

一种适合低连接度拓扑的集中式保护路由机制*

谭晶, 罗军舟, 李伟

(东南大学 计算机科学与工程学院, 江苏 南京 210096)

通讯作者: 谭晶, E-mail: tanjing@seu.edu.cn

摘要: 在集中式路由中, 由路由控制平台统一计算路由表进行分发, 路由器不再具备决策能力, 需要预先构建一种具备保护功能的路由机制, 使得路由器的下游路径失效后都有立即可用的备份路径, 确保报文的最小损失, 已有的集中式保护路由机制在低连接度拓扑上保护效果不佳. 为了解决该问题, 提出了一种适合低连接度拓扑的集中式域内保护路由机制, 允许失效处的相邻节点在没有可用路径时将报文返回至其上游节点, 由有可用备份路径的上游节点通过备份路径发送, 确保单个节点或连接失效后报文的最小损失. 证明了为给定拓扑构建最优保护路由的问题是一个 NP-hard 问题, 并且提出了解决该问题的三阶段启发式算法. 在各种类型的拓扑中验证了启发式算法的性能. 实验结果表明, 该方法优于已有保护路由方案.

关键词: 域内路由; 集中式; NP-hard

中图法分类号: TP301 **文献标识码:** A

中文引用格式: 谭晶, 罗军舟, 李伟. 一种适合低连接度拓扑的集中式保护路由机制. 软件学报, 2013, 24(3): 575-592. <http://www.jos.org.cn/1000-9825/4250.htm>

英文引用格式: Tan J, Luo JZ, Li W. Centralized intra-domain protection routing mechanism for low-connectivity topology. Ruanjian Xuebao/Journal of Software, 2013, 24(3): 575-592 (in Chinese). <http://www.jos.org.cn/1000-9825/4250.htm>

Centralized Intra-Domain Protection Routing Mechanism for Low-Connectivity Topology

TAN Jing, LUO Jun-Zhou, LI Wei

(School of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

Corresponding author: TAN Jing, E-mail: tanjing@seu.edu.cn

Abstract: In centralized routing, routing tables are computed by a control platform, but routers do not make decisions anymore. It is necessary to build backup paths for each router so that packets can be forwarded after failure. Existing protecting routing schemes cannot work well in low-connectivity topology. To solve this problem, a method for building centralized protection routing is proposed in this paper. In this method, packets can be pushed back to the upstream nodes and sent by the backup paths of upstream nodes if current node has no valid paths after failure. The problem of building optimal protection routing for a given topology is proved to be NP-hard and a heuristic algorithm is proposed. The algorithm is verified in various topologies and the experimental results show that it is better than existing schemes.

Key words: intra-domain routing; centralized; NP-hard

Internet 采用完全分布式的路由决策方式, 路由决策功能被部署在路由器上, 路由器既进行路径选择又要转发报文. 这种完全分布式的决策方式保证了路由的可靠性, 然而却导致了许多的不一致, 带来了路由环路等问

* 基金项目: 国家重点基础研究发展计划(973)(2010CB328104); 国家自然科学基金(60903161, 60903162, 61003257, 61070158, 61070161, 61070210); 国家科技支撑计划(2010BAI88B03, 2011BAK21B02); 高等学校博士点学科专项科研基金(20110092130002); 江苏省自然科学基金(BK2008030); 江苏省网络与信息安全重点实验室资助项目(BM2003201); 教育部计算机网络与信息集成重点实验室(东南大学)资助项目(93K-9)

收稿时间: 2011-11-30; 定稿时间: 2012-04-09

题,加剧了网络控制的复杂度^[1].并且,越来越多的控制逻辑被添加到路由器上,导致了路由器负载越来越大,限制了其报文转发功能的进一步提升.

为了解决完全分布式路由带来的问题,集中式路由重新引起了研究者的兴趣,特别是在可扩展性较易保证的域内路由中.如图 1 所示,在集中式域内路由^[2-4]中,决策逻辑从路由器上剥离,在每个自治域中构建一个逻辑集中的路由控制平台进行路由决策.路由器负责收集拓扑信息并且向路由控制平台通告,路由控制平台根据当前的网络拓扑生成每个路由器的路由表,并且将路由表发送至每个路由器上进行配置,路由器根据配置好的路由表转发报文.逻辑集中的路由控制方式减小了路由器的工作负载,使得路由器更加专注于分组转发;同时,集中式域内路由避免了分布式域内路由中设定公共连接权值从而间接影响路由的控制方式,直接为每个目的地址计算独立的路由树,提高了路由控制的效率.

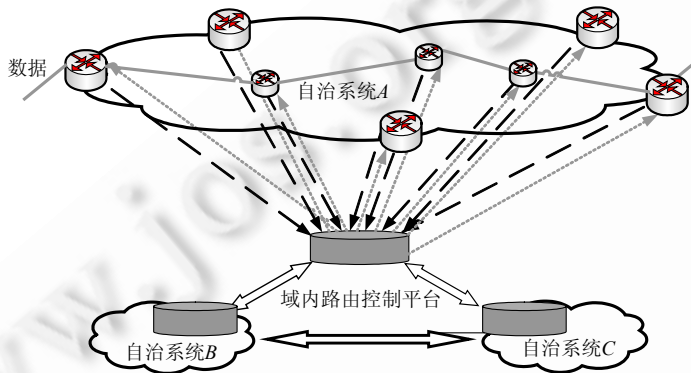


Fig.1 Centralized intra-domain routing

图 1 集中式域内路由示意图

虽然集中式域内路由有着许多优点,但其缺点也同样明显,比如,较慢的失效反应速度.在集中式域内路由中,失效的相邻节点感知到失效后,产生失效通告发送至控制平台,控制平台生成新的路由树,并且对受影响的路由器进行配置.在该阶段中,失效的上游节点可能失去到达目的节点的路径并且导致报文的丢失.解决该问题的可行方法为,在路由器上预先构建下游路径失效后的备份路径,当某条下游路径失效后在控制节点完成对路由的重新配置前暂时启用备份路径进行传输,保证报文的最小损失.在集中式路由中,路由器被极大地简化了,采用隧道等方式构建备份路径会增大路由器的负载,降低传输的效率.较好的失效应对方式为,构建一种具备失效保护功能的路由,在失效后快速切换至备份路径且不需要改变报文的转发方式.在基于目的地址的报文转发方式下,最佳的保护路由构建方式为,在每个路由器的路由表中预先为每个目的地址设定备份下一跳,路由器感知到基本下一跳失效后本地激活备份下一跳进行传输,最大程度地避免失效后报文的损失.由于网络中最常见的失效为单个节点或者连接失效^[3],构建单个节点或者连接失效后的保护路由有着尤其重要的意义.在本文以下的研究中,假设网络中同时最多只有一个节点或者一条连接失效.

已有研究者对该问题进行了研究.MARA^[5]采用增大节点下一跳的数目构建保护路由,使得尽可能多的节点有两个以上的下一跳.然而,如果相邻节点间只有一条边,无论拓扑连接度多高,MARA 中至少有一个节点无法得到保护.Kwong 等人提出的 PR 算法在下一跳的基础上引入了 joker 连接的方式构建保护路由^[3].Joker 连接在网络中没有失效时为一条无向连接,只有在失效发生后其才被设定为一条有向连接.在相同的拓扑条件下,PR 能够比 MARA 保护更多的节点,且如果拓扑的连接度足够高,PR 能够保护所有的节点.然而,MARA 和 PR 对拓扑连接度要求较高,需要较大的连接冗余度,增大了网络建设的成本.

为了提升保护路由在低连接度拓扑上的有效性,降低网络建设的成本,本文对 Kwong 的方法进行了扩展,将回溯的思想引入域内保护路由构建中,提出了一种适合低连接度拓扑的域内保护路由机制.该机制的主要思想为,允许失效处的相邻节点在没有可用路径时将报文返回至其上游节点,由有可用备份路径的上游节点通过

备份路径发送,确保在单个节点或者连接失效后报文的最小损失.本文的主要贡献是:

- (1) 将回溯思想引入域内保护路由构建中,提出了一种适用于低连接度拓扑的集中式域内保护路由机制,在较低连接度拓扑上具备较好的保护效果;
- (2) 证明了为给定拓扑构建最优保护路由,使得最多的节点能够被保护的问题为一个 NP-hard 问题;
- (3) 提出了一种解决该问题的三阶段启发式算法.

本文第 1 节介绍相关工作.第 2 节对适合低连接度拓扑的保护路由机制的主要思想进行阐述并且对为给定拓扑构建最优保护路由的问题进行形式化描述.第 3 节证明为给定拓扑构建最优保护路由问题为一个 NP-hard 问题,并且分析最优保护路由与拓扑特性间的关系.第 4 节提出一个构建最优保护路由的启发式算法.第 5 节验证所提启发式算法的性能.第 6 节总结全文并提出未来工作.

1 相关工作

IPFRR(IP fast re-routing)^[6]包括了当前主要的域内路由可靠性方案,其主要目的是为确保检测到失效后路由的快速恢复.已有 IPFRR 成果分为两类:不改变基于目的地址的报文转发方式以及采用其他的报文转发方式.

第 1 类的 IPFRR 成果主要有 ECMPs^[7],LFA^[8],DIV_R^[9],O2^[10],PR^[3]以及 MARA^[5].其中,ECMPs,LFA 以及 DIV_R 为分布式的路由算法,MARA,O2 以及 PR 为集中式的路由方案.ECMPs 是对 OSPF 算法的改进,其主要思想是,如果某个节点有多条代价相等的下游最优路径,则该节点同时采用多条路径进行传输.然而,ECMPs 中权值的设定较为复杂,其保护效果较差.LFA 采用一些规则确保邻居节点不会选择对方成为备份节点,从而避免了构建备份路径时产生环路.然而,LFA 并不适用在集中式路由中.DIV_R 通过一个分布式的算法计算一个度量指标最大化每个节点的下一跳的数目.DIV_R 可以有效地应对连接失效,但是其无法有效应对节点失效.MARA 通过为每个节点构建多个下一跳的方式构建保护路由,其提出最大化每个节点到目的节点的连接度,使得整体路由保护效果最好.然而,即使网络连接度再高,MARA 中至少有一个节点无法得到保护.相比于 MARA,O2 与 PR 除了采用直接下一跳,还引入了 joker 连接构建保护路由,确保了在相同连接度条件下能够比 MARA 保护更多的节点.O2 与 PR 的主要区别为,PR 比 O2 多考虑了网络性能的需求.但是,O2 与 PR 不适用于连接度较低的网络.

第 2 类的 IPFRR 方案主要有文献[11–14],其希望对报文进行标记或者对路由器的转发方式进行改变,从而提高路由应对失效的能力.比如,文献[11]提出了基于接口的报文转发方式,在发生失效时能够避免环路.然而,该方法会增大路由表规模,降低路由查找的效率.文献[12]提出预先在每个路由器上构建多个路由配置方案,每个配置应对不同的连接失效,在失效发生后所有路由器从一种配置转换到另一种配置中.但是,该方法复杂度较高,路由器需要存储各种配置方案且在各方案间切换,负载较大.文献[13]提出了构建绕过失效的隧道,确保失效后最少的报文损失报文,但该方法会带来额外的路径构建开销.文献[14]提出在报文头中标记网络的失效信息,使得其他路由器能够诊断出网络失效位置并且选择合适的备份路径进行传输.该方法的主要缺点是需要对报文进行标记,降低了报文的处理速度,增大了路由器的负载.

此外,有研究者采用回溯的方式进行域间路由保护.Kushman 等人提出了一种可回溯的 BGP 路由保护方案,确保在失效后只要网络物理上连通报文就不会丢失^[15].其提出在路径失效后,下游节点可以将上游节点收到的报文返回至上游节点,由上游节点通过其他路径将报文发送.该方法能够保证只要网络在物理上是连通的,报文总能够送达目的节点.该工作对于域内路由的研究有一定的启发性.

2 适合低连接度拓扑的域内保护路由问题描述

为了便于理解,本节首先对保护路由机制采用的 3 种保护方式进行了说明,在此基础上给出域内保护路由问题的形式化定义.

根据文献[3]的阐述,网络中最常见的失效为单个节点或连接失效,在本文以下的研究中,我们假设网络中同时只有一个节点或连接失效.

在本文的机制中,下游路径失效后路由器有 3 种可行的保护方式,按照触发优先级由高到低依次为:

- 通过其他下一跳传送;
- 通过与邻居间的 joker 连接**进行传送;
- 将报文回溯至上游节点,由上游节点通过备份路径传送.

图 2 分别给出了这 3 种路由保护方式的示例.图 2(a)中在没有失效的情况下节点 1 包含两个下一跳:节点 2 和节点 4;连接 1-2 失效后,节点 1 可以直接通过另一个下一跳节点 4 发送报文,保证报文的最小损失.图 2(b)中节点 5 包含基本下一跳节点 d 以及一个 joker 连接 5-4,连接 5- d 失效后,节点 5 采用 joker 连接 5-4 进行传输.图 2(c)中给出了连接 6- d 失效后,节点 6 通过回溯下一跳节点 4 进行保护.图 2(c)中,在没有失效的情况下节点 6 只有基本下一跳节点 d ,没有其他可用的基本下一跳以及 joker 连接.一旦连接 6- d 失效,节点 6 会向上游节点 4 发出其下游路径失效通告,并且将收到的报文都发送至节点 4,节点 4 收到节点 6 的通告后,采用 joker 连接 4-5 进行传输.在此期间,所有发送到节点 6 的报文都被 6 发回至节点 4 并通过路径 4-5- d 发送,实现最小损失的报文传输.

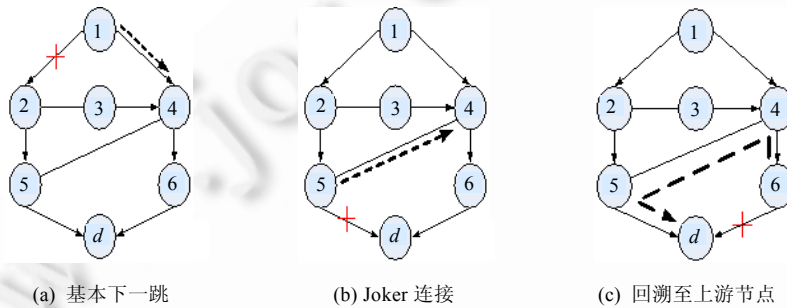


Fig.2 Three protection routing schemes in this paper

图 2 本文包括的 3 种路由保护方式

基于以上 3 种保护方式,本文解决的问题为:对于给定拓扑,如何构建最优的保护路由,使得最多数目的路由器至少被一种方式所保护.下面我们对问题进行形式化定义.

采用无向图 $G=(V,E)$ 表示域内拓扑,其中,节点表示路由器,节点间的边表示路由器间的连接.采用 $R_d=(V,E_d)$ 表示目的地址为 d 的网络路由, E_d 通过给 E 中的部分边赋予方向获得.采用 $\langle i,j \rangle$ 表示端点分别为 i 和 j 的无向边, (i,j) 表示从 i 指向 j 的有向边.

定义 1. 在单个节点或连接 f 失效后,剩余图和以 d 为目的的节点的剩余路由分别采用 G^f 和 R_d^f 标记. G^f 和 R_d^f 通过从 G 和 R_d 中移走失效部件得到. $N_{G^f}(i)$ 表示在 G^f 中节点 i 的邻居节点集.

定义 2. 在路由 R_d 中,如果存在从 $i \sim j$ 的路径,则 i 是 j 的上游节点, j 是 i 的下游节点.

定义 3(基础下一跳(PNH)). 如果 $\langle i,j \rangle \in E_d$,则节点 j 称为 i 的基础下一跳.

定义 4(joker 下一跳(JNH)). 在路由 R_d 中,如果单个节点或者连接失效影响了节点 i 的 PNH,存在一个节点 $k \in N_{G^f}(i)$ 使得如下 3 个条件满足,则 k 为 i 的 JNH:

- 1) 在 R_d^f 中,节点 k 不是 i 的上游节点;
- 2) 在 R_d^f 中,节点 k 不是 i 的下游节点;

** Joker 连接在没有失效时为无向连接,只有在某个端节点没有下游路径时,才会变成有向连接.例如在图 2(b)中,连接 5-4 为一条 joker 连接.在没有失效的情况下,如果该连接被给定方向,则节点 5 和节点 4 中的一个必定得不到保护.而在无失效情况下,该连接不给定方向,节点 5 感知到连接 5- d 失效后设定路由指向节点 4,节点 4 感知到连接 4-6 失效后设定路由指向节点 5,只要连接 5- d 和连接 4-6 不同时失效,节点 4 和节点 5 都能得到保护.

3) 在 R_d^f 中,节点 k 和它的所有下游节点都有至少一个 PNH.

定义 5(回溯保护下一跳(BPNH)). 在路由 R_d 中,如果单个节点或连接 f 失效影响了节点 i 的 PNH,存在一个节点 $k \in N_{G^f}(i)$ 使得如下两个条件满足,则 k 为 i 的 BPNH:

- 1) 在 R_d^f 中,节点 k 是 i 的上游节点;
- 2) 在 R_d^f 中,节点 k 和它的所有下游节点都有至少一个 PNH;或者节点 k 存在一个 JNH 节点 J_k ,且在 R_d^f 中 J_k 和其所有下游节点都有至少一个 PNH.

假定在路由 R_d 中最多可以回溯 n 跳,则定义 6 给出了节点 i 被保护的条件.

定义 6. 在路由 R_d 中, $i \neq d$,如果节点 i 的 PNH 失效,以下 3 个条件中至少一个满足,则 i 是被保护的:

- 1) i 存在其他可用的 PNH;
- 2) i 存在至少一个 JNH;
- 3) i 的上游 $n-1$ 跳内存在一个节点 k , k 有一个 BPNH.

定义 7. 对于给定目的节点 d 的拓扑 G , R_d 是最优的当且仅当其中被保护的节点数目最大.

3 NP-hard 证明及性质分析

在本节中,首先证明了为给定拓扑构建最优保护路由从而使得最多的节点被保护的问题为一个 NP-hard 问题,然后分析了最优保护路由与拓扑特性之间的关系.

3.1 NP-hard 证明

首先证明了当允许回溯一跳时,本文的最优保护路由构建问题为一个 NP-hard 问题.通过对回溯一跳问题证明过程的简单改进,可以证明回溯 $n(n>1)$ 跳时的最优保护路由构建问题也是 NP-hard 的.

回溯一跳时的 NP-hard 证明通过对文献[3]的 NP-hard 证明过程的改进得到,具体思路为,将 3-SAT 问题^[16]与回溯一跳的最优保护路由问题在多项式时间内相互转化,证明回溯一跳的最优保护路由问题也是一个 NP-hard 问题.

设定 Φ 是包含 s 个变量 x_1, x_2, \dots, x_s 和 q 个子句 P_1, P_2, \dots, P_q 的 3CNF 表达式.

比如, $\Phi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_4) \wedge (x_1 \vee x_3 \vee \overline{x_4})$ 包含 4 个变量 x_1, x_2, x_3, x_4 和 3 个子句

$$P_1 = x_1 \vee x_2 \vee x_3, P_2 = \overline{x_1} \vee \overline{x_2} \vee x_4, P_3 = x_1 \vee x_3 \vee \overline{x_4}.$$

首先对 Φ 中的子句以及变量分别构造一些子图,将它们组合起来构建对应于 Φ 的连接图.

a) 对每个子句 $P_j = X_j \vee Y_j \vee Z_j$, 构造如图 3 所示的结构 C_j , 子句中的 X_j, Y_j, Z_j 分别对应图中的 X_j, Y_j 及 Z_j .

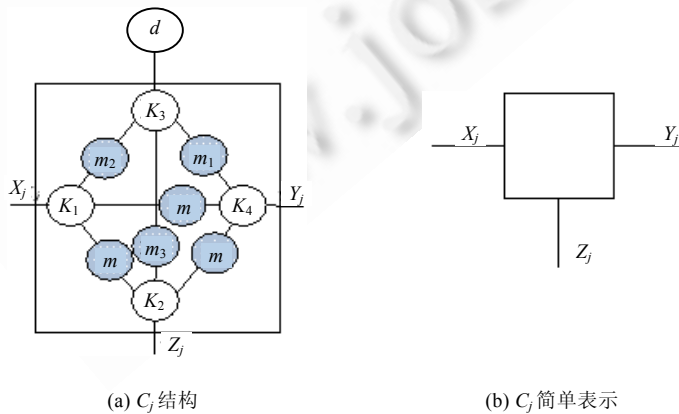


Fig.3 Structure of C_j and its simple representation

图 3 C_j 结构及其简单表示

引理 3.1. 如果 C_j 中的每个节点都被保护,则图 3 中的 X_j, Y_j, Z_j 中至少有一条从 C_j 流出.

证明:通过反证法对该引理进行证明.假设 C_j 中 X_j, Y_j, Z_j 都不从 C_j 流出,如果 K_3-d 失效,基于本文的 3 种路由保护方式,最大的影响范围为节点 m_1, m_2 以及 m_3 ,不会对 X_j, Y_j, Z_j 的方向产生影响.根据之前的假设, X_j, Y_j, Z_j 中没有连接从 C_j 流出.在该情况下, C_j 中的所有节点都无法流出,一定无法到达 d, C_j 中的节点一定是无法被保护的.因此假设不成立,引理 3.1 得证. \square

b) 对于每个变量 x_i ,构造如图 4 所示的结构 u_i .

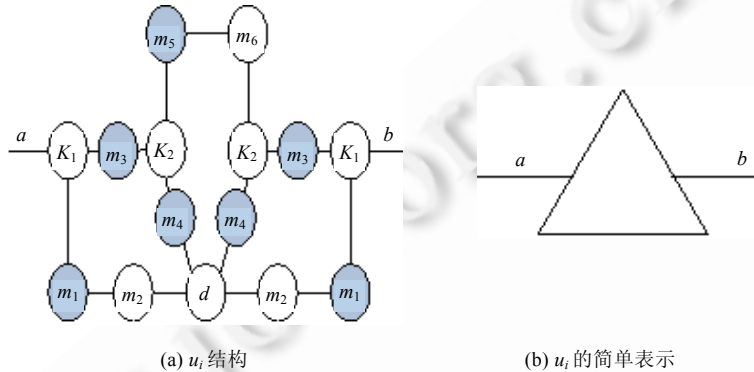


Fig.4 Structure of u_i and its simple representation

图 4 u_i 结构及其简单表示

引理 3.2. 如果 u_i 中的每个节点都被保护,则图 4 中的 a 或者 b 必须从 u_i 流出.

证明:同样通过反证法证明.假设 a 和 b 都不从 u_i 流出,从该假设导出矛盾.

首先说明 K_1 不能指向 m_1 .如果 K_1 指向 m_1, m_1 为了保证其有一个出度必须指向 m_2 ,同样地, m_2 必须指向 d .在此情况下,一旦连接 m_2-d 失效,则 m_2 即使回溯至 m_1 也无法得到保护.

根据假设条件 a 和 b 都不能流出,且 K_1 不能指向 m_1 ,则 K_1 只能指向 m_3, m_3 为了保证其有一个出度一定指向 $K_2. K_2$ 有两种选择,指向 m_4 或者 m_5 .如果 K_2 指向 m_4 ,则 m_4 必定指向 d ,为了使得连接 m_4-d 失效后 m_4 能够得到保护,连接 K_2-m_5 必须被设定为 joker 连接, m_5 的出度必定指向 m_6 ;如果 K_2 指向 m_5 ,则 m_5 也必定指向 m_6 .综合上述两种情况, m_5 必定指向 m_6 .由于 u_i 左右对称,可以推导出 m_6 必定也指向 m_5 .因此矛盾得出, a 和 b 都不流出的假设不成立,引理 3.2 得证. \square

c) 定义一种如图 5 所示的连接器来连接上面构建的 C_j 和 u_i .

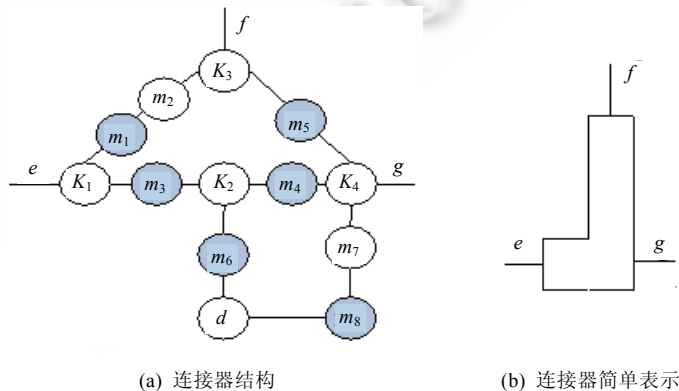


Fig.5 Structure of connector and its simple representation

图 5 连接器及其简单表示

引理 3.3. 设 e 为一个进入连接器的 PNH, 则如果连接器中的每个节点都被保护, f 不能进入且 g 必须流出. 证明:

(1) 首先证明如果 e 为进入 PNH, 则 f 不能进入.

假设 e 和 f 都为进入连接器的 PNH. 如果 K_3 指向 m_2 , 则 m_2 必定指向 m_1 , m_1 必定指向 K_1 , 则一旦连接 m_1-K_1 失效, 节点 m_1 将无法得到保护. 因此, K_3 不能指向 m_2 . 如果 m_2 指向 K_3 , 则 K_3 必定指向 m_5 , m_5 必定指向 K_4 , 则一旦连接 m_5-K_4 失效, 即使 m_5 回溯至上一跳节点 K_3 , K_3 也没有可用的 PNH 或者 JNH, m_5 将无法得到保护. 因此, m_2-K_3 必定为 joker 连接. 在此情况下, m_2 必定指向 m_1 , m_1 必定指向 K_1 . 由于 e 为流入 PNH, 则 K_1 必定指向 m_3 , m_3 指向 K_2 . 在该情况下, 一旦 m_3-K_2 失效, 则节点 m_3 无法得到保护. 因为即使其回溯至上一跳 K_1 , K_1 也没有可用的 PNH 或者 JNH. 因此, 如果 e 和 f 都为流入 PNH, 则总有节点无法得到保护, 导出矛盾. 也即, 如果 e 为流入 PNH, 则 f 不能为流入 PNH.

(2) 证明如果 e 为流入 PNH, g 必须流出.

假设 g 不是流出 PNH. 由于 e 为流入 PNH, 如果 K_1 流向 m_1 , 则 m_1 必须流向 m_2 且 m_2 必须流向 K_3 . 在该情况下, 一旦 m_2-K_3 失效, 则 m_2 将无法得到保护. 因此, K_1 必定流向 m_3 , m_3 流向 K_2 . 如果 K_2 指向 m_6 , 则连接 K_2-m_4 必定被设定为一条 joker 连接, 否则 m_6 将无法得到保护, 即 m_4 必定流向 K_4 ; 如果 K_2 指向 m_4 , m_4 也必定流向 K_4 . 综合以上两种情况, m_4 必定流向 K_4 . 由于在任何情况下 K_4 都不可能流向 m_7 , 否则将会造成节点 m_8 无法得到保护. 由于已经假设了 g 不是流出 PNH, 则节点 K_4 必定流向 m_5 , m_5 流向 K_3 , K_3 不能流向 m_2 , 否则将会造成 m_1 无法得到保护, K_3 必须由 f 流出. 在该情况下, 如果 m_1 指向 K_1 , 则会造成 m_3 在连接 m_3-K_2 失效后无法得到保护; 如果 m_1 流向 m_2 , 将会造成节点 K_3 在连接 f 失效后无法得到保护. 因此, 如果 g 不流出, 总有节点无法得到保护, 则连接器无法得到保护.

(3) 下面说明当 e 为流入 PNH 时该连接器的一个最优保护路由是存在的.

如图 6 所示, 在 e 为进入 PNH 的情况下, 如果 g 流出, 则存在最优的保护路由方案.

综合情形(1)~情形(3), 引理 3.3 得证. □

引理 3.4. 如果 f 是连接器的流入 PNH, 则可以找到一个最优保护路由, 其中, e 为流出 PNH, g 并不使用.

证明: 如图 7 所示, 如果 f 为连接器的流入 PNH, 可以找到一个最优保护路由, 其中, e 为流出 PNH, g 并不使用.

引理 3.4 得证. □

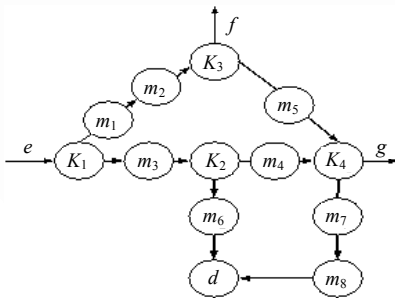


Fig.6 Optimal protection routing when e is an inflow PNH of connector

图 6 e 为流入 PNH 时连接器最优保护路由

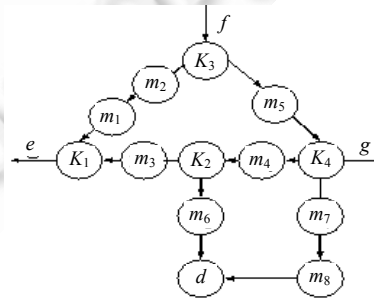


Fig.7 Optimal protection routing when f is an inflow PNH of connector

图 7 f 为流入 PNH 时连接器最优保护路由

如图 8 所示, 通过连接器将 u_i 以及 C_j 连接, 构建图 G . 构造规则为: 如果子句 P_j 包含变量 x_i , 则将 C_j 中对应于 x_i 的端通过连接器连接至 u_i 的左端; 如果 P_j 包含 \bar{x}_i , 则将 C_j 中对应于 \bar{x}_i 的端口通过连接器连接至 u_i 的右端.

引理 3.5. 如果 ϕ 存在一个真值分配, 则图 G 中每个节点都是被保护的.

证明: 如果 ϕ 存在一个真值分配使其为真, 则每个子句 C_j 中至少存在一个变量的值为真, 设定值为真的变量从 C_j 中流出. 如果 u_i 为真, 则如图 9 所示设定路由; 如果 \bar{u}_i 为真, 则将图 9 中的箭头反向. 根据引理 3.1~引理 3.4,

图 9 中的每个部件中的节点都是被保护的,引理 3.5 得证. □

引理 3.6. 如果 G 中每个节点都是被保护的,则 Φ 存在一个真值分配使其为真.

证明:首先说明图 10 中的情形不可能发生.由于 u_i 要得到保护,其必须有一条边流出.不失一般性,假设在该配置中 u_i 指向左边,则根据引理 3.3,左边第一个连接器的 f 不能流入,且 g 必须流出.由于连接器串联,根据引理 3.3,每个连接器的 f 必定都不能流入,与图 10 的情形相矛盾,因此图 10 的情形不可能发生.

通过如下方式构建真值分配:如果 C_j 中的某个变量指向连接器,则设定该变量的值为真.由于 G 中每个节点都被保护,根据引理 3.1,其至少包含一个变量指向连接器,即至少有一个变量的值为真,该真值分配一定使得 Φ 为真,引理 3.6 得证. □

定理 3.1. 当允许回溯一跳时,最优保护路由构建问题为一个 NP-hard 问题.

证明:根据引理 3.5、引理 3.6,定理 3.1 得证. □

定理 3.2. 当允许回溯 n 跳时,最优保护路由构建问题也是一个 NP-hard 问题.

证明:回溯 n 跳的 NP-hard 证明可以通过将图 3、图 4 以及图 5 中的深色节点替换为图 11 的图形即可.由于篇幅所限,具体证明过程不再赘述. □

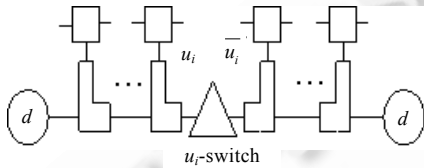


Fig.8 Structure of G

图 8 G 构建方法

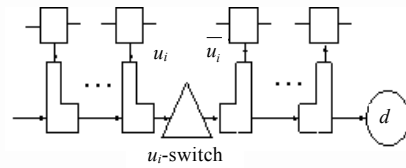


Fig.9 Routing scheme when u_i is true

图 9 u_i 为真时路由方案

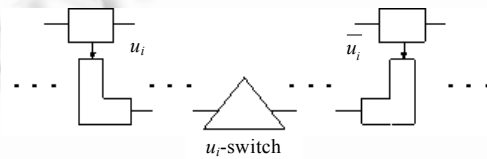


Fig.10 An impossible routing configuration

图 10 不可能的路由方案图

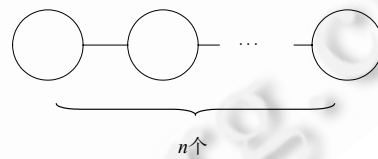


Fig.11 Series of n nodes

图 11 n 个节点的串联

3.2 性质分析

定理 3.3. 给定 $n(n \geq 3)$ 个节点的图 G ,如果图中节点可以组成一个环且节点间的最小连接度为 $k(k \leq \lceil n/2 \rceil)$,则允许回溯 $\lceil (n-1)/2 \rceil - k + 1$ 跳能够确保图中每个节点都能被保护.

证明:文献[3]已经证明了当节点数为 n 、最小连接度为 $k(k \geq \lceil n/2 \rceil)$ 时,直接通过 PNH 以及 joker 连接两种方式可以确保所有的节点都能得到保护.在本定理中,我们分析了 $k \leq \lceil n/2 \rceil$ 时,为了确保图中所有节点被保护所需的最小回溯跳数.

采用 $N(i)$ 表示编号为 i 的节点.既然节点可以组成一个环,将这 n 个节点按照图 12 进行路由配置.

为了简单起见,我们仅仅说明当 n 为偶数时的情形,当 n 为奇数时的情况可以通过对 n 为偶数时情况稍微变换得到.图 12(a)中存在两条不相交路径:

- (1) $N(\lceil (n-1)/2 \rceil) \rightarrow N(\lceil (n-1)/2 \rceil - 1) \rightarrow \dots \rightarrow N(1) \rightarrow d;$
- (2) $N(\lceil (n-1)/2 \rceil + 1) \rightarrow N(\lceil (n-1)/2 \rceil + 2) \rightarrow \dots \rightarrow N(n-1) \rightarrow d.$

由于图 12(a)为对称的,我们仅仅分析节点 $N(i)(1 \leq i \leq \lceil (n-1)/2 \rceil)$ 的保护方式. $N(i)$ 的下游连接 $N(i-1)-N(i)$ 或者节点 $N(i-1)$ 节点失效后 $N(i)$ 有 3 种可能的保护途径:

- (1) 如果节点 $N(i)(1 \leq i \leq \lceil (n-1)/2 \rceil)$ 与节点 $N(j)(\lceil (n-1)/2 \rceil < j \leq n-1)$ 相连,则设定该连接为 joker 连接,节点 $N(i)$ 的下游路径失效后其可以通过 joker 连接 $N(i)-N(j)$ 发送报文,该情况下不需要回溯;

- (2) 如果 $N(i)(1 \leq i \leq \lceil (n-1)/2 \rceil)$ 与 d 以及 $N(1)$ 至 $N(i-2)$ 中的任何一个节点 $N(t)$ 相连,则 $N(i)$ 可以通过连接 $N(i)-N(t)$ 发送报文,也不需要回溯;
- (3) 最坏情况下,节点 $N(i)(1 \leq i \leq \lceil (n-1)/2 \rceil)$ 不与 $N(1)$ 至 $N(i-2)$ 以及 $N(\lceil (n-1)/2 \rceil)$ 至 $N(n-1)$ 中的任何一个节点相连,其只与 $N(i+1)-N(\lceil (n-1)/2 \rceil)$ 中的节点相连,则 $N(i)(1 \leq i \leq \lceil (n-1)/2 \rceil)$ 的下游路径失效后其只能顺着 $N(i+1), N(i+2), \dots, N(\lceil (n-1)/2 \rceil)$ 向上回溯至某个有可用备份路径的上游节点,最多回溯至节点 $N(\lceil (n-1)/2 \rceil)$ 一定有可用的备份路径。

只有在第 3 种情况下节点需要回溯,下面我们分析第 3 种情况下所需的最大回溯跳数。既然节点间最小连接度为 k ,最坏的情况为节点 $N(1)$ 与节点 $N(2), N(3), \dots, N(k)$ 相连,其一跳内最多能够回溯至节点 $N(k)$,节点 $N(k)$ 又需要通过 $\lceil (n-1)/2 \rceil - k$ 跳回溯至 $N(\lceil (n-1)/2 \rceil)$ 节点才有可用路径。在此情况下,最大回溯跳数为 $\lceil (n-1)/2 \rceil - k + 1$ 。

当 n 为奇数时, $N(i)(1 \leq i \leq \lceil (n-1)/2 \rceil)$ 回溯至节点 $N(\lceil (n-1)/2 \rceil)$ 后,可以通过 joker 连接 $N(\lceil (n-1)/2 \rceil) - N(\lceil (n-1)/2 \rceil + 1)$ 发送报文,问题同样取为求从 $N(i)(1 \leq i \leq \lceil (n-1)/2 \rceil)$ 回溯至节点 $N(\lceil (n-1)/2 \rceil)$ 的最大跳数。与当 n 为偶数时类似,最大回溯跳数同样为 $\lceil (n-1)/2 \rceil - k + 1$ 。定理 3.3 得证。 □

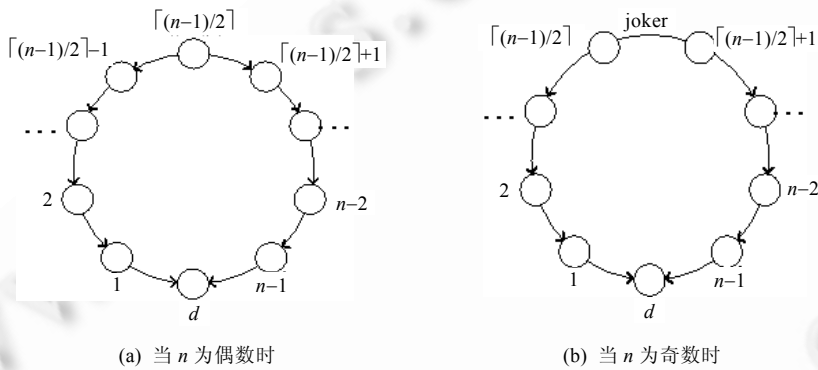


Fig.12 Cycle composed of nodes

图 12 节点构成的环

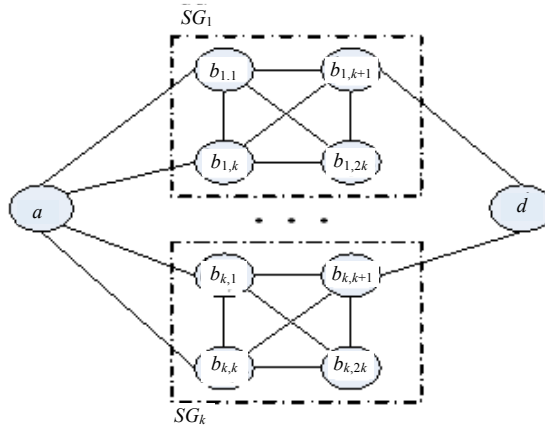
定理 3.4. 对于节点数目为 n 的图 G ,如果移除任意一个节点或连接后剩余图仍然连通,最大回溯 $n-2$ 跳能够确保 G 中的所有节点都能够得到保护。

证明:我们采用反证法对该定理进行证明。采用 R_d 标记以 d 为目的地址的最优保护路由,则 R_d 中每条连接一定是有向连接或 joker 连接,且每个节点至少包含一个 PNH。基于有向连接对 R_d 中的所有节点进行拓扑排序^[17],排序结果记为 $N_1 \rightarrow N_2 \rightarrow \dots \rightarrow N_i \rightarrow \dots \rightarrow N_{n-1} \rightarrow d$ 。假设在 R_d 中允许回溯 $n-2$ 跳的情况下仍然有一个节点 $N_i (1 \leq i \leq n-1)$ 无法得到保护,则表明 N_i 回溯至上游的 N_1-N_{i-1} 中的任何一个节点都无法得到保护,也即 N_1-N_{i-1} 中的任何一个节点都没有 PNH 或 JNH 指向 $N_{i+2}-d$ 中的任何一个节点,也即在图 G 中去除节点 N_{i+1} 后剩余图并不连通。该假设与先决条件中移除任意一个节点和连接后剩余图仍然是连通的相矛盾,因此定理 3.4 得证。 □

定理 3.5. 对任意正整数 k ,在允许回溯 1 跳的情况下,至少存在一个 k 边连接(k -link-connected)图 G ,其中至少有一个除 d 外的节点无法得到保护。

证明:通过给出一个 k 边连接图且其中有一个除 d 外的节点无法得到保护的实例来证明该定理。按照如下规则构建实例:构建 k 个全连接的子图 $SG_i(1 \leq i \leq k)$,每个子图 SG_i 包含 $2k$ 个节点 $\{b_{i,1}, b_{i,2}, \dots, b_{i,2k}\}$ 。按照图 13 所示将这些子图连接成图 G ,其中,节点 a 和 $b_{i,1} \dots b_{i,k}(1 \leq i \leq k)$ 相连,很明显, G 为 k 边连接图。

a 要得到保护,其至少有一个出度。不失一般性,假设 a 指向 SG_1 ,则 SG_1 不能指向 a ,否则将会造成环路。 $b_{1,k+1}$ 必须指向 d ,否则 SG_1 将不能达到 d 。在此情况下,如果 SG_1-d 失效,由于只允许回溯一跳,其最多回溯至节点 $b_{1,1}$ 和 $b_{1,k}$ 。根据前面的分析, $b_{1,1}$ 和 $b_{1,k}$ 不能指向 a ,则在此情况下, SG_1 中的节点无法到达 d 。在该图中,节点 $b_{1,k+1}$ 无法得到保护,即图 G 中至少存在一个除 d 外的节点无法得到保护,定理 3.5 得证。 □

Fig.13 The topology when $k=2$ 图 13 $k=2$ 时拓扑实例

4 构建保护路由的启发式算法

在第 3 节中已经证明了为给定拓扑构建最优保护路由为一个 NP-hard 问题,因此无法在多项式时间内找到该问题的最优解.本节给出了一个三阶段的启发式算法进行最优保护路由的计算.

4.1 算法主要思想

启发式算法的主要目的为在给定拓扑中保护尽量多的节点.启发式算法分为 3 个阶段:在第 1 阶段中,为尽可能多的节点构建两个以上的 PNH;在第 2 阶段中,将 PNH 数量大于 2 节点的部分 PNH 转化为 JNH,增大其邻居的保护性,从而增大被保护节点的数目;在第 3 阶段中,为仅有一个 PNH 并且没有 JNH 保护的节点构建回溯保护,最大程度地提高保护节点的数目.

4.2 算法描述

启发式算法具体流程见算法 1,其中,BPEN 函数实现第 1 阶段功能,PTJ 函数实现第 2 阶段的功能,BB 函数实现第 3 阶段的功能.在函数 PTJ 和 BB 中,不同的节点处理顺序对最终解的质量有影响,因此在算法 1 的第 3 行~第 10 行的每轮循环中,以不同的节点处理顺序重复执行 PTJ 和 BB 函数,以期通过改变节点的处理顺序获得更好的解.为了取得不同的节点处理顺序,在每轮迭代中保存之前所有轮中节点的处理顺序,在本轮 BPEN,PTJ 以及 BB 函数中选取处理节点时与之前所有轮的节点处理顺序进行比对,确保本轮的节点处理顺序与以前所有轮不同.算法 1 最大循环 P 轮,选取 P 轮的最优结果,其中, F_d 表示路由 R_d 中被保护节点的个数.

1) BPEN 函数

MA 排序^[18]在为每个节点构建尽可能多的下游 PNH 上有着较好的效果,因此在 BPEN 中采用 MA 排序为每个节点构建 PNH.BPEN 的详细过程在函数 1 中给出,其中, $PNH(v_i)$ 表示节点 v_i 的所有 PNH 节点集, $d(v,S)$ 表示节点 v 与集合 S 间的连接数目.函数 1 的第 3 行~第 7 行对给定拓扑进行 MA 排序,在第 8 行~第 14 行中根据 MA 排序的结果对拓扑中的边定向.函数 1 的执行结果为一个有向无环图,图中的每个边都被设定了方向.

Algorithm 1. Building protection routing.

Input: $G=(V,E)$;

Output: R .

- 1) **Foreach** $d \in V$ **do**
- 2) $R_d^* = BPEN(G,d)$
- 3) **Repeat**

- 4) $R_d^{**} = PTJ(G_d, R_d^*)$
- 5) $R_d^{***} = BB(G_d, R_d^{**}, n)$
- 6) **If** $F_d^{***} > F_d$ **then**
- 7) $F_d \leftarrow F_d^{***}, R_d \leftarrow R_d^{***}$
- 8) **End**
- 9) Diversification on search order in PTJ and BB
- 10) **Until** F_d has no improvement after P rounds
- 11) **End**
- 12) **Return** R

Function 1. $BPEN(G, d)$.

Input: $G=(V, E)$, destination $d \in V$;

Output: R_d^* .

- 1) $s \leftarrow d, S = \{s\}, T = V \setminus \{s\}$
- 2) $i \leftarrow 2$
- 3) **While** $i \leq |V|$ **do**
- 4) Choose $v \in T$ with the largest $d(v, S)$
- 5) $v_1 \leftarrow v, S = S \cup \{v\}, T = T \setminus \{v\}$
- 6) $i = i + 1$
- 7) **End while**
- 8) **Foreach** $(v_1, v_2) \in E$ **in** G
- 9) **If** $order(v_1) > order(v_2)$ **then**
- 10) $PNH(v_1) \leftarrow v_2$
- 11) **Else**
- 12) $PNH(v_2) \leftarrow v_1$
- 13) **End**
- 14) **End**
- 15) **Return** R_d^*

2) PTJ 函数

该函数的详细过程在函数 2 中给出,其中, $PNH(v_i), JNH(v_i)$ 分别表示节点 v_i 的 PNH 集合以及 JNH 集合, $PN(v_i)$ 表示节点 v_i 的 PNH 的数目, $JN(v_i)$ 表示节点 v_i 的 JNH 的数目。

在为节点 v_i 构建 JNH 时,需要保证节点 v_i 的 JNH 的下游路径不会通过节点 v_i 的 PNH。例如在图 14 中,当设定 v_s 为 v_e 的 joker 下一跳时,需要保证 v_s 的下游路径不会经过 v_k 。因为当节点 v_e 感知到连接 v_e-v_k 失效时,其并不知道究竟是连接 v_e-v_k 断开还是节点 v_k 失效,如果其 joker 下一跳 v_s 经过 v_k 且节点 v_k 失效,则 v_e 即将发送至 v_s 流量也会丢失。对应地,在 PTJ 的第 6 行将 v_k 移除,考察连接 v_s 是否还有到达 d 的可达路径。

3) BB 函数

需要注意的是,在允许回溯 n 跳的情况下,可能无法为节点指定一个通用的 BPNH,一个节点需要不同的 BPNH 对应不同的下游连接失效。例如在图 15 中,如果连接 v_3-v_4 失效,则回溯至节点 v_2 后就有可用的下游路径;而如果节点 v_4 通告连接 v_4-v_5 失效,则回退至 v_2 后仍然需要回退至节点 v_1 。因为虽然 v_4 感知到连接 v_4-v_5 失效,失效的真正原因可能为节点 v_5 失效,在此情况下,路径 $v_2-v_5-v_6$ 并不可用。因此,节点 2 上对应连接 v_4-v_5 的 BPNH 为 v_1 ,而对 v_3-v_4 失效直接采用下一跳 v_3 传送即可。对应地,在 BB 函数的第 20 行,需要特别指定节点 v_n 对应于连接 v_i-v_k 失效的回溯保护节点为 v_k 。该函数的详细过程在函数 3 中给出,其中, n 表示允许回溯的最大跳数, $BN(v_i)$ 表示

v_i 的 BPNH 的数目.

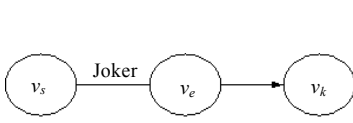


Fig.14 An example of joker link
图 14 Joker 连接构建示例

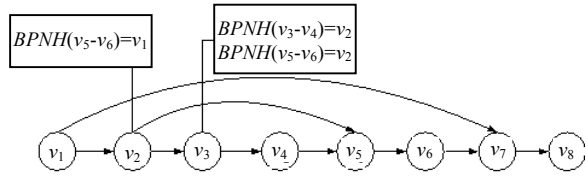


Fig.15 An example of BPNH
图 15 BPNH 构建示意图

在有 N 个节点、 M 条连接的网络中,算法 1 执行 $P \times N$ 轮,每轮 PTJ 的循环次数为 M 次,BB 的最大循环次数为 $N \times N$ 次,则算法 1 的最大执行次数为 $P \times N \times (M + N \times N)$.因此,算法 1 的时间复杂度为 $O(n^3)$.

4.3 算法实例

为了便于理解,在图 16 中给出算法 1 的一个执行实例.

在图 16(a)中,节点旁的数字表示在 MA 排序中节点被选择的次序.在图 16(b)中,连接 1-2 和连接 4-5 被设定为 joker 连接,从而使得节点 2 和节点 5 都得到保护.在图 16(c)中,为剩余的未保护节点 6 构建回溯保护,其只要回溯至节点 4 就能够得到保护.

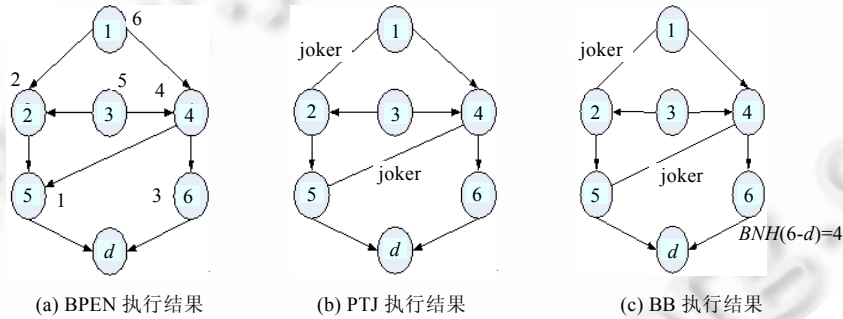


Fig.16 An example of heuristic algorithm
图 16 启发式算法实例

Function 2. $PTJ(G_d, R_d^*)$.

Input: $G_d=(V, E_d), R_d^*$;

Output: R_d^{**} .

- 1) $R_d^{**} \leftarrow R_d^*$
- 2) **Foreach** $\langle v_s, v_e \rangle \in E$ in R_d^*
- 3) **If** $(PN(v_s) > 1 \ \&\& \ PN(v_e) = 1 \ \&\& \ JN(v_e) = 0)$ **then**
- 4) $R'_d \leftarrow R_d$
- 5) Denote v_k as PNH of v_e
- 6) Remove v_k from R'_d
- 7) Remove $\langle v_s, v_e \rangle$ from R'_d
- 8) Add $\langle v_e, v_s \rangle$ in R'_d
- 9) **If** there is no cycle in R'_d and there is a path from v_s to d in R'_d **then**
- 10) $PNH(v_e) \leftarrow v_s$

- 11) $PNH(v_s) \leftarrow v_e$
- 12) Transform $\langle v_s, v_e \rangle$ to (v_s, v_e) in R_d^{**}
- 13) Delete v_e from $PNH(v_s)$
- 14) **End**
- 15) **Else**
- 16) **End**
- 17) **Return** R_d^{**}

Function 3. $BB(G_d, R_d^{**}, n)$.

Input: $G_d=(V, E_d)$, R_d^{**} , n ;

Output: R_d^{***} .

- 1) $R_d^{***} \leftarrow R_d^{**}$
- 2) **Foreach** $((v_i \in V \setminus d) \ \&\& \ (PN(v_i)=1) \ \&\& \ (JN(v_i)=0) \ \&\& \ (BN(v_i)=0))$ **do**
- 3) Denote an array $a[\cdot]$ for storing nodes
- 4) Add v_i to $a[\cdot]$
- 5) **Foreach** $v_p \in a[\cdot]$ and v_j is upstream of v_p in R_d **do**
- 6) **If** $(PN(v_j)=1)$ **then**
- 7) Add v_i to $a[\cdot]$
- 8) **Continue**
- 9) **End**
- 10) **Foreach** path P from v_j to v_i in R_d^d **do**
- 11) $R_d' \leftarrow R_d^{***}$
- 12) Let v_k denote current PNH of v_i
- 13) Remove v_k from R_d'
- 14) Reverse the direction of edges in P
- 15) **End**
- 16) **If** $(PN(v_j) \geq 2)$ **then**
- 17) **If** no circle exists and there is a path from v_j to d in R_d' **then**
- 18) **Foreach** node v_n on P and v_x is
- 19) the upstream node of v_n on P **do**
- 20) $BPNH(v_n, v_i - v_k) \leftarrow v_x$
- 21) **End**
- 22) $BN(v_i)=1$
- 23) **End**
- 24) **End**
- 25) **If** $(PN(v_j)=1 \ \&\& \ JN(v_j) \geq 1)$ **then**
- 26) **Foreach** $v_p \in JNH(v_j)$ **do**
- 27) $R_d' \leftarrow R_d^{***}$
- 28) Add $\langle v_j, v_p \rangle$ to R_d'
- 29) **If** no circle exists and there is a path from v_j to d in R_d' **then**
- 31) **Foreach** node v_n and its upstream node v_n on P **do**
- 32) $BPNH(v_n, v_i - v_k) \leftarrow v_x$

```

33)          End
34)          BN(i)=1
35)          End
36)          End
37)          End
38)          End
39) End
40) Return  $R_d^{***}$ 

```

5 性能评估

采用 TP 表示本文提出的保护路由算法,在不同连接度的拓扑上分别将 TP 与 PR^[13],MARA^[5]及 ECMPs^[7]的保护效果进行了对比,比较各种方案中被保护的路由器的个数.PR 是集中式路由中最新的保护路由算法.ECMPs 是大规模应用在域内路由中的多路径路由算法.MARA 是多路径领域的最新研究成果,主要目的是增强路由的可靠性.不失一般性,在 ECMPs 中设定每条边的权值都为 1.

5.1 实验环境

1) 网络拓扑:为了验证启发式算法在真实网络拓扑下的性能,采用了 Rocketful^[19]项目组提供的 AS1239 (315 节点,972 条连接)、AS3257(161 节点,328 连接)以及 AS1221(108 节点,153 连接)的真实网络拓扑进行实验.在这些拓扑中,有部分连接度为 1 的节点,这些节点无法得到保护.为了提高实验的准确性,在每个拓扑中迭代地删除连接度为 1 的节点及与其相连的边,得到每个 AS 的主干拓扑分别采用 B_AS1239(284 节点,941 条连接)、B_AS3257(115 节点,282 连接)以及 B_AS1221(50 节点,97 连接)表示,在主干拓扑中进行实验.此外,为了验证在更低连接度拓扑上算法的性能,我们采用 BRITE^[20]拓扑生成器生成了两个仿真拓扑 Sim1 以及 Sim2.Sim1 包含 50 个节点以及 85 条连接,Sim2 包含 50 个节点以及 70 条连接.在表 1 中对各实验拓扑的节点连接度属性进行了归纳,其中, D_i 表示连接度为 i 的节点数目, D_{avg} 表示节点的平均连接度.通过表 1 还可以看到,这 5 个拓扑的平均节点连接度从高到低,具备较高的代表性.

Table 1 Characteristics of experimental topologies

表 1 实验拓扑连接度属性

拓扑	B_AS1239	B_AS3257	B_AS1221	Sim1	Sim2
D_1	0	0	0	0	0
D_2	69	35	25	19	28
D_3	52	18	3	11	12
D_4	26	19	6	10	4
$D_i > 4$	145	43	16	10	6
所有节点	284	115	50	50	50
D_{avg}	6.63	4.91	3.88	3.4	2.8

5.2 实验结果及分析

1) 实验 1:保护效果实验

设定算法 1 循环的次数 $P=20$ (该值在保护效果与算法执行时间间达到均衡,具体原因在实验 2 中阐述),即循环 20 轮取最优解.同时,为了验证不同的回溯跳数对保护效果的影响,分别设定函数 3 中最大回溯跳数为 1,3,5,验证允许回溯 1,3,5 跳时 TP 的保护效果.在每个拓扑下,依次以其中的每个节点作为目的节点,运行各种算法构建其他所有节点到该节点的保护路由,统计各算法下被保护节点的个数.实验结果在图 17~图 21 中给出.其中,TP1,TP3,TP5 分别表示最多回溯 1 跳、3 跳以及 5 跳时 TP 算法的结果.

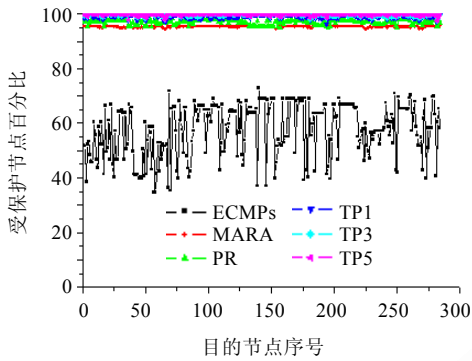


Fig.17 Experimental results of B_AS1239

图 17 B_AS1239 实验结果

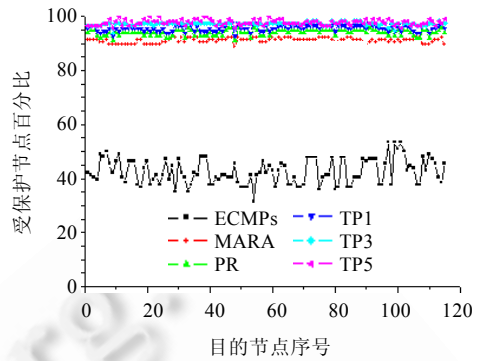


Fig.18 Experimental results of B_AS3257

图 18 B_AS3257 实验结果

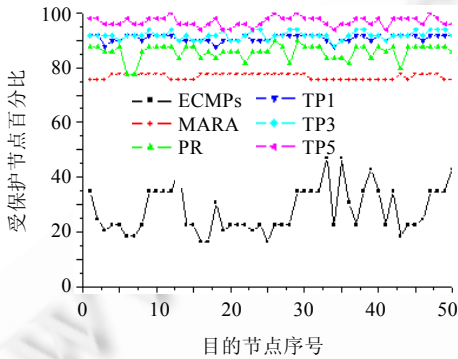


Fig.19 Experimental results of B_AS1221

图 19 B_AS1221 实验结果

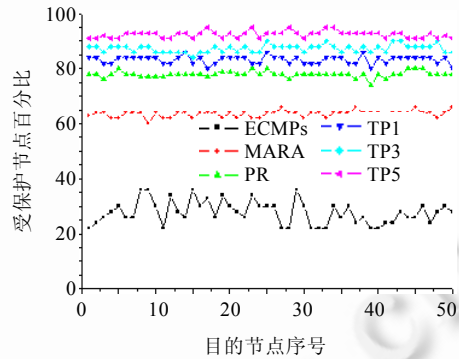


Fig.20 Experimental results of Sim1

图 20 Sim1 实验结果

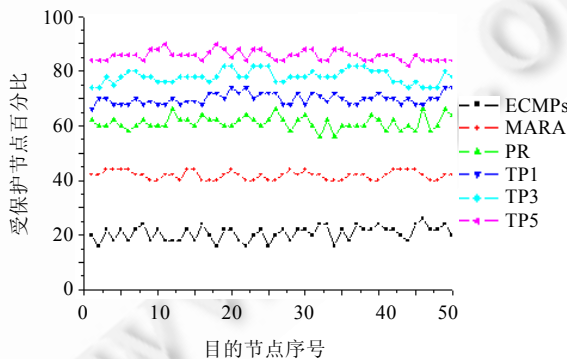


Fig.21 Experimental results of Sim2

图 21 Sim2 实验结果

通过图 17 可以看到,在连接度较高的拓扑 B_AS1239(平均节点连接度 6.63)中,TP,PR 以及 MARA 算法的效果差别不大,MARA 保护的节点比例大约在 95%,PR 保护的节点比例大约在 97%,TP1~TP5 的保护效果逐步提升,在 TP5 时接近 100%。随着拓扑连接度的逐步降低,在 B_AS3257(平均节点连接度 4.91),B_AS1221(平均节点连接度 3.88),Sim1(平均节点连接度 3.4)以及 Sim2(平均节点连接度 2.8)中,TP,PR,MARA 以及 ECMPs 保护节点的比例都有所下降,但 TP 下降得较慢,PR,MARA 以及 ECMPs 与 TP 间的差距逐步增大。在图 21 给出的 Sim2

拓扑的实验结果中,EMCPs 大约保护 20%的节点,MARA 大约保护 40%的节点,PR 大约保护 60%的节点,TP1 保护 70%的节点,TP3 保护 77%的节点,TP5 大约保护 85%的节点.

综合图 17~图 21 各个拓扑中各种算法的效果可以看到,TP 有着最好的实验效果,在每种拓扑中都能够保护最多的节点,在低连接度的拓扑上其优势更为明显,并且随着可回溯跳数的增大,TP 中被保护的节点比例逐步增大.PR 效果其次,MARA 比 PR 少保护部分节点,ECMPs 在每个拓扑下保护效果都是最差的.在平均节点连接度大于 4 的拓扑上,TP,PR 以及 MARA 都能够保护 90%以上的节点,其差距不大,TP 大约比 PR 多保护 3%的节点,PR 比 MARA 多保护 3%的节点,ECMPs 大约比 MARA 少保护 40%的节点;在平均节点连接度大约为 3~4 的拓扑上,TP 比 PR,MARA 以及 ECMPs 有着较为明显的优势,在仅仅回溯一跳的情况下,TP 大约比 PR 多保护 7%的节点,PR 大约比 MARA 多保护 17%的节点,ECMPs 大约比 MARA 少保护 25%的节点.

2) 实验 2:算法性能实验

此外,为了验证不同的迭代次数对算法性能的影响,我们依次设定算法 1 中的 P 为 5,10,20,30,50,考察不同的算法迭代次数对算法效果以及性能的影响.为了节省篇幅,我们在 B_AS1239,B_AS1221 以及 Sim2 上分别运行 TP1,TP3 以及 TP5,考察平均被保护节点比例以及算法的执行时间.这 3 个拓扑的连接度由高至低,具备较好的代表性,节点数目也不相同,其中,B_AS1239 节点数目较多,B_AS1221 以及 Sim2 节点数目较少.实验结果在表 2~表 7 中给出.我们的实验配置是:处理器为主频为 2.2GHz 的 Pentium(R) Dual-Core CPU T4400,内存 1G,算法的开发与执行环境为 NetBeans IDE 6.8,主要开发工具采用了 Java 的 JGraph 开发包.

通过实验结果可以看出,随着迭代次数的增多,产生的最终路由的保护效果越好.随着迭代次数的增多,算法的执行时间越来越长,但基本处于同一数量级.在节点数量较少的 B_AS1221 以及 Sim2 上,在各种迭代次数下 TP 的计算时间最长不超过 11s.在 B_AS1239 上,由于节点数目较多,随着迭代次数的增多,计算时间大约需要几百秒.通过表 2~表 7 我们可以看出,在 3 个拓扑下,迭代次数大于 20 时,保护节点的比例增长规模已经很小,这种趋势在 B_AS1221 以及 Sim2 上体现得尤为明显.因此,综合不同 P 值下算法的效果以及性能,我们选取了 $P=20$ 作为本文实验 1 的迭代次数.

Table 2 Ratio of protected nodes on B_AS1239 when P is given different values

表 2 B_AS1239 上不同 P 值下 TP 保护节点比例

P (%)	5	10	20	30	50
TP1	98.278 9	98.612 6	99.076 9	99.091 3	99.113 4
TP3	98.714 5	99.103 7	99.619 5	99.729 2	99.781 5
TP5	98.839 2	99.321 7	99.856 9	99.884 3	99.886 4

Table 3 Execution time of TP on B_AS1239 when P is given different values (s)

表 3 B_AS1239 上不同 P 值下 TP 执行时间 (s)

P	5	10	20	30	50
TP1	61	118	242	389	643
TP3	73	125	267	465	773
TP5	78	132	283	512	829

Table 4 Ratio of protected nodes on B_AS1221 when P is given different values

表 4 B_AS1221 上不同 P 值下 TP 保护节点比例

P (%)	5	10	20	30	50
TP1	90.207 1	90.263 5	91.385 2	91.415 7	91.442 5
TP3	92.103 4	93.326 3	93.521 7	93.797 5	93.831 8
TP5	96.869 6	97.421 2	97.963 1	98.153 6	98.157 3

Table 5 Execution time of TP on B_AS1221 when P is given different values (s)表 5 B_AS1221 上不同 P 值下 TP 执行时间 (s)

P	5	10	20	30	50
TP1	2	3	4	6	8
TP3	2	3	4	6	10
TP5	2	3	5	7	11

Table 6 Ratio of protected nodes on Sim2 when P is given different values表 6 Sim2 上不同 P 值下 TP 保护节点比例

P (%)	5	10	20	30	50
TP1	67.745 6	68.203 5	69.826 0	71.234 3	71.321 7
TP3	75.212 2	76.324 6	78.130 4	78.791 8	79.287 5
TP5	85.023 6	85.652 1	85.826 0	86.178 2	86.282 5

Table 7 Execution time of TP on Sim2 when P is given different values (s)表 7 Sim2 上不同 P 值下 TP 执行时间 (s)

P	5	10	20	30	50
TP1	2	3	4	6	8
TP3	2	3	4	6	10
TP5	2	3	5	7	11

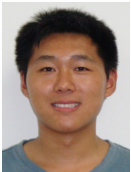
6 总结及未来工作

构建失效后的备份路径是保证集中式路由可靠性的重要手段.本文提出了一种适合低连接度拓扑的域内保护路由构建方法,保证在单个节点或者连接失效的情况下报文的最少损失.证明了为给定拓扑构建最优保护路由问题为一个 NP-hard 问题,并且给出了三阶段启发式算法,在真实的网络拓扑下验证了启发式算法具备较好的性能.本文的工作不但促进了集中式路由的理论研究,而且具备了较大的实际意义,可以降低网络的冗余度,从而节约网络建设成本.基于本文的工作还有许多有意义的工作,比如可以分析在节点重要性不同的网络中如何构建高效的路由保护方案.

References:

- [1] Fortz B, Thorup M. Internet traffic engineering by optimizing OSPF weights. In: Proc. of the IEEE INFOCOM 2000. Israel: IEEE Press, 2000. 519–528. [doi: 10.1109/INFCOM.2000.832225]
- [2] Greenberg A, Hjalmtysson G, Maltz DA, Myers A, Rexford J, Xie G, Yan H, Zhan J, Zhan H. A clean slate 4D approach to network control and management. ACM SIGCOMM Computer Communication Review, 2005,35(5):41–54. [doi: 10.1145/1096536.1096541]
- [3] Kwong KW, Gao L, Guerin R, Zhang ZL. On the feasibility and efficacy of protection routing in IP networks. IEEE/ACM Trans. on Networking, 2011,19(5):1543–1556. [doi: 10.1109/TNET.2011.2123916]
- [4] Caesar M, Caldwell D, Feamster N, Rexford J. Design and implementation of a routing control platform. In: Proc. of the 2nd Symp. on Networked Systems Design & Implementation (NSDI 2005). Boston: ACM Press, 2005. 15–28.
- [5] Ohara Y, Imahori S, Meter RV. MARA: Maximum alternative routing algorithm. In: Proc. of the IEEE INFOCOM 2009. Brazil: IEEE Press, 2009. 298–306. [doi: 10.1109/INFCOM.2009.5061933]
- [6] Shand M, Bryant S. IP fast reroute framework. Internet Draft, 2009. <http://tools.ietf.org/html/draft-ietf-rtgwg-ipfrr-framework-13>
- [7] Moy J. OSPF version 2. IETF, RFC 2328, 1998.
- [8] Atlas A, Zinin A. Basic specification for IP fast reroute: Loop-Free alternates. IETF RFC 5286, 2008.
- [9] Ray S, Guerin R, Kwong KW, Sofia R. Always acyclic distributed path computation. IEEE/ACM Trans. on Networking, 2010, 18(1):307–319. [doi: 10.1109/TNET.2009.2025374]

- [10] Schollmeier G, Charzinski J, Kirstädter A, Reichert C, Schrodi K, Glickman Y, Winkler C. Improving the resilience in IP networks. In: Proc. of the IEEE Workshop on High Performance Switching and Routing. Torino: IEEE Press, 2003. 91–96. [doi: 10.1109/HPSR.2003.1226686]
- [11] Nelakuditi S, Lee S, Yu Y, Zhang ZL, Chuah CN. Fast local rerouting for handling transient link failures. IEEE/ACM Trans. on Networking, 2007,15(2):359–372. [doi: 10.1109/TNET.2007.892851]
- [12] Kvalbein A, Hansen AF, Cčić T, Gjessing S, Lysne O. Multiple routing configurations for fast IP network recovery. IEEE/ACM Trans. on Networking, 2009,17(2):473–486. [doi: 10.1109/TNET.2008.926507]
- [13] Sényedi G, Szilagyi P, Retvari G, Csaszar A. IP fast reroute: Lightweight not-via without additional addresses. In: Proc. of the IEEE INFOCOM 2009. Brazil: IEEE Press, 2009. 2771–2775.
- [14] Lakshminarayanan K, Caesar M, Rangan M, Anderson T, Shenker S, Stoica I. Achieving convergence-free routing using failure-carrying packets. In: Proc. of the SIGCOMM. Kyoto: ACM Press, 2007. 241–252.
- [15] Kushman N, Kandula S, Katabi D, Maggs B. R-BGP: Staying connected in a connected world. In: Proc. of the 4th Symp. on Networked Systems Design and Implementation (NSDI 2007). Berkeley: USENIX Association, 2007. 341–354. [doi: 10.1145/1282380.1282408]
- [16] Kullmann O. New methods for 3-SAT decision and worst-case analysis. TCS, 1999,223(1-2):1–72. [doi: 10.1016/S0304-3975(98)00017-6]
- [17] Diestel R. Graph Theory. 3rd ed., Springer-Verlag, 2005.
- [18] Fujishige S. A maximum flow algorithm using MA ordering. Operations Research Letters, 2003,31(3):176–178. [doi: 10.1016/S0167-6377(02)00237-7]
- [19] Spring N, Mahajan R, Wetherall D, Anderson T. Measuring ISP topologies with rocketfuel. IEEE/ACM Trans. on Networking, 2004,12(1):2–16. [doi: 10.1109/TNET.2003.822655]
- [20] Medina A, Lakhina A, Matta I, Byers J. BRITE: An approach to universal topology generation. In: Proc. of the 9th Int'l Symp. in Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2001). Washington: IEEE Press, 2001. 346–353. [doi: 10.1109/MASCOT.2001.948886]



谭晶(1984—),男,江苏南通人,博士生,主要研究领域为下一代互联网,路由机制.
E-mail: tanjing@seu.edu.cn



李伟(1978—),男,博士,副教授,CCF 高级会员,主要研究领域为下一代网络体系结构,服务计算.
E-mail: xchlw@seu.edu.cn



罗军舟(1960—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为下一代网络体系结构,协议工程,网络安全与管理,网格与云计算,服务计算.
E-mail: jl原因@seu.edu.cn