

面向 IP 快速路径切换的 OSPF 冗余路径算法

商荣亮, 张晓哲, 郇苏丹

(国防科学技术大学 计算机学院, 湖南 长沙 410073)

摘要:在 IP 网络中, 当某链路或者节点发生故障时, 通过路由协议的收敛来绕开故障的链路或节点。对 OSPF 路由协议, 这个时间至少为 5 秒, 期间经过故障节点或链路的流量将会被丢弃, 绝大多数的应用可以承受这种程度的延迟。但是, 对延迟敏感的应用如 VoIP 而言, 这种量级的延迟是很难为用户所接受的。基于现有的 OSPF 路由协议的最短路径树 (SPT) 算法, 提出一种支持 IP 快速重路由的多冗余路径树计算算法。算法计算除最短路径外至少一条不相交无环备份路径, 保证在最短路径的链路或节点故障时, 通过快速切换到备份路径, 以提高 IP 网络的故障收敛时间。

关键词:开放最短路径优先; 双向转发检测; 快速重路由; 快速路径切换; 备份路径

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2011)06-0001-03

OSPF Redundant Path Algorithm of a Fast Path Switching for IP

SHANG Rong-liang, ZHANG Xiao-zhe, LI Su-dan

(School of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: In the IP network, bypass the failed link or node through the convergence of routing protocol when a link or node failed. This time at least 5 seconds for OSPF routing protocol, during the fault the packet will be dropped, most of the applications can accept this level of delay. However, delay-sensitive applications such as VoIP, the delay of this magnitude is difficult to accept for users. Based on the existing routing protocols of shortest path tree (SPT) algorithm, proposed to support multiple redundant path tree calculation algorithm for IP fast reroute. In addition to calculating the shortest path, the algorithm calculates at least one disjoint acyclic backup path, when link or node of the shortest path fail, ensure that the packet switches to the backup path quickly to improve IP network fault convergence time.

Key words: OSPF; BFD; fast reroute; fast path switching; backup path

0 引言

IP 网络本身具有一定的健壮性, 因为诸如 OSPF 这样的内部网关协议被设计可在网络设备失效发生后, 网络域内所有路由器间通过洪泛链路状态通告报文 (Link State Advertisement, LSA) 来交互信息, 以此更新链路状态数据库。然而通过路由重收敛重建新的有效路由的处理过程时间较长, 并且在网络还没有收敛到一个新的状态之前, 流量会由于路由无效而被丢失。针对路由收敛慢、收敛过程中分组丢失或延迟等问题, 越来越多的研究者开始关注网络路由的快速收敛问题研究。文中提出一种支持 IP 快速重路由的多冗余路径树计算算法。其重点解决 OSPF 路由协议从网络节点/链路故障发生开始, 直到重新计算最短路径生成树 (SPT) 完成协议收敛, 这段时间内的 IP 报文转发的中

断问题。

1 相关研究

为了提高网络故障收敛时间, 目前业界正在研究的新方法是提前计算备用路由, 即当路由探测到链路节点故障时并不是立即扩散路由信息来进行路由计算, 而是利用备份路径替换失效路径, 先在本地直接修复故障。在整个路由完成重新收敛期间, 一直使用事前确定的备用路由指导转发, 这样流量中断时间将大大缩短, 等于探测邻接故障与采用备用路由替换失效路由的时间之和。这种利用由本地预先决定的修复路径对失效链路或路由节点提供保护的技术就称为 IP 快速重路由 (IP FRR)。基于 IP 备份路径的快速重路由有多种方案^[1-5], 包括无环替代路径 (LFA)、重地址配置 (NVA)、接口选择转发 (ISF)、多拓扑路由 (MTR) 等。其中, LFA 占用网络过多资源, 且在复杂组网的情况下路由保护功能不够完善; NVA 方案还不够成熟, 有较多细节问题待讨论; MTR 的实现依赖多拓扑路由技术的发展, 且实现方法较复杂; 而 ISF 虽然技术方案比较成熟, 但其应用于当前流行的分布式架构路由设

收稿日期: 2010-11-10; 修回日期: 2011-02-10

基金项目: 国家自然科学基金资助项目 (60803153)

作者简介: 商荣亮 (1981-), 男, 辽宁营口人, 硕士研究生, 研究方向为路由查找算法; 郇苏丹, 博士, 副教授, 硕士生导师, 研究方向为高性能路由器和路由协议。

备当中仍是一个较大的挑战,可行性较低。

双向转发检测(BFD)^[6,7]是一种独立的检测机制,承载在所检测的数据传送协议中,可用于多种协议层和传送方式;它还能够提供一种用于物理接口、数据链路,甚至转发引擎故障检测的低开销、短时延机制。此外,BFD 的检测时间和开销可以根据需要进行调整,能够灵活适应不同协议的不同要求。因此,采用双向转发检测可以进一步提高故障响应速度和收敛时间。

2 基于 OSPF 的 IP 快速路径切换实现框架

图 1 给出了基于 OSPF 的 IP 快速路径切换实现框架,其主要原理是利用 BFD(RFC5880)对某条关键路径(即最短路径)进行快速故障检测,当关键路径上的某链路或者节点发生故障时,BFD 会话将进入 Down 状态,这时将该状态变化通知上层协议(OSPF)。当 OSPF 接收到 BFD 会话 Down 消息之后,即将通过该条路径的流量切换到预先计算好的备份路径上去,然后将路由下发与切换保存到转发表。

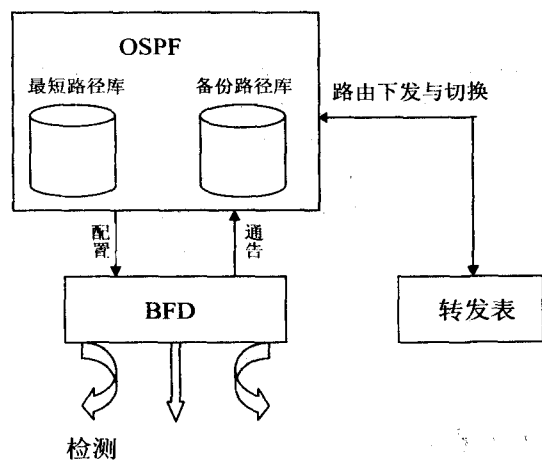


图 1 基于 OSPF 的 IP 快速路径切换实现框架图

3 支持 IP 快速重路由的多冗余路径树计算算法

3.1 算法描述

假设图 $G=(V,E)$ 有 n 个确定的顶点,即 $V=\{v_1, v_2, \dots, v_n\}$ 。用邻接矩阵 $A_{(n,n)}$ 作为存储结构。规定:当顶点 i 和顶点 j 之间有直接边,且权值为 w_{ij} 时, $A_{[i][j]} = w_{ij}$;当顶点 i 和顶点 j 之间没有直接边时, $A_{[i][j]} = 0$ 。

初始化: $P_{(len)} = \Phi$ ($len = 1, 2, \dots, maxlen$), $maxlen$ 取一个适当的值。 $B_{(i)} = \Phi$ ($i = 1, 2, \dots, n$)。

步骤 1 循环,当 $i = 2$ 到 n 时进行;

若 $A_{[1][i]} \neq 0$, 则 $P_{(1)} = P_{(1)} \cup \{v_1 \rightarrow v_i\}$; $B_{(i)} = B_{(i)} \cup \{v_1 \rightarrow v_i\}$;

步骤 2 循环,当 $len = 2$ 到 $maxlen$ 时进行;

2.1 令 $S = P_{(len-1)}$;

2.2 循环,令 β 等于 S 中的每个路径;

2.2.1 令 v_a 等于 β 中的最后一个顶点;

2.2.2 循环,当 $i = 1$ 到 n 时进行;

若 $A_{[a][i]} \neq 0$ 且 v_i 不是 β 路径中的顶点,则 $P_{(len)} = P_{(len)} \cup \{\beta \rightarrow v_i\}$; $B_{(i)} = B_{(i)} \cup \{\beta \rightarrow v_i\}$;

步骤 3 循环,当 $i = 2$ 到 n 时进行;

计算 $B_{(i)}$ 中各个路径的 $cost(v_1, v_i)$ 值,按 $cost$ 值从小到大排序;

步骤 4 循环,当 $i = 2$ 到 n 时进行;

4.1 令 a 等于 $B_{(i)}$ 中的第一个路径,即最短路径;

4.2 循环,令 b 等于 $B_{(i)}$ 中除第一个路径以外的每个路径;

4.2.1 若路径 b 与路径 a 中除 v_1 和 v_i 顶点外,没有公共顶点,则转到 4.2.2,否则删除路径 b ;

4.2.2 令 v_e 为路径 b 中的第二个顶点,若 $cost(a) + cost(v_1, v_e) > cost(b) - cost(v_1, v_e)$, 则路径 b 为路径 a 的备份路径,否则删除路径 b ;

步骤 5 结束。

3.2 算法的时间复杂度分析

第一步循环需要 $n - 1$ 次;第二步循环需要 $(maxlen - 1) * m_1 * n$ 次;第三步循环需要 $n - 1$ 次;第四步循环需要 $(n - 1) * m_2$ 次;

其中: m_1, m_2 为路径的个数^[8,9], 且 $m_1 < n(n - 1)$; $m_2 < \frac{n(n - 1)}{2}$; $maxlen \leq n$ 。

因此,总共需要的次数小于

$$y = n - 1 + (maxlen - 1) * m_1 * n + n - 1 + (n - 1) * m_2 < n - 1 + (n - 1) * n(n - 1) * n + n - 1 + (n - 1) * \frac{n(n - 1)}{2} = n^4 - \frac{3}{2}n^3 + \frac{5}{2}n - 2$$

算法的时间复杂度为 $O(n^4)$ 。

3.3 实例

下面给出一个实例。图 2 为加权无向图。应用此算法求备份路径。

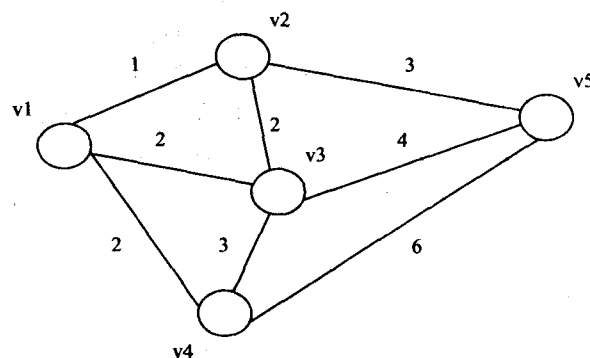


图 2 算法实例

已知: $n = 5$; 取 $maxlen = 5$;

则:长度为 1 的路径:

$$P_{(1)} = \{v_1 \rightarrow v_2, v_1 \rightarrow v_3, v_1 \rightarrow v_4\};$$

长度为 2 的路径:

$$P_{(2)} = \{v_1 \rightarrow v_2 \rightarrow v_3, v_1 \rightarrow v_2 \rightarrow v_5, v_1 \rightarrow v_3 \rightarrow v_2, v_1 \rightarrow v_3 \rightarrow v_4, v_1 \rightarrow v_3 \rightarrow v_5, v_1 \rightarrow v_4 \rightarrow v_3, v_1 \rightarrow v_4 \rightarrow v_5\};$$

长度为 3 的路径:

$$P_{(3)} = \{v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4, v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_5, v_1 \rightarrow v_2 \rightarrow v_5 \rightarrow v_3, v_1 \rightarrow v_2 \rightarrow v_5 \rightarrow v_4, v_1 \rightarrow v_3 \rightarrow v_2 \rightarrow v_5, v_1 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5, v_1 \rightarrow v_3 \rightarrow v_5 \rightarrow v_2, v_1 \rightarrow v_3 \rightarrow v_5 \rightarrow v_4, v_1 \rightarrow v_4 \rightarrow v_3 \rightarrow v_2, v_1 \rightarrow v_4 \rightarrow v_3 \rightarrow v_5, v_1 \rightarrow v_4 \rightarrow v_5 \rightarrow v_2, v_1 \rightarrow v_4 \rightarrow v_5 \rightarrow v_3\};$$

长度为 4 的路径:

$$P_{(4)} = \{v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5, v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_5 \rightarrow v_4, v_1 \rightarrow v_2 \rightarrow v_5 \rightarrow v_3 \rightarrow v_4, v_1 \rightarrow v_2 \rightarrow v_5 \rightarrow v_4 \rightarrow v_3, v_1 \rightarrow v_3 \rightarrow v_2 \rightarrow v_5 \rightarrow v_4, v_1 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_2, v_1 \rightarrow v_4 \rightarrow v_3 \rightarrow v_2 \rightarrow v_5, v_1 \rightarrow v_4 \rightarrow v_3 \rightarrow v_5 \rightarrow v_2, v_1 \rightarrow v_4 \rightarrow v_5 \rightarrow v_2 \rightarrow v_3\};$$

长度为 5 的路径: $P_{(5)} = \Phi$;

源顶点 v_1 到目的顶点 v_1 的所有路径: $B_{(1)} = \Phi$;

源顶点 v_1 到目的顶点 v_2 的所有路径:

$$B_{(2)} = \{v_1 \xrightarrow{1} v_2, v_1 \xrightarrow{2} v_3 \xrightarrow{2} v_2, v_1 \xrightarrow{2} v_4 \xrightarrow{3} v_2, v_1 \xrightarrow{2} v_3 \xrightarrow{4} v_5 \xrightarrow{3} v_2, v_1 \xrightarrow{2} v_4 \xrightarrow{6} v_5 \xrightarrow{3} v_2, v_1 \xrightarrow{3} v_2 \xrightarrow{2} v_4 \xrightarrow{3} v_3 \xrightarrow{4} v_5 \xrightarrow{3} v_2, v_1 \xrightarrow{3} v_2 \xrightarrow{2} v_4 \xrightarrow{6} v_5 \xrightarrow{4} v_3 \xrightarrow{2} v_2\};$$

源顶点 v_1 到目的顶点 v_3 的所有路径:

$$B_{(3)} = \{v_1 \xrightarrow{2} v_3, v_1 \xrightarrow{1} v_2 \xrightarrow{2} v_3, v_1 \xrightarrow{2} v_4 \xrightarrow{3} v_3, v_1 \xrightarrow{1} v_2 \xrightarrow{3} v_5 \xrightarrow{4} v_3, v_1 \xrightarrow{2} v_4 \xrightarrow{6} v_5 \xrightarrow{4} v_3, v_1 \xrightarrow{1} v_2 \xrightarrow{3} v_5 \xrightarrow{6} v_4 \xrightarrow{3} v_3, v_1 \xrightarrow{2} v_4 \xrightarrow{6} v_5 \xrightarrow{3} v_2 \xrightarrow{2} v_3\};$$

源顶点 v_1 到目的顶点 v_4 的所有路径:

$$B_{(4)} = \{v_1 \xrightarrow{2} v_4, v_1 \xrightarrow{2} v_3 \xrightarrow{3} v_4, v_1 \xrightarrow{1} v_2 \xrightarrow{2} v_3 \xrightarrow{3} v_4, v_1 \xrightarrow{3} v_2 \xrightarrow{1} v_3 \xrightarrow{6} v_4, v_1 \xrightarrow{1} v_2 \xrightarrow{3} v_5 \xrightarrow{4} v_3 \xrightarrow{2} v_4, v_1 \xrightarrow{4} v_3 \xrightarrow{3} v_5 \xrightarrow{2} v_4, v_1 \xrightarrow{6} v_5 \xrightarrow{1} v_2 \xrightarrow{2} v_3 \xrightarrow{4} v_4, v_1 \xrightarrow{4} v_5 \xrightarrow{6} v_4, v_1 \xrightarrow{2} v_3 \xrightarrow{2} v_2 \xrightarrow{3} v_5 \xrightarrow{6} v_4\};$$

源顶点 v_1 到目的顶点 v_5 的所有路径:

$$B_{(5)} = \{v_1 \xrightarrow{1} v_2 \xrightarrow{3} v_5, v_1 \xrightarrow{2} v_3 \xrightarrow{4} v_5, v_1 \xrightarrow{1} v_2 \xrightarrow{2} v_3 \xrightarrow{4} v_5, v_1 \xrightarrow{2} v_3 \xrightarrow{2} v_2 \xrightarrow{3} v_5, v_1 \xrightarrow{2} v_4 \xrightarrow{6} v_5, v_1 \xrightarrow{2} v_3 \xrightarrow{2} v_2 \xrightarrow{3} v_5, v_1 \xrightarrow{6} v_5, v_1 \xrightarrow{2} v_3 \xrightarrow{3} v_4 \xrightarrow{2} v_5, v_1 \xrightarrow{2} v_4 \xrightarrow{3} v_3 \xrightarrow{2} v_5, v_1 \xrightarrow{3} v_2 \xrightarrow{2} v_3 \xrightarrow{6} v_5, v_1 \xrightarrow{1} v_2 \xrightarrow{2} v_3\};$$

$$\xrightarrow{3} v_4 \xrightarrow{6} v_5\};$$

利用算法中的 4.2.1 和 4.2.2 两个判断条件,最后得出:

顶点 v_1 到顶点 v_2 的最短路径为 $\{v_1 \rightarrow v_2\}$; 备份路径为 $\{v_1 \rightarrow v_3 \rightarrow v_2\}$;

顶点 v_1 到顶点 v_3 的最短路径为 $\{v_1 \rightarrow v_3\}$; 备份路径为 $\{v_1 \rightarrow v_2 \rightarrow v_3, v_1 \rightarrow v_4 \rightarrow v_3\}$;

顶点 v_1 到顶点 v_4 的最短路径为 $\{v_1 \rightarrow v_4\}$; 备份路径为 $\{v_1 \rightarrow v_3 \rightarrow v_4\}$;

顶点 v_1 到顶点 v_5 的最短路径为 $\{v_1 \rightarrow v_2 \rightarrow v_5\}$; 备份路径为 $\{v_1 \rightarrow v_3 \rightarrow v_5\}$ 。

4 分析比较

4.1 传统 OSPF 收敛时间

一个典型的 OSPF 收敛时间包括: $N * (\text{LSA 接收时间} + \text{发送定时器时间} + \text{传输时延}) + \text{SPF 计算定时器时间}$, 其中 N 为网络跳步数, RFC 推荐的 SPF 计算定时器时间为 $5s^{[10]}$; 在不考虑路由协议分组丢失的情况下, 每跳所花费时间在 $10ms \sim 100ms$ 之间; RFC 推荐的当前互联网中最短的发送定时器时间为 $275ms^{[10]}$; LSA 在网络中洪泛时, 每前进一跳需要的时间在 $10ms \sim 100ms$ 之间。

综合以上, 可估算出 OSPF 的收敛时间, 如表 1 所示。

表 1 采用传统 OSPF 路由协议的收敛时间统计表

N	LSA 接收时间	发送定时器时间	传输时延	SPF 计算定时器时间	OSPF 收敛时间
1	10ms	275ms	10ms	5000ms	5295ms
2	10ms	275ms	10ms	5000ms	5590ms
3	10ms	275ms	10ms	5000ms	5885ms
4	10ms	275ms	10ms	5000ms	6180ms
5	10ms	275ms	10ms	5000ms	6475ms

4.2 支持 IP 快速重路由的多冗余路径树计算算法的中断时间

采用支持 IP 快速重路由的多冗余路径树计算算法, 其中断时间为 BFD 故障检测时间和路由下发与切换时间之和。BFD 检测间隔为 $50ms$, 3 个检测间隔能够发现故障。通过测试得知本算法中路由下发与切换时间约为 $200ms$ 。

综合以上, 可估算出中断时间约为 $350ms$ 。与传统 OSPF 收敛相比时间大大缩短, 提高了 IP 网络的可靠性。

5 结束语

文中提出了一种支持 IP 快速重路由的多冗余路径树计算算法, 该算法结合 OSPF 协议中的最短路径

(下转第 8 页)

组发给 P2, P2 对 VPN 标签分组的处理过程与 P1 一样,只是在 P2 中将 VPN 标签分组的外层标签 9 换为 11,继续将 VPN 标签分组往下游的 P3 路由器转发,P3 的转发模块依据它的 IML 表采用倒数第二跳弹出机制将外层标签去掉,通过查询内层标签,知道要将只剩下内层标签 30 的 VPN 标签分组转发给 PE2,PE2 收到该分组后,通过查询,知道它是自己在 VPN1 中给 15.19.2.0/8 网络分配的唯一标签,PE2 会将分组的 VPN 标签去掉,将 VPN 分组恢复成 IP 分组,PE2 路由器通过查询 IP 路由表,将该分组转发到 CE4 路由器,CE4 路由器将该分组转发到 15.19.2.0/8 网段。

5 结束语

MPLS 具有高效、快速转发的显著特点,在 MPLS 网络中架构 BGP/MPLS VPN,通过在 PE 路由器中创建多个 VRF 实现不同 VPN 之间路由信息隔离,从而保证数据报运载的安全。标签交换路由器(P 路由器及 PE 路由器)分为控制平面和数据转发平面,在控制平面,通过扩展 BGP 协议和 LDP 协议传递路由信息,在标记转发信息库产生 VRF、FTN、ILM 表,并将表下发到的 MPLS 转发处理模块,在数据转发平面,在核心路由器中只进行外层标签的交换,在转发到本地的 PE 路由器时才去掉内层标签,以 IP 方式将数据报转发到目标网段。

(上接第 3 页)

树算法,实现了备份路由^[11-13]的动态计算。同时,结合双向转发检测技术,对链路与节点的故障作出及时响应,从而提高了 IP 网络的故障收敛时间。

由于现有的网络大多还是基于纯 IP 技术构造的,因此纯 IP 网络的保护和恢复具有很好的实际意义。虽然纯 IP 网络的无连接特性给网络的保护恢复带来了一定的困难,但是在原有比较成熟的网络协议的基础上添加新的功能或者调整网络参数以增强 IP 网络的生存性是有实际意义的,尤其是 IP 网络的快速重路由机制具有很好的潜力,许多方法和技术有待进一步研究。

参考文献:

- [1] Atlas A, Zinin A. Basic Specification for IP Fast-Reroute: Loop-free Alternates [C]// In: Proceedings of 7th Annual Conference on Evolutionary Programming, Germany: [s. n.], 1998:209-215.
- [2] Bryant S, Shand M. IP Fast Reroute Using Notvia Addresses [J]. Science, 1995, 220(5): 671-680.
- [3] Nelakuditi S. IP Fast Reroute with Interface Specific Forwarding [R]. Columbia: University of South Carolina, 2007.

参考文献:

- [1] Juniper Networks. BGP/MPLS VPN Fundamentals [S]. RFC 2547, 2000.
- [2] 中华人民共和国通信行业标准. 多协议标记交换 (MPLS) (总体技术要求), YD/T [S/OL]. 2000. <http://wenku.baidu.com/view/63ad13bff121dd36a32d828f.html>.
- [3] 江晓峰, 高兴锁, 周海涛. 基于 MPLS 的 VPN 技术探究 [J]. 计算机技术与发展, 2006, 16(3): 53-55.
- [4] Rosen E, Viswanathan A, Callon R. Multiprotocol Label Switching Architecture [S]. IETF RFC3031, 2001.
- [5] Muthukrishnan K, Malis A. A core MPLS VPN architecture [S]. IETF RFC2917, 2001.
- [6] Agarwal P, Akyol B A. TTL Processing in MPLS Networks [S]. RFC3443, 2002.
- [7] Pepelnjak I, Guichard J. MPLS 和 VPN 体系结构 [M]. 赵斌, 陈文飞, 徐鸿文, 译. 北京: 人民邮电出版社, 2003.
- [8] Bonica R, Tappan D, Gan D. ICMP Extensions for Multiprotocol Label Switching [S]. RFC4950, 2000.
- [9] 罗恒洋. IPsec 在 MPLS VPN 中的应用 [J]. 计算机技术与发展, 2009, 19(3): 36-39.
- [10] 冯 径. 多协议标签交换技术 [M]. 北京: 人民邮电出版社, 2002.
- [11] 吴 伟. 下一代 IP 网络技术保障—多协议标签交换技术 [M]. 北京: 清华大学出版社, 2002.
- [12] Andersson L, Doolan P, Feldman N, et al. LDP Specification [S]. RFC 3036, 2001.
- [4] Apostolopoulos G. Multi-topology protection: promises and problems [R]. [s. l.]: FORTH, 2007.
- [5] 吴 江, 赵慧玲. 下一代的 IP 骨干网络技术 [M]. 北京: 人民邮电出版社, 2001.
- [6] BFD 技术白皮书 [M]. 华为技术有限公司, 2008.
- [7] Hundessa L, Domingo-Pascual J. Reliable and fast rerouting mechanism for a protected label switched path [C]// Global Telecommunications Conference, 2002. GLOBE COMM. [s. l.]: IEEE, 2002: 1608-1612.
- [8] 谢 政. 网络算法与复杂性理论 [M]. 第 2 版. 长沙: 国防科技大学出版社, 2003.
- [9] 高 市, 姜 虹. 有向图无负权全路径算法研究 [J]. 长春大学学报 (自然科学版), 2009, 19(1): 57-59.
- [10] Moy J. OSPF Version 2 [EB/OL]. 1998. <http://www.ietf.org/rfc/rfc2328.txt>.
- [11] Schollmeier G, Charzinski J, Kirstdter A, et al. Improving the resilience in IP networks [C]// Proceedings of High Performance Switching and Routing. USA: IEEE press, 2003: 91-96.
- [12] Xiao Bin, Cao Jiannong. Dynamic Update of Shortest Path Tree in OSPF [C]// Proceedings of the 7th International Symposium on Parallel Architectures. [s. l.]: IEEE, 2004.
- [13] 张民贵, 刘 斌. IP 网络的快速故障恢复 [J]. 电子学报, 2008, 36(8): 1595-1602.