

# 一种快捷的 MPLS 网络负载均衡动态路由算法<sup>\*</sup>

唐治果 李乐民 虞红芳

(电子科技大学宽带光纤传输与通信网技术教育部重点实验室 成都 610054)

**摘要** 针对 MPLS 网络引入了一种快捷的有带宽保证的负载均衡动态路由算法(FDRA),该算法通过提出预期负载的概念和链路上的剩余容量来定义链路的成本,然后采用以链路成本为基础的最短路径算法选路,使得网络快捷地建立尽可能多的有带宽保证的路由,并且使这些路由均衡地通过网络。在与其他算法相比较中,该算法在业务请求拒绝率和重路由性能方面有更好的表现,特别是路由的建立时间快捷。

**关键词** 负载均衡,流量工程,MPLS

## A Fast Load Balancing Dynamic Routing Algorithm in MPLS Networks

TANG Zhi-Guo LI Le-Min YU Hong-Fang

(Key Lab of Broadband Optical Fiber Transmission and Communication Networks, UESTC, Chengdu 610054)

**Abstract** A Fast Dynamic Routing Algorithm(FDRA) with bandwidth guaranteed is introduced for MPLS networks. This algorithm defines link cost through expecting load to be presented and link residual bandwidth, then chooses a path through the shortest path algorithm based on link cost. The main objective of FDRA is to swiftly set up paths as many as possible with bandwidth guaranteed, and make these paths go through evenly the network. Simulation shows the algorithm is better than other algorithms in reject ratio of the traffic requests and performance of rerouting the traffic requests when a link is failure, especially in computing time cost.

**Keywords** Load balancing, Traffic engineering, MPLS

## 1 引言

Internet 应用的指数增长已经给网络传输带来了很重的负担,加之现有网络使用的选路算法有缺陷,使得网络的网速缓慢,网络容易出现堵塞。虽然增加更多的网络资源可以暂时缓解网络拥塞,但是从长远来看,增加网络资源来解决网络拥塞,会增加网络运营商的成本,而网络的综合性能没有得到增强,因而不是一个有效经济的解决方案,会损害运营商利益。而且,用户逐渐要求网络运营商提供不同等级的服务,这加重了网络管理和操作的复杂度。网络运营商需要的是协调、控制和有效利用现有资源来满足用户需求,以实现资源利用最大化和利润最大化。幸运的是,新出现的多协议标签交换技术(MPLS)能在不增加网络资源的前提下,采用好的负载均衡路由算法,从而使链路负载均衡,减轻网络资源争夺和提高全网资源利用率,提高网络整体性能,这就是流量工程(TE)。

针对网络运营商的 MPLS 网络,怎样在主干网或者传输网中建立满足预留带宽需求的路径,是减轻网络资源争夺和提高全网资源利用率的关键。MPLS 技术使标签交换路径(LSP)能沿着具体的路径准确地路由,一条 LSP 通过标签集合来识别,使数据包在网络入口路由器处被封装,贴上标签。包沿着已知的 LSP 路径转发,到达网络目的出口路由器处拆除封装,完成包在 MPLS 网络中的传送。这种方法很容易通过使用新的路由算法实现流量工程,从而解决运营商的问题。为此本文提出了 FDRA 路由算法,该算法的目标就是有效地利用现有的网络资源,采用良好的策略安排业务通过网络,使网络拥塞最小化。

建立有带宽保证的路由问题已经在其他地方研究过<sup>[1~4]</sup>。下面先分析最小跳路由算法 MHA、最宽最短路径算法 WSP<sup>[1]</sup>、最小干扰路由算法 MIRA<sup>[2]</sup>、PBR 算法<sup>[3]</sup> 和 LCBR 算法<sup>[4]</sup>等的思想和存在的问题。

最简单的小跳路由算法(MHA)是在源和目的节点之间建立一条有最少链路数的可行路径。这种算法是基于 Dijkstra 算法,只用了链路状态信息,简单而有效,但是很容易使网络中一些链路出现瓶颈效应,而其他一些链路负载很轻,从而导致网络利用率很低。

最宽最短路径算法(WSP)<sup>[1]</sup> 是选择跳数最少的最短可行路径,如果存在多条跳数相同的最短路径,则在其中选择路径可用带宽最多的一条路径。它利用了链路状态信息和一些辅助的容量信息,没有考虑在进出节点对之间选路对其他节点对的影响,也没有考虑进出节点对之间或者链路上的流量,它只关心单条链路的当前可用带宽和被路由的业务带宽需求,既不关心也不知道它对其他进出节点对的影响以及影响程度有多大。因而此种算法网络容易产生瓶颈效应,导致堵塞,网络利用率较低。

最小干扰路由算法(MIRA)<sup>[2]</sup> 是比最小跳算法(MHA)和最宽最短路径算法(WSP)更好的算法,它的关键思想是在源和目的节点对之间选择一条对路由未来业务产生最小干扰的路径。MIRA 算法中关键链路的定义是当链路容量减少一个单位时,源目的节点对之间的最大流也减少。该算法的目标是选择包含尽可能少的关键链路的路径。MIRA 关键是利用了进出节点对,一般可以避免瓶颈效应。但这种算法有它的弱点:算法在建立每条路由时,都要采用最大流算法,计算所有节点对之间的最大流,以此确定哪些链路是关键链路。

<sup>\*</sup>国家自然科学基金项目(基金号:60302010)。唐治果 在读研究生,主要研究方向:光纤传输与通信网技术;李乐民 院士。

这样就大大增加了算法的计算复杂度,需要大量的计算,耗时,在仿真结果中可以证明。另外,MIRA 仅简单地考虑在进出路由器对之间最大可用带宽的减少,没有考虑路由跳数以及链路的影响。因而,即使在路由器对之间,网络有充足的剩余带宽来路由一条 LSP 请求,MIRA 也可能拒绝这条 LSP 请求,这也是造成算法的性能不是很好的原因。

对于 PBR 算法<sup>[3]</sup>和 LCBR 算法<sup>[4]</sup>,它们在计算中都需要网络的统计流量作为对未来流量分布的粗略预测,依赖进出路由器对之间的流量统计,但由于网络流量是突发的,非常动态,进出路由器对之间的统计流量不容易获得,而且可能不可靠,所以会影响算法的可靠性。

对于所有的 LSP 路由算法,要计算具体的路由,入口路由器需要知道当前网络拓扑和链路剩余容量。假设网络拓扑已知,链路剩余带宽可通过有服务质量的扩展的 OSPF<sup>[5]</sup>或 IS-IS<sup>[6]</sup>、扩展的 RSVP-TE<sup>[7]</sup>或 CR-LDP<sup>[8]</sup>等协议对有限制的路由广播链路剩余带宽而获得,它是唯一的动态可用信息。

通过上述分析我们知道,一个好的路由算法要能适应网络拓扑和流量条件的改变,并且只有在网络发生拥塞时才能充分体现出它的优点。为了克服以上各种算法中存在的缺点,利用现有的仅能利用的信息,本文提出了快捷的动态路由算法 FDRA,它考虑了进出节点对位置的影响、最大流流经每条链路的影响、每条链路上可能经过的路由的影响以及链路剩余带宽的影响,并且采用预先计算的方法减少动态路由的计算复杂度,从而使得该算法高效快捷。

本文第 2 部分具体描述文中提出有带宽保证的 FDRA 算法;第 3 部分通过仿真比较该算法与其他算法的性能;最后总结本文的要点。

## 2 路由算法 FDRA 的描述

### 2.1 算法模型

在网络算法研究中,常用图论工具来描述。采用一个  $N$  点、 $E$  边的无向加权连通图  $G(N, E, B)$  作为网络的模型,其中  $N$  代表网络中路由器的集合, $E$  代表网络中路由器之间的链路集合, $B$  表示网络中链路的容量的集合。用  $P$  表示进出路由器对的集合,图中每个节点都有唯一的标识。 $s$  和  $d$  表示图  $G$  中的任意两点,其中  $s, d \in N$ ,  $(s, d)$  表示进出路由器对,  $(s, d) \in P$ 。  $R_l$  表示任意时刻链路  $l$  的剩余容量,  $n$  表示节点个数,  $m$  表示链路数。

### 2.2 具体算法

受文[2]、[4]中思想的启发,首先定义期望负载这个概念来测量链路关键性。也就是说,在不同的源目的节点对之间传输一定量的业务时,传输业务多的链路将比传输业务少的链路更关键。将定义用数学公式来具体化,假设在源和目的节点对  $(s, d)$  之间总共有  $x$  条网络路由,  $y$  表示在  $x$  条网络路由中通过链路  $l$  的路由数。那么定义源目的节点对  $(s, d)$  之间的链路  $l$  的期望负载为:

$$\Phi_l(s, d) = \frac{y}{x} f_l(s, d) \quad (1)$$

在链路  $l$  上的总的期望负载  $\Phi_l$  被定义为如下公式:

$$\Phi_l = \sum_{(s, d) \in P} \Phi_l(s, d) \quad (2)$$

式中  $f_l(s, d)$  表示在离线情况下节点对  $(s, d)$  之间的网络最大流中通过链路  $l$  的那部分最大流,也就是最大流的一部分。在这里,链路关键性  $\Phi_l(s, d)$  是静态的,因为它是由网络拓扑决定,只有当网络拓扑改变时才改变。因而,  $\Phi_l$  的值也是静

态的,只需离线预先计算,只有当网络拓扑发生变化时才重新计算。

为了能确定具体的路由路径,必须把每条链路上的总的负载期望转化为每条链路的成本,这样才能通过最短路径算法找出所需的路由路径。因此,具体路由一条 LSP 请求  $r(s1, d1, b)$ , 此处  $s1$  表示入口路由器节点,  $d1$  表示出口路由器节点,  $b$  表示 LSP 请求的需求带宽。首先定义每一条链路的动态成本,公式如下:

$$\cos t(l) = \frac{\Phi_l}{R_l} \quad (3)$$

从式中可以看出,有小的剩余容量  $R_l$  或者大的期望负载  $\Phi_l$  的一条链路  $l$  在 LSP 路由中使用是更昂贵的。

综上所述,要找出一条 LSP 请求  $r(s1, d1, b)$  的路由路径,具体算法步骤如下:

(1) 确定进出节点对  $(s, d)$  之间的所有路由  $x$  和通过每条链路的路由  $y$  以及通过每条链路的流  $f_l(s, d)$ , 根据式(1)得出  $\Phi_l(s, d)$ 。

(2) 对于所有进出节点对  $(s, d)$ , 确定  $\Phi_l(s, d)$ , 根据式(2)得出  $\Phi_l$ 。

(3) 根据式(3)确定网络中每条链路的动态成本  $\cos t(l)$ 。

(4) 确定在整个网络中满足需求带宽的链路,把链路剩余带宽  $R_l$  小于需求带宽  $b$  的链路去除,得到新的网络即余留网  $G'$ 。

(5) 使用上面计算出的  $\cos t(l)$  作为链路  $l$  的权值,在余留网  $G'$  中用 Bellman-Ford 最短路径算法确定节点对  $(s1, d1)$  之间的最短路径。

(6) 从  $s1$  到  $d1$  沿着最短路径路由带宽需求为  $b$  单位的 LSP,更新链路上的剩余带宽。

(7) 返回到步骤(3),为下一个 LSP 请求做好准备。

## 3 仿真分析

### 3.1 仿真模型

为了研究本文提出的路由算法 FDRA 的性能,使用 C++ 语言编写了几种算法的仿真程序。仿真中使用了 15 个节点的网络拓扑(如图 1),采用无向图,即每条链路都是双向的。进出路由器为  $(S1, S2, S3, S4, D1, D2, D3, D4)$ , 在图中已经标出。图 1 中容量分布不均,粗线链路代表 4800 个单位带宽,细线链路代表 1200 个单位带宽,分别代表 OC-48 和 OC-12。LSP 请求的入口路由器节点在  $S1 \sim S4$  之间任意选择,出口节点在  $D1 \sim D4$  之间任意选择,带宽需求为 1~3 之间的任意整数(包括 1 和 3)个单位。每个 LSP 请求建立后,它的生命期是永久的。

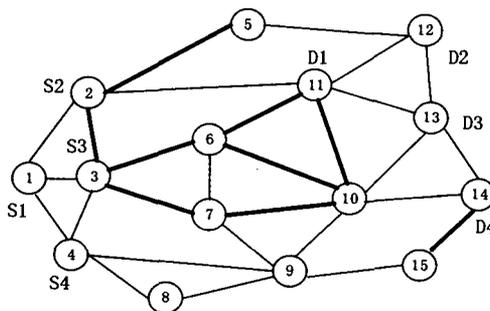


图 1 15 个节点的网络拓扑图

在仿真中,采用最小跳算法 MHA、最宽最短路径算法 WSP、最小干扰路由算法 MIRA 和本文提出的 FDRA 路由算法,以比较使用不同的算法建立多个业务请求后网络总的被接受请求的带宽、业务请求的阻塞数和重路由性能。此处的网络总的被接受请求的带宽是指在业务请求下建立的满足带宽要求的路由的带宽总和。如果在相同的业务请求数下网络总的被接受请求的带宽多,表明网络的资源利用率高,算法的性能好。同样地,业务请求阻塞数是指被网络拒绝路由的业务数。如果业务请求阻塞数高,则表明网络负载平衡不足,网络容易堵塞,资源利用率不高,相应地算法性能不好。在网络有可用带宽的情况下,算法的重路由性能越高,表明算法的性能越好。

### 3.2 仿真结果分析

针对不均匀容量网络拓扑图 1,得出几种算法在建立 12000 个 LSP 请求下网络的 LSP 请求阻塞数(见图 2)和所有进出节点对总的被接受请求的带宽(见图 3)。从图中可以看出,LSP 请求数在 1000 以后,MHA 的性能比其他算法的性能都差;在 2000 以后,WSP 算法的性能也开始变差;MIRA 算法在 8500 以后性能变得比 FDRA 算法要差,本文提出的 FDRA 算法的请求阻塞数曲线一直处于最低,而进出节点对之间总的被接受请求的带宽曲线一直处于最上方,说明 FDRA 算法综合考虑了进出节点对、路由路径以及链路的影响,性能更好,使得更多的路由负载均衡地通过网络,提高了网络资源的利用率。通过比较得出,在 LSP 请求拒绝率和 LSP 请求下进出节点对被接受请求的总带宽方面,本文提出的 FDRA 算法的路由性能最好,也说明了 FDRA 算法有很好的路由性能和负载均衡能力。

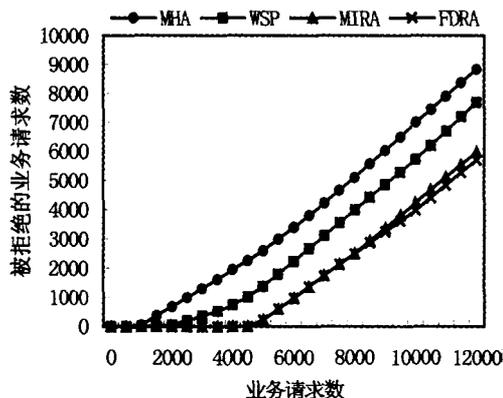


图 2 请求阻塞率

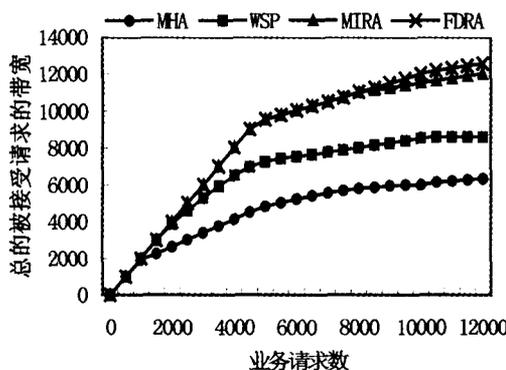


图 3 总的被接受请求的带宽

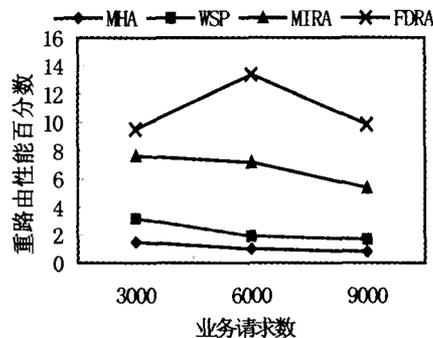


图 4 算法重路由性能

另外,一种算法的好坏不但要考虑算法的路由性能,还要考虑当链路失效时算法的重路由性能,这样才能更好地适应网络运行的需要。为了比较各种算法的重路由性能,仍然采用网络拓扑图 1,比较 LSP 请求数在 3000、6000、9000 时链路失效后各种算法的重路由性能(见图 4)。因为每个业务的路由不同,成功路由的路由数也不同,在相同的 LSP 请求数下网络进出节点对之间所余留可用最大流也不同,选择某一条链路失效来计算重路由成功率带有很大的随机性,并且重路由成功率与网络剩余带宽有关,只考虑重路由成功率得出的数据不能可靠地反映算法的性能。所以,需要采用一种比较合理的比较方法。本文定义重路由性能来比较算法的重路由性能。用公式表示为:

$$\text{算法的重路由性能} = \text{重路由的成功率} \times \text{进出节点对的最大流利用率}$$

在计算算法重路由成功率和进出节点对的最大流利用率时,每次只考虑一条链路失效,让通过该失效链路上的所有 LSP 业务重新路由,并且遍历网络拓扑图中的每一条链路,最后得出所有链路的总的重路由的路由数和总的成功重路由的路由数以及成功重路由业务的带宽总和。那么算法的重路由成功率用公式表示为:

$$\text{算法的重路由成功率} = \frac{\text{总的成功重路由的路由数}}{\text{总的重路由的路由数}}$$

而进出节点对的最大流利用率用公式表示为:

$$\text{进出节点对的最大流利用率} = \frac{\text{平均每条链路失效后成功重路由业务的带宽}}{\text{所有进出节点对的余留可用最大流}}$$

式中平均每条链路失效后成功重路由业务的带宽就是成功重路由业务的带宽总和与链路数的比值。所有进出节点对的余留可用最大流就是在路由一定数量的 LSP 业务后网络的所有进出节点对所剩余的可用最大流。

从图 4 中可以看出,MHA 算法的重路由性能最差,其次是 WSP 算法,MIRA 算法的重路由性能比 MHA 算法和 WSP 算法都好,但次于 FDRA 算法。本文提出的 FDRA 算法的重路由性能曲线位于最上方,在这几种算法中,性能是最好的。

### 3.3 算法复杂度比较

路由算法 FDRA 分为离线阶段和在线阶段,离线阶段的算法复杂度比较高。但是由于离线阶段是预先计算的,只有当网络拓扑或者链路发生变化时才需要重新计算,所以对动态路由不会产生影响。最关键的是在线阶段的路由算法复杂度,因为在线阶段,路由器的计算能力有限,而且路由器的大部分时间需要去处理大量的到达离开业务,没有更多的时间

来处理算法计算,所以要求在线路由算法越简单快捷越好。表1列出了各种算法的计算复杂度和每条路由的计算时间。为了数据的可靠性,这里的每条路由的计算时间是对每种算法在相同的运行环境下运行20次得出的平均统计时间,而每次都是计算100个业务请求得出的每个业务请求所花费的平均时间。

表1 算法复杂度和每条路由的计算时间

算法名称	算法复杂度	每条路由的建立时间 ( $\times 10^{-3}$ s)
MHA	$m+n \log n$	2.8
WSP	$2(m+n \log n)$	5.5
MIRA	$(mn^2+1)(m+n \log n)+m$	30389.1
FDRA	$2m+n \log n$	3.1

从表1可以看出,新算法FDRA比WSP、MIRA的计算复杂度都要低,与最快的MHA算法接近。需要说明的是,表中的数值只是相对值,与计算机的性能和网络拓扑图有关。但是也可以看出,FDRA算法计算复杂度比MIRA要低3~4个数量级,这对于任务繁重的路由器来说是很有必要的。MIRA算法的计算复杂度使它不能用于实用的路由算法,不过它为路由算法的深入研究提供了方向。而FDRA算法在线计算的计算复杂度接近MHA算法,可以在实际的路由器上应用,故而FDRA算法的总体性能比MHA、WSP和MIRA要好很多。

综上所述,FDRA算法的计算速度快、路由拒绝率低、重

路由性能好,是快捷高效的动态负载均衡算法。

**结束语** 本文针对MPLS提出了一种快捷的有带宽保证的动态负载均衡路由算法FDRA,具体描述了FDRA算法。通过仿真分析,结果显示该算法在业务拒绝率和重路由性能方面有很好的表现,并且在计算时间花费上表现出了很好的性能。

## 参考文献

- 1 Guerin R, Orda A, Williams D. QoS Routing Mechanisms and OSPF Extensions. In: Proc. IEEE GLOBECOM'97, phoenix, AZ, 1997, 3:1903~1908
- 2 Kodialam M S, Lakshman T V. Minimum Interference Routing with Applications to MPLS Traffic Engineering. IEEE INFOCOM, 2000, (2):884~893
- 3 Suri S, Waldvogel M, Bauer D, et al. Warkhede, Profile-Based Routing and Traffic Engineering, Computer Communications, 2003, 26(4):351~365
- 4 Gopalan K, Chiueh Tzi-cker, Lin Yow-jian. Load Balancing Routing Algorithm with bandwidth-delay guarantees. IEEE Communications Magazine, 2004; 108~113
- 5 Katz D, Yeung D, Kompella K. Traffic engineering extensions to OSPF. Work in progress, Internet Draft, draft-katz-yeung-ospf-traffic-05.txt, 2001
- 6 Li T, Smit H. IS-IS extensions for traffic engineering. Work in progress, Internet Draft, draft-ietf-isis-traffic-04.txt, 2001
- 7 Berger L. GMPLS Signaling Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions, IETF RFC 3473, Jan. 2003
- 8 Ashwood-Smith P, Berger L. GMPLS Signaling Constraint-based Routed Label Distribution Protocol (CR-LDP) Extensions, IETF RFC 3472, Jan. 2003

(上接第57页)

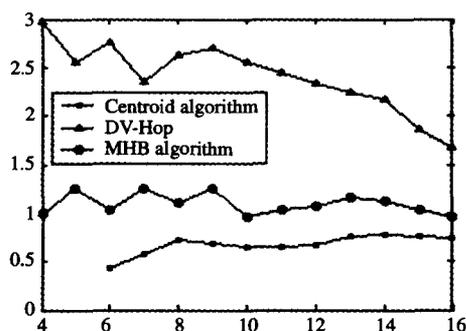


图9 最大定位误差

**结论** 针对无线传感器网络的网络结构特点及经典Range-free定位方法的局限性,本文提出了一种新的基于多跳导标的分布式节点定位方法,该方法可以有效解决无线传感器网络中少导标节点情况下节点定位中的定位代价的经济性和定位结果的可靠性之间的平衡问题。MHB方法和经典的基于导标节点的网络定位方法相比,主要区别在于能够选择合适的导标三角形,使得本节点尽可能位于三角形之内,大大减少了定位估计误差,提高了定位结果的稳定性。该方法的主要特点在于:(1)能够根据局部网络拓扑结构动态选择合适导标节点,进行协作定位;(2)不依赖复杂的测距方法和优化计算,具有较小的计算代价,能满足无线传感器网络节点资源受限的要求。仿真实验结果同样表明,MHB算法具有很好的自适应性、扩展性和鲁棒性,在大规模无线传感器网络的实际应用中具有较好的应用前景。

## 参考文献

- 1 Chong C Y, Kumar S P. Sensor Networks: Evolutions, Opportu-

- 2 nities, and Challenges. In: Proc. of the IEEE, 2003, 91(8): 1247~1256
- 2 He T, Stankovic J A, Lu C, et al. SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks. In: Proc. of 23th IEEE Intl. Conf. on Distributed Computing Systems. Providence, RI, United States, 2003. 46~55
- 3 Ko Young-Bae, Vaidya N H. Location-Aided Routing(LAR) in Mobile Ad Hoc Networks. 2000, 6(4):307~321
- 4 Hightower J, Borriello G. Location systems for ubiquitous computing. IEEE Journal on Computer, 2001, 38(6): 57~66
- 5 Bulusu N, Heidemann J, Estrin D. GPS-less Low Cost Outdoor Localization for Very Small Devices. IEEE Personal Communications Magazine, 2000, 7(5):28~34
- 6 Niculescu D, Nath B. DV-based positioning in ad hoc network. Journal of Telecommunication Systems, 2003, 22(1): 267~280
- 7 He T, Huang CD, Blum BM, et al. Range-Free localization schemes in large scale sensor networks. In: Proc. of the 9th Annual Intl. Conf. on Mobile Computing and Networking. San Diego, CA, United States, 2003. 81~95
- 8 Savvides A, Park H, Srivastava MB. The bits and flops of the N-hop multilateration primitive for node localization problems. In: Proc. of the 1st ACM Intl Workshop on Wireless Sensor Networks and Applications. Atlanta, GA, United States, 2002. 112~121
- 9 Shang Y, Rumi W, Zhang Y, et al. Localization from connectivity in sensor networks. IEEE Transaction on Parallel and Distributed Systems, 2004, 15:961~974
- 10 Nasipuri A, Li K. A directionality based location discovery scheme for wireless sensor networks. In: First ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, GA, United States, 2002. 105~111
- 11 Priyantha N B, Balakrishnan H, Demaine E, et al. Anchor-free distributed localization in sensor networks. In: Proc. of the First Intl Conf on Embedded Networked Sensor Systems. Los Angeles, CA, United States, 2003. 340~341
- 12 Wang Shu, Liu Kezhong, Hu Fuping, et al. A Distributed Network Localization Scheme Motivated by Graph Rigidity Theory. In: Proc. of IEEE 12th Intl. Conf. on Telecommunication. Cape-town, South Africa, 2005. 423~427
- 13 Langendoen K, Reijers N. Distributed localization in wireless sensor networks: A quantitative comparison. The Intl Journal of Computer and Telecommunications Networking, 2003, 43(4): 499~518