

# 基于带宽和跳数的流量工程动态路由选择算法研究

薛希俊,孙雨耕,刘振肖

(天津大学电气自动化与能源工程学院,天津 300072)

**摘 要:** 随着网络规模和网络硬件的快速发展,支持流量工程的路由器解决方案已经成为一种趋势,而流量工程动态路由选择算法则是流量工程方案的一个重要组成部分.本文分析了流量工程路由算法的研究现状,提出了以跳数和带宽作为量度的流量工程路由算法数学模型,并给出了路由算法规则.最后,通过计算机仿真验证了该算法的可行性和相对于只采用现行最短路径优先路由选择算法的优越性.

**关键词:** 流量工程;路由算法;带宽;跳数

**中图分类号:** TP393.01 **文献标识码:** A **文章编号:** 0372-2112(2002)02-0274-05

## Traffic Engineering Dynamic Routing Based on Bandwidth and Hops

XUE Xi-jun, SUN Yu-geng, LIU Zhen-xiao

(School of Electrical Engineering and Energy, Tianjin University, Tianjin 300072, China)

**Abstract:** With development of network scale and hardware, router solution project based on Traffic Engineering has been more and more popular. And in the Traffic Engineering solution project, Traffic Engineering dynamic routing is an important part. So, we have analysed the current situation, and presented TE dynamic routing math model, which takes bandwidth and hops into account and presents routing rules. In the end, by the computer simulation, we validate the feasibility of the routing and advantage of the current shortest path priority routing.

**Key words:** traffic engineering; routing; bandwidth; hops

### 1 引言

在传统的路由器解决方案中,路由的选择是通过以跳数或延迟为量度的最短路径优先算法实现的<sup>[1,6]</sup>.用最短路径来转发流量的本质缺陷在于:(1)如果从一个源节点到目的节点的流量超过了最短路径的容量,最短路径将变得拥塞,但同时这两点之间可能有一条更长的路径没有被充分使用;(2)在来自不同源节点的最短路径在一条链路上重叠的情况下,如果通过该链路的总流量超过了该链路的容量,那么就会发生拥塞<sup>[3]</sup>.随着网络规模的增长,这种情况会变得越来越严重.克服这种缺陷的唯一措施是对网络实施流量工程.实施流量工程的目的是使ISP(Internet 服务供应商)可以对其网络中的业务量进行精确的控制,从而提高资源利用率,为用户提供更好的服务.当前世界上大型的网络设备供应商(如Cisco、Juniper 等公司)都在积极地研究和开发具有流量工程能力的路由设备<sup>[1]</sup>.但是到目前为止,国际上还没有成熟的流量工程解决方案,尤其是在流量工程的动态实现方面,图1描述的是一种最流行的流量工程动态解决方案,该方案将现行的动态路由选择和流量工程动态路由选择作为两个独立的模块.其基本思想是在运行现行动态路由选择算法的网络中,当业务流量对现行最短路径产生竞争时,我们采用阈值技术,适时地激

活流量工程动态路由选择算法,并根据路由结果建立一条或多条显式路径,将部分流量转移到这些路径上,从而缓解业务流量对链路的竞争,使流量在网络上更加均匀地分配,最终减小网络的拥塞概率.

现在,最有效的显式路径技术是MPLS(Multiprotocol Label Switching 多协议标记交换),采用MPLS技术生成的显式路径称为LSP(Label Switched Path 标记交换路)<sup>[3,4]</sup>.通常情况下,IP包根据现行动态路由选择模块得到的路由表进行转发;在特殊情况下,(如:网络中流量分布很不平衡),流量工程动态路由选择模块被激活,从而得到一条或多条LSP,并将部分IP流量通过LSP的转发表进行转发,以期最终优化流量在网络上的分布,提高网络的性能<sup>[3]</sup>.图2是应用流量工程之前,流量在网络上的分布情况.图3是应用流量工程,并生成两条LSP(0—3—6—9, 2—3—8—9)之后,流量在网络上的分布.从中可以看出流量工程优化了流量在网络上的分布,缓解了流量对同一链路的竞争,提高了网络的性能.

流量工程动态路由算法是流量工程路由模块中最重要的部分.针对MPLS网络,研究流量工程动态路由算法的约束条件和目标函数,从而建立算法的数学模型和算法规则对于动态实现基于MPLS的流量工程具有特别重要的意义.

下面几部分分别研究了流量工程动态路由选择算法的数学模型、算法规则,并且根据该规则提出了一种可行的流量工程路由选择算法,最后通过仿真验证了该算法的可行性、相对于当前网络协议(如 OSPF 或 ISIS)所采用的路由选择算法的优越性。

## 2 流量工程路由选择算法的数学模型

当前,自动进行流量工程的一个重要算法工具是基于约束的路由选择(Constraint-based routing)<sup>[4]</sup>。它是从服务质量路由选择(QoS routing)发展而来的。QoS 路由选择算法就是指一套协议和算法,它们能够选择同时满足多个约束并达到高网络吞吐率要求的路径<sup>[2]</sup>。基于约束路由选择算法则需要考虑更多的约束和策略<sup>[3,9]</sup>。本文研究的基于带宽和跳数的流量工程动态路由选择算法便属于基于约束路由选择算法的范畴。

研究表明,基于约束路由选择算法必须考虑网络的拓扑信息、流量的 QoS 要求、链路的资源供应和管理策略<sup>[1,3]</sup>。在本文中,用无向加权图  $G =$

$(V, E)$  表示网络,其中  $V$  为网络中所有路由器节点的集合,  $E$  表示路由器之间链路的集合。由于流量工程的目的是在现有网络上配置一些 LSP 以便调整网络上的流量分布,所以,在源节点  $s$  激活流量工程路由算法时,需要给出转移流量的带宽要求(即 LSP 的带宽要求)  $\text{req. } B$ 。链路的资源供应情况主要是指链路的可预留带宽,它可以表示为链路的二元加权值  $(b_{i,j}, b_{j,i})$ ,其中  $b_{i,j}$  表示链路  $(i, j)$  从  $i$  到  $j$  方向的可预留带宽,同理  $b_{j,i}$  表示链路  $(i, j)$  从  $j$  到  $i$  方向的可预留带宽。管理策略是指链路的颜色、优先级等问题,在动态路由算法中,只考虑链路的颜色。

在建立算法的目标函数之前,先简要介绍一下路径量度的选择问题<sup>[2,5]</sup>。

路径的量度主要有费用、跳数、带宽、可靠性、延迟和抖动。路由选择算法就是选择那些优化了一个或多个这些量度的路径。接下来,我们看一个定义。

**定义** 设  $d(i, j)$  表示链路  $(i, j)$  的量度,则对于任一路径  $P = (i, j, k, \dots, l, m)$ ,如果满足条件  $d(P) = d(i, j) + d(i, k) + \dots + d(l, m)$ ,则量度  $d$  是加法型的;如果满足条件  $d(P) = d(i, j) * d(i, k) * \dots * d(l, m)$ ,则  $d$  是乘法型的;如果满足条件  $d(P) = \min\{d(i, j), d(i, k), \dots, d(l, m)\}$ ,则  $d$  是凹型的。

根据这个定义,量度延迟、抖动、费用和跳数是加法型,可靠性是乘法型,带宽是凹型。

研究还得到定理:

**定理** 计算受两个或多个加法型与/或乘法型量度约束

的最优路径是 NP 完全的。即,用延迟、费用、抖动、跳数和损失概率率中的两个或多个作为量度,并同时使这些参数最优是 NP 完全问题。

据此,得到结论 1:

**结论 1** 流量工程动态路由算法一般只考虑带宽和跳数,原因如下:

- 由于一个流的延迟和抖动能够被收集的带宽和路径的跳数确定,所以如果需要,延迟和抖动约束可以映射到带宽和跳数约束上<sup>[2]</sup>。根据定理可知:延迟、费用、抖动、跳数和损失概率率这些量度中两个或多个不能同时作为量度,并使这些量度参数最优,否则算法是 NP 完全的。此外,因为一个流穿过的跳数越多,它消耗的资源越多,所以一条路径的跳数是非常重要的。例如,一条两跳的 1-Mb/s 流消耗的资源是一跳的 1-Mb/s 流消耗资源的两倍。因此,在加法型量度中,只选择跳数。

- 许多实时“应用”(如在线播放)对带宽要求非常严格。因此,带宽量度是一个非常重要的性能指标。

从上述分析可知,流量工程动态路由算法必须满足的 QoS 约束是可预留带宽和 LSP 的跳数约束,必须满足的管理策略约束是 LSP 的颜色约束。由此,构造算法的数学模型如下:

最小化:  $\text{hops}(\text{paths}(s, d)), \text{paths}(s, d) \subset T(s, d)$

约束条件为:

$$\left[ \begin{array}{ll} \text{hops}(p(s, d)) & H \\ \text{bandwidth}(p(s, d)) & B \\ \text{bandwidth}(\text{paths}(s, d)) & \text{req. } B \\ \text{num}(\text{paths}(s, d)) & N \\ \text{color}(p(s, d)) & \subset \text{COLOR} \end{array} \right. \quad \left[ \begin{array}{ll} p(s, d) & \text{paths}(s, d) \\ p(s, d) & \text{paths}(s, d) \\ \text{paths}(s, d) & \subset T(s, d) \\ \text{paths}(s, d) & \subset T(s, d) \\ p(s, d) & \text{paths}(s, d) \end{array} \right.$$

其中,  $s$  表示源节点,  $d$  表示目的节点,  $p(s, d)$  表示  $s$  到  $d$  的一条路径,  $T(s, d)$  表示  $s$  到  $d$  的所有路径的集合;  $\text{paths}(s, d)$  表示  $T(s, d)$  的一个子集,  $\text{hops}(\text{paths}(s, d))$  表示  $\text{paths}(s, d)$  路径集的链路数,由于一条路径的链路数等于其跳数,而且在单路径路由中,跳数是一个重要的量度,因此为统一起见,也可以称路径集的链路数为路径集的跳数;  $\text{bandwidth}(p(s, d))$  表示路径  $p(s, d)$  的可预留带宽;  $\text{num}(\text{paths}(s, d))$  表示组成路径集的路径条数。  $H, B, \text{req. } B, N$  都是常数,分别表示路径的最大长度(即最大跳数)、对链路的最低带宽要求、待转移流量对带宽的要求、所求路径的最多条数;  $\text{COLOR}$  表示允许的颜色集合,可以由网络管理员设定,  $\text{color}(p(s, d))$  表示组成路径  $p(s, d)$  的所有链路的颜色集合。

因为提高一个高带宽流量的平均吞吐量的技术是采用多路径传输,一条采用多路径传输的“会话应用”比单路径传输的“会话应用”能达到更高的性能,所以在该数学模型中,我们求解的是一个满足带宽约束的路径集,该路径集可能包括一条或多条路径<sup>[8]</sup>。此外,在得到的 LSP 路径中应该避免出现过长、过窄的路径。为此,我们增加了路径最大跳数约束和路径的最低带宽(从而也是链路的最低带宽)要求。

点对之间的流量以多路径传输时,会增加源节点和目的节点对流量调度的开销,因此,应该限制最终建立的 LSP 路径

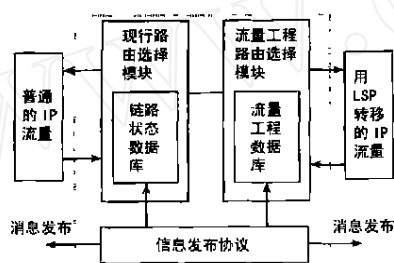


图1 流量工程在路由器中的结构示意图

条数.

最后一个约束是颜色约束,该约束的使用可以给网络管理员以更大的控制权,以期通过控制链路颜色属性来控制 LSP 所穿过的区域.

### 3 算法规则

对于上节提出的算法模型,由于求两点间的第  $K$  条最短路的时间复杂度是  $O(|V|^3)^{[10]}$ ,从而求解一组无带宽约束的由  $K$  条路径组成的最短路径集的时间复杂度是  $O(K|V|^3)$ ,考虑带宽约束后,  $K$  不再是一个常量,可以表示为:  $K=f(V, E)$ ,与此同时,最优解不一定是无带宽约束时的最短路径集,因此求解的时间复杂度便很高,甚至可能是 NP 问题.为此,我们首先研究了单路径模型的路由算法规则,在此基础上给出了路径集算法的较优路由规则.

**命题 1** 如果  $B = \text{req. } B$  条件成立,则上节提出的多路径模型转化为单路径模型:

最小化:  $\text{hops}(p(s, d)), p(s, d) = T(s, d)$

约束条件为:

$$\begin{cases} \text{hops}(p(s, d)) \leq H, & p(s, d) = T(s, d) \\ \text{bandwidth}(p(s, d)) \geq B, & p(s, d) = T(s, d) \\ \text{color}(p(s, d)) \in \text{COLOR}, & p(s, d) = T(s, d) \end{cases}$$

对于单路径模型,可以采用改进的 dijkstra 算法来求解<sup>[7]</sup>.

在下面的算法规则中,我们考虑的是链路,因此我们给出了命题 2.

**命题 2** 根据路径的可预留带宽定义:

$$\text{bandwidth}(p(s, d)) = \{\min(\text{bandwidth}(i, j)) | (i, j) \in p(s, d), p(s, d) = T(s, d)\}$$

和约束

$$\text{bandwidth}(p(s, d)) \geq B$$

得到:

$$\text{bandwidth}(i, j) \geq B, (i, j) \in p(s, d)$$

其中,  $\text{bandwidth}(i, j)$  表示链路  $(i, j)$  从  $i$  到  $j$  方向的可预留带宽.然后,将节点编号,并定义结构:

```
struct node . record
{
    int hops
    int pre . node
    boolean ontree
}
```

其中,  $\text{hops}$  表示从源节点  $s$  到该节点的最优路径的跳数;  $\text{pre . node}$  表示从源节点到该节点的最优路径上,该节点的相邻节点;  $\text{ontree}$  表示节点是否被处理过了,该变量可以取 TRUE 或 FALSE, TRUE 表示处理过了, FALSE 表示还没有处理.

单路径路由规则如下:

(1) 任一节点  $\text{node} \in V$ , 令  $\text{node . hops} = \infty$ ;

$\text{node . pre . node} = \text{null}$ ;

$\text{node . ontree} = \text{FALSE}$

(2) 设  $S$  为一集合, 令  $\text{node1} = s$ ,

$S = \emptyset, \text{node1 . ontree} = \text{TRUE}, \text{node1 . hops} = 0$ ;

(3) 当  $\text{node1} = d$  时, 对每一节点  $\text{node2} \in V$ , 做

若  $\text{bandwidth}(\text{node1}, \text{node2}) \geq B$

且  $\text{color}(\text{node1}, \text{node2}) \in \text{COLOR}$ , 则做

若  $\text{node2 . ontree} = \text{FALSE}$ , 则做

若  $\text{node2 . hops} > \text{node1 . hops} + 1$ ,

且  $\text{node1 . hops} + 1 = H$ , 则做

$S = S \cup \{\text{node2}\}$ ;

$\text{node2 . hops} = \text{node1 . hops} + 1$ ;

$\text{node2 . pre . node} = \text{node1}$ ;

}

否则做

若  $S = \emptyset$ , 则  $\text{node1} = d$ ;

否则{

从  $S$  中任取一值赋给  $\text{node1}$ , 即  $\text{node1} \in S$ ;

$S = S - \{\text{node1}\}$

}

}

}

}

}

(4) 从节点  $d$  开始, 根据结构  $\text{node . record}$  回溯得到一条路径  $p(s, d)$ .

当  $\text{req. } B > B$  时, 采用以下路由规则, 以期快速得到一组较优解. 多路径路由规则: 设  $\text{PATHS}(s, d)$  为  $s$  到  $d$  的一个路径集, 令  $\text{PATHS} = \emptyset$ , 则重复做:

若  $\text{num}(\text{PATHS}(s, d)) < N$ , 且

$\text{bandwidth}(\text{PATHS}(s, d)) < \text{req. } B$ , 则做

若应用单路径规则可以得到一个解  $p(s, d)$ , 则

$\{\text{PATHS}(s, d) = \text{PATHS}(s, d) \cup \{p(s, d)\}\}$ ;

对任一  $(i, j) \in p(s, d)$ , 做

$\text{bandwidth}(i, j) = \text{bandwidth}(i, j) -$

$\text{bandwidth}(p(s, d))$ ;

}

否则, 退出;

}

证明算法的正确性: 本文提出的单路径路由规则本质上是从 dijkstra 最短路算法改进而来的<sup>[10]</sup>. 不同点在于: 此处链路的距离是 1 (即以跳数为距离), 另外增加了带宽约束、颜色约束和路径跳数约束. 如果将每条链路表示为两条方向相反的有向边, 那么可以将网络表示为一个有向图, 每条有向边的权值表示相应链路传输方向的可预留带宽, 从而可知条件  $\text{bandwidth}(\text{node1}, \text{node2}) \geq B$  相当于对不满足带宽要求的链路传输方向进行了剪除操作,  $\text{color}(\text{node1}, \text{node2}) \in \text{COLOR}$  相当于对不满足颜色要求的链路方向进行了剪除操作. 因此, 整个路由规则相当于剪除不满足约束的有向边之后, 用 dijkstra 最短路算法求解最佳路径. 其时间复杂度与 dijkstra 算法相当. 因此, 是正确的. 单路径路由规则正确时, 多路径路由规则的正确性是显而易见的.

### 4 算法的实现与仿真

根据上节中提出的算法规则, 对 NS 仿真软件进行了进一步开发, 实现了网络的带宽信息发布、算法的触发、路由选择. NS 没有统计链路带宽的能力, 为此, 在链路仿真对象中增加了对带宽统计的能力, 并通过泛洪算法进行信息发布. NS

中的仿真是针对无连接网络的,对数据包无特殊的处理.而本算法是应用于 MPLS 网络的,因此进一步开发了处理标记流的程序.此外,还扩充了节点仿真对象的能力,使其能够监测相邻链路的带宽使用状况,当链路的使用率超过一定的阈值时(我们采用的是 95%),触发本算法,根据前面提出的路由算法进行路由选择,进而建立部分 LSP.

在仿真实验中,采用了美国国家骨干网作为仿真实验的拓扑.图 2 和图 3 采用的都是这种拓扑,其中,圆圈表示路由器,圆圈间的连线表示链路.链路的两个方向的带宽都是 2M.链路采用同样的颜色.

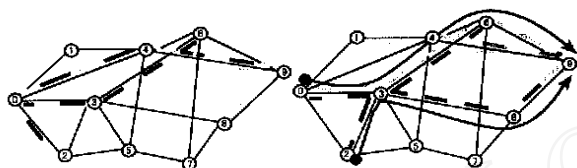


图 2 采用 SPF 算法时,流量在网络上的分布

图 3 采用本算法时,流量在网络上的分布

表 1 仿真所用流量矩阵

编号	源节点	目的节点	表示颜色	带宽要求
Cbr0	2	9	蓝色	1200k
Cbr1	4	9	红色	900k
Cbr2	8	1	绿色	500k
Cbr3	5	6	黄色	1000k
Cbr4	6	0	黑色	1000k
Cbr5	0	9	紫色	1200k

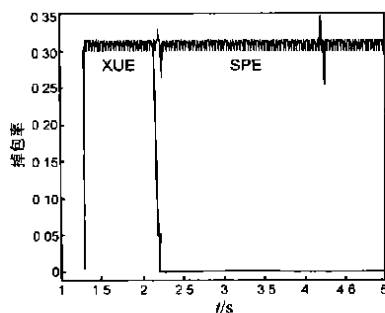


图 4 链路 49 在两种算法下的丢包率变化

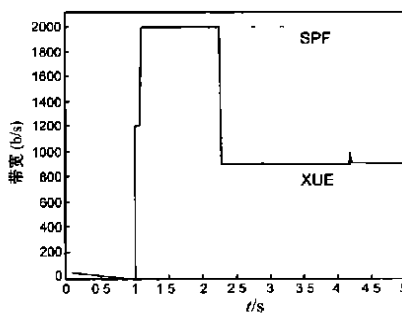


图 5 链路 49 在两种算法下的带宽变化

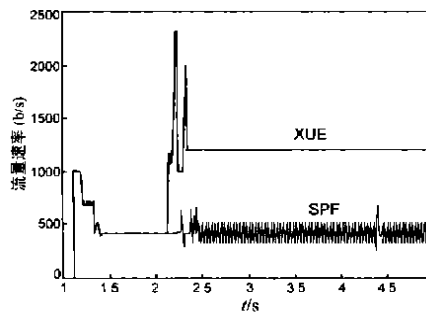


图 6 链路 49 上一个流量在两种算法下的速率变化

## 6 结论

和目前只采用现行最短路径优先路由选择算法(如链路状态路由选择算法)相比,得到以下结论:

(1) 基于带宽和跳数的流量工程路由选择算法是现行路由选择算法的补充,它只在必要时在部分路由器上被激活,因此节约了大量的计算开销.

(2) 仿真结果表明,基于带宽和跳数的流量工程路由选择算法对于缓解由于流量对资源竞争引起的丢包具有显著的效果.

(3) 仿真结果表明,基于带宽和跳数的流量工程路由选择算法明显地改善了流量在整个网络上的分布状况,避免了部

图 2 和图 3 采用了相同的网络拓扑和数据源(如表 1 所示),各种数据源产生的流用不同的颜色表示(黑白文档用不同的灰度表示).由于它们采用的路由算法不同,使得流量在网络上的分布也出现了不同.图 2 中,网络采用了现行的最短路径优先选择算法 SPF,数据源 Cbr5、Cbr1 和 Cbr0 产生的流(路径分别为:0-4-9,4-9,2-0-4-9)都通过链路 49,Cbr5 和 Cbr0 产生的流都通过链路 04,由此使得链路 04 和 49 非常拥挤,同时有些链路却没有被使用(如 38).在图 3 所示的实验中,仿真开始时,网络上流量的分布与图 2 一致,此时由于链路 04 和 49 使用率非常高,由此激活了本文提出的流量工程路由选择算法 XUE,并对 Cbr5 和 Cbr0 产生的流进行了重新路由(路径分别为:0-3-6-9,2-3-8-9),使得流量在网络上分布更加均匀,从而缓解了链路 04 和 49 的拥挤状况,减小了网络的拥塞概率.

## 5 数据分析和结论

最后,我们对仿真数据进行了分析,发现采用 XUE 算法后,网络中链路的掉包率得到了极大的改善,如图 4(描述了链路 49 在两种算法下的掉包率)所示.采用 SPF 算法时比较拥挤的链路在采用 XUE 算法后,链路的使用带宽降低了,如图 5(描述了链路 49 在采用两种算法时使用带宽的变化)所示.采用 XUE 算法增加了点对间的流量,如图 6(描述了通过链路 49 的一个流量在两种算法下的速率变化)所示.

分链路过分使用的同时,其它链路的不充分使用.

(4) 仿真结果也表明,基于带宽和跳数的流量工程路由选择算法可以增加网络的吞吐量.

## 参考文献:

- [1] Juniper Networks. Traffic engineering for the new public network [EB/OL]. {http://www.juniper.net}, 1999. 1.
- [2] Q Ma. QoS routing in the integrated services networks [D]. Ph. D. thesis. Carnegie Mellon University, 1998. 1.
- [3] Xipeng Xiao, Alan Hannan, Brook Bailey. Traffic engineering with MPLS in the Internet [J]. IEEE Network Magazine, 2000, 14 (2): 28 - 33.
- [4] Ddaniel O. Awduche, Joe Malcolm, Johnson Agogbua, Mike O Dell. Re-

- quirements for traffic engineering over mpls [ EB/OL ]. RFC2702. { <http://www.ietf.org/rfc/rfc2702.txt> }, 1998, 10.
- [ 5 ] Zheng Wang, Jon Crowcroft. Bandwidth-delay based routing algorithms [ A ]. IEEE Globecom 95 [ C ]. Singapore, 1995, 11.
- [ 6 ] Jeff Doyle. TCP/ IP 路由技术 [ M ]. 清华大学出版社, 1999, 4.
- [ 7 ] Jose Joaquin Garcia-Lunar-Aceves. A minimum-hop routing algorithm based on distributed information [ J ]. Computer Networks and ISDN Systems 1988/ 89 16:367 - 382.
- [ 8 ] Curtis Villanizar. MPLS optimized multipath (MPLS - OMP) [ EB/OL ]. draft-villanizar-mpls-omp-01. { <http://ardnoc41.canet2.net/mpls/drafts/draft-villanizar-mpls-omp-01.txt> } 1999, 2.
- [ 9 ] Bala Rajagopalan, Qingming Ma. An overlay model for constraint-based routing [ EB/OL ]. draft-rajagopalan-CR-overlay-00. txt. { <http://community.roxen.com/developers/idecs/drafts/draft-rajagopalan-CR-overlay-00.html> }, 1999, 1.
- [ 10 ] E. 米涅卡. 网络和图的最优算法 [ M ]. 中国铁道出版社, 1998, 4.

### 作者简介:



**薛希俊** 1975 年生于山西省交城. 2000 年于天津大学自动化学院获硕士学位, 现在天津大学自动化学院攻读博士学位. 研究方向为通信网(含计算机网络)的性能优化、流量工程研究和网络规划.



**孙雨耕** 1940 年出生. 教授、博士生导师. 研究方向为: 图论与系统优化、电路理论(含电网网络的 CAD)、可靠通信网优化设计、通信网络中的流量控制工程等.

www.cnki.net