基于增量最短路径优先的域内高效路由保护算法

耿海军[1] 尹 霞[2]

(山西大学软件学院 太原 030006)[1] (清华大学信息科学技术学院计算机科学与技术系 北京 100084)[2]

摘 要 学术界提出利用LFC(Loop-Free Criterion,LFC)规则来解决网络中所有可能出现的单链路故障情形,但是已有的针对LFC的实现方式的计算开销随着网络节点平均度的增加而增加,给路由器带来了大量的额外负担。针对该问题,文中研究如何降低LFC实现方式的计算开销,提出了一种基于增量最短路径优先(Incremental Shortest Path First,i-SPF)的域内高效路由保护算法(Efficient Intra-domain Routing Protection Algorithm Based on i-SPF,ERPISPF)。理论证明ERPISPF的计算开销远远小于构造一棵最短路径树的计算开销,并且可以为任意源-目的对计算出所有符合LFC规则的下一跳集合。实验结果表明,与LFC方案相比,ERPISPF的计算开销降低了93%左右,并且与LFC拥有相同的故障保护率。

关键词 实时应用,路由保护,最短路径树,增量最短路径优先,LFC规则,网络故障,路由可用性

中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.08.019

Efficient Intra-domain Routing Protection Algorithm Based on i-SPF

GENG Hai-jun[1] YIN Xia[2]

(School of Software Engineering,Shanxi University,Taiyuan 030006,China)[1]

(Department of Computer Science & Technology,School of Information Science and Technology,Tsinghua University,Beijing 100084,China)[2]

Abstract LFC(Loop-Free Criterion)has been proposed to cope with all the single link failure scenarios. However,the existing LFC implementation algorithms are time-consuming and require a large amount of router CPU resources. Therefore,this paper studied how to reduce the computational overhead of the LFC implementation,and proposed an efficient Intra-domain routing protection algorithm based on i-SPF (ERPISPF). Theoretical analysis indicates that the time complexity of ERPISPF is less than that of constructing a shortest path tree,and ERPISPF can compute all the valid LFC next-hop sets for any source-destination pairs. The experiment results show that ERPISPF reduce more than 93% computation overhead compared with the LFC,and can provide the same protection ratio with LFC.

Keywords Real-time applications,Routing protection,Shortest path Tree,i-SPF,LFC rule,Network failures,Network availability

1 引言

互联网的飞速发展使其成为全球最重要的通信基础设施之一。因此,越来越多的应用程序部署在互联网上,人们对互联网的依赖达到了前所未有的程度。互联网在设计之初主要支持一些非实时应用,但是目前许多实时应用程序[1-3]都部署在互联网上,例如电话会议、视频、即时通信、互联网金融和多人在线游戏等。因为实时应用对网络时延和丢包率更加敏感,所以这些应用对网络的可用性提出了更加苛刻的要求。

互联网中经常会出现故障[4-6]。路由协议一般通过收敛来应对故障,但是在收敛时可能会遇到路由黑洞或者路由环路[7-10]。因此如何提高域内路由的可用性[11-12]成为了亟待解决的科学问题,这使得越来越多的科研工作者投身于研究如何提升网络快速应对故障的能力。

业界一般采用被动恢复方案和路由保护方案[13]来提高网络中的路由可用性。被动恢复方案[14-18]主要通过改变路由协议的一些参数来加速路由收敛,但是这些方案可能造成网络路由不稳定。路由保护可以分为非逐跳转发和逐跳转发两种方式。非逐跳转发方式需要利用一些辅助机制,如Not-Via[19]、隧道和MPLS[20]等,这些辅助机制对默认协议的改动比较大,不容易部署在实际的互联网。LFC[21]是一种较为经典且受到关注的逐跳转发路由保护方案,但是目前实现LFC算法的时间复杂度随着网络节点平均度的增加而增加,因此已有实现方式无法满足大规模网络的需求。针对LFC存在的上述问题,本文提出一种高效的基于逐跳方式的路由保护方案ERPISPF,该方案不仅具有较高的计算效率,其算法复

到稿日期:2018-02-20 返修日期:2018-06-05 本文受国家自然科学基金(61702315),网络与交换技术国家重点实验室(北京邮电大学)开放课题(SKLNST-2018-1-19)资助。

耿海军 男,博士,讲师,主要研究方向为软件定义网络、路由算法和网络体系结构,E-mail:ghj123025449@163.com(通信作者);**尹 霞** 女,博士,教授,主要研究方向为网络体系结构、路由算法、SDN、协议测试。

杂度小于构造一棵最短路径树的复杂度，并且与 LFC 具有同样的故障保护率。ERPISPF 的高效性源于：该算法依赖于计算节点构造的以该节点为根的最短路径树，然后在该树的基础上执行 i-SPF 算法，最后求出该节点到其他所有节点的满足 LFC 规则的下一跳集合。

2　国内外研究现状

文献[22]提出通过路由偏转算法来提高网络的可用性，该算法首先通过 3 个 Loop-free 来条件计算所有节点之间的备份下一跳，然后利用 Tag 技术从所有备份下一跳中选择合适的下一跳转发报文。然而，该算法由于实现开销较大，并且对转发平面的改动较大，因此难以在实际网络中进行部署。

因特网工程任务组(The Internet Engineering Task Force，IETF)提出利用 LFC 规则来缓解由于链路故障造成的报文丢失的问题。LFC 以其简单和容易部署的特点得到了互联网厂家[23-24]的支持。但是 LFC 算法的复杂度随着网络节点平均度的增加而增加，不利于算法的扩展。文献[25]提出利用 TBFH 来降低算法复杂度，该方案为每个节点计算两个到目的地址的备份下一跳。虽然 TBFH 降低了 LFC 的算法复杂度，但是其故障保护率远远小于 LFC 的故障保护率。针对 LFC 的实现方式复杂度高的问题，我们在文献[26]中提出利用 DMPA 方案来降低 LFC 算法的时间复杂度。虽然 DMPA 的计算时间小于 TBFH 和 LFC 的计算时间，但是其故障保护率依然低于 LFC 的故障保护率。上述方案没有很好地权衡算法开销和故障保护率之间的关系，因此本文研究了一种高效的针对 LFC 的实现方案，该方案不仅具有较高的执行效率，并且与 LFC 具有同样的故障保护率。

针对 LFA 路由可用性低的问题，文献[27]提出利用 Not-Via 可以保护网络中所有的单故障情形。文献[28-29]深入研究了网络拓扑结构和 LFC 故障保护率之间的关系，通过调整链路权值或者增加网络中的链路来提高 LFC 的故障保护率。因此，如果将本文提出的算法运行在通过上述研究成果改变后的网络拓扑中，就可以将故障保护率提高到 100%。

3　网络模型和问题描述

3.1　网络模型

我们用图 $G=(V,E)$ 表示一个网络拓扑结构，其中 V 为该拓扑中节点的集合，E 为该拓扑中边的集合。对于 $\forall v\in V$，$N(v)$ 表示该节点的所有邻居节点，$spt(v)$ 为以节点 v 为根的最短路径树，$D(v,x)$ 表示在 $spt(v)$ 中节点 x 的所有子孙节点。对于 $\forall (i,j)\in E$，$w(i,j)$ 为该边对应的代价；对于 $\forall x,y\in V(x\neq y)$，$cost(x,y)$ 表示这两个节点之间的最短路径对应的代价，$dn(x,y)$ 表示节点 x 到节点 y 的默认下一跳，$bn(x,y)$ 表示节点 x 到节点 y 的备份下一跳的集合。

3.2　问题描述

IETF 提出利用 LFC 规则来应对网络中的单链路故障。下面详细介绍 LFC 规则，其具体内容可以表示为：对于目的地址 d，节点 c 可以将报文发送给其邻居节点 x，当且仅当满足 $cost(x,d)<cost(c,d)+cost(c,x)$。

对于目的地址 d，为了在节点 c 上实现 LFC 规则，节点 c 需要知道 $cost(c,d)$，$cost(c,x)$ 和 $cost(x,d)$ 的数值。节点 c 可以通过 $spt(c)$ 获得 $cost(c,x)$ 和 $cost(c,d)$ 的值，为了得到 $cost(x,d)$ 的值，节点 c 需要计算一棵以节点 x 为根的最短路径树。但是当节点 c 有 k 个邻居节点时，则需要构造 k 棵最短路径树来获得其所有邻居到目的节点的最小代价。因此，上述实现方法的时间复杂度随着计算节点度数的增加而增加，扩展性较差，不容易在实际网络中部署。根据上述计算过程可知，对于目的地址 d，如果节点 c 可以快速计算出其所有邻居对应的 $cost(x,d)(x\in N(c))$ 值，则可以得到高效的 LFC 实现算法。下面给出本文需要解决的关键问题。

问题：对于目的地址 d 和节点 c，如果给定 $spt(c)$，是否可以找到一个算法来快速计算出其所有邻居对应的 $cost(x,d)(x\in N(c))$ 值。该算法具有以下两个特征：

1)计算时间远远小于构造一棵最短路径树的时间；

2)可以计算出节点 c 到节点 d 的所有符合 LFC 规则的下一跳集合。

4　ERPISPF 算法

4.1　算法描述

为了高效地解决 3.2 节中提出的问题，首先介绍两个定理，并证明它们的正确性。

定理 1　已知计算节点 c 和以其为根的最短路径树 $spt(c)$。对于网络中的任意节点 $d(d\neq c,d\neq x)$，如果 $d\notin D(spt(c),x)$ 和 $d\in D(spt'(c),x)$ 同时成立，则 $cost(x,d)<cost(c,d)+cost(c,x)$。其中 $x\in N(c)$，$spt'(c)$ 表示将链路 (c,x) 的代价调整为 $-w(c,x)$ 时对应的新的以节点 c 为根的最短路径树。

证明：假设在 $spt(c)$ 中，节点 c 到节点 d 的默认下一跳为 $y(y\neq x)$，则可以得到 $cost(c,d)=cost(c,y)+cost(y,d)$。因为 $d\in D(spt'(c),x)$，所以 $cost'(c,d)=cost(c,x)+cost(x,d)$，其中 $cost'(c,d)$ 表示在 $spt'(c)$ 中，节点 c 到节点 d 的代价。由于在 $spt'(c)$ 中，$cost(c,x)=-w(c,x)$，则：

$$cost'(c,d)=cost(x,d)-w(c,x) \tag{1}$$

因为 $d\notin D(spt(c),x)$ 和 $d\in D(spt'(c),x)$，所以：

$$cost'(c,d)<cost(c,d) \tag{2}$$

合并式(1)和式(2)可以得到 $cost(x,d)<cost(c,d)+w(c,x)=cost(c,d)+cost(c,x)$。因此，该定理成立。

定理 2　已知计算节点 c 和以其为根的最短路径树 $spt(c)$。对于网络中的任意节点 $d(d\neq c,d\neq x)$，如果 $d\notin D(spt(c),x)$ 和 $d\in D(spt'(c),x)$ 同时成立，则节点 x 可以作为节点 c 到节点 d 的备份下一跳，即 $bn(c,d)=bn(c,d)\cup\{x\}$。其中 $x\in N(c)$，$spt'(c)$ 表示将链路 (c,x) 的代价调整为 $-w(c,x)$ 时对应的新的以节点 c 为根的最短路径树。

证明：由定理 1 可知，$cost(x,d)<cost(c,d)+cost(c,x)$，根据 LFC 规则可知节点 x 可以作为节点 c 到节点 d 的备份下一跳，因此该定理成立。

上述定理给出了计算符合 LFC 规则的备份下一跳的方法。从定理 1 和定理 2 可知，在节点 c 上，如果需要判断其邻居节点 x 可以作为哪些节点的备份下一跳，则将链路 (c,x) 的代价调整为 $-w(c,x)$ 即可；对于某个节点 d，如果在 $spt(c)$ 中，$d\notin D(spt(c),x)$，在 $spt'(c)$ 中，$d\in D(spt'(c),x)$，则节点

x 可以作为节点 c 到节点 d 的备份下一跳。同理，可以判断节点 c 的其余邻居节点的情况。

算法 ERPISPF 的伪代码如算法 1 所示。

算法 1 ERPISPF

```
Input:G=(V,E)和 spt(c)
Output: ∀v∈V,bn(c,v)
1. FOR x∈N(c) DO
2.   weight←w(c,x)
3.   w(c,x)←w(c,x)
4.   利用 i-SPF 算法重新构造 spt'(c)
5.   FOR v∈V,v≠c,v≠x DO
6.     IF v∉D(spt(c),x) and v∈D(spt'(c),x) then
7.       bn(c,v)=bn(c,v)∪{x}
8.     ENDIF
9.   ENDFOR
10.  w(c,x)←weight
11. ENDFOR
```

算法 ERPISPF 给出了计算节点 c 到网络中所有其他节点的备份下一跳的执行过程。算法的输入为网络拓扑结构和以节点 c 为根的最短路径树 $spt(c)$，输出为节点 c 到网络中其余所有节点的备份下一跳集合。对于节点 c 的任意邻居节点 x，首先将二者之间链路 (c,x) 的代价调整为 $-w(c,x)$（算法第 3 行），然后利用 i-SPF 算法重新构造 $spt'(c)$（算法第 4 行）。对于网络中任意节点 v，如果在 $spt(c)$ 中，节点 v 不是节点 x 的子孙节点，并且在 $spt'(c)$ 中，节点 v 是节点 x 的子孙节点，则根据定理 2 可知，节点 x 可以作为节点 c 到节点 v 的备份下一跳（算法第 5－9 行），最后恢复链路 (c,x) 的代价（算法第 10 行）。

4.3 算法性能分析

本节分析算法的性能，包括算法的时间复杂度和计算出的备份下一跳的数量。通过定理 3 可知，ERPISPF 的计算开销小于构造一棵最短路径树的时间。通过定理 4 可知，ERPISPF 可以计算出所有满足 LFC 规则的备份下一跳。下面给出定理 3 和定理 4，并且证明它们的正确性。

定理 3 算法 ERPISPF 的时间复杂度为 $O(|E|\lg|V|)$。

证明：为了计算节点 c 到网络中其他所有节点的满足 LFC 规则的备份下一跳，算法 ERPISPF 需要调用 k 次 i-SPF 算法，其中 k 为节点 c 的邻居数量。当链路 (c,l) 的代价调整为 $-w(c,l)$ 时，用 $M(l)$ 表示当链路代价发生变化时受影响的节点的数量，用 $P(l)$ 表示受影响的边的数量。因此算法对应的时间复杂度为 $\sum_{l=1}^{k}M(l)\lg M(l)\leqslant\lg|V|*\sum_{l=1}^{k}M(l)=O(|E|\lg|V|)$，即 ERPISPF 的时间复杂度小于构造一棵最短路径树的时间复杂度。

定理 4 算法 ERPISPF 可以计算出所有满足 LFC 规则的节点 c 到网络中其他节点的备份下一跳集合。

证明：用反证法来证明该定理。假设存在一个节点 v，$v\neq x$，$v\neq c$，节点 $x(x\in N(c))$ 满足 LFC 条件 $cost(x,v)<cost(c,v)+cost(c,x)$，但是利用算法 ERPISPF 计算出的节点 c 到节点 v 的备份下一跳集合不包括节点 x。根据算法 ERPISPF 可知，当计算节点 x 可以作为某些节点的备份下一跳时，首先将链路 (c,x) 的代价调整为 $-w(c,x)$，然后根据 i-SPF 构造新的最短路径树 $spt'(c)$，由于 $cost(x,v)<cost(c,v)+cost(c,x)$，则在 $spt'(c)$ 中节点 v 必定为节点 x 的子孙节点，根据定理 1 和定理 2 可知，节点 x 为节点 c 到节点 v 的备份下一跳，与前提假设矛盾，即该定理成立。

5 实验和结果分析

5.1 实验

本节将通过实验来测试算法的性能，下面首先介绍实验采用的网络拓扑结构，然后给出评价度量和实验方法。

(1)网络拓扑

为了通过实验来验证算法的性能，本文在多种网络拓扑结构上运行了 LFC，TBFH，DMPA 和 ERPISPF。实验用到的拓扑结构包括真实网络拓扑结构 Abilene[1)]、测量网络拓扑结构[30]、利用 Brite[2)] 软件生成的网络拓扑结构。

1)Abilene 是一个简单的拓扑结构，由 11 个节点和 14 条边构成。

2)测量拓扑由 Rocketfuel 项目公布，可以从网络中获取相应的拓扑数据。表 1 列出了测量拓扑的具体参数。

表 1 Rocketfuel 拓扑

Table 1 Rocketfuel topology

AS 号码	AS 名称	节点数量	链路数量
1221	Telstra	108	153
1239	Sprint	315	972
1755	Ebone	87	162
3257	Tiscali	161	328
3967	Exodus	79	147
6461	Abovenet	128	372

3)Brite 是由波士顿大学开发的一款生成拓扑的软件。表 2 列出了该软件的详细运行参数。

表 2 Brite 生成拓扑结构的参数设置

Table 2 Parameters setting for Brite topology

Model	N	HS	LS	m	Node Placement	GrowthType	alpha	beta	BWDist	BwMin-BwMax	model
Waxman	20-1000	1000	100	2-25	Random	Incremental	0.15	0.2	Constant	10.0-1024.0	Router-only

(2)评价度量

本文的评价度量包括算法的计算开销和故障保护率。其中计算开销可以表示为算法需要构造最短路径树的数量，故障保护率可以表示为网络中可以被保护的链路数量与网络中所有链路数量的比值。

(3)实验方法

本文利用 C++语言实现了 LFC，TBFH，DMPA 和 ERPISPF，然后在上述拓扑结构中分别运行这 4 种算法，实验结

1) https://www.internet2.edu/products-services/advanced-networking

2) http://www.cs.bu.edu/brite/

果为 20 次计算结果的平均值。

5.2　计算开销

首先描述不同算法在真实拓扑和测量拓扑中的结果。图 1 给出了在真实拓扑和测量拓扑中不同算法对应的计算开销。从中可以看出：LFC 的计算开销最大，TBFH 需要计算两棵最短路径树，DMPA 需要计算一棵最短路径树，ERPISPF 的计算开销小于 LFC，TBFH 和 DMPA 的计算开销，其计算开销远远小于构造一棵最短路径树的计算开销。

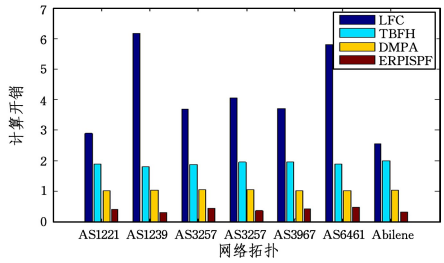

图 1　真实拓扑和测量拓扑中不同算法的计算开销

Fig. 1　Computation overhead of different algorithms in real topology and measured

图 2 给出了不同算法的计算开销与拓扑之间的关系，从中可知，随着拓扑的增加，不同算法的计算开销基本不变，ERPISPF 的计算开销远远小于 LFC，TBFH 和 DMPA 的计算开销。

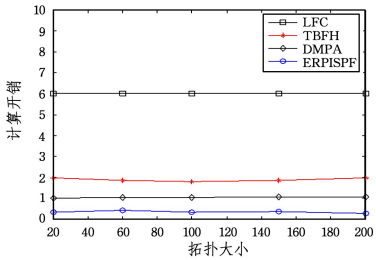

图 2　不同算法的计算开销与拓扑大小的关系

Fig. 2　Relationship between computation overhead and topology size of different algorithms

图 3 给出了不同算法的计算开销与节点平均度之间的关系，从中可知，随着节点平均度的增加，LFC 算法的计算开销随之增加，而 TBFH，DMPA 和 ERPISPF 的计算开销与节点平均度几乎没有关系。在 4 个算法中，ERPISPF 的计算开销仍然是最小的。

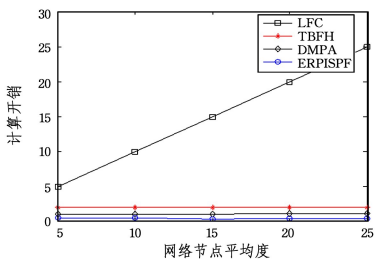

图 3　不同算法的计算开销与节点平均度的关系

Fig. 3　Relationship between computation overhead and average node degree of different algorithms

通过上述实验可知，与 LFC 方法相比，ERPISPF 大约可以降低 94％的计算开销。

5.3　故障保护率

图 4 给出在真实拓扑和测量拓扑中不同算法对应的故障保护率，从中可以看出 LFC 和 ERPISPF 的故障保护率是相同的，TBFH 和 DMPA 的故障保护率明显小于 LFC 和 ERPISPF 的故障保护率。

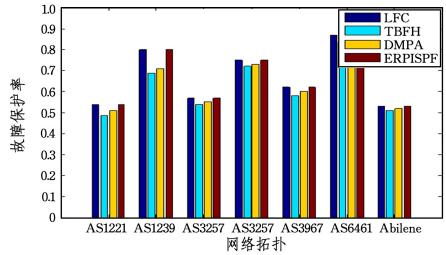

图 4　真实拓扑和测量拓扑中对应的故障保护率

Fig. 4　Protection rate in real to pology and measured

图 5 给出不同算法的故障保护率与拓扑大小之间的关系，从中可知，LFC 和 ERPISPF 的故障保护率明显优于 TBFH 和 DMPA，并且二者具有相同的性能。

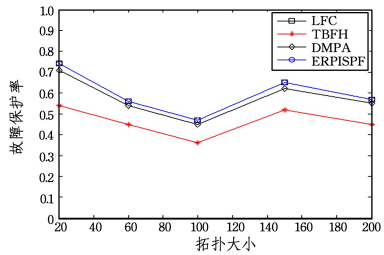

图 5　不同算法的故障保护率与拓扑大小的关系

Fig. 5　Relationship between protection rate and topology size of different algorithms

图 6 给出不同算法的故障保护率和网络节点平均度之间的关系，从中可知，随着网路节点平均度的增加，4 个算法对应的故障保护率都随之增加，LFC 和 ERPISPF 依然具有同样的故障保护率。

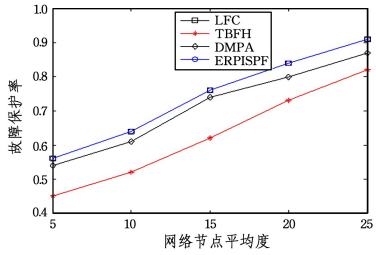

图 6　网络节点平均度与故障保护率的关系

Fig. 6　Relationship between protection rate and average node degree

结束语　针对已有实现 LFC 算法的复杂度较高的问题，本文深入研究了已有方案的缺陷，提出了一种基于 i-SPF 的高效路由保护方案 ERPISPF。该方案只需要在已有的最短路径树的基础上根据链路代价变化动态地调整该树，然后在新的最短路径树的基础上计算符合 LFC 规则的备份下一跳集合。实验结果表明，与 LFC 相比，ERPISPF 不仅具有较小的计算开销，并且具有相同的故障保护率。因此 ERPISPF 更适合在实际网络中部署，从而为 ISP 提供一种兼顾执行效率和故障保护率的域内路由保护方案。

参 考 文 献

[1] ANTONAKOPOULOS S,BEJERANO Y,KOPPOL P. Full Protection Made Easy: The DisPath IP Fast Reroute Scheme [J]. IEEE/ACM Transactions on Networking, 2015, 23(4): 1229-1242.

[2] XU A,BI J,ZHANG B. Failure Inference for shortening traffic Detours[C]// Proceedings of the International Symposium on Quality of Service. Beijing,China,2016:1-10.

[3] KRIST P. Scalable and Efficient Multipath Routing:Complexity and Algorithms[C]// Proceedings of the IEEE International Conference on Network Protocols. San Francisco,CA,2015:376-385.

[4] YANG Y,XU M,LI Q. Tunneling on demand: A lightweight approach for IP fast rerouting against multi-link failures[C]// Proceedings of the International Symposium on Quality of Service. Beijing,China,2016:1-10.

[5] ELHOURANI T,GOPALAN A,RAMASUBRAMANIAN S. IP fast rerouting for multi-link failures[C]// Proceedings of the Conference on Computer Communications. Toronto, Canada, 2014:2148-2156.

[6] MARKOPOULOU A,IANNACCONE G,BHATTACHARYYA S. Characterization of failures in an operational ip backbone network[J]. IEEE/ACM Transactions on Networking, 2008,16(4):749-762.

[7] GOPALAN A,RAMASUBRAMANIAN S. IP Fast Rerouting and Disjoint Multipath Routing With Three Edge-Independent Spanning Trees[J]. IEEE/ACM Transactions on Networking, 2016,24(3):1336-1349.

[8] XU M,HOU M,WANG D,et al. An efficient critical protection scheme for intra-domain routing using link characteristics[J]. Computer Networks,2013,57(1):117-133.

[9] FRANCOIS P,BONAVENTURE O. Avoiding transient loops during the convergence of link-state routing protocols[J]. IEEE/ACM Transactions on Networking, 2007, 15(6): 1280-1292.

[10] NELAKUDITI S,LEE S,YU Y,et al. Fast local rerouting for handling transient link failures[J]. IEEE/ACM Transactions on Networking,2007,15(2):359-372.

[11] YALLOUZ J,ROTTENSTREICH O,BABARCZI P,et al. Optimal Link-Disjoint Node-"Somewhat Disjoint" Paths[C]// Proceedings of the IEEE International Conference on Network Protocols. Singapore,2016:1-10.

[12] KWONG K,GAO L,GUERIN R,et al. On the feasibility and efficacy of protection routing in IP networks[J]. IEEE/ACM Transactions on Networking,2011,19(5):1543-1556.

[13] LEE S,YU Y,NELAKUDITI S,et al. Proactive vs Reactive Approaches to Failure Resilient Routing[C]// Proceedings of the IEEE International Conference on Computer Communications. Hong Kong,China,2004:1-9.

[14] FRANCOIS P,BONAVENTURE O. Avoiding transient loops during the convergence of link-state routing protocols[J]. IEEE/ACM Transactions on Networking, 2007, 15(6): 1280-1292.

[15] FRANCOIS P,FILSFILS C,EVANS J,et al. Achieving sub-second igp convergence in large ip networks[J]. Computer Communication Review,2005,35(3):35-44.

[16] CLAD F,MERINDOL P,VISSICCHIO S,et al. Graceful Router Updates for Link-State Protocols[C]// Proceedings of the IEEE International Conference on Network Protocols. Göttingen,Germany,2013:1-10.

[17] ZHANG B,WU J,BI J. RPFP:IP fast reroute with providing complete protection and without using tunnels[C]// Proceedings of the International Symposium on Quality of Service. Montreal,Canada,2013:1-10.

[18] XU M,YANG Y,LI Q. Selecting Shorter Alternate Paths for Tunnel-based IP Fast ReRoute in Linear Time[J]. Computer Networks,2012,56(2):845-857.

[19] ENYEDI G,RÉTVÁRI G,SZILÁGYI P,et al. IP Fast Re-Route:Lightweight Not-Via without Additional Addresses[C]// Proceedings of the Conference on Computer Communications. Rio de Janeiro,Brazil,2009:157-168.

[20] ZHANG M,LIU B,ZHANG B. Multi-Commodity Flow Traffic Engineering with Hybrid MPLS/OSPF Routing[C]// Proceedings of the Global communications Conference. Honolulu, USA,2009:1-6.

[21] ATLAS A K,ZININ A. Basic Specification for IP Fast Reroute: Loop-Free Alternates[S]. RFC 5286,2008:1-31.

[22] YANG X,WETHERALL D. Source selectable path diversity via routing deflections[C]// Proceedings of the ACM Special Interest Group on Data Communication. Pisa,Italy,2006:159-170.

[23] GENG H J,SHI X G,WANG Z L,et al. Efficient implementation method for LFA[J]. Journal of Software,2018,29(12):3904-3920. (in Chinese)
耿海军,施新刚,王之梁,等. LFA 算法的一种高效实现方法[J]. 软件学报,2018,29(12):3904-3920.

[24] Junos OS Routing Protocols Configuration Guide [OL]. https://www.juniper.net/documentation/en_US/junos12.3/information-products/topic-collections/release-notes/12.3/junos-release-notes-12.3.pdf.

[25] MARKOPOULOU M P,FRANCOIS P,BONAVENTURE O,et al. An efficient algorithm to enable path diversity in link state routing networks [J]. Computer Networks, 2011,55(5):1132-1149.

[26] GENG H,SHI X,YIN X,et al. Dynamic distributed algorithm for computing multiple next-hops on a tree[C]// Proceedings of the IEEE International Conference on Network Protocols. Göttingen,Germany,2013:1-10.

[27] ENYEDI G,RÉTVÁRI G,SZILÁGYI P,et al. IP Fast Re-Route: Lightweight Not-Via[C]// Proceedings of the IEEE International Conference on Computer Communications. Rio De Janeiro,Brazil,2009:1-9.

[28] RÉTVÁRI G,et al. IP fast ReRoute: Loop Free Alternates revisited[C]// Proceedings of the IEEE Conference on Computer Communications. Shanghai,China,2011:2948-2956.

[29] RETVARI G,CSIKOR L,TAPOLCAI J,et al. Optimizing IGP link costs for improving IP-level resilience[J]. Computer Communications,2013,36(6):645-655.

[30] SPRING N,MAHAJAN R,WETHERALL D,et al. Measuring isp topologies with rocketfuel[C]// IEEE/ACM Transactions on Networking. 2004:2-16.