析对其作出了改进。与传统策略相比,新策略算法简单灵活,具有一定的实用性。

**关键词:**共享风险链路组;双路由;可变链路权重

**中图分类号:** TN915.08

**文献标识码:** A

**文章编号:** 1673-629X(2009)12-0131-04

## A Strategy of Double Routes Selecting Based on SRLG - Disjoint

SUN Chen, BAI Xian-yi

(Jiangnan Institute of Computing Technology, Wuxi 214083, China)

**Abstract:** In computer communication networks, failures of links usually cut off a mass of connections, and then serious loss would be caused. So, pre-distributing double routes as working route and backup route for a connection request is an important approach to improve survivability of networks, and one of the key issues is to make sure the two routes are SRLG(Shared Risk Link Group)-disjoint. Several traditional strategies of double routes selecting based on SRLG-disjoint are presented, aiming at the disadvantages of them, a new strategy is put forward, simulation and validation for the strategy are presented based on topology model built in this paper, finally, some improvements of the strategy are made based on theoretical analysis of the result. Compared with traditional strategies, the new strategy is practical to a certain extent with simple and flexible algorithm.

**Key words:** shared risk link group; double routes; variable link weight

## 0 引 言

计算机通信网络的可靠性问题十分复杂<sup>[1-3]</sup>,抗毁性<sup>[4]</sup>是其中尤为重要的一点。当为某一连接请求分配资源时,为其选择工作、备用双路由是提高网络抗毁性的重要途径。目前的网络中,一根光缆常常包含多根光纤,当一根光纤链路发生故障,极有可能是因为一整根光缆发生了断裂。即网络链路存在一定的故障相关性,这种故障相关性可以用 SRLG<sup>[5]</sup>的概念来描述。因此,为连接请求分配的工作、备用路由不宜包含处于同一 SRLG 的链路。而基于 SRLG 分离的双路由选择的最简单方法是:当选好工作路由以后,将工作路由经过的链路以及这些链路所处的 SRLG 中所有的链路全部删除,然后再进行备用路由的选择。这种算法称为工作路由由优先(APF<sup>[6]</sup>, Active Path First)算法。这种

方法较为简单,且保证了工作路由与备用路由的 SRLG 分离,但存在一个明显缺点,即容易造成备用路由无法获取的情况,如图 1 所示。以从 0 到 5 号节点的连接请求为例来对这种情况进行描述。

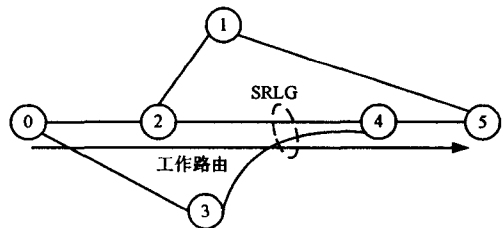


图 1 APF 算法导致的问题

从图 1 可以看出,在删除工作路由经过的 SRLG 中所有链路(即链路 0-2,2-4,3-4,4-5)之后,0 到 5 号节点之间便不存在任何路由。但实际上,0 和 5 号节点之间可以找到两条 SRLG 分离的路由,即 0-3-4-5 和 0-2-1-5。因此,这是由 APF 算法的缺陷所导致的“算法陷阱<sup>[6]</sup>”。解决 APF 算法陷阱问题的传统策略主要包括:(1)基于前 K 条最短路由<sup>[7]</sup>

收稿日期:2009-03-30;修回日期:2009-06-04

作者简介:孙 晨(1981-),男,河北蠡县人,硕士研究生,研究实习员,研究方向为网络技术。

(KSP)的双路由选择策略;(2)基于 SRLG 计入权重法<sup>[8]</sup>(WSRLG)的双路由选择策略。这两种算法有效性高,但缺点在于算法复杂度较高,且容易出现由于  $K$  值确定不当而造成浪费计算量的情况。

## 1 基于可变链路权重 SRLG 分离的双路由选择策略

基于已有算法的缺点,提出一种基于链路可变权重 SRLG 分离的双路由选择策略。基于可变链路权重 SRLG 分离的双路由选择 (VLW - SDDPS: Variable Link Weight SRLG Disjoint Double Paths Selection)策略本质上是一种基于 Dijkstra 算法<sup>[9]</sup>的双路由选择策略。这种策略的设计基于四个原则:

(1)SRLG 分离;(2)选择的路由要尽量趋于最优;(3)避免由于盲目地删除链路而引起的请求阻塞(算法陷阱);(4)尽量减少计算量。这种策略的原理如图 2 所示。

图 2 给出了一个 8 节点的网络拓扑结构,以 0-7 号节点的连接请求为例,将这种算法的思想描述如下:

(1)对于每条链路,标记出它所属的 SRLG 集合。其中链路 1-2、1-5、1-6 同处于 SRLG<sup>7</sup>, 0-3、2-6 同处于 SRLG<sup>8</sup>, 4-7、5-7 同处于 SRLG<sup>9</sup>,其余的链路分别标记为 SRLG<sup>1</sup> - SRLG<sup>6</sup>,并为每一个 SRLG 建立一个列表。

(2)利用 Dijkstra 算法选择节点 0 到节点 7 之间的最短路由(0-1-5-7),然后将路由所经过链路拆开,分别放入相应的 SRLG 列表中。

(3)将最短路由上的所有链路的权重分别乘以权重系数  $\alpha$  ( $\alpha > 1$ ),然后更新网络拓扑信息。

(4)再次利用 Dijkstra 算法选择一条当前节点 0 到节点 7 之间的最短路由(0-2-4-7),并将路由所经过的链路拆开放入相应的 SRLG 列表中。

(5)如果本次选择的路由上的链路发现之前已有其它链路(包括本身)与其共处在一个 SRLG 列表中,则将这些链路的权重乘以权重系数  $\alpha$ ,然后更新网络拓扑信息。

(6)将当前选择出的所有已选路由进行比较,如果发现有两条 SRLG 分离的路由,则报告选路成功,宣告算法结束;如果发现超过了算法约束时间( $T_{\max}$ , 用来控制算法运行时间),则宣告算法结束并报告阻塞;否则转到(4)。

相对基于 KSP 算法以及基于 SRLG 计入权重法

的双路由选择策略,此策略的优点是算法复杂度低,计算效率较高。

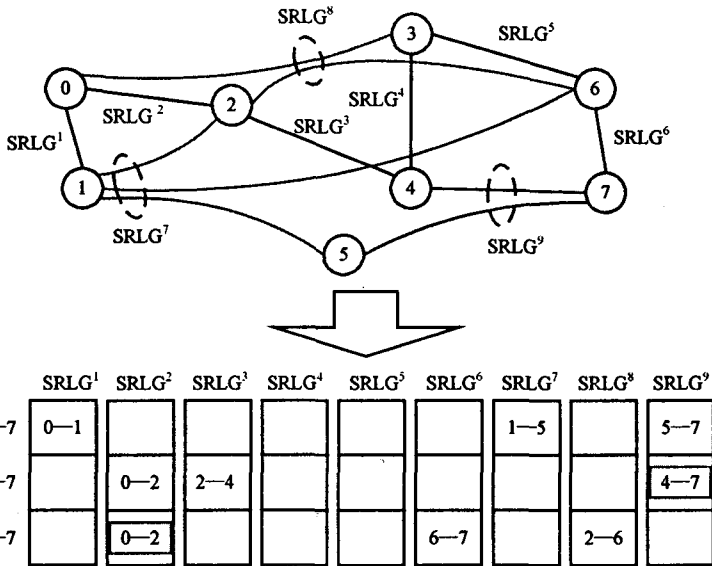
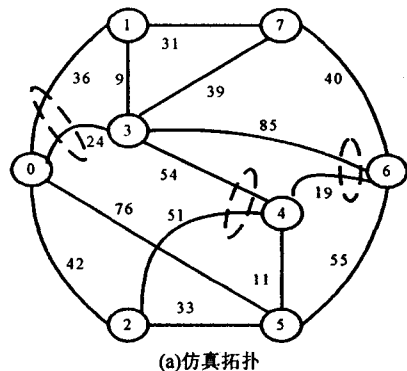


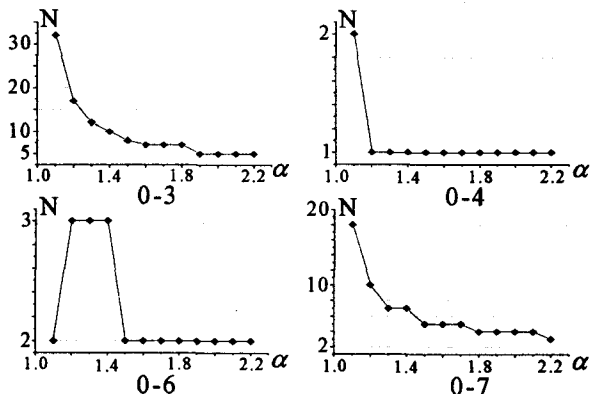
图 2 VLW-SDDPS 策略

## 2 算法仿真及结果分析

首先,构造一个 8 节点的网络拓扑,如图 3(a)所示。图中被虚线包含的链路处于同一个 SRLG。以选出 SRLG 分离的双路由所需要的循环次数  $N$  (第二次选路时开始计数)作为仿真的结果。 $N$  越小则说明算



(a) 仿真拓扑



(b) 仿真结果

图 3 算法仿真

法的效率越高。

在仿真的过程中,将 0 号节点作为源节点,3、4、6、7 号节点分别作为目的节点;同时,将  $\alpha$  的取值范围值定在 [1.1, 2.2]。仿真中所用到的编程语言为 C 语言。

利用仿真结果绘制得到  $\alpha - N$  的关系图,如图 3(b) 所示。由图 3(b) 可以看出,  $\alpha$  的取值与  $N$  有着密切的关系,一般来说,  $\alpha$  的取值越大,其迭代次数越少,这一点是合理的,因为  $\alpha$  的值越大,被重复遍历的 SRLG 中链路的权重增加就会越快,从而使得利用 Dijkstra 算法计算出的路由变化会更频繁,因此反复冗余遍历同一链路的次数就会越少,选中 SRLG 分离的路由就会越快。需要说明的是,在为 0 - 6 号请求选路的过程中,当  $\alpha$  为 1.1 时,  $N$  的值要小于  $\alpha$  为 1.2、1.3、1.4 的值,这一点是  $\alpha$  与拓扑结构共同造成的(在后文将详细说明),但随着  $\alpha$  的继续增大,  $N$  的值最终仍呈现缩减的趋势。

3 路由遍历完全性的约束条件

在上一节中,对基于可变链路权重的 SRLG 分离双路由选择算法进行了描述和仿真,并得出论断:  $\alpha$  的取值越大,则得到结果的过程越快。然而有可能存在这样一种情况:当  $\alpha$  值取定,算法开始运行之后,即使链路权重在不断变化,也不能够遍历所有可能的路由。若这种情况出现,将产生一种新的算法陷阱。图 3(b) 中为 0 到 6 号节点连接请求选路的仿真图在某种程度上暗示了这种可能性的存在,事实上,这种情况是确实存在的,这种情况可定义为“路由遍历非完全性”。下面利用仿真对这种情况进行说明。首先,构造一个如图 4(a) 所示的拓扑图。

在仿真的过程中,将 0、4 号节点分别设为源、宿节点。  $\alpha$  的取值范围是 [1.1, 2.2]。另外,在算法中加入约束时间  $T_{max}$ ,一旦算法发现其计算时间超过了约束时间  $T_{max}$ ,则认为网络中不存在满足条件的双路由,继而强行终止搜索,报告阻塞。在仿真中,规定  $T_{max}$  的值为 20。仿真结果仍然用  $\alpha - N$  的关系图来表示,如图 4(b) 所示。可以看出,当  $\alpha$  的取值在 [1.1, 2.0] 的范围内变化时,迭代次数  $N$  随着  $\alpha$  的增大呈减小的趋势。这一特点与前一个仿真类似,与理论分析吻合。但是在  $\alpha > 2.0$  之后,算法突然报告阻塞,当分别增加了  $T_{max}$  和  $\alpha$  值之后再进行测试,发现仍然阻塞。从这一点,可以推测出这种情况应和  $\alpha$  的取值有关系。经过对仿真过程的分析,发现上述推测是正确的。当  $\alpha$  的值超过 2.0 之后,算法会在 0 - 1 - 2 - 3 - 4、0 - 5 - 2 - 6 - 4 以及

0 - 7 - 8 - 4 三条路由之间循环遍历,直到循环次数上限  $T_{max}$ 。而由于这三条路由两两之间都是 SRLG 相关的,因此均不满足条件,所以算法最终报告阻塞。显而易见,这是由于  $\alpha$  的选取过大引起的,  $\alpha$  取值过大导致了 0 - 5 - 2 - 3 - 4 这条路由的权重始终无法处于所有路由中的最小位置,因此始终无法被算法选择到。从这一个仿真可以得到这样的一个结论:  $\alpha$  的值并不是越大越好。  $\alpha$  的取值和网络拓扑结构对算法的路由遍历非完全性存在着一个联合制约的关系。

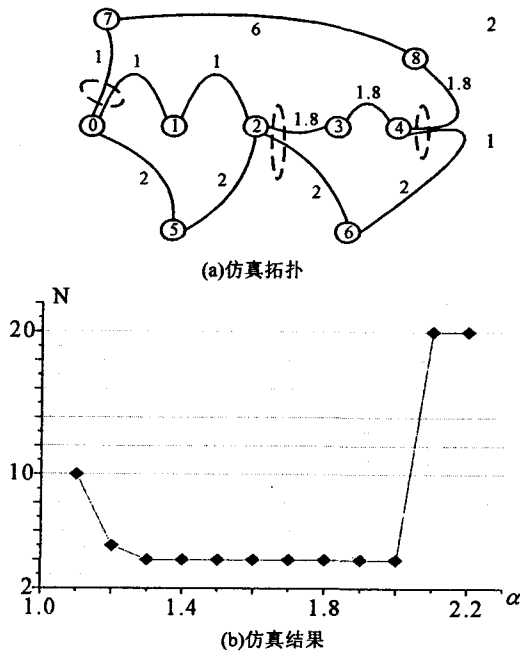


图 4 路由遍历非完全性仿真

下面对这种约束关系作一个更为详细的分析。为了更好地说明问题,在此构造出一个网络路由分解的抽象模型,如图 5 所示。

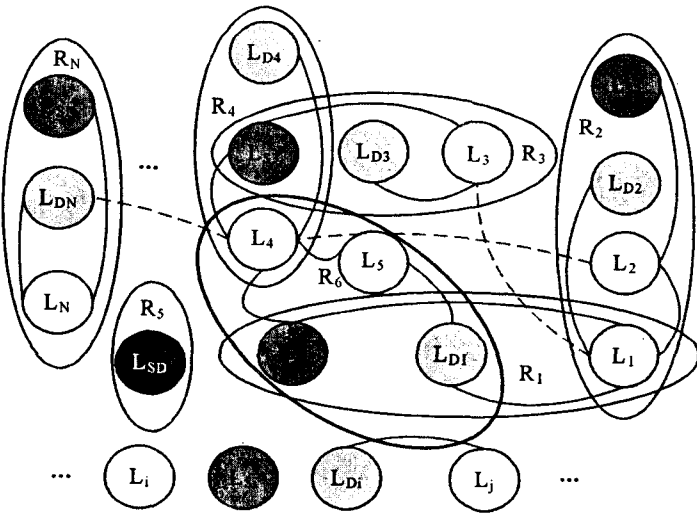


图 5 网络路由分解图

对照图 5,可以作这样的定义:图中的圆圈代表了

整个网络拓扑的每个链路,普通的链路标号为  $L_i (i = 1, 2, 3, \dots, N \dots)$ 。

另外有三种比较特殊的链路:第一种是连接源节点和目的节点的链路,标号为  $L_{SD}$ ;第二种是一端连接源节点的链路,标号为  $L_{Si} (i = 1, 2, 3, \dots, N \dots)$ ;第三种是一端连接目的节点的链路,标号为  $L_{Di} (i = 1, 2, 3, \dots, N \dots)$ 。这些链路被包含在相应的源节点到目的节点的路由里。每一个由源节点到目的节点的完整路由都用一个椭圆区域来表示,标号为  $R_i (i = 1, 2, 3, \dots, N \dots)$ 。图中的被虚线连接的链路表示它们同处于一个 SRLG 中。

在这种情况下,只考虑  $R_3$ 、 $R_4$  以及  $R_6$ 。若开始  $R_3$ 、 $R_4$  的代价都小于  $R_6$ ,那么利用 Dijkstra 最短路由算法计算之后,不论得到哪一条路由,在对其链路权重加权的同时, $R_6$  的权重也会随之增加,这样就有可能使得  $R_6$  的权重一直无法处于三条路由中最小的位置上,算法也就无法对其进行遍历,这一点是网络拓扑本身的原因;另外,还有一方面是  $\alpha$  取值的原因(如图 4(b))。这两个问题构成了对算法中路由遍历完全性的约束条件。然而网络拓扑一旦给定便无法改变,所能改变的只有  $\alpha$ 。于是, $\alpha$  的恰当选取是唯一能够提高算法有效性的方法。

#### 4 基于约束条件的算法改进

经过上一节的论述可以看出, $\alpha$  的取值十分重要。显然, $\alpha$  的值必须大于 1;此外, $\alpha$  的取值不宜过小,如果  $\alpha$  的值选取过小,将可能造成某条路由被多次重复遍历,导致计算量的浪费;同时, $\alpha$  的取值也不宜过大。

另外,对于算法约束时间  $T_{\max}$  的选取也是一个关键问题。 $T_{\max}$  可以根据系统时钟确定,也可以根据算法迭代次数确定。与权重系数  $\alpha$  一样, $T_{\max}$  的确定也需要从多方面加以考虑:首先,如果取值过小,将会造成算法的“误判”,即网络中确实存在两条(以上)SRLG 分离的路由,但是由于某些原因(如  $\alpha$  取值过小造成循环次数增加),算法还没有遍历到这些路由便被终止;其次,如果这个时间的选取过大,则会拖延连接建立时间。

由上述分析可知,在基于链路可变权重的 SRLG 分离双路由选择策略中,权重系数  $\alpha$  与算法约束时间  $T_{\max}$  之间存在着很强的制约关系。基于这种性质,可采用  $\alpha$  循环缩减算法。这种算法基于 VLW - SDDPS 策

略,图 6 给出了算法的流程。

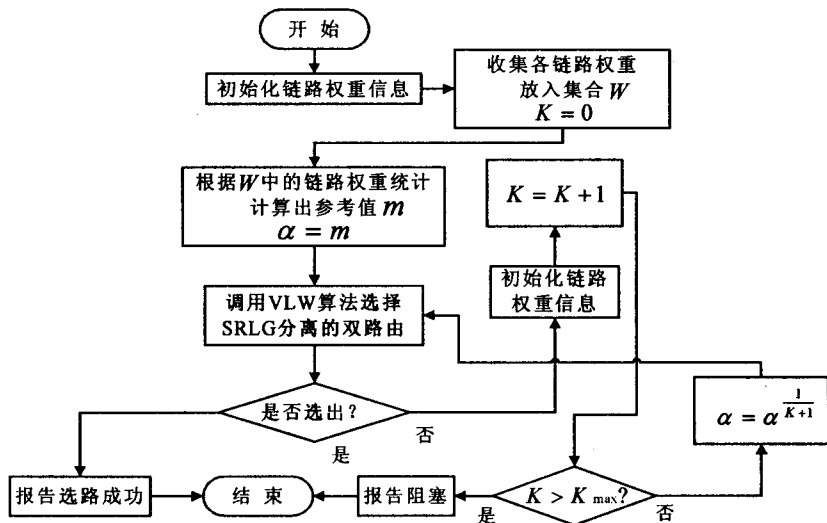


图 6 基于 VLW - SDDPS 策略的循环缩减算法

其中, $m$  为由各链路权重统计计算得出的  $\alpha$  初值, $K$  为循环测试次数, $K_{\max}$  为最大循环测试次数,即循环  $K_{\max}$  次之后仍未找到满足要求的双路由,则报告阻塞。 $m$  值的选取可以依据(1) 式来确定。

$$m = \xi \left( \frac{\sum_{i=0}^{N-1} \frac{|w_i - w_{\text{avg}}|}{w_{\text{avg}}}}{N} + 1 \right) \quad (\xi > 1) \quad (1)$$

式中, $w_i$  为第  $i$  条链路的权重, $w_{\text{avg}}$  为所有链路的平均权重, $\xi$  为权重系数参数,其取值应大于 1,这样就保证了  $m$  大于 1。确定  $m$  值的方法依据了这种思想:如果各链路的权重差别较大,为了减少冗余循环加权的次数,则希望  $\alpha$  的值大一些;如果各链路的权重差别较小,在保证较少的冗余循环加权的同时,为了尽量满足算法约束条件(即尽量不出现漏选路由的情况),则希望  $\alpha$  的值小一些。这里确定的  $m$  值是一个估算值,并不一定能够完全避免由于  $\alpha$  的取值不当而引起的阻塞。因此,在算法流程中加入了循环,如果某一次利用 VLW - SDDPS 策略无法选择两条 RLG 分离的路由,则  $K$  值递增,并对  $\alpha$  取  $1/(K+1)$  次方,这样在  $\alpha$  值减小后继续利用 VLW 算法寻找 SRLG 分离的路由,最终找到满足条件的双路由。然而,与 VLW - SDDPS 策略中的约束时间  $T_{\max}$  一样,这种循环也需要引入一个门限时间进行控制。否则若网络中确实不存在满足条件的双路由,整个算法则会发生死锁。在此,可以利用  $K_{\max}$  作为时间门限。即算法如果重复测试  $K_{\max}$  次(从第二次开始)之后仍然无法找到 SRLG 分离的链路,则宣告请求阻塞。经测试,对于图 4(a) 给出的拓扑,只要  $K_{\max}$  的值足够大,则能够避免算法陷阱的出现。在此给出算法不落入陷阱的情况下, $\xi$  与  $K_{\max}$  之间的约束

(下转第 149 页)

CheckBox,下拉框等控件的操作。

```
录制开始时间: 2009-2-26 10:19:59
任务名称: 文件压缩任务
您的操作是:
列表框选择的内容--test.txt
选择工具栏“添加”按钮
弹出“压缩文件名和参数”对话框;
“压缩后删除源文件”--设为--ON
“压缩方式”选择“存储”
“压缩文件格式”--ZIP--设为--ON
在“压缩文件名和参数”对话框内点击“确定”按钮
**点击按钮--确定--后,各控件默认内容为
【压缩文件名】的内容为【test.zip】
【压缩方式】的内容为【存储】
【压缩分卷大小,字节】的内容为【】
【更新方式】的内容为【添加并替换文件】
点击“stop”按钮
录制结束时间: 2009-2-26 10:20:33
```

图6 WinRAR 压缩设置记录结果

通过实验,两实例录制的结果符合图1中测试文档中的规范要求,完全可以直接用于测试文档的编写,可以复制需要的部分或全部内容,快速客观完成测试文档中测试步骤的填写工作,对于GUI测试文档的编写提供了极大的方便。

### 3 结束语

编写测试文档是测试过程中的一个重要的组成部分,目前市场上测试文档辅助工具的欠缺促成我们的研究。所实现的 AutoDoc 工具作为 GUI 软件测试文档编写的辅助工具,可使文档编写更加快捷方便、客观准确,可较好地解决测试人员的现实需求。下一步的研究将针对马里兰大学 Memon 博士的 GUI 软件逆工

程以及覆盖标准的基础上<sup>[7,8]</sup>,设计出手动测试和自动测试相结合的测试模型,使软件更加贴近实际应用。

### 参考文献:

- [1] 王泓. 软件测试文件的编制[J]. 电子产品可靠性与环境试验, 2005, 44(5): 63-64.
- [2] 李菲, 晏海华, 郝建营. 基于 XML 的通用文档生成方法研究[J]. 计算机工程与设计, 2007, 28(20): 4836-4838.
- [3] Steven J, Chanda P, Fleck B, et al. A Capture/Replay Tool for Observation-Based Testing[M]. [s.l.]: ACM, 2000.
- [4] 戴春达, 符红光. Win32 中钩子的实现技术及应用[J]. 计算机应用, 2002, 22(8): 72-74.
- [5] 陈浩, 邹北骥, 江文, 等. 基于消息机制的自动化黑盒测试方法研究[J]. 计算机应用研究, 2004(10): 33-35.
- [6] Petzold C. Windows 程序设计[M]. 北京博彦科技发展有限公司译. 北京: 北京大学出版社, 1999.
- [7] Memon A M. Reverse Engineering of Graphical User Interface for Testing[C]// Proceedings of the 10th Working Conference on Reverse Engineering (WCRE'03). [s.l.]: IEEE, 2003.
- [8] Memon A M, Soffa M L, Pollack M E. Coverage criteria for GUI testing[C]// ESEC/FSE-9: Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering. New York, NY, USA: [s.n.], 2001: 256-267.

(上接第134页)

关系,见(2)式。

$$\xi < \frac{\alpha_{\text{gate}}^{K_{\max}+1}}{A} \quad (\xi > 1, K_{\max} \text{ 为正整数}) \quad (2)$$

其中,  $A$  代表了(1)式括号中的算式,  $\alpha_{\text{gate}}$  是  $\alpha$  的取值门限, 当网络拓扑一定时,  $A$  和  $\alpha_{\text{gate}}$  为大于1的常量; 如果  $\alpha$  的取值不超过此门限, 算法就不会落入陷阱。经测试同时可知,  $\xi$  取值不宜过大, 否则会增加算法的运行时间。一般来说, 建议在链路代价数量级差别较小的网络中,  $\xi$  取值只需比1稍大即可。

### 5 结束语

在网络信息量与日剧增的当代, 提高网络抗毁能力的重要性日益凸显。SRLG 分离的双路由选择策略为网络抗毁性问题提供了一个较为有效的解决方案。针对几种传统策略, 文章中提出的新策略简单灵活, 效率较高, 具有一定的实用价值。

### 参考文献:

- [1] 陈太一, 陈常嘉. 通信网的可靠性[J]. 电信科学, 1992, 8

(5): 18-25.

- [2] 冯海林. 网络系统中可靠性问题的研究[D]. 西安: 西安电子科技大学, 2004.
- [3] 张俊良. 复杂网络可靠性研究[D]. 大连: 大连理工大学, 2006.
- [4] 刘啸林. 网络抗毁性研究介绍[J]. 计算机应用与软件, 2007, 24(6): 135-137.
- [5] Mannie E. Generalized multi-protocol label switching (GMPLS) architecture[S]. IETF Network Working Group. Request for Comments: 3945 Standards Track, 2004.
- [6] Xu Dahai, Xiong Yizhi, Qiao Chunming, et al. Trap avoidance and protection schemes in networks with Shared Risk Link Groups[J]. IEEE Lightwave Technology, 2003, 21(11): 2683-2693.
- [7] Yen J. Finding the k shortest loopless paths in a network[J]. Management Science, 1971, 17(11): 712-716.
- [8] Oki E, Matsuura N, Shiimoto K. A disjoint path selection scheme with shared risk link groups in GMPLS networks[J]. Communications Letters, IEEE, 2002, 6(9): 406-408.
- [9] 谢希仁. 计算机网络[M]. 第4版. 北京: 电子工业出版社, 2003: 434-435.