

摘 要

在因特网初期发展阶段，TCP/IP协议被设计为可保证报文的成功传输和按序投递，同时辅以重收敛技术用于当网络拓扑变化时确保路由的健壮性。在现实网络环境中，设备发生失效是不可避免的。在失效所触发的路由重收敛过程完成之前，由于全网路由器对于新的拓扑结构的认知不一致，会引发丢包、环路、延迟、拥塞等一系列问题。近年来诸如IPTV、IP电话、网络游戏等新型业务大量涌现，重收敛过程的速度已不能满足这些实时性业务的服务需求，因此研究路由快速恢复技术具有一定的意义和价值。

本文的研究对象是多拓扑路由（Multi-topology Routing, MTR）技术。MTR技术由互联网工程任务组（Internet Engineering Task Force, IETF）提出，用于在IP网络中设备发生短时失效后快速恢复业务的传输路径。MTR的核心思想是：基于网络的物理拓扑创建一组逻辑拓扑，称为多拓扑（Multi-topology, MT）或备份拓扑，负责启动恢复操作的路由器选择可保护该失效的多拓扑，在此拓扑对应的路由表中查找下一跳，并将该拓扑编号写入数据包首部，其余路由器根据数据包携带的拓扑标识和目的地址进行路由转发。本文围绕这一技术，通过仿真评估相关的路由转发方案，研究并提出多拓扑生成算法，并将权重优化应用于解决路由恢复后的负载均衡问题。

第一章介绍网络设备失效的分布特征以及现有的IP网络路由恢复技术。在第二章中阐述MTR技术的基本原理及实现方案：首先介绍用于保护节点和保护链路的两类多拓扑，然后采用保护链路的多拓扑，对于全局和局部这两种业务路径的恢复方案，以及单次或升序这两种路由在拓扑间切换的策略分别进行对比分析。

基于现有技术方案，进行以下研究及改进：（1）多拓扑结构的设计。第三章首先介绍三种用于生成多拓扑的算法，然后针对双链路失效情形，提出一种可降低拓扑所需承载业务量的多拓扑生成算法。并以路径跳数及链路负载作为性能指标，对比算法性能；（2）多拓扑链路权重的优化。在第四章中首先提出用于求解多拓扑权重的整数线性规划（Integer Linear Programming, ILP）模型，然后结合遗传算法设计优化拓扑权重的方案，并测试对比了优化权重前后的链路利用率。

第五章介绍了使用OPNET工具搭建的仿真平台，第六章总结全文。

关键词：多拓扑路由，快速恢复，多拓扑设计，链路权重优化

ABSTRACT

In the early stage of Internet, the TCP/IP protocol suite was developed to ensure successful transmission and in-order delivery of packets, supplemented by the re-convergence process to update the network state and ensure robust routing after changes in the network topology. In the real network environment, failures in network equipments are inevitable. Before the re-convergence process triggered by the failures is fully completed, packet loss, loops, delay, congestion and other issues may occur since all routers haven't attached to a common network topology. More recently, new type of applications like IPTV, IP telephony and online games are being offered through the Internet. Re-convergence process does not operate in a time-scale that is compliant with the stringent requirements of these real time applications, so researching route schemes that offer fast failure handling does make sense to some extent.

The main theme for this thesis is the Multi-topology Routing (MTR) technology, which has been proposed by the Internet Engineering Task Force (IETF). MTR is a scheme that can be used to achieve fast recovery path from transient failures in IP networks. Its basic principle is to build a certain number of logical topologies based on network physical graph, named as Multi-topology (MT) or backup topology. The router, which is responsible for initiating the recovery action, maps the failure scenario to its corresponding backup topology, finds the next hop in the associated routing table and marks the backup topology's identifier in the header of IP packets. Then other routers forward the packets by inspecting both the topology identifier and the destination address in the packet header. This thesis focuses on the technology, evaluates the relevant routing and forwarding schemes through simulation, studies and proposes algorithms for generating multi-topologies. Besides, optimization of weight is applied to settle the load balancing issue after redirecting traffic to an alternate route using MTR.

The first chapter introduces the distribution characteristics of failures and the existing route recovery schemes in IP networks. Chapter II researches on the basic principles and technical specifications of MTR. Firstly, two types of multi-topology are introduced, which are used for protecting nodes and links respectively. Then focusing

on the protection of the link failures, both global and local recovery strategies, as well as the single or sort ascending path redirecting schemes are compared and analyzed through simulation.

Based on the existing technology, further researches and improvements are as follows: (1) Multi-topology design. The third chapter introduces three kinds of algorithms for building backup topologies. One MT generation algorithm is proposed, which has good performance on decreasing the traffic load that every topology required to carry under double-link failure scenarios. As performance metrics, path length and link load are calculated to evaluate those algorithms. (2) Link Weight Optimization. In Chapter IV, firstly an integer linear programming (ILP) model is formed, and then a solution is proposed combined with the Genetics Algorithm to optimize the link weight of multi-topologies. The maximum link utilization before and after link weight optimization has been tested and compared.

The fifth chapter describes the simulation platform constructed by OPNET tools. Chapter VI summarizes the full text.

Keywords: multi-topology routing, fast recovery, multi-topology design, link weight optimization

图目录

图 1-1	原始拓扑及其备份拓扑.....	6
图 1-2	MTR 恢复方案的研究与实现流程.....	8
图 2-1	用于节点保护的多拓扑.....	11
图 2-2	用于链路保护的多拓扑.....	12
图 2-3	链路失效后业务路径的全局恢复方案.....	14
图 2-4	链路失效后业务路径的局部恢复方案.....	14
图 2-5	采用局部恢复方案时业务路径发生交叠的示例.....	15
图 2-6	采用不同业务恢复方案后业务传输路径跳数对比.....	17
图 2-7	采用不同业务恢复方案后链路最大负载的对比.....	18
图 2-8	多链路失效时产生环路示例.....	19
图 2-9	仅允许路径切换一次备份拓扑时数据包的转发流程.....	21
图 2-10	允许路径按备份拓扑编号升序切换拓扑时数据包的转发流程.....	23
图 2-11	双链路失效情形下两种切换拓扑方案的丢包率对比.....	24
图 3-1	在拓扑总数不同的备份拓扑集合中路径跳数的分布率.....	34
图 3-2	局部恢复模式下采用不同个数的备份拓扑集合时链路最大负载的对比.....	35
图 3-3	使用 MRPH 算法生成的备份拓扑后业务传输路径的总跳数	37
图 3-4	全局恢复模式下 MRPH 与 MLDF 算法生成 MT 的负载对比	38
图 3-5	局部恢复模式下 MRPH 与 MLDF 算法生成 MT 的负载对比	39
图 3-6	全局恢复模式下 MRPH 与 MLDF 算法生成 MT 的传输路径总跳数对比	40
图 3-7	局部恢复模式下 MRPH 与 MLDF 算法生成 MT 的传输路径总跳数对比	40
图 4-1	ECMP 路由示意图.....	43
图 4-2	遗传算法生成下一代种群的过程.....	52
图 4-3	父代交叉变异过程的伪码.....	53
图 4-4	遗传算法求解备份拓扑权重的算法流程图.....	54
图 4-5	全局恢复模式下优化权重前后的最大链路利用率对比.....	56
图 4-6	局部恢复模式下优化权重前后的最大链路利用率对比.....	56
图 5-1	US 网络物理拓扑图	60
图 5-2	MTR 仿真平台网络节点模型.....	63

图 5-3	泊松数据源的进程状态机.....	64
图 5-4	节点核心模块的进程状态机.....	65
图 5-5	节点核心模块的数据包转发流程.....	65
图 5-6	中央控制器的节点模型.....	66
图 5-7	中央控制器的进程状态机.....	66

表目录

表 3-1	四种拓扑生成算法求得的备份拓扑.....	33
表 4-1	在所有单失效情形下使用 3 个 MT 时链路最大利用率对比.....	57
表 4-2	在所有单失效情形下使用 21 个 MT 时链路最大利用率对比.....	57
表 5-1	US 网络业务量需求矩阵(Gbit/s).....	60
表 5-2	US 网络链路长度(km).....	61

缩略词表

英文缩写	英文全称	中文释义
AMPL	A Modeling Language for Mathematical Programming	一种数学建模语言
AS	Autonomous System	自治系统
DDoS	Distribution Denial of Service	分布式拒绝服务攻击
ECMP	Equal-Cost Multi-Path	等价多径
FIB	Forwarding Information Base	转发表
FIR	Failure Insensitive Routing	失效不敏感路由
GA	Genetic Algorithm	遗传算法
IANA	Internet Assigned Numbers Authority	互联网地址分配委员会
IETF	Internet Engineering Task Force	互联网工程任务组
ILP	Integer Linear Programming	整数线性规划
IOS	Internetwork Operating System	网际操作系统
IP	Internet Protocol	网际协议
IS-IS	Intermediate System to Intermediate System	中间系统到中间系统
LSA	Link State Advertisement	链路状态通告
MPLS	Multi-Protocol Label Switching	多协议标签交换
MRC	Multiple Routing Configurations	多路由配置
MT	Multi-topology	多拓扑
MTR	Multi-topology Routing	多拓扑路由
O2	Outdegree 2	二出度路由
OPNET	Optimized Performance Network Engineering Tool	网络仿真工具

缩略词表

OSPF	Open Shortest Path First	开放式的最短路径优先
QoS	Quality of Service	服务质量
RFC	Request For Comments document	请求注解文档
RRL	Resilient Routing Layers	弹性路由层
SPF	Shortest Path First	最短路径优先
TCP	Transport Control Protocol	传输控制协议
TOS	Type Of Service	服务类型
VoIP	Voice over IP	IP 语音

与本文所提算法相关的重要缩略词

Minimum	Minimum number of MT	最小化多拓扑总数
MLDF	Minimum MT Load under Double link Failure	最小化双链路失效情形下多拓扑的负载
MRPH	Minimum Recovery Path Hop	最小化恢复路径跳数
Rich	Rich MT connectivity	增加多拓扑连通度

独 创 性 声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

签名： 贾冰 日期：2010 年 5 月 30 日

论 文 使 用 授 权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

签名： 贾冰 导师签名： 王 斌
日期：2010 年 5 月 30 日

第一章 绪论

1.1 引言

近年来互联网对于传输实时业务的需求已日趋提高。诸如 VoIP (Voice over Internet Protocol)、流媒体、网游、视频会议等业务对于网络的传输延迟和可靠性等指标都有着严格的要求^[1]。同时,解决由于网络设备故障而导致的连通性失效成为大型通讯系统每天要应对的棘手问题。

IP 网络本身是健壮的,因为诸如开放式的最短路径优先 (Open Shortest Path First, OSPF) 或中间系统到中间系统 (Intermediate System to Intermediate System, IS-IS) 这样的路由协议被设计为在网络设备发生失效后可以采用路由重收敛机制,即网络域内所有路由器之间通过洪泛链路状态通告报文来更新邻接链路状态,重新计算有效路由并更新转发表 (Forwarding Information Base, FIB)。然而在网络还没有收敛到一个新的状态之前,会造成一段时间内的路由不稳定:即由于不是所有路由器都在同一时刻接收到链路状态更新信息并启用新的转发表,导致数据包可能会因路由无效而被丢弃,或者传输路径产生暂时的环路,即微环 (micro loops),引发时延增加等问题。传统的路由重收敛机制由于重建新路由的处理过程很慢,已不能很好的满足新型业务的服务质量 (Quality of Service, QoS) 需求。

因此,在提供无连接服务的 IP 网络中需要一种快速恢复机制,该机制应具备以下特点:(1) 先应式的 (proactive); (2) 局部的; (3) 尽量避免引发拥塞。这种机制应该是先应式的——即在检测到一处失效之前,一条备份路由已经是可用的,当传输路径上设备发生失效时可以实现快速反应;这种机制还应是局部的,从而可以减少被告知失效的路由器的数量——只有需要更改路由的路由器才会被告知,其余的路由器可一如既往地转发业务流;此外还应注意使用备份路径承载待恢复的业务时不会引起链路拥塞。

下面概述网络设备失效的种类及分布特征、现有的 IP 网络恢复技术以及本文的主要研究内容。

1.2 链路失效情形概述

网络是使用通信线路和各种连接设备将计算机互联,按照协议实现通信的系

统。良好优质的网络环境是人们之间实现有效沟通和资源共享的基础，然而网络链路失效的情形在现实生活中是不可避免的。IP 网络的链路失效定义为在 IP 层上、或其下层发生的失效。网络失效的症状是两个直连路由器的连通性受到破坏，在逻辑上可以看作是两个设备终端之间的链路失效。

引起网络链路失效的原因是多种的，其分布特征也非常复杂，具有一定的规律性和随机性。路由器硬件故障、光纤断裂、光器件故障、软件漏洞、病毒侵入以及人为的协议配置错误及维护不当等等都可能会导致链路失效。人们通过对真实网络环境的监测，大致总结了链路失效的成因和分布特征，下面分别予以介绍。

1.2.1 链路失效成因

链路失效产生的原因可分为两类：由维护操作行为所导致的链路失效和由非维护操作行为导致的链路失效。

由人为的维护操作不当所造成的链路失效，是指由于维修的调度安排不当、协议配置错误、防火墙设置不当等引发的链路通信故障。

由非维护操作行为所导致的链路失效的成因主要包括：（1）硬件原因：通常是指由架构网络的设备，如网卡、网线、路由、交换机、调制解调器等硬件设备引起的网络断链；（2）软件原因：软件漏洞也易使网络设备运行异常；（3）病毒入侵：恶劣频繁的网络攻击，如分布式拒绝服务（Distribution Denial of Service, DDoS）攻击、蠕虫病毒等，能够在短时间内造成大量网络设备瘫痪。（4）自然灾害：网络链路还面临着自然老化，以及来自如地震、火山、洪水等所导致的设备断电、损毁等风险。

对于由非维护操作行为所导致的链路失效，按照同一时段内发生失效的链路的数量，可将其划分为单链路失效和多链路失效。其中单链路失效是指在某段时间域内仅有一条链路发生失效，而多链路失效是指多条链路并发失效，主要是由路由器或光设备的故障所引发的共享失效风险的多条链路同时失效。

1.2.2 链路失效分布特征

人们以 Sprint 网络为监测对象，收集并统计了为期 6 个月的链路状态信息^[2]，发现在这一时间域内所发生的链路失效情形中，20%的失效是由于维护行为不当所导致。在剩余 80%的非人为维护因素所引发的失效情形中，30%为多条链路并发失效，70%为单链路失效。对于这部分单失效情形，有 55%集中在网络中 3%的链路

集合中，即非维护操作行为导致的高频单链路失效情形占有失效情形的 30.8%。

通过对真实网络失效情形数据进行统计分析及绘图，发现可以使用 Weibull 分布函数来拟合网络链路失效后的恢复时间分布特性。大多数失效是短时的^[3]：50% 的失效情形持续时间少于一分钟，40% 的失效持续时间介于 1 分钟与 20 分钟之间，仅有 10% 的失效情形持续时间超过 20 分钟。

1.3 恢复技术概述

恢复 (recovery) 是指在设备发生失效后网络返回到一个可运作的状态的过程。网络在失效后可以提供并维持可接受的服务水平的能力称为网络的恢复力或弹力 (resilience)^[4]。

对于具有恢复力的网络，要求其通信协议能够在设备发生失效后使用其它路径保持业务传输，这一功能可由网络层协议（例如 IP）或底层协议来完成。人们更倾向于在网络层实现恢复，因为应用于底层的恢复机制不能解决与路由器软件或硬件相关的失效行为，此外由于网络层中几条逻辑链路可能共享底层的同一物理光纤，因此网络层的恢复比底层的恢复具有更细粒度。

IP 网络泛指基于 IP 协议的分组交换网络，在网络层转发数据包可以采用无连接或面向连接机制：基于 IP 地址的逐跳转发机制是无连接的转发机制，基于路径标签转发的多协议标签交换 (Multi-Protocol Label Switching, MPLS) 是面向连接的转发机制。由于不是所有的 IP 网络都支持 MPLS，并且 MPLS 需要预先建立大量带有路径标签的备份路径，建立和管理这些备份路径需要一定的代价，因此本文重点研究应用于使用无连接转发机制的 IP 网络中的恢复技术。

目前，多数大型网络使用 OSPF 路由协议作为域内路由协议^[5]，当网络拓扑发生变化时 OSPF 采用重收敛机制重建路由。在下面小节中依次介绍 OSPF 路由重收敛的过程及其不足，可应用于 IP 网络快速恢复的候选方案，以及本文的研究对象 MTR 技术。

1.3.1 OSPF 路由重收敛过程及其不足

OSPF 是链路状态路由协议，用于在自治系统 (Autonomous System, AS) 内部维护路由器的邻接信息并决策路由。在采用 OSPF 协议的 IP 网络中，路由器之间通过发送 Hello 报文建立邻接关系，使用链路状态通告 (Link State Advertisement, LSA) 报文交换网络拓扑信息，更新链路状态数据库，根据数据库中所存储的拓扑

结构信息使用 SPF 算法计算最短路径树，构造路由表和转发表。

SPF (Shortest Path First, 最短路径优先) 算法又称为 Dijkstra 算法，每个路由器使用此算法计算自身到其它路由器的最短路径，这些路径的集合在结构上可视为一棵以该路由器为根的最短路径树。路由算法计算最短路径所使用的计量标准称为权重 (weight) 或度 (metric)，通过对每条链路自动配置或人工赋以一个范围为 [1,65535] 的整数，用来描述使用此链路传输业务时所需的代价 (cost)。使用路径所经过的所有链路的权重之和来衡量路径代价，业务源宿节点间具有最小代价的路径称为最短路径。评定代价的可用标准包括跳数 (即路径所经过的路由器个数)、延迟 (数据包的传输时间)、地理距离、容量等。Cisco 公司推荐按照链路容量的倒数设置权重。

OSPF 协议对于网络拓扑结构的变化具有动态的自适应特性，即当拓扑发生变化时 (例如增减路由器，设备发生故障等)，将触发重收敛过程使同一区域内所有路由器所维护的链路状态数据库信息重新达到一致。OSPF 的路由重收敛过程是指——路由器使用 Hello 报文探测与邻居节点的邻接关系，当探测到连续 3 次丢失 Hello 报文时，使用 LSA 将变化通告给所有与其相邻的路由器。全网所有路由器以洪泛的方式发送 LSA，更新自身所维护的链路状态数据库，新的拓扑结构信息扩散至全网，直至所有路由器的链路状态数据库信息相同，即使全网所有路由器对网络拓扑的认知达到一致。

然而 OSPF 的路由重收敛过程仍存在不足。首先，重收敛过程的时间量级为秒级，其每一阶段 (如探测失效、创建 LSA、洪泛、最短路径计算、更新路由信息库、更新转发信息库) 的完成都需要一定时间。例如在探测失效阶段，RFC (Request For Comments document, 请求注解文档) 中规定发送 Hello 报文的时间间隔为 10 秒^[6]，这一时间量级是实时应用服务所不能接受的。此外，在重收敛过程未完成之前常常伴随着一段时期内的路由不稳定和丢包^[7]。

人们就如何提高重收敛过程的速度做了很多尝试^[8,9]——例如缩短发送 Hello 报文的时间间隔以减少探测时间；在拓扑稳定时期减少 LSA 的更新数量以提高失效后洪泛 LSA 的效率；使用原始的最短路径树作为输入数据以提高计算新的最短路径树的速度等。然而即使应用了这些方法，仍不能使重收敛的时间量级满足新的实时服务的需求。此外由于网络中大多数失效是短时的^[2]，过于快速的触发重收敛过程仍会导致路由抖动，增加了网络的不稳定性，当多失效发生时这一情形将更为严重。

OSPF 的路由重收敛过程是反应式的 (reactive) 和全局的，当探测到失效发生

后才建立恢复路由，并且需要向网络中所有路由器通告拓扑变化信息。为了应对短时失效，兼顾恢复速度与网络的稳定性，亟待提出先应式的、局部的恢复机制，即预先建立备份路由，在失效发生后仅由部分路由器负责恢复业务传输。

1.3.2 IP 网络快速恢复技术介绍

为了解决网络设备发生失效后路由重收敛机制所导致的服务质量降级，人们提出了一系列应用于 IP 网络的快速恢复技术^[10]。这些恢复技术不依赖于洪泛链路状态通告来重建路由，而是为可能发生的失效情形预先计算可用的备份路由。

二出度（Outdegree 2，O2）路由的基本思想是构造网络使所有节点到所有目的地都有两个有效的下一跳^[11,12]。正常情况下流量分配给这两个下一跳，当失效发生后它们互为对方的下一跳。为了避免产生路由环路，在网络无失效情形下一些链路不参与到达某些目的地的报文转发，只是被用于备份路由。O2 机制的缺点是为了确保存在两个无环的下一跳，在无失效情形下使用的路由通常不是最优的。此外为了尽可能保护更多的节点或链路，这一恢复机制对网络的连通程度要求较高。在连通度较低的网络中，O2 机制将无法保护某些链路和节点。

失效不敏感路由（Failure Insensitive Routing，FIR）可以对任意连通性的网络进行单链路保护^[13]或节点保护^[14]。其基本原理是路由器不是通过链路状态通告被明确告知失效信息，而是通过报文到达了一个不寻常的接口这一现象来推断出有链路失效发生。通过预先计算出不寻常的报文转发现象对应哪些可能的链路失效，建立可以避免经过失效链路的基于接口的路由表，路由器根据报文的目的地和接收报文的接口查找下一跳。FIR 的主要缺陷是：①当多条链路并发失效时转发路径可能产生环路。使用基于“黑名单”的转发方式可解决这一问题^[15]，但采用此办法后 FIR 无法保证可对任意连通性的网络进行单链路保护；②需要基于一个非标准化的路由协议支持 FIR 机制。

IETF 提出了 not-via 恢复机制，可以保证对任意连通性的网络进行单链路或单节点保护^[16]。其主要思想是当传输路径上发生故障时，通过隧道将数据包绕过失效设备继续传输，即“不经过”失效设备（not-via 由此得名）。例如为了保护节点 N，“不经过 N”的特殊地址将分别指派给节点 N 的每一个邻居节点。当节点 N 失效后，检测到失效的节点将在移除节点 N 的拓扑中计算到达“不经过 N”的地址的最短路径，原本应转发给节点 N 的数据包被封装，发往原始工作路径上失效设备的下游节点 X，在节点 X 处数据包被解封装，按照正常情况下的路径传送到目

的地。Not-via 的不足之处在于：①需要使用隧道，对数据包进行封装和解封装增加了路由器的负担；②路由器需要在转发表中存储大量的 not-via 地址。对于链路总数为 E 的网络，需要存储 $2E$ 个 not-via 地址（即每一条链路的两端点各对应一个 not-via 地址）；③当失效发生后需要以 not-via 地址作为隧道的终点，计算到达此地址的最短路径，增加了运算量。

1.3.3 多拓扑路由技术——MTR

近年来继多个版本的相关草案被提出后，多拓扑路由技术 MTR 被正式列入 RFC^[17,18]。这些 RFC 描述了为使用多拓扑路由技术而对 OSPF 和 IS-IS 协议所进行的一些扩展，从而可定义一些独立的逻辑拓扑，用于为不同类型的业务计算路径。此外，Cisco 公司的网际操作系统（Internetwork Operating System, IOS）支持对多拓扑路由技术的配置^[19]，用于传输如音频、视频、数据等不同类型的业务。

多拓扑路由的主要思想是基于网络的物理拓扑（或称为原始拓扑），预先配置多个逻辑拓扑，称为多拓扑或备份拓扑。每个备份拓扑同原始拓扑相比，都缺少一部分节点或链路，缺少的这部分节点或链路称作在这个备份拓扑中被保护。构造备份拓扑的基本原则是使原始拓扑中的每个节点或链路都可以至少被一个备份拓扑所保护。网络中的路由器为每个备份拓扑分别维护一张路由表。

在数据包的转发过程中，由业务的源节点将原始拓扑标识（即编号）写入数据包首部。当检测到某网络设备发生失效后，对于在原始拓扑中工作路径经过此失效设备的业务，需要为其重新计算恢复路径以保证业务的连续传输。由负责恢复业务传输路径的路由器选择可保护此失效设备的备份拓扑，将数据包所携带的拓扑标识更新为此备份拓扑编号，用来告知其它路由器转发此数据包时所应使用的拓扑，路由器在此拓扑所对应的路由表中根据目的地址查找下一跳并转发数据包。采用 MTR 技术可在检测到失效后瞬时恢复业务的传输路径，避免了在路由重收敛过程未完成之前可能造成的丢包和路由环路等问题。MTR 技术可应用于自治系统的域内恢复和域间恢复^[20]。

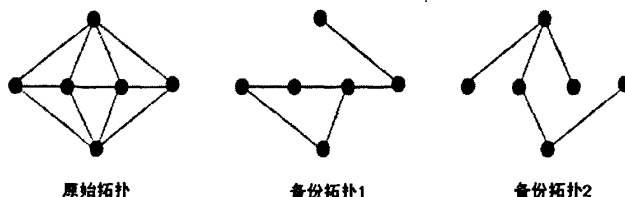


图 1-1 原始拓扑及其备份拓扑

图 1-1 是一个使用两个备份拓扑来保护原始拓扑中链路的例子。挪威 Simula Research 实验室的 Amund Kvalbein^[21]针对“减少路由器存储的信息量、增加备份拓扑连通度和应对多链路并发失效”这三个指标，提出了三种用于构造链路保护的备份拓扑的算法^[22]。实验证明当网络设备失效后使用备份拓扑上的恢复路径传输业务时，将导致链路上负载被重新分配，在一些情况下会造成拥塞。

因此，探索如何构造生成具有更好性能的备份拓扑，以及解决采用备份拓扑恢复业务传输之后的负载均衡问题成为了今后的研究方向。

1.4 本文主要工作及结构安排

本文以多拓扑路由（MTR）技术作为研究对象，在该技术目前发展状况的基础上，对路由转发方案进行了性能的定量评估及分析，改进和提出拓扑生成算法，并设计优化拓扑权重的方案。本文的主要贡献及创新点可归纳为以下三点：

（1）通过仿真测试对比多拓扑路由技术的相关路由转发方案；

（2）对比分析现有的三种生成备份拓扑的算法，并对其中一种算法加以改进。通过建立 ILP 数学模型，提出一种生成备份拓扑的新方法，使得在双链路失效情形下，可降低备份拓扑所需承载的业务量；

（3）建立求解多拓扑权重的线性规划模型。使用遗传算法设计备份拓扑链路权重的求解方案，解决采用备份拓扑恢复业务传输后的负载均衡问题。采用优化后的权重，可降低所有单链路失效情形下链路利用率的最大值。

本文后续章节内容安排如下：

第二章介绍多拓扑的种类以及恢复方案。包括以下三部分内容：①备份拓扑种类——按照所保护的失效设备类型，分为用于节点保护和用于链路保护的备份拓扑；②业务恢复方式——根据负责启动失效恢复的节点在网络中所处位置，分为全局恢复和局部恢复；③路由在备份拓扑间的切换方式——在多链路失效场景下，仅允许使用一次备份拓扑单次切换恢复路径或允许按照拓扑编号升序多次使用备份拓扑。在本章中选取用于链路保护的备份拓扑作为实验拓扑，以 OPNET 为仿真工具，模拟数据包在单、双链路失效场景下的动态转发过程。对于②和③这两类候选方案，以链路负载、恢复路径跳数及丢包率作为性能指标，通过仿真进行方案性能的对比分析。

第三章研究生成备份拓扑的算法。首先介绍现有的三种拓扑生成算法：①可生成数量最少的备份拓扑的 Minimum 算法；②可生成连通度更大的备份拓扑的

Rich 算法；③可最小化恢复路径总跳数的 MRPH 算法。对于 MRPH 算法，通过添加约束条件加以改进，使得每条链路仅被一个备份拓扑所保护。然后通过建立 ILP 模型，提出一种用于生成备份拓扑的 MLDF 算法。该算法旨在分别针对全局和局部这两种恢复方案生成备份拓扑，使得在双链路失效情形下可最小化备份拓扑所需承载的业务量。在本章中以负载及跳数作为性能指标，对比分析了采用四种算法所求得的备份拓扑的性能。

第四章研究备份拓扑的权重设计方案，从而进一步改善将 MTR 技术应用于失效恢复后的性能。以最小化所有单链路失效情形下的全网链路最大利用率为优化目标，提出了求解备份拓扑链路权重的 ILP 模型，并采用遗传算法求解权重。通过仿真测试对比了优化权重前后的链路利用率。并同允许业务在多条可行路径间按任意比例分流的传输方案性能进行对比。

第五章介绍了采用 OPNET 工具搭建的仿真平台，用以测试本文所研究的方案和算法性能。

第六章对全文进行总结。指出了研究中的不足和下一步研究方向。

图 1-2 为实现 MTR 技术方案的整体框架。

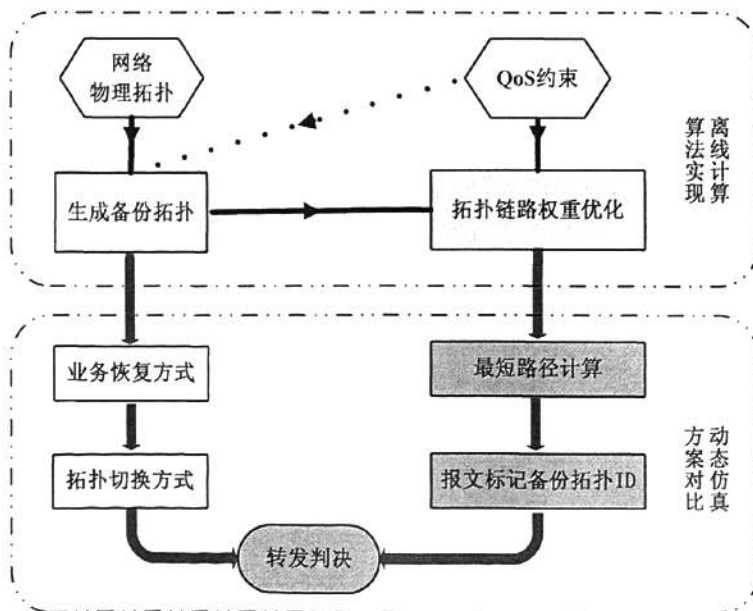


图 1-2 MTR 恢复方案的研究与实现流程

第二章 MTR 基本原理及方案的研究

2.1 研究背景

2.1.1 RFC4915 简介

RFC4915^[17]描述了为使用多拓扑路由（MTR）技术而对现有的 OSPF 路由协议所进行的扩展。旨在基于 IP 网络的物理拓扑，定义一组独立的逻辑拓扑，用来为单播、多播以及其它不同类型的业务计算不同路径。路由器为每个拓扑分别计算最短路径树，维护不同的路由表和转发表。

在 RFC4915 中，主要介绍了以下几方面的技术规范：

（1）建议重新定义数据包格式中的服务类型（Type of Service, TOS）字段；

OSPF 协议的数据包封装在 IP 报文中传输。RFC4915 提出可以重新定义 LSA 报文中的 TOS 字段，用于存储多拓扑编号（MT-ID），通过通告同一条物理链路对应于不同 MT-ID 的链路权重值，来控制该链路可以参与到哪些拓扑中用于承载业务。定义若某条链路在某个拓扑中的权重值为 65535，则视为此拓扑中不包含这条链路。

（2）MT-ID 的有效取值范围；

TOS 字段为 8bit，MT-ID 有效值范围是 0 至 127。所保留的 MT-ID 数值具体含义如下：

0 ——用于通告与缺省拓扑（即物理拓扑）相对应的权重值。

1 ——用于通告与缺省多播拓扑相对应的权重值。

2 ——用于 IPv4 域内的管理。

3-31 ——由国际因特网地址分配委员会（Internet Assigned Numbers Authority, IANA）分配。

32-127 ——为扩展，实验或私有用途而保留。

（3）路由器的协同；

提出增加一个称为“DefaultExclusionCapability”的路由器配置参数，通过对该参数的激活或禁用，可实现业务在支持 MTR 的路由器与不支持 MTR 的路由器之间平滑传输。

(4) OSPF 协议 5 种 LSA 的报文格式。

此外, 在 RFC4915 中, 关于如何将数据包与对应的拓扑相关联, 转发数据包过程中多个拓扑之间的信息如何交互, 以及网络安全性方面都未予以讨论。RFC5120^[18]提议在数据包的转发判决过程中, 可根据 IP 报文首部字段或数据字段内容选择对应的转发表。

2.1.2 多拓扑类型

当网络设备发生失效时, 可以采用 MTR 技术为受失效所影响的业务计算恢复路径, 选择不包含该失效设备的备份拓扑, 即选择可保护此失效设备的备份拓扑来传输业务。同一备份拓扑中可以保护多个网络设备, 同时必须确保网络的每个设备至少可在一个备份拓扑中被保护。

按照被保护的网络设备类型, 可将多拓扑 (MT) 分为用于保护节点的多拓扑和用于保护链路的多拓扑两种。本小节分别介绍这两类多拓扑, 并在下文中重点研究在链路失效情形下采用保护链路的多拓扑实现业务恢复的相关算法和方案。

2.1.2.1 用于保护节点的多拓扑——MRC

多路由配置 (Multiple Routing Configurations, MRC) 是用于在节点/链路失效情形下实现路由快速恢复的多拓扑^[23,24]。MRC 的主要思想是: 基于网络的原始拓扑生成一组备份拓扑, 每个备份拓扑均包含原始拓扑的所有节点和部分链路。通过配置不同备份拓扑的链路权重, 使得在每个备份拓扑中, 某些节点的邻接链路会因权重值过大而不会被 SPF 算法选作恢复路径上的链路, 从而使得业务的恢复路径不经过失效设备。

在 MRC 中, 按照所配置的链路权重值, 可将备份拓扑中被保护节点的邻接链路分为以下两类:

①被孤立的链路 (isolated link): 链路权重值设为无穷大 (在 OSPF 路由协议中该值通常设置为 65535), 使得此链路不会用于承载业务。

②被限制的链路 (restricted link): 链路权重值设为足够大 (建议设置为原始拓扑中所有链路权重之和), 使得仅当被保护的节点是数据包的源节点或目的节点时, 才会使用此链路传输业务。

为了保证可以在单节点失效后实现业务恢复, 要求用于节点保护的备份拓扑 MRC 满足以下条件:

(1) 如果某个节点在某备份拓扑中被保护, 那么此节点在这个备份拓扑中不

能参与业务传输。此外，业务流必须能够离开或到达被保护的节点。

(2) 如果某条链路在某备份拓扑中被孤立，那么此链路在该拓扑中不能参与业务传输。

(3) 在每个备份拓扑中，所有节点对之间都必须存在这样一种路径——路径不经过任何被保护的节点或被孤立的链路。

(4) 原始拓扑中的每个节点必须至少在一个备份拓扑中被保护，每条链路必须至少在一个备份拓扑中被孤立。

图 2-1 是 MRC 的构造示意图。

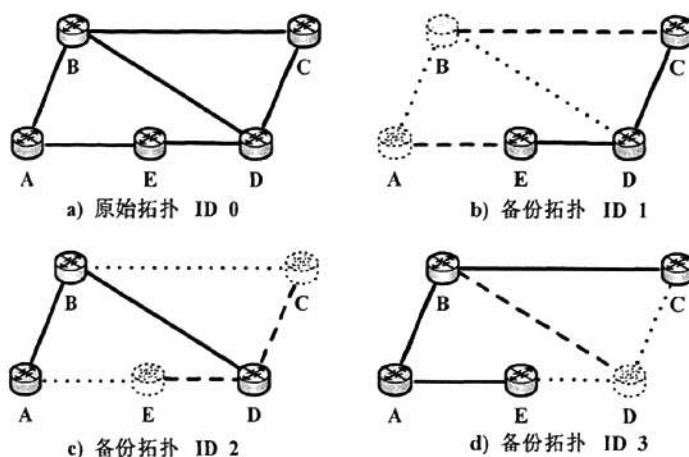


图 2-1 用于节点保护的多拓扑

如图 2-1 所示，图中 a) 为网络的原始拓扑，由 5 个节点和 6 条链路组成，链路视为双向。原始拓扑编号为 0。图 b)、c)、d) 为基于原始拓扑所生成的 3 个用于保护节点的备份拓扑，编号依次为 1、2、3。备份拓扑所保护的节点用白色标注，被孤立的链路用点线标注，被限制的链路用虚线标注。其中备份拓扑 1 保护的节点为 A 和 B，备份拓扑 2 保护的节点为 C 和 E，备份拓扑 3 保护节点 D。

假定网络原始拓扑的链路权重均为 1，使用 SPF 算法计算最短路径。设节点 A 到 C 之间有业务需要传输，由图 a) 可知该业务的原始路径为 A-B-C。若节点 B 发生失效，则使用保护该失效节点 B 的 1 号备份拓扑将业务传输到目的地，业务的传输路径为 A-E-D-C。

2.1.2.2 用于保护链路的多拓扑——RRL

弹性路由层 (Resilient Routing Layers, RRL) 是用于在链路失效情形下实现路由快速恢复的多拓扑^[22]。RRL 的主要思想是：基于原始拓扑生成一组备份拓扑，

若原始拓扑的某条链路不存在于某个备份拓扑中，则称此链路在该备份拓扑中被保护。一个备份拓扑可以保护原始拓扑的多条链路，原始拓扑中的每条链路至少应在一个备份拓扑中被保护。

在 OSPF 路由协议中，通过 LSA 向全网节点（路由器）通告同一链路在不同拓扑中的权重值，将权重值为 65535 视为此链路被保护。网络中所有节点为每个备份拓扑分别维护一张路由表，当节点探测到链路发生失效后，原本应经过失效链路传输的业务将使用保护此失效链路的备份拓扑上的恢复路径传输。

图 2-2 是 RRL 的构造示意图。

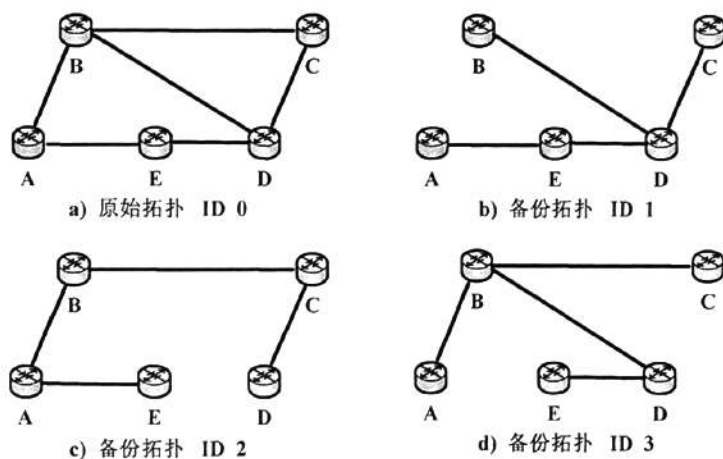


图 2-2 用于链路保护的多拓扑

如图 2-2 所示，图 a) 为网络的原始拓扑，由 5 个节点 6 条链路组成，拓扑编号为 0。图 b)、c)、d) 为基于原始拓扑所生成的 3 个用于保护链路的备份拓扑。其中备份拓扑 1 保护的链路为 A-B 和 B-C，备份拓扑 2 保护的链路为 B-D 和 E-D，备份拓扑 3 保护的链路为 A-E 和 C-D。由于被保护的链路在备份拓扑中不用于承载业务，因此在图中不予画出。

2.2 链路失效后的业务恢复方案

最初 MTR 技术被提出可用于在流量工程领域解决网络链路的拥塞控制问题^[25]。即当节点检测到链路发生拥塞后，可以将适量的业务使用备份拓扑重新寻径路由，以降低网络的链路利用率。

关于多拓扑应用于拥塞控制，有全局法和局部法两种解决方案——全局法是

基于已知业务量需求的前提下，离线计算重载链路，通过指派业务在其源点以一定的分流比通过多个拓扑上的路径传输，来避免路径经过重载链路，从而预先避免发生拥塞；局部法是在线实时调整业务的传输路径，即为路由器转发模块的缓冲区队列长度设置一个阈值，当队列长度大于阈值时视为连接转发端口的链路发生拥塞，则将业务使用与此链路相对应的备份拓扑上路径传输，从而缓解拥塞。

受此 MTR 应用的启发，Amund Kvalbein 建议在将 MTR 技术应用于网络设备失效后的重路由时，可以按照失效后负责启动恢复操作的节点在业务路径上的位置，将业务的恢复方案分为全局恢复和局部恢复。

全局恢复是指把原本应经过失效设备传输的业务，在此业务的入口节点（ingress node），即业务源点处开始使用备份拓扑上的恢复路径传输；局部恢复是指业务按照原始工作路径传输，当传输至检测到邻接设备发生失效的节点，即失效设备的上游节点（upstream node）时，由此节点负责计算业务在对应备份拓扑上到达目的节点的恢复路径，并予以转发。

下面采用图 2-2 中的原始拓扑和备份拓扑，分别举例说明当链路发生失效后这两种业务的恢复方案，并通过仿真对比其性能。

2.2.1 全局恢复

为了实现业务路径的全局恢复，要求业务的源节点可在业务路径上设备发生故障后快速启动恢复操作。研究者们建议可在网络中令一节点作为“中央控制器”——即负责收集全网节点所检测到的链路状态更新信息，当发生链路失效后，中央控制器向受失效所影响的业务的源点通告失效链路编号。业务的源点收到失效链路信息后启动恢复操作，查找可保护此失效链路的备份拓扑，将备份拓扑编号写入数据包首部字段，然后按照此备份拓扑的转发表将数据包交付给下一跳。网络中其它节点在收到数据包后，解析拓扑标识，查找该备份拓扑的转发表，将数据包逐跳传输至业务宿点。

图 2-3 是业务全局恢复方案的示例，采用的网络原始拓扑和备份拓扑为图 2-2 中 a) 和 b)。假定原始拓扑中链路权重值均为 1，使用 SPF 算法为业务计算最短路径。设节点 A 到 C 之间有业务需要传输，该业务在原始拓扑中的工作路径为 A-B-C。若链路 B-C 发生故障，采用全局恢复方案时，由业务源点 A 启动恢复操作，查找可保护失效链路的备份拓扑为 1 号备份拓扑，在此备份拓扑中业务从源点到宿点的传输路径为 A-E-D-C。节点 A 将数据包中拓扑标识由原始拓扑编号 0 更新为

备份拓扑编号 1，然后把数据包转发给恢复路径上的下一跳节点 E。

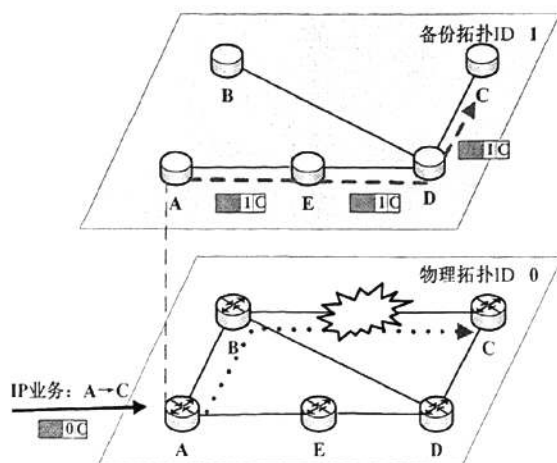


图 2-3 链路失效后业务路径的全局恢复方案

2.2.2 局部恢复

业务路径的局部恢复方案是指当节点检测到其邻接链路发生失效后，由检测到失效的节点，即失效链路的上游节点启动恢复操作，查找用于保护此失效链路的备份拓扑的编号，更新报文字段中的拓扑标识，将业务交付给备份拓扑中路径的下一跳节点继续进行传输。同全局恢复相比，局部恢复方案不需要中央控制器，而是由检测到失效的节点直接在本地切换传输业务所使用的备份拓扑。

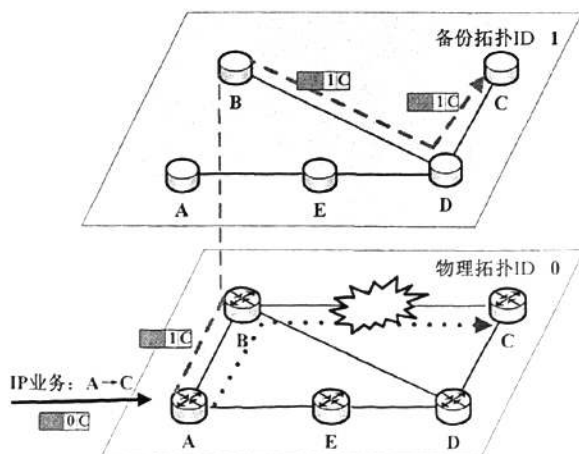


图 2-4 链路失效后业务路径的局部恢复方案

图 2-4 是业务局部恢复方案的示例，采用的网络原始拓扑和备份拓扑如图 2-2

中 a)和 b)所示。节点 A 到 C 之间的业务在原始拓扑中的工作路径为 A-B-C。若节点 B 检测到链路 B-C 发生故障，采用局部恢复方案时，节点 B 将更新数据包所携带的拓扑标识，使用保护该失效链路 B-C 的备份拓扑将数据包继续传输到目的节点 C。业务在备份拓扑中从失效链路的上游节点 B 到目的节点 C 的路径为 B-D-C。发生链路失效后该业务在网络中的传输路径为 A-B-D-C。

不同于 MTR 全局恢复方案，局部恢复方案规定业务在从源点到失效链路上游节点之间的传输路径保持不变，仍使用原始拓扑中路径，业务传输至失效链路上游节点处才被切换至备份路径传输。业务从源到宿的传输路径由于分段采用原始拓扑和备份拓扑上的路径，因此传输路径所经过的链路可能会发生交叠。

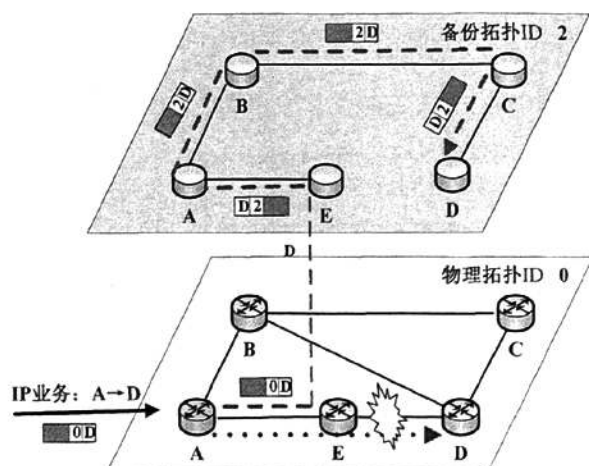


图 2-5 采用局部恢复方案时业务路径发生交叠的示例

图 2-5 是采用局部恢复方案后业务传输路径发生交叠的示例，采用的网络原始拓扑和备份拓扑如图 2-2 中 a)和 c)所示。假定节点 A 到 D 之间有业务需要传输，该业务在原始拓扑中的工作路径为 A-E-D。若节点 E 检测到链路 E-D 发生故障，采用局部恢复方式时，由失效链路的上游节点 E 启动恢复操作，使用保护该失效链路 E-D 的 2 号备份拓扑上路径将数据包继续传输到目的节点 D。业务在备份拓扑中从节点 E 到目的节点 D 的路径为 E-A-B-C-D。发生链路失效后该业务在网络中的传输路径为 A-E-A-B-C-D。即数据包在转发过程中会两次经过链路 A-E 传输。

2.2.3 业务恢复方案性能对比

OSPF 的路由重收敛是指当网络设备发生故障后，全网节点通过交换 LSA 更

新链路状态数据库，最终对整个网络的拓扑结构的认知重新达到统一的过程。重收敛过程结束后，网络各节点计算业务路径所使用的拓扑是在原始拓扑中仅去掉失效设备后的拓扑。MTR 的核心思想是指当失效发生后，由业务的源节点（全局恢复）或失效设备的上游节点（局部恢复）为原本应通过失效设备传输的业务计算新的恢复路径，所使用的拓扑是预先配置的一组备份拓扑。

下面通过仿真，以路径跳数和链路负载为性能指标，对：①OSPF 路由重收敛；②MTR 全局恢复；③MTR 局部恢复这三种方案的性能进行对比分析。

本节中仿真的相关参数设置如下：网络的原始拓扑图采用图 5-1 的 US 网络拓扑，该拓扑包含 14 个节点和 21 条双向链路。业务量需求矩阵（单位 Gbit/s）和链路的物理长度（单位 km）分别按照表 5-1、5-2 中参数设置。节点对 st 之间的业务量需求定义为业务源节点 s 每秒钟产生的以节点 t 为目的的节点的业务量。链路权重采用“ $1 + \text{链路的物理长度} \times 10^{-6}$ ”，即设置为整数 1 加上一个正比于传播延迟的小数偏移量，使用 SPF 算法计算业务路径。

网络的失效模型设置为单链路失效，21 条链路对应的 21 种单链路失效情形在仿真时间域内相互独立发生。在测试 OSPF 重收敛方案性能时，使用的拓扑是在原始拓扑中去掉当前失效链路后所构成的拓扑。在测试 MTR 全局恢复和局部恢复方案性能时，所使用的备份拓扑为用 3.1.1 节介绍的 Minimum 算法生成的拓扑，可以保护 US 网络所有链路的备份拓扑的最小个数为 3 个，备份拓扑结构详见表 3-1。

2.2.3.1 业务传输路径跳数的对比

当某链路失效后，对于在原始拓扑中的工作路径不经过该失效链路的业务，仍按照原始路径传输。因此在评估 OSPF 路由重收敛、MTR 全局以及局部恢复这三种方案的恢复路径跳数性能时，本小节仅以受失效链路所影响的那部分业务作为研究对象，统计业务传输路径的跳数。

图 2-6 是在无失效情形以及有失效发生后分别采用上述三种恢复方案的业务传输路径跳数（即数据包从源点传输到宿点所经过的节点数目）。图中横坐标为失效链路编号，取值范围为[1,21]。纵坐标代表传输路径的平均跳数，即“所有受当前失效链路所影响的业务的传输路径跳数之和/ 路径总数”。

对于原始工作路径经过某条链路的所有业务，在下列场景：①无链路发生失效；②该链路失效后采用 OSPF 路由重收敛；③该链路失效后采用 MTR 进行业务的全局恢复；④该链路失效后采用 MTR 进行业务的局部恢复，这 21 种链路失效情形下业务传输路径的平均跳数的最大值依序为 2.58824、4、5.46154、6.46154。

对于所有可能发生的失效链路情形，业务传输路径的平均跳数的均值依次为 2.37065、3.47175、4.1679、4.79845。

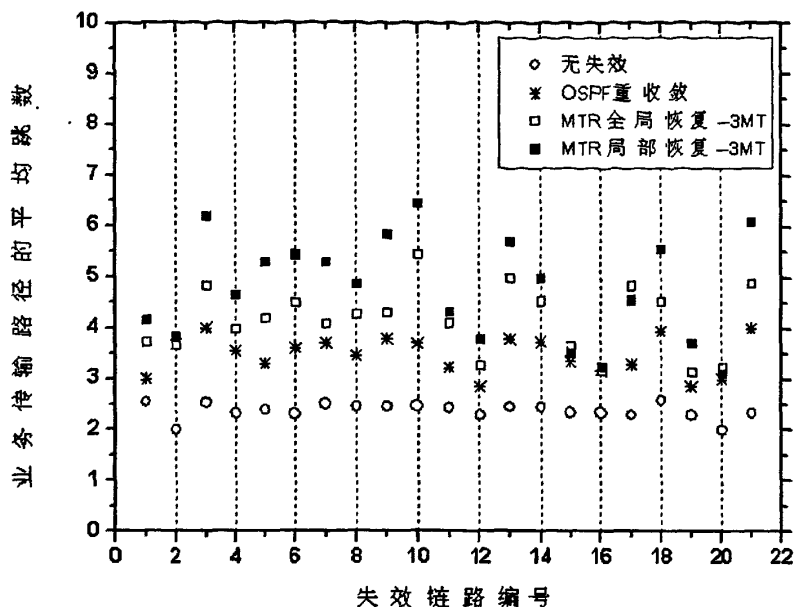


图 2-6 采用不同业务恢复方案后业务传输路径跳数对比

由上述数据可知，业务在原始拓扑中的工作路径跳数最小，发生链路失效后使用 MTR 恢复后所获得路径的跳数大于 OSPF 重收敛，且采用 MTR 局部恢复方案后路径跳数大于全局恢复方案。

这是因为：当网络中无链路发生失效时，业务基于原始拓扑使用 SPF 算法求解最小代价路径，本节仿真中采用的链路权重为 $1 + \text{某小数}$ ，因此所求得的最小代价路径也就是最小跳数路径；当发生链路失效后，采用 OSPF 路由重收敛时全网节点在仅去掉此条失效链路的拓扑上重新计算业务路径；采用 MTR 方案计算业务恢复路径时，所使用的备份拓扑同原始拓扑相比，不止保护一条链路（在本例中使用的一组共 3 个备份拓扑，每个备份拓扑分别保护原始拓扑的 7 条链路）。由于采用 OSPF 路由重收敛时网络更新后的拓扑比 MTR 恢复方案使用的备份拓扑连通度更大，即有更多链路可以参与业务路径的计算，因此增加了获得跳数较小的恢复路径的几率。

此外，正如 2.2.2 小节中所述，MTR 局部恢复方案是业务从源节点起仍保持使用原始路径传输，直至失效链路上游节点处才切换至备份路径传输，业务在备份拓扑上的新路径可能会与原始路径发生交叠。因此使用局部恢复所求得的业务路

径跳数会略大于使用全局恢复后的业务路径跳数。

2.2.3.2 链路负载的对比

对于“①无链路发生失效；②某链路失效后使用 OSPF 路由重收敛；③某链路失效后采用 MTR 进行业务的全局恢复；④某链路失效后采用 MTR 进行业务的局部恢复”这四种场景，通过仿真对比在每种场景下全网所有链路的负载（即链路所需承载的业务量大小）。

图 2-7 中横坐标为失效链路的编号，纵坐标代表采用相应恢复方案后，原始拓扑中所有链路的负载的最大值，单位为 Gbit/s。

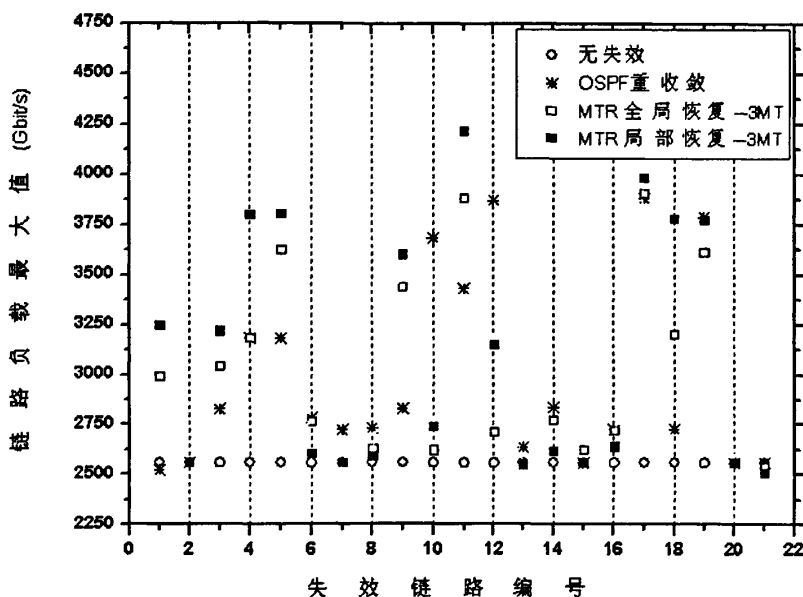


图 2-7 采用不同业务恢复方案后链路最大负载的对比

网络无失效发生时，全网链路负载最大值为 2559.64。对于 21 种单链路失效情形，分别采用 OSPF 路由重收敛、全局 MTR、局部 MTR 时，全网链路最大负载的峰值依序为 3883.18、3912.02、4218.64。全网链路负载最大值在所有可能发生的失效情形下的平均值依序为 2980.49、2976.26、3100.11。

当链路失效后计算业务的恢复路径时，由于 OSPF 路由重收敛更新后的拓扑比 MTR 使用的备份拓扑连通度更大，从而 OSPF 路由重收敛使用 SPF 算法计算业务路径时可供选择的链路范围更大，故使用 OSPF 路由重收敛后全网链路的最大负载可能低于使用 MTR 进行路由恢复后链路的最大负载。此外，MTR 局部恢复与全

局恢复方案相比, 由于业务的传输路径可能会发生链路交叠, 因此链路负载过大的几率也随之增加。

2.3 路径在多拓扑间的切换方案

在网络中采用 MTR 技术实现故障后路由的快速恢复时，负责启动恢复操作的节点查找到保护此失效链路的备份拓扑，将数据包的拓扑标识字段内容由原始拓扑编号更新为对应的备份拓扑编号，使用此备份拓扑为受失效所影响的业务计算恢复路径并转发给备份的下一跳节点。如果网络发生多条链路并发失效，在备份路径上传输的业务可能会在传输过程中再次遇到下一跳转发节点不可达的情况。此时若采用与新的失效链路相对应的备份拓扑计算恢复路径并予以转发，可能会导致数据包的传输路径产生环路。

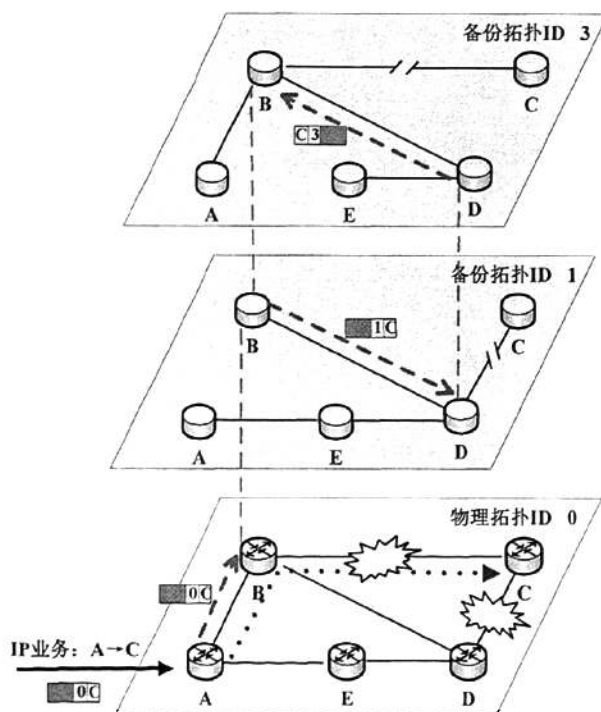


图 2-8 多链路失效时产生环路示例

图 2-8 是网络发生多链路并发失效后业务的传输路径产生环路的示例, 采用的网络原始拓扑和备份拓扑如图 2-2 中 a) 和 b)、d) 所示。假定节点 A 到 C 之间有业务需要传输, 此业务在原始拓扑中的工作路径为 A-B-C。若此时链路 B-C 和 D-C

同时失效。若采用 MTR 局部恢复方案,数据包传输至节点 B 时,节点 B 将更新数据包所携带的拓扑标识,使用保护该失效链路 B-C 的 1 号备份拓扑将数据包沿备份拓扑上路径 B-D-C 继续传输。数据包传输至节点 D 后,由于节点 D 检测到链路 D-C 失效,该节点将按照可保护失效链路 D-C 的 3 号备份拓扑上的路径 D-B-C 继续转发数据包,用于保护失效链路 D-C 的恢复路径将再次经过失效链路 B-C。因此链路发生失效后该业务在网络中的传输路径为 A-B-D-B-D-.....。

为防止路由产生环路时数据包被无限次转发,在 IP 数据包中有一个长度为 8bit 的 TTL (Time To Live) 字段,限制了数据包传输路径可以经过的最多路由器数为 255,路由器转发数据包时将此字段中数值减 1, TTL 值为 0 时丢弃该数据包。路由环路将导致丢包和链路拥塞等问题。

MTR 所使用的每个备份拓扑可分别为受失效影响的业务提供无环路由,但在多链路失效情形下,使用多个备份拓扑协同计算所得的恢复路径可能会产生环路。为避免这一问题发生,人们提出了这样一种解决方案——由于多链路失效情形仅占有失效情形的 30% (如 1.2.2 小节所述),此外,每个备份拓扑可保护原始拓扑的多条链路,从而使得若并发失效的链路恰好可被同一个备份拓扑所保护时,不需要采用多个备份拓扑来计算业务的恢复路径,也就不会产生环路。因此建议规定数据包在从源到宿的传输过程中,至多仅能使用一个备份拓扑用于路径恢复^[22]。即如果某数据包已经被路由器按照备份拓扑上的恢复路径传输,若传输路径上再次出现失效链路,则路由器直接将此数据包丢弃。

“仅允许切换一次备份拓扑”这一方案使得路由器的转发逻辑较为简洁,但代价是在某些失效情形下必然造成丢失一定数量的数据包。为了可在更为复杂的网络链路失效场景下使用 MTR 获得更好的业务恢复性能,提议采用另一种方式切换备份拓扑^[26]。其主要思想是:按照备份拓扑编号升序的顺序在相应备份拓扑中计算恢复路径,若已查询至编号值最大的备份拓扑仍未找到可用的备份下一跳,则将数据包字段中当前拓扑编号加 1,然后随机选取一条未失效的邻接链路转发业务。由于这种切换拓扑的策略其转发规则具有随机性,因此本文将此方案进一步简化:当已尝试至编号值最大的备份拓扑仍未找到可用的下一跳时,则直接丢弃数据包不予随机转发。

下面对于方案“业务在传输的全过程中仅允许切换一次备份拓扑”与“按照备份拓扑编号升序切换”,分别介绍其路由转发流程,并通过仿真对这两种方案在双链路失效场景下的丢包率加以对比分析。

2.3.1 仅切换一次备份拓扑

图 2-9 说明了在第五章所介绍的仿真平台中，模拟双链路失效场景，当限制数据包在传输全过程中切换拓扑次数不超过一次时数据包的转发过程。

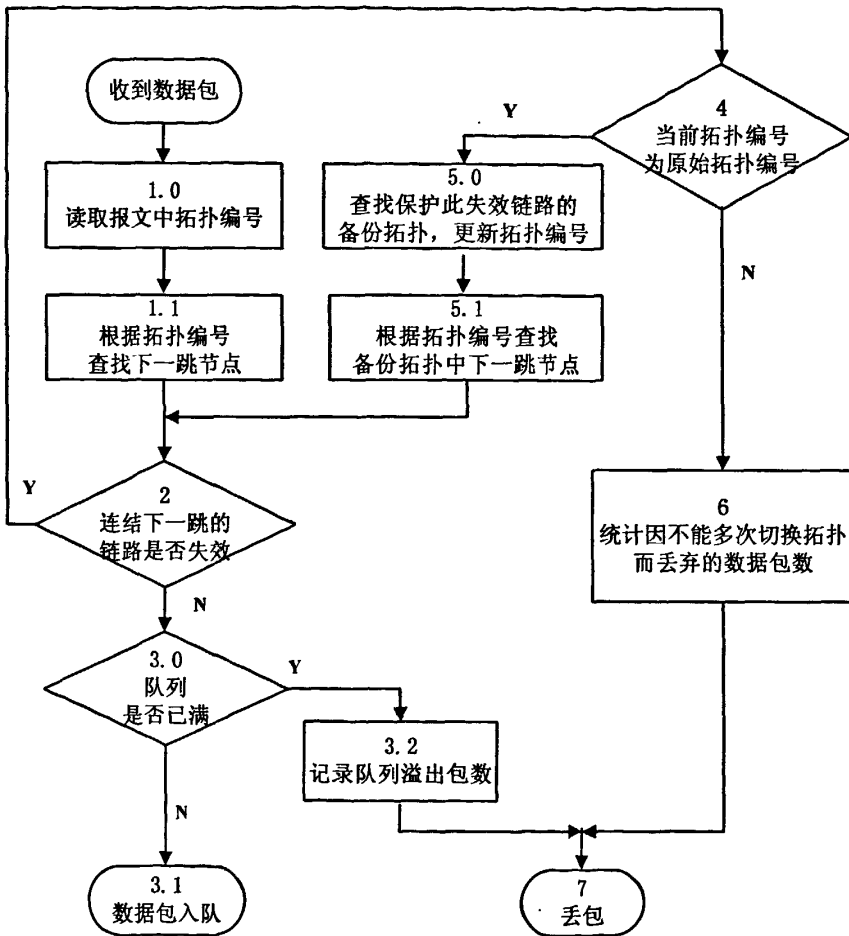


图 2-9 仅允许路径切换一次备份拓扑时数据包的转发流程

流程图从节点接收到数据包开始，当该节点不是此数据包的目的节点时，对数据包的具体处理流程如下：

步骤 1.0：当节点产生数据包，或从邻居节点收到待转发的数据包时，将数据包解封装提取其拓扑编号信息。步骤 1.1：根据当前拓扑编号，读取对应拓扑的转发表，查找下一跳转发节点；

步骤 2：读取本地节点邻接链路的状态信息，判断业务路径上连结下一跳节点

的链路是否失效。如果失效，则转步骤 4；否则转步骤 3.0，数据包准备进入节点转发模块中的发射机队列；

步骤 3.0：判断此时队列长度是否已经达到所设置的阈值。如果队列已满，转步骤 3.2，记录溢出数据包的个数，继而进入步骤 7 将数据包丢弃；若队列缓冲区未满，则数据包进入队列等待转发；

步骤 4：由于连结下一跳节点的链路失效，数据包由步骤 2 转入此步骤。判断此数据包当前传输路径所使用的拓扑的编号。若当前使用的是原始拓扑，转步骤 5.0；否则，即当前拓扑编号为备份拓扑编号，说明此数据包在转发过程中路径已经切换过拓扑，进入步骤 6，统计因不能再次使用备份拓扑来绕开失效链路被转发的数据包个数，转步骤 7，丢弃数据包；

步骤 5：进入此步骤的为原始拓扑上工作路径的下一跳节点不可达的数据包，查询保护此失效链路的备份拓扑编号，并将此编号值重新写入此数据包的拓扑编号字段，继续执行步骤 5.1，在备份拓扑的转发表中根据目的节点查找下一跳。转步骤 2，进一步判断备份下一跳节点是否可达。

2.3.2 按照拓扑编号升序切换

图 2-10 介绍了在多链路失效情形下，业务传输路径切换拓扑的另一种方案：即不限定业务在传输全过程中路由切换拓扑的次数，但限定按照备份拓扑编号从小到大的顺序为业务在备份拓扑中计算可用的下一跳转发节点，从而可以避免业务路径在几个拓扑间反复切换导致路由成环。

下面为在对此方案进行仿真时数据包的处理流程：

步骤 1.0~步骤 7：同 2.3.1 节。

步骤 4：进入此步骤的为传输路径上的下一跳转发节点不可达的数据包，若数据包使用原始拓扑传输，则转步骤 5.0 和步骤 5.1，查找备份拓扑中的下一跳节点；否则，转步骤 9.0。

步骤 8：判断本地节点与备份下一跳节点之间链路的状态。若备份下一跳可用，即链路未失效，则转步骤 3.0，数据包准备进入节点转发模块中的发射机队列；若备份的下一跳节点不可达，此时不同于 2.3.1 节中所述方案，不是直接丢弃数据包，而是转入步骤 9.0 开始继续在备份拓扑集合中寻找新的可行的备份下一跳。

步骤 9.0：判断，若当前拓扑编号已经等于备份拓扑编号的最大值，转步骤 6，统计因数据包字段中备份拓扑编号已达到最大值而被丢弃的数据包个数；若当前

备份拓扑编号未达到最大值，则转步骤 9.1，将当前拓扑编号加 1，然后执行步骤 9.2，继续在新的备份拓扑中查找下一跳转发节点。

此方案同 2.3.1 节的方案相比，增加了多链路并发失效后业务多次使用备份拓扑计算恢复路径的机会，但由于为了避免产生环路，限定了按拓扑编号升序切换至编号最大值后即丢包，因此仍会造成一定量的数据包丢失。在 2.3.3 节将对这两种方案的丢包率性能进行对比。

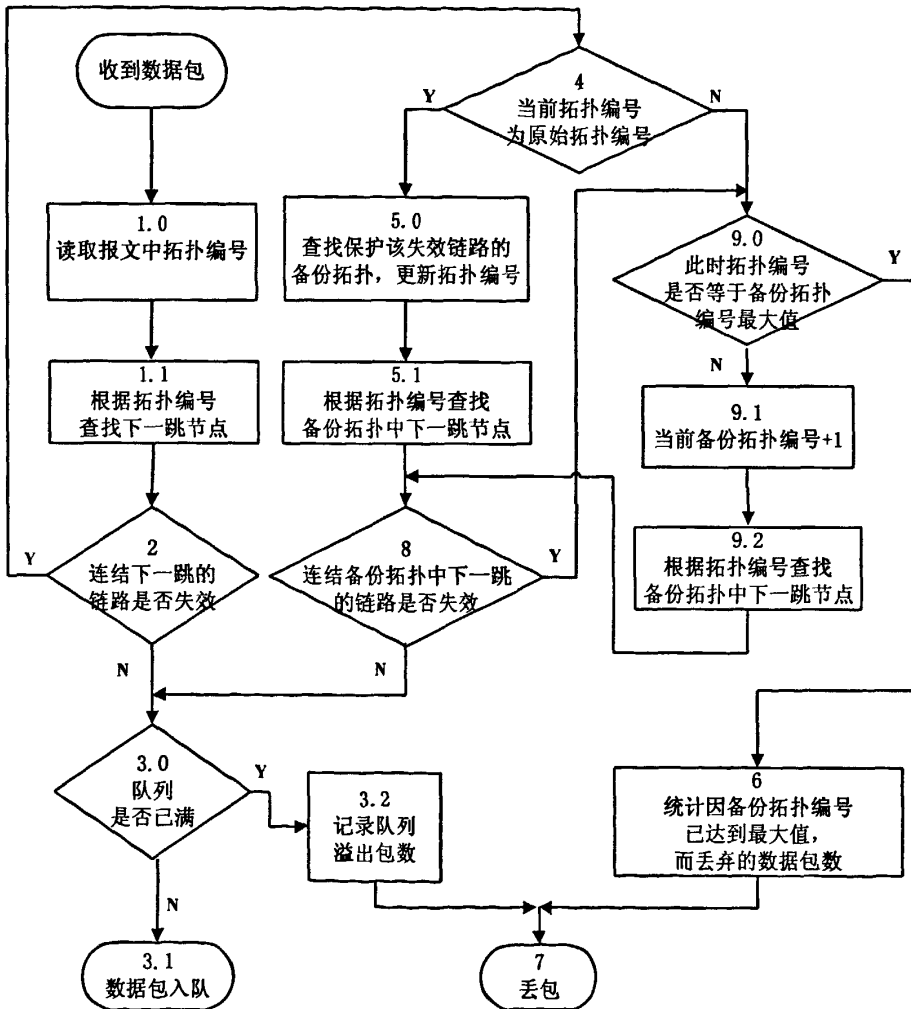


图 2-10 允许路径按备份拓扑编号升序切换拓扑时数据包的转发流程

2.3.3 路径切换拓扑方案的丢包率性能对比

下面以丢包率为性能指标,对于 2.3.1 和 2.3.2 节中介绍的关于传输路径切换拓扑的两种方案,通过使用第五章介绍的仿真平台比较其性能。

仿真所使用的原始拓扑图和链路权重设置同 2.2.3 节。全网任意节点对之间每秒钟需要传输的业务量设置为单位 1。网络的失效模型设置为双链路失效,即单位时间网络中有两条链路同时失效,在仿真时间域内,每种双链路失效情形仅发生一次,并且双链路失效情形之间相互独立。业务恢复方案采用 MTR 局部恢复。网络的备份拓扑采用 3.1.2 节介绍的 Rich 算法生成,共三组备份拓扑,每组中子拓扑的个数依次为 3、4、5 个,备份拓扑结构详见表 3-1。

全网节点总数 V 为 14 个,链路总数 L 为 21 条,则双链路失效情形总数 $C_L^2=210$ 种。每种失效情形下全网的业务量大小为 $14 \times 13=182$,仿真全过程节点发包总数为 $182 \times 210=38220$ 。对于上述两种路径切换拓扑的方案,分别通过仿真统计丢包数。当业务的传输路径上有多条链路失效时,产生丢包的原因分别为:业务已经使用过一次备份拓扑传输;业务当前所采用的备份拓扑编号已达到最大值。对于仿真时间域内所有发生的双链路失效情形,丢包率=丢包总数/发包总数。

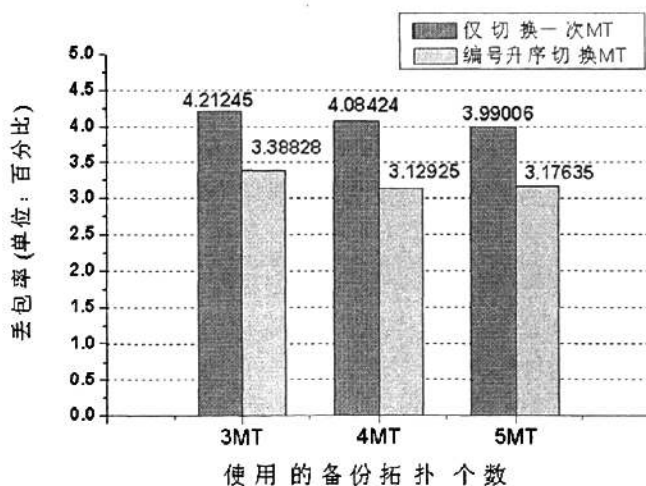


图 2-11 双链路失效情形下两种切换拓扑方案的丢包率对比

图 2-11 给出了双链路失效情形下,使用不同个数的备份拓扑集合,在业务路径分别采用单次切换和升序切换备份拓扑这两种方案时的丢包率情况。从图上可以看出,按照拓扑编号升序的路径切换方案同仅允许切换一次备份拓扑的方案相比,由于增加了为受失效影响的业务成功找到可用下一跳转发节点的几率,因此

在丢包率方面有一定改善。从第一组数据结果显示，限制切换单次备份拓扑时丢包率为 4.21%，当允许按编号升序多次切换拓扑时丢包率降至 3.39%。

此外，从图中可知当业务路径采用同一种切换拓扑方式时，基于原始拓扑预先构造的备份拓扑数目越多，丢包率越小。以采用单次切换拓扑的方案统计数据为例，分别采用 3、4、5 个备份拓扑时的丢包率依次为 4.21%、4.08%、3.99%。由于在本节仿真中所使用的备份拓扑，限定了原始拓扑的每条链路仅在一个备份拓扑中被保护。因此可计算出这三组备份拓扑集合中子拓扑的链路数分别约为 14、16、17。可知采用的备份拓扑个数越多，每个备份拓扑的连通度越大，即可用于参与最短路径计算的候选链路范围增大，得到跳数更小的备份路径的几率也随之增加，业务的恢复路径再次遭遇链路失效的可能性也随之减小，因而丢包率随备份拓扑个数增加而减少。

2.4 本章小结

本章主要介绍并研究了 MTR 技术应用于失效后实现业务快速恢复的相关原理及解决方案。

首先介绍了在 RFC 中为使用 MTR 技术而对 OSPF 路由协议的扩展，以及多拓扑的两种类型。

然后对于全局和局部恢复这两种方案，从恢复路径跳数和链路负载两个角度加以评估，仿真结果表明在多数情况下全局恢复所得到的路由方案比局部恢复具有更好的路径跳数和链路负载性能。但从网络管理的角度考虑，局部恢复方案仅需要在检测到邻接链路失效的本地节点处将下一跳不可达的业务使用备份路径传输，而全局恢复则需要在网络中增设一个中央控制器，当检测到失效后由中央控制器通知业务的源节点，将原本应经过失效链路传输的那部分业务切换至备份拓扑上传输，增加了管理的复杂度。

此外，针对双链路失效情形，分别采用两种业务路径切换拓扑的方案，以不同个数的备份拓扑集合为仿真拓扑，对丢包率进行了统计。仿真结果显示，通过允许路径多次切换拓扑以及增加备份拓扑个数，可以降低丢包率。

第三章 多拓扑生成算法研究

多拓扑（又称为备份拓扑）是应用MTR技术的载体，与失效后业务路径恢复的性能直接相关。因此研究如何构造备份拓扑以更好地满足QoS需求，成为本文的主要研究方向之一。

3.1 研究背景

本章研究用于生成可在链路失效后实现业务路径恢复的备份拓扑的算法。

生成用于保护链路的备份拓扑的主要思想是：在每个备份拓扑中，通过将某些链路的权重值设置为无穷大，使业务的恢复路径将不会经过这些链路，从而达到应对链路失效的目的。由于备份拓扑只包含原始拓扑的部分链路，因此备份路径的跳数可能会略大于采用全网 OSPF 路由重收敛所得到的新路径的跳数，路径跳数增益其具体值取决于网络链路权重、原始拓扑结构以及所采用的备份拓扑总数。可以通过增加使用的备份拓扑数目来减小恢复路径的跳数，然而将更多的备份拓扑用于失效恢复意味着路由器需要维护更多的链路状态信息，同时也增加了路由器计算和装载路由转发表的工作负荷。因此在设计多拓扑时需要在降低备份拓扑总数以及减小路径跳数这两方面之间进行折衷，并可以结合其它 QoS 需求，生成根据实际需求所定制的多拓扑。

人们提出了三种构造备份拓扑的算法^[22]，分别为 Minimum 算法、Rich 算法和 Sparse 算法。三种算法的侧重点不同：Minimum 算法用于生成个数最少的备份拓扑，从而可以减少路由器存储的额外信息量；Rich 算法是通过增加备份拓扑的个数来增加备份拓扑的连通度；Sparse 算法是指在备份拓扑中尽可能使更多的链路被保护，通过生成更为稀疏的备份拓扑以增大任意两条链路在同一个备份拓扑中被保护的概率，从而可以更好地应对多条链路并发失效的情况。

Minimum 算法和 Rich 算法限定原始拓扑中的每条链路仅被一个备份拓扑保护。而 Sparse 算法的主要思想是在备份拓扑中尽可能地“去掉”更多的链路，因此原始拓扑中的每条链路可能被多个备份拓扑所保护。实验验证了采用 Sparse 算法所生成的备份拓扑应用失效时，业务的传输路径跳数大于另两种算法^[22]。备份拓扑结构越稀疏，可能参与承载待恢复业务的链路数目也更为有限，产生链路拥

塞的几率也随之增加。此外,当有多个备份拓扑可保护同一失效链路时,关于业务在多个备份拓扑上如何分流转发目前也没有很好的解决方案。因此本文研究的多拓扑生成算法以“规定原始拓扑中的每条链路仅被一个备份拓扑所保护”为前提,对于 Sparse 算法不予以进一步讨论。

多拓扑的设计问题也可归结为整数线性规划(ILP)问题。线性规划是指把待求解的实际问题转化为数学关系式,用以描述并求解决策方案。建立线性规划模型的三个要素是:①寻求要确定的决策变量,即问题的解;②制定需要优化(求极大或极小)的目标函数;③指出解必须满足的约束条件。线性这一特性要求模型中每个决策变量对目标函数值的贡献与决策变量的值成比例。满足约束和目标的决策变量的值称为可行解,求解线性规划问题的目标就是找到最好的可行解,即最优解。整数线性规划是指要求某些或全部变量只取整数值的线性规划。

Scheffel 提出了用于求解备份拓扑的 ILP 模型,可保证生成的备份拓扑所提供的恢复路径总跳数最小^[27]。为便于描述,在本文中暂将此 MT 生成算法简称为 MRPH (Minimum Recovery Path Hop, 最小化恢复路径跳数)算法。

上述算法都是在不考虑业务量需求大小的前提下,在保证各节点对之间均存在有效路径的基础上,调整链路在一组备份拓扑中的分布以达到一定的目标需求。因此在本文中基于流量工程的思想,在假定网络业务量需求已知的前提下,使用 ILP 数学建模,提出了一种新的生成备份拓扑的解决方案,下文中简称为 MLDF (Minimum MT Load against Double link Failure, 最小化双链路失效情形下多拓扑的负载)算法。

下面在 3.1.1~3.1.3 节中依次简要介绍 Minimum 算法、Rich 算法和 MRPH 算法,在 3.2 节中提出 MLDF 算法的 ILP 模型,并在 3.3 节中对比分析了 4 种多拓扑生成算法的性能。

3.1.1 Minimum 算法

Minimum 算法的主要思想是尽可能使用最少的备份拓扑个数,来保护网络中的每条链路。在每个备份拓扑中,网络的任意节点对之间均有路径可达,即备份拓扑图为连通图。

拓扑图的割边是指这样一种链路,即如果在该拓扑图中去掉此链路,将破坏图的连通性,即无法保证任意节点对之间有路径相连。因此判决备份拓扑能否保护某条链路,也就是判断该链路是否是这个备份拓扑的割边。

Minimum 算法的具体步骤如下所示:

已知网络原始拓扑 G 的链路集合 L , 建立集合 S , S 用于存放原始拓扑的割边和已生成的备份拓扑所保护的链路。

步骤 1: 初始化备份拓扑集合 MT 为空集。计算原始拓扑 G 的割边集合并以此初始化集合 S 。变量 i 初始化为 1。

步骤 2: 若 $S=L$, 即当原始拓扑中所有链路都被备份拓扑保护时, 转步骤 5。否则, 将当前待生成的备份拓扑 MT_i 按照 G 进行初始化, 集合 L' 用于存放链路集合 L 中除去集合 S 中链路之外的其它链路。

步骤 3: 对于 L' 中的每条链路 l , 如果链路 l 不是当前备份拓扑 MT_i 的割边, 则将链路 l 在备份拓扑 MT_i 中去掉, 并将链路 l 添入集合 S 。

步骤 4: 将已生成的备份拓扑 MT_i 存入备份拓扑集合 MT 。 $i+1$, 转步骤 2, 生成下一个备份拓扑, 继续在新的备份拓扑中尝试保护集合 L' 中剩余链路。

步骤 5: 此时已生成可保护原始拓扑 G 中所有链路的一组备份拓扑 MT , 下面进行调整以使各备份拓扑所保护的链路数基本均等。每次从链路数最多的备份拓扑 (记作 MT_{max}) 中选取不破坏其连通性的链路 (即非割边链路), 将其移到链路数最少的备份拓扑 (记作 MT_{min}) 中, 重复此操作直至 MT 中所有拓扑的链路数基本相等。平衡备份拓扑中链路的数目是为了避免“在某些备份拓扑中仅有很少的几条链路被保护, 而在其它备份拓扑中有大量链路被保护”。

3.1.2 Rich 算法

Rich 算法其最终目标也是生成可保护原始拓扑所有链路的一组备份拓扑, 但与 Minimum 算法不同的是, Rich 算法旨在通过增加生成备份拓扑的个数, 使得在备份拓扑中可以保留更多链路, 从而使在网络发生某种失效情形后采用备份拓扑计算业务的恢复路径时, 有更多链路可供选择。

Rich 算法的具体步骤如下所示:

已知网络原始拓扑 G , 计算 G 的非割边链路集合 L 。

步骤 1: 调用 Minimum 算法, 求得可保护所有链路的备份拓扑的最小个数 N 。

步骤 2: 读取指定生成的备份拓扑个数 K , $K \in [N, L]$ 。用原始拓扑 G 初始化备份拓扑集合 $\{MT_1, \dots, MT_K\}$ 中的每个拓扑。采用集合 L 初始化待保护的链路集合 L' 。变量 i 初始化为 1。

步骤 3: 若 L' 为空集, 程序结束。否则, 从集合 L' 中取出一条链路 l , 令 $j=i$ 。

步骤 4: 判断 l 是否为当前备份拓扑 MT_i 的割边。若不是, 则从 MT_i 中移除链路 l , 即将 l 在备份拓扑 MT_i 中的权重值设为无穷大, 然后更新 $i = i\%K + 1$, 转步骤 3。如果 l 是备份拓扑 MT_i 的割边, 则将 i 更新为 $i\%K + 1$, 转步骤 5。

步骤 5: 若 $i \neq j$, 转步骤 4。否则, 提示无法生成指定的 K 个备份拓扑, 算法终止。

假定网络原始拓扑的链路总数为 L , 指定需生成备份拓扑的数量 K , 则采用此算法可得到 K 个备份拓扑, 每个备份拓扑大约包含 $L - (L/K)$ 条链路。MTR 技术使用的备份拓扑数目越多, 路由器需要储存的相关数据就越多, 路由表和转发表的计算量也随之增加。

3.1.3 MRPH 算法

Scheffel 提出了两种 ILP 模型, 分别生成可用于业务路径局部恢复和全局恢复的备份拓扑^[27]。优化目标是对于所有可能发生的单链路失效情形, 最小化受失效影响的所有业务的恢复路径跳数之和。

此算法的前提是假设网络任意节点间的业务量需求值为单位 1, 且链路权重均为 1, 使用 SPF 算法求解路径。预先计算业务在原始拓扑中的工作路径, 并以此作为输入参数。

使用此 ILP 模型所生成的备份拓扑, 不能保证原始拓扑的每条链路只由单个备份拓扑对应保护。由于在前两小节中介绍的 Minimum 算法和 Rich 算法, 以及将在 3.2 节中提出的 MLDF 算法, 这三种算法所生成的备份拓扑均具有同样的特点: 对于原始物理拓扑中的每条链路, 有且仅有一个备份拓扑保护此链路。为了基于相同的备份拓扑基本特性来对比分析不同算法所生成的备份拓扑的性能, 对于文献^[27]中所提出的 ILP 模型, 添加约束条件式(3-1), 使得所生成的备份拓扑中, 原始拓扑的每条链路仅被一个备份拓扑所保护。并在本文中将修改后的 ILP 模型简称为 MRPH 算法。式(3-1)中符号含义详见文献^[27]。

$$\sum_{i \in I} \text{TopProtFail}_{i,e} \leq 1, \forall e \in E \quad (3-1)$$

3.2 MLDF 算法

在 3.1 节中所介绍的多拓扑生成算法, 都未考虑网络的业务量需求。使用备份拓扑进行业务恢复的性能受以下因素所影响: ①备份拓扑的构造 (拓扑个数、连

通度等)；②网络的业务量需求；③恢复路径在备份拓扑上的分布（可通过设置链路权重来优化调整）；④路由转发方案。因此如果业务量需求可测，则可以针对更为具体的优化目标制定备份拓扑的生成方案。

当网络链路 l 发生失效后，采用 MTR 技术时，原来在原始拓扑上采用该链路传输的业务量（记作 $Load_l^0$ ）将会被切换到可保护此链路的备份拓扑上传输。因而在网络发生失效后，某个备份拓扑是否承载待恢复的业务取决于该备份拓扑是否保护此条失效链路。如果某备份拓扑保护此条失效链路 l ，则该备份拓扑需承载的待恢复的业务量为 $Load_l^0$ 。

当网络发生双链路失效时，两条失效链路在网络无失效时承载的业务量，都将被加载到保护失效链路的对应备份拓扑上传输。如果保护这两条失效链路的备份拓扑不同，则需要重路由的业务将在不同的备份拓扑上传输，也即在不同的链路集合中计算恢复路径。但如果保护这两条失效链路的备份拓扑恰好是同一个拓扑，那么待恢复的业务将全部加载到此备份拓扑上，两部分待恢复的业务都将在同一个备份拓扑的链路集合中计算恢复路径，增加了链路发生拥塞的几率。

因此为了在双链路失效情形下降低网络链路发生拥塞的概率，提出以下解决方案：

假定已知网络原始拓扑、链路权重和业务量需求，使用 SPF 算法可计算出在无失效情形下每条链路负载。尝试生成这样一组备份拓扑：对于每个备份拓扑所独立保护的每种双链路失效情形，降低在此失效情形下拓扑所需承载的业务量。

此问题可详细描述为：

设原始拓扑编号为 0，指定备份拓扑的总数 M ，则备份拓扑编号为 1~ M 。原始拓扑的链路集合 L ，链路视为单向且权重已知。已知业务量需求，由 SPF 算法可求得无失效时原始拓扑每条链路 l 的负载 $Load_l^0$ 。

每个备份拓扑中保护的链路集合为 A_m ， $L = \sum_{m=1}^M A_m$ 。对于某个备份拓扑，若该拓扑所保护的链路集合 A_m 中的链路 l_i 和 l_j 同时失效，则此备份拓扑所需承载的业务量大小为 $Load_{l_i} + Load_{l_j}$ 。

求解满足下列条件的一组备份拓扑，即求解 $\{A_m\}$ ，使得

$$\text{Minimize } \text{Max}(Load_{l_i} + Load_{l_j}) \quad , \forall l_i \in A_m, l_j \in A_m, l_i \neq l_j, m \in M \quad (3-2)$$

此问题可以通过建立整数线性规划模型来加以描述。在下文中将此方法简称为 MLDF 算法，即可最小化双链路失效情形下备份拓扑所需承载业务的多拓扑生

成算法。

设原始拓扑 G 的编号 0, 备份拓扑集合 MT 的编号为 $1 \sim M$ 。原始拓扑的节点集合 V , 链路集合 L , 链路视为单向。假设当发生链路失效时, 同一相邻节点对之间对称方向上的两条链路均发生失效。链路 l 的起点 $F(l)$, 终点 $H(l)$ 。使用矩阵 D 描述具有业务需求的源节点与宿节点。ILP 模型定义如下:

1. 常量

- ① M ——指定生成的备份拓扑个数。
- ② $SafeLinkNum$ ——指定每个备份拓扑中至少应保护的链路数目。
- ③ $Load_l^0$ ——网络无失效时链路 l 所承载的业务总量。
- ④ $Gpath_{st}$ ——网络无失效时, 源宿节点对 st 之间的业务在原始拓扑上的工作路径所包含的链路集合, $st \in D$ 。

2. 变量

- ① $APP_{st}^{f,m}(l)$ ——布尔值。为 1: 当链路 f 失效后, 在备份拓扑 m 中链路 l 在业务 st 的恢复路径上。为 0: 不在恢复路径上。
- ② SL_l^m ——布尔值。为 1: 链路 l 可被备份拓扑 m 保护, 即备份拓扑 m 中不含链路 l 。为 0: 链路 l 不被备份拓扑 m 保护, 即链路 l 存在于备份拓扑 m 中。
- ③ $Demand(m)$ ——对于备份拓扑 m 可独立保护的所有双链路失效情形, 备份拓扑 m 所需承载的业务量的最大值, $Demand(m) \geq 0$ 。
- ④ T ——优化目标。表示在所有备份拓扑可独立保护的双链路失效情形下, 备份拓扑所需承载的业务量的最大值, $T \geq 0$ 。

3. 优化目标

$$\text{Minimize } T$$

4. 约束条件

(a) 采用 MTR 进行全局恢复时的流量守恒方程。

$$\sum_{m \in M, l \in L, F(l)=s} APP_{st}^{f,m}(l) - \sum_{m \in M, l \in L, H(l)=s} APP_{st}^{f,m}(l) = 1, \quad \forall st \in D, f \in GPath_{st} \quad (3-3)$$

$$\sum_{l \in L, F(l)=n} APP_{st}^{f,m}(l) = \sum_{l \in L, H(l)=n} APP_{st}^{f,m}(l), \quad \forall st \in D, f \in GPath_{st}, m \in M, n \in N, n \neq s \cap n \neq t \quad (3-4)$$

$$\sum_{m \in M, l \in L, H(l)=t} APP_{st}^{f,m}(l) - \sum_{m \in M, l \in L, F(l)=t} APP_{st}^{f,m}(l) = 1, \quad \forall st \in D, f \in GPath_{st} \quad (3-5)$$

(a') 采用 MTR 进行局部恢复时的流量守恒方程。

$$\sum_{m \in M, l \in L, F(l)=F(f)} APP_{st}^{f,m}(l) - \sum_{m \in M, l \in L, H(l)=F(f)} APP_{st}^{f,m}(l) = 1, \quad \forall st \in D, f \in GPath_{st} \quad (3-6)$$

$$\sum_{l \in L, F(l)=n} APP_{st}^{f,m}(l) = \sum_{l \in L, H(l)=n} APP_{st}^{f,m}(l), \quad \forall st \in D, f \in GPath_{st}, m \in M, n \in N, n \neq F(f) \cap n \neq t \quad (3-7)$$

$$\sum_{m \in M, l \in L, H(l)=t} APP_{st}^{f,m}(l) - \sum_{m \in M, l \in L, F(l)=t} APP_{st}^{f,m}(l) = 1, \quad \forall st \in D, f \in GPath_{st} \quad (3-8)$$

(b) 确保如果某备份拓扑可保护某种失效情形，那么此失效情形对应的两个方向上的失效链路都可被此备份拓扑保护。

$$SL_{f^+}^m + SL_{f^-}^m \geq 2 * APP_{st}^{f,m}(l), \quad \forall st \in D, f \in GPath_{st}, m \in M, l \in L \quad (3-9)$$

(c) 限制若某链路 l 被备份拓扑 m 保护，即 $SL_l^m=1$ ，则当链路 l 发生失效后，链路 l 不会位于业务的恢复路径上。

$$APP_{st}^{f,m}(l) \leq 1 - SL_l^m, \quad \forall st \in D, f \in GPath_{st}, m \in M, l \in L \quad (3-10)$$

(d) 限制原始拓扑的每条链路 l 至多只能被一个备份拓扑所保护。

$$\sum_{m \in M} SL_l^m \leq 1, \quad \forall l \in L \quad (3-11)$$

(e) 保证每个备份拓扑所保护的链路数目尽量相等。

$$\sum_{l \in L} SL_l^m \geq SafeLinkNum, \quad \forall m \in M \quad (3-12)$$

(f) 定义对于每个备份拓扑，在该拓扑可独立保护的所有双链路失效情形下此拓扑应承载的业务量的最大值。

$$Demand(m) \geq Load_{l_i}^0 * SL_{l_i}^m + Load_{l_j}^0 * SL_{l_j}^m, \quad \forall m \in M, l_i \in L, l_j \in L, l_i \neq l_j \quad (3-13)$$

(g) 定义优化目标。

$$T \geq Demand(m), \quad \forall m \in M \quad (3-14)$$

由上述 ILP 模型中约束条件(a)和(b)~(g)可联合求解应用于全局恢复的备份拓扑，使用约束条件(a')和(b)~(g)可联合求解应用于局部恢复的备份拓扑。

3.3 多拓扑生成算法的性能分析

在本节中，对于前两节所介绍和提出的以下四种多拓扑生成算法的性能加以对比：①Minimum 算法；②Rich 算法；③MRPH 算法；④MLDF 算法。

首先采用图 5-1 中的 US 网络作为原始拓扑, 此网络包含 14 个节点和 21 条双向链路。以表格形式详细给出使用不同算法分别生成的备份拓扑图。

表 3-1 中第一行为算法名称, 第二行为使用该算法生成的一组备份拓扑中备份拓扑的个数 K 。表格的第一列为链路编号, 其余每列分别表示采用对应算法生成的一组备份拓扑。表格中的数值为备份拓扑编号, 即表示该链路在第几个备份拓扑中被保护。例如: 在下表中第 3 行第 2 列的单元格数值为 3, 表示在采用 Minimum 算法生成的一组共 3 个备份拓扑中, 可保护编号为 1 的链路的备份拓扑编号为 3。

表 3-1 四种拓扑生成算法求得的备份拓扑

算 法	Minimum	Rich			MRPH										MLDF			
					全局 恢复					局部 恢复					全局 恢复		局部 恢复	
K Link ID	3	4	5	21	3	4	5	6	7	3	4	5	6	7	3	4	3	4
1	3	1	1	1	3	1	4	2	5	1	2	2	3	3	2	3	1	2
2	1	2	2	2	1	4	5	4	7	3	1	3	6	2	3	4	3	3
3	3	3	3	3	1	4	5	4	1	3	1	3	3	5	1	2	1	4
4	1	4	4	4	2	1	4	2	6	2	3	2	6	3	1	3	1	2
5	2	1	5	5	1	4	5	4	7	3	1	3	5	5	2	2	3	1
6	1	2	1	6	3	3	2	3	6	2	4	5	4	6	1	2	1	1
7	1	3	2	7	3	3	1	1	5	2	3	4	2	7	1	4	2	4
8	2	4	3	8	2	1	4	2	5	1	2	2	6	3	2	2	3	3
9	1	1	4	9	3	3	1	1	4	2	3	4	2	1	2	3	2	2
10	2	2	5	10	3	3	1	1	4	1	4	4	2	7	3	3	2	2
11	1	3	1	11	1	1	2	5	3	3	2	5	5	4	3	4	2	4
12	2	4	2	12	3	2	3	3	2	1	4	1	1	1	1	3	2	2
13	2	1	3	13	1	4	3	5	2	1	2	1	1	4	3	1	3	1
14	3	2	4	14	1	4	5	4	1	3	1	3	3	2	3	1	2	4
15	1	3	5	15	3	2	3	6	3	1	4	1	1	7	2	1	3	3
16	2	4	1	16	1	4	5	3	7	3	2	3	5	1	1	4	1	4
17	2	1	2	17	2	4	5	4	7	2	1	3	4	5	2	3	3	2
18	3	2	3	18	1	2	3	6	2	1	2	1	1	2	1	1	1	3
19	3	3	4	19	2	3	1	1	4	2	3	4	2	6	2	1	1	3
20	3	4	5	20	2	1	4	5	1	2	1	2	3	4	3	2	2	1
21	3	1	1	21	2	2	2	6	3	3	4	5	4	6	3	4	3	1

3.3.1 Minimum 算法与 Rich 算法的性能对比

在本小节中，以路径跳数和链路负载作为性能指标，对于 3.1.1 和 3.1.2 节中介绍的 Minimum 算法和 Rich 算法所生成的备份拓扑的性能加以对比。

仿真采用的原始拓扑为图 5-1 中的 US 网络，该拓扑包含 14 个节点和 21 条双向链路。备份拓扑采用：①Minimum 算法生成的一组共 3 个备份拓扑；②Rich 算法生成的三组备份拓扑，每组中备份拓扑个数分别为 4、5 和 21。备份拓扑的结构详见表 3-1。链路权重的设置同 2.2.3 节，使用 SPF 算法计算路径。

下面在原始拓扑和上述四组拓扑总数不同的备份拓扑中，计算 US 网络任意两节点间的路径，并统计路径的跳数。图 3-1 中横坐标 x 值为路径的跳数，纵坐标表示路径跳数为 x 的路径数目占所有路径总数的百分比。

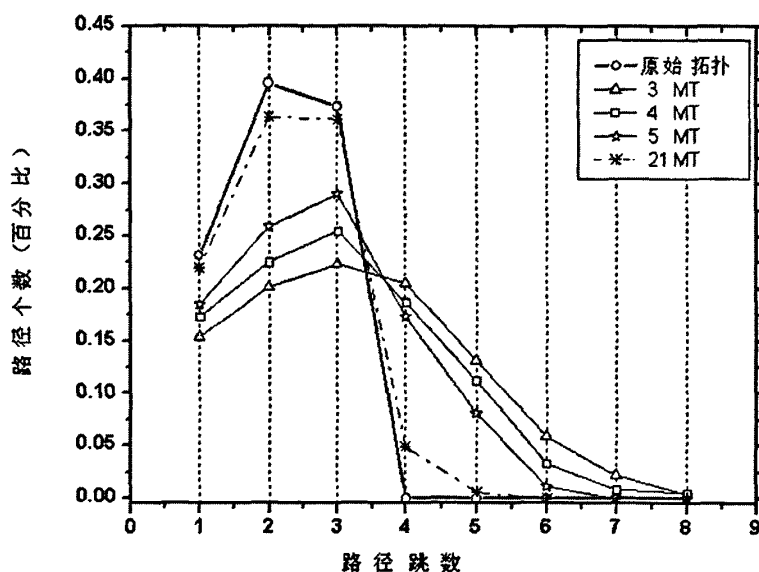


图 3-1 在拓扑总数不同的备份拓扑集合中路径跳数的分布率

由图 3-1 可知，业务路径的跳数随着备份拓扑个数的增加而减少。备份拓扑总数越大，平均每个备份拓扑所需保护的链路就越少，导致备份拓扑的连通度更高，即有更多链路有机会参与最短路径的计算，增加了获得更短路径的几率。对于原始拓扑和 3、4、5、21 个备份拓扑，网络任意节点对间路径跳数的均值依次为 2.14 和 3.24、2.99、2.74、2.26。

使用 21 个备份拓扑时，每个备份拓扑只保护原始拓扑中的一条链路，等价于 OSPF 路由重收敛更新后的拓扑，此时业务路径的跳数值最接近于无链路失效时原

始拓扑中的路径跳数。然而 MTR 技术使用的备份拓扑数越多，路由器所需维护的拓扑数据信息也就越多。因此可以根据实际需求，选择生成适当个数的备份拓扑。

下面分析备份拓扑总数对链路负载的影响。

业务量需求矩阵（单位 Gbit/s）按照表 5-1 中参数设置。业务路径采用局部恢复方案。在每种单链路失效情形下，分别使用 Minimum 算法和 Rich 算法所生成的 3、4、5、21 个备份拓扑图计算业务的恢复路径，统计全网链路负载的最大值和平均值。在图 3-2 中，横坐标为失效链路的编号，纵坐标代表网络中所有链路的负载的最大值，单位为 Gbit/s。

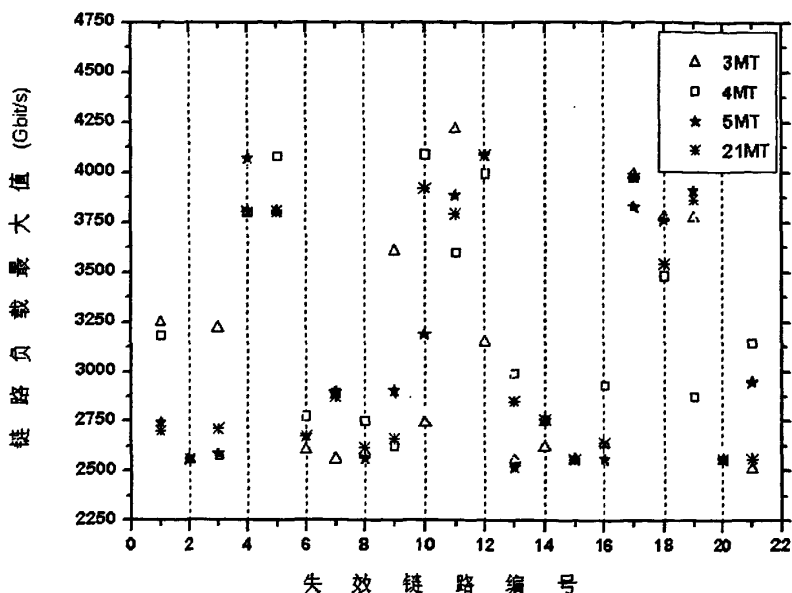


图 3-2 局部恢复模式下采用不同个数的备份拓扑集合时链路最大负载的对比

由图中数据可知，使用 3、4、5、21 个备份拓扑用于业务路径的快速恢复时，全网链路的最大负载在所有可能发生的单链路失效情形下的峰值依次为 4218.64、4093.29、4087.3、4087.3，平均值依次为 3100.11、3154.26、3112.05、3119.51。此外，在仿真中统计出全网链路的平均负载在所有可能发生的失效情形下的平均值依序为 1435.6、1421.75、1427.11、1398。

由于链路负载受备份拓扑总数、备份拓扑结构、链路权重和业务需求量等多种因素联合影响，使得链路负载与备份拓扑个数之间不存在绝对的线性比例关系。例如分别采用 3 个和 4 个备份拓扑时，就所有单失效情形下链路最大负载的峰值而言，采用 4 个备份拓扑时此值为 4093.29，同使用 3 个备份拓扑时的负载峰值

4218.64 相比,性能更优。然而对于所有单失效情形下链路最大负载的均值,采用 4 个备份拓扑时最大负载均值为 3154.26,性能劣于采用 3 个备份拓扑时的负载均值 3100.11。

当使用的备份拓扑总数在一定范围内波动时(例如本例中的 3、4、5 个),不能断言备份拓扑数量对于网络链路负载的影响规律。但从以上数据可以看出,当备份拓扑总数显著增加时,例如本例中的 21 个同 3 个备份拓扑相比,每个备份拓扑更为稠密,扩大了计算路径时的候选链路集合,因而增加了平衡链路负载的几率。

因此当构造备份拓扑时,可将备份拓扑总数纳入需要考虑的因素,但是在根据实际需求选取拓扑总数的取值范围之后,还应结合其它因素联合设计性能更好的备份拓扑。

3.3.2 MRPH 算法的性能分析

Minimum 算法和 Rich 算法所生成的备份拓扑,可以保证网络任意节点对间均有路径可达,即所生成的拓扑图连通,但是这两种算法在设计拓扑时仅将拓扑总数纳入考虑因素。为了针对不同的业务恢复方案及 QoS 需求,制定相应的拓扑生成方案,改进了 MRPH 算法,并提出 MLDF 算法。

MRPH 算法和 MLDF 算法在生成备份拓扑时,考虑了全局恢复和局部恢复这两种业务恢复方案的特性——对于全局恢复,所生成的备份拓扑可保证在业务的源宿节点间均有路径可达;对于局部恢复,所生成的备份拓扑可保证从失效链路的上游节点到业务的目的节点之间均有路径可达。

此外,这两种算法分别具有不同的优化目标:MRPH 算法可以生成恢复路径跳数最小的备份拓扑;MLDF 算法可保证在双链路失效情形下,将备份拓扑所需承载待恢复的业务量降至最小。

在本小节中测试改进后的 MRPH 算法所生成的备份拓扑性能,并将在下一节中对比 MRPH 算法和 MLDF 算法的性能。

以 US 网络作为原始拓扑,使用 MRPH 算法分别对于全局恢复、局部恢复两种方案生成共 10 组备份拓扑,拓扑总数指定为 3~7 个。此算法的优化目标是:在采用指定的业务恢复方案时,最小化所有业务在备份拓扑上的恢复路径总跳数。由于该算法旨在对路径跳数进行优化,对应的 ILP 模型不涉及业务量需求的大小,因此假定全网所有节点对之间均有业务需求,业务的大小为单位 1。模拟单链路失

效场景，链路权重的设置同 2.2.3 节，采用 SPF 算法计算路径。

使用 MRPH 算法的 ILP 模型求解用于全局恢复的备份拓扑时，计算出的最优目标值为受失效影响的业务从源到宿的路径跳数总和。而在求解用于局部恢复的备份拓扑时，所得的最优目标值为受失效影响的业务从失效链路上游端点到业务目的节点的路径跳数总和。在本节中，当求解用于局部恢复的备份拓扑时，将所得的最优目标值与“受失效影响的业务在原始拓扑工作路径上从业务源点到失效链路上游节点的路径的跳数”相叠加之后，再与全局恢复 ILP 模型的最优目标值相对比。即对比的是——使用 MRPH 算法生成备份拓扑，业务恢复方案分别采用全局恢复和局部恢复时，所有受单链路失效所影响的业务从源点到宿点的传输路径的跳数之和。

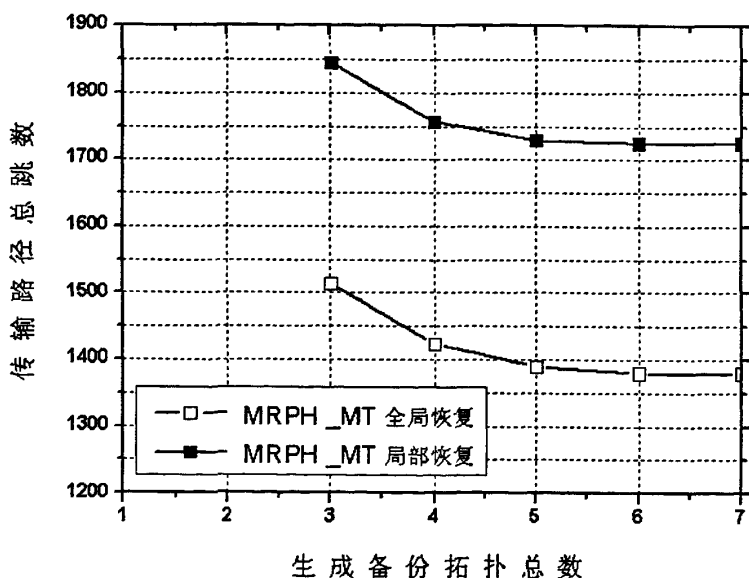


图 3-3 使用 MRPH 算法生成的备份拓扑后业务传输路径的总跳数

图 3-3 中横坐标为使用 MRPH 算法生成的每组备份拓扑集合中所包含的拓扑总数，纵坐标表示对于所有可能发生的单链路失效情形，受失效所影响的所有业务从源到宿的传输路径跳数的总和。采用全局恢复方案所求得的路径跳数值用空心点描绘，采用局部恢复方案所求得的路径跳数值用实心点标注。

根据图 3-3 中所示数据以及 2.2.3.1 节的分析，可知与局部恢复方案相比，采用全局恢复方案时业务的传输路径跳数更小。此外从所得数据可以看出，使用的备份拓扑个数越多，路径跳数总和越小。当采用的备份拓扑个数等于 6 个时，恢复路径总跳数已达到最小值，此时再增加备份拓扑个数对于路径跳数性能已无改

善。采用全局恢复方案，使用 6 个备份拓扑可达到的传输路径总跳数的最小值为 1380；采用局部恢复方案，使用 6 个备份拓扑后传输路径跳数最小值为 1725。

3.3.3 MRPH 算法与 MLDF 算法的性能对比

MRPH 算法和 MLDF 算法都可以分别针对全局恢复和局部恢复这两种业务恢复方案生成备份拓扑。MLDF 算法旨在将每个备份拓扑在可独立保护的那些双失效情形中备份拓扑所需承载的最大业务量降至最小。下面对于 MRPH 算法和 MLDF 算法所生成的备份拓扑，从拓扑所需承载的业务量以及传输路径的总跳数这两方面加以对比。

原始拓扑、业务量需求矩阵（单位 Gbit/s）和链路权重均按照 2.2.3 节中的仿真参数进行设置，使用 SPF 算法预先计算业务在原始拓扑中的工作路径。采用 MRPH 算法和 MLDF 算法分别对于全局恢复、局部恢复两种业务恢复方案各生成 2 组备份拓扑，所包含的备份拓扑个数为 3 和 4。

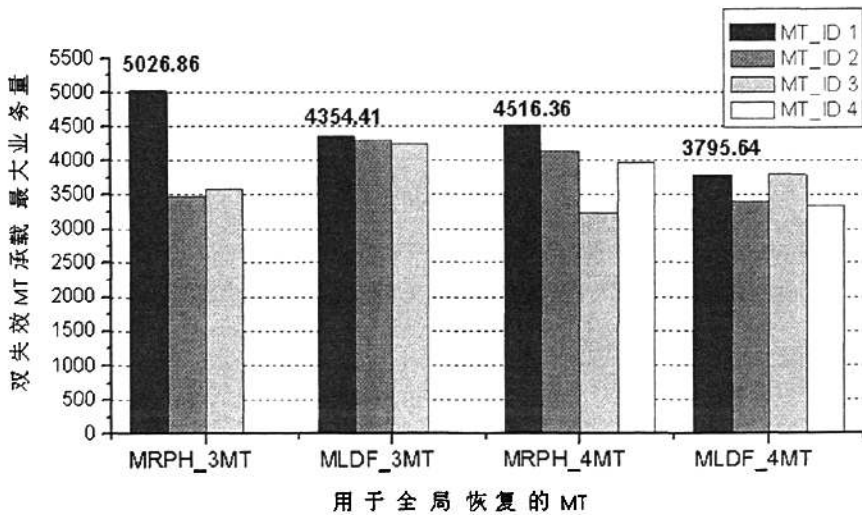


图 3-4 全局恢复模式下 MRPH 与 MLDF 算法生成 MT 的负载对比

在图 3-4 中，对比使用两种算法生成的用于全局恢复方案的备份拓扑，在双链路失效情形下，每个备份拓扑所需承载的最大业务量。横坐标轴为使用的算法名称以及生成的备份拓扑个数。一组备份拓扑中每个拓扑承载的业务量使用不同颜色标注。纵轴为对于该组中每个备份拓扑，在此拓扑可独立保护的双链路失效情形下，此拓扑所需承载的待恢复业务的最大值。

以生成 3 个备份拓扑为例，如图 3-4 所示，在发生双链路失效时使用 MRPH

算法所生成的备份拓扑组中,编号为 1 的备份拓扑需要承载的业务为 5026.86,而使用 MLDF 算法所生成的备份拓扑中,备份拓扑需要承载的最大业务量为 4354.41,并且 3 个备份拓扑所需承载的业务量基本均等。

图 3-5 为对于局部恢复方案,使用 MRPH 算法和 MLDF 算法所生成的备份拓扑,在双链路失效情形下所需承载的最大业务量的对比。同样,MLDF 算法在降低双失效情形下拓扑承载业务量这一方面的性能优于 MRPH 算法——同 MRPH 算法相比,分别采用 3 个和 4 个备份拓扑,MLDF 算法可使拓扑所需承载的最大业务量降低 13.38%和 23.43%。

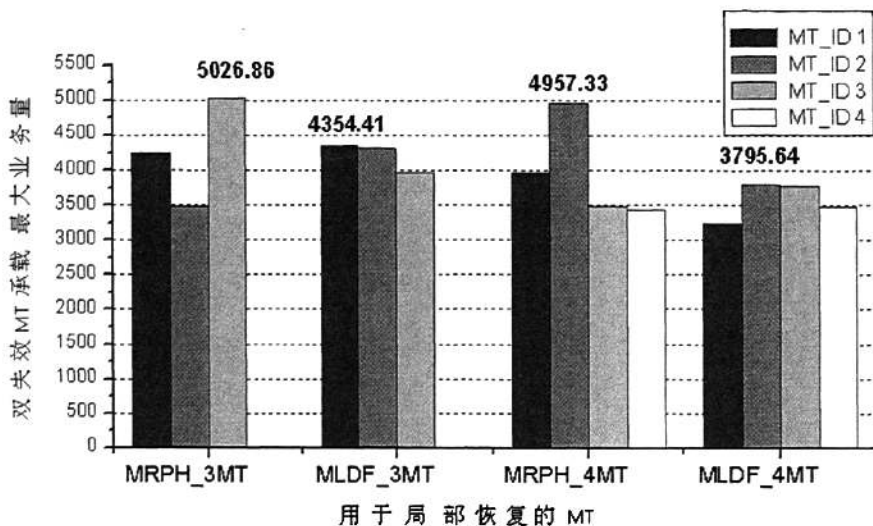


图 3-5 局部恢复模式下 MRPH 与 MLDF 算法生成 MT 的负载对比

和 MRPH 算法相较而言,采用 MLDF 算法所生成的备份拓扑可实现在双链路失效时,更好地均衡待恢复的业务量在多个备份拓扑之间的分布。由于 MRPH 算法可以保证将该算法所生成的备份拓扑应用于快速恢复后,业务的恢复路径总跳数最小。为分析 MLDF 算法所生成拓扑的恢复路径跳数与最优解的差距,下面按照 3.3.2 节中设置仿真参数,以单链路失效作为失效场景,在路径跳数方面对比两种算法所生成的备份拓扑的性能。

如图 3-6 所示,分别采用 3 个和 4 个备份拓扑用于业务路径的全局恢复时,使用 MRPH 算法求得的业务传输路径总跳数的最小值为 1514 和 1424。对于采用 MLDF 算法所生成的两组备份拓扑,为所有受失效影响的业务所提供的传输路径的总跳数分别为 1858 和 1638,相对于最优解增加了 22.72%和 15.03%。

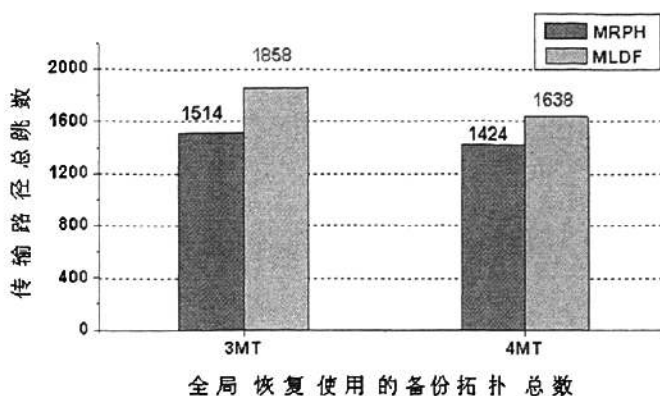


图 3-6 全局恢复模式下 MRPH 与 MLDF 算法生成 MT 的传输路径总跳数对比

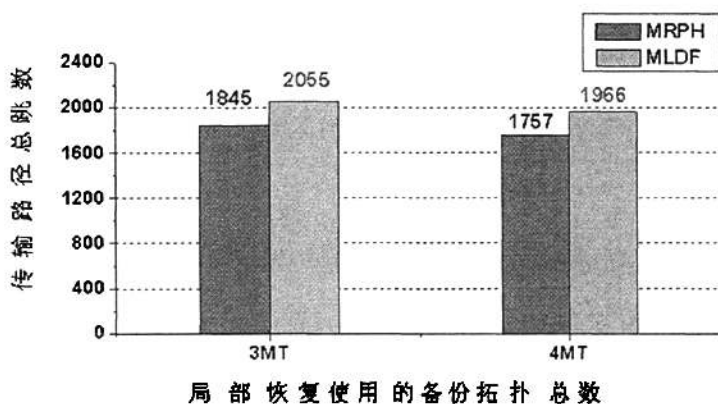


图 3-7 局部恢复模式下 MRPH 与 MLDF 算法生成 MT 的传输路径总跳数对比

如图 3-7 所示，分别采用 3 个和 4 个备份拓扑用于业务路径的局部恢复时，使用 MRPH 算法求得的业务恢复路径总跳数的最小值为 1845 和 1757。若应用 MLDF 算法生成两组备份拓扑，业务传输路径总跳数分别为 2055 和 1966，相对于最优解增加了 11.38% 和 11.90%。

由此可见，MRPH 算法与 MLDF 算法所生成的备份拓扑在降低路径跳数和均衡拓扑负载这两方面各有一定优势，因此可根据不同的 QoS 需求选取具有相应优化目标的拓扑生成算法。

3.4 本章小结

本章研究用于生成备份拓扑的算法并进行性能对比。

Minimum 算法用于生成可以保护原始拓扑中所有链路的最小个数的备份拓扑；相对于 Minimum 算法，Rich 算法用于生成连通度更大的备份拓扑。实验证明增加备份拓扑的个数，可以减少业务传输路径的跳数。

然后对于文献^[27]提出的 ILP 模型加以改进，在指定拓扑总数的前提下，对于局部恢复和全局恢复方案，分别求得所有受失效影响的业务其恢复路径总跳数的最小值。发现当备份拓扑个数增加到一定值之后，传输路径总跳数再无改善。

此外，针对备份拓扑可独立保护的双链路失效情形，提出了一种新的拓扑生成方案，可用于降低备份拓扑所需承载的待恢复业务量的最大值。并针对传输路径总跳数，以及双失效情形下拓扑所需承载的最大业务量这两方面指标，对后两种拓扑生成算法的性能加以对比。由所得到的数据可以看出两种算法对于各自所针对的优化目标分别具有良好的性能。

第四章 多拓扑权重规划算法研究

与失效后路径恢复相伴的问题是流量如何分配。理想的网络规划方案应该是合理地分配可用链路上的流量负载，使得最小化发生链路拥塞的几率。因此设计快速恢复技术时，也应考虑如何使被恢复的业务流在网络中的传输路径上重新分布后不会导致链路负载过大。

本章研究如何求解一组拓扑的链路权重，使得在采用 MTR 技术实现链路失效后业务路径的快速恢复时，可获得较优的负载均衡效果。首先在 4.1 节中简要介绍现有的技术方案，然后在 4.2 节中对于本章要求解的问题加以描述。4.3 节中对于 MTR 应用于失效恢复后最小化网络最大链路利用率的这一权重设计问题，给出了 ILP 描述。在 4.4 节中采用遗传算法，对多拓扑的权重设计问题加以求解。

4.1 研究背景

链路的权重值通常按照链路的容量的倒数或物理长度来设置^[28]，其取值范围为 1 到 65535 之间的整数，代表在该链路上传输数据包所需的代价。节点转发数据包时，采用 SPF 算法为数据包计算到目的节点的具有最小代价的路径。

等价多径 (Equal-Cost Multi-Path, ECMP) 是通过多条等价路径发送数据包的一项路由技术^[29]。为了更好的实现负载均衡，包括 OSPF 和 IS-IS 在内的路由协议，可支持 ECMP 路由，即如果数据包到达同一目的节点有多条代价相等的候选路径，则路由器按照“平等选择多路径”来进行路由^[30]。

路由器的转发模块使用下一跳节点来区分具有最小代价值的多条路径，本地节点将数据流按照到目的节点的下一跳个数进行等分，即在转发每个数据包时，路由器按照轮询 (Round-Robin) 的方式决策该数据包使用哪一个下一跳进行转发。

图 4-1 是使用 ECMP 技术进行路由转发的示例。假定路由器 A 有大小为 120 的业务量需要传输到路由器 F。网络的链路权重均为 1。则 A 和 F 之间有 3 条等价路径，分别为：①A-B-D-F；②A-B-E-F；③A-C-F。图中箭头方向表示数据包在链路上的传输方向，箭头上的数值表示该链路承载的业务量大小，即链路负载。路由器 A 在到 F 的等价路径上具有两个下一跳：路由器 B 和路由器 C，因此 A 到 F 的业务 120 在 A-B 和 A-C 这两条链路上均分传输。

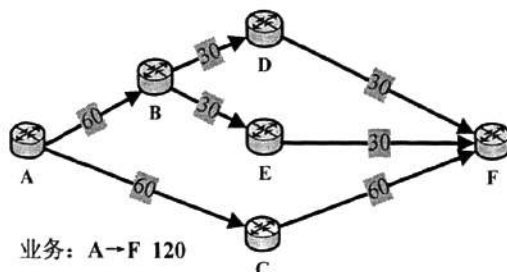


图 4-1 ECMP 路由示意图

研究者们致力于在可对业务量矩阵进行估计的前提下找到一组链路权重以获得良好的负载均衡效果。已经被证实通过启发式算法调整链路权重，对于给定的业务量矩阵可以得到接近于最优解的负载均衡效果^[31]。

可以将禁忌搜索 (Tabu Search) 算法应用于链路权重的求解问题^[32]，此方法可使求得的链路权重在无失效情形下和发生任意单链路失效的情形下都具有良好性能。该方法的问题是需要大量计算来预测在所有可能发生的失效情形下的流量分配，因而被提议可用于对一些关键链路失效的情形进行链路权重优化^[33]。

线性规划算法也可应用于求解基于 ECMP 路由时的最优链路权重^[34,35]，但是此两种方法所提出的 ILP 模型只能用于求解无失效情形下的单个拓扑的链路权重。

在可对节点间业务需求量进行估计的前提下，求解链路权重使得采用 ECMP 路由规则时业务的源宿节点对之间具有可行路由，同时获得良好的负载均衡效果，是一个 NP 问题^[36]，因此对于规模较大的网络拓扑其运算量很大。

4.2 问题描述

MTR 用于失效后网络业务路径的快速恢复时，链路负载及路径跳数等性能受以下四方面因素影响：业务量矩阵、节点转发业务所采用的路由规则、多拓扑（即备份拓扑）结构、链路权重。

在第二章中，介绍了采用 MTR 技术时实现业务路径恢复的两种方案和传输路径切换拓扑的两种方案，在第三章中，介绍了生成多拓扑的系列算法。如果采用拓扑数目较少的多拓扑集合，可以减少路由器储存的相关拓扑信息，同时也降低了路由器转发判决的复杂性。但是 MTR 所使用的多拓扑个数越少，多拓扑连通度越低，由于采用多拓扑作为业务恢复路径的载体时，业务仅由多拓扑中的部分链路承载，因此链路发生拥塞的几率也越大。因此必须根据实际的网络拓扑选择适

当的多拓扑个数。此外，在多拓扑总数及结构已经确定的前提下，为了进一步降低链路拥塞的概率，从而对 MTR 快速恢复技术的性能进一步优化，可以通过调整多拓扑的链路权重以达到负载均衡的目的。

本章主要研究的问题是假定已知网络的原始拓扑、备份拓扑和节点间业务量需求，在网络发生单链路失效的情形下，基于已求得的无失效情形下网络链路负载信息，如何设置每个备份拓扑的权重，从而使得在所有单链路失效情形下，使用这些备份拓扑进行业务路径恢复时，将发生链路拥塞的可能性降至最低。此问题可归结为：求解备份拓扑链路权重，最小化所有单链路失效情形下的网络链路的最大利用率。

4.3 多拓扑权重设计问题的 ILP 描述

多拓扑权重设计问题可归结为整数线性规划问题，本节使用 ILP 模型对其加以描述。下面给出本节中所用到的符号的定义：

原始拓扑的节点集合 V ，链路集合 L ，单链路失效情形集合 F 。链路视为单向。链路 l^+ 和 l^- 为同一相邻节点对之间两个方向上的链路， l^+ 或 l^- 发生故障后对应的失效情形均为 f ，其中 $l^+ \in L$ ， $l^- \in L$ ， $f \in F$ 。即假定若发生链路故障，则同一对相邻节点间两个方向上的链路同时失效，对应于一种失效情形。链路 l 的起点记作 $F(l)$ ，终点 $H(l)$ 。所有以节点 v 为终点的链路集合为 $I(v)$ ，所有以节点 v 为起点的链路集合为 $O(v)$ 。

备份拓扑集合 MT ，备份拓扑总数 M 。原始拓扑编号 0，备份拓扑编号 1~ M 。使用矩阵 D 描述具有业务传输需求的源宿节点对。

参数 $INFINITY$ 设置为一个很大的整数。

1. 常量

- ① c_l ——链路 l 的容量。
- ② d_{st} ——源宿节点对 st 间的业务量需求值。 $s \in V$ ， $t \in V$ ， $s \neq t$ 。
- ③ $x_{st}^0(l)$ ——从 s 到 t 的业务在原始拓扑中链路 l 上的分流比， $0 \leq x_{st}^0(l) \leq 1$ 。
- ④ SL_l^m ——布尔值。为 1：链路 l 可被备份拓扑 m 保护，即备份拓扑 m 中不含链路 l 。为 0：链路 l 不被备份拓扑 m 保护，即链路 l 存在于备份拓扑 m 中。
- ⑤ $Move_{st}^f(l)$ ——某失效场景 f 下，链路 l 上承载的源宿节点对 st 间的业务量需要切换至备份拓扑上传输的百分比。
- ⑥ $xu_{st}^0(v)$ ——从节点 s 到 t 的业务在原始拓扑中汇聚到节点 v 的分流比之和，

且 $0 \leq x_{st}^0(v) \leq 1$ 。

2. 变量

① R ——优化目标：最小化所有可能发生的单链路失效情形下的全网链路利用率的最大值， $R \geq 0$ 。

② $U(f)$ ——某失效场景 f 下全网链路利用率的最大值， $U(f) \geq 0$ 。

③ u_l^f ——某失效场景 f 下链路 l 的利用率， $0 \leq u_l^f \leq 1$ 。

④ w_l^m ——链路 l 在第 m 号备份拓扑上的权重值。 $1 \leq w_l^m \leq INFINITY$ ，且 w_l^m 为整数。

⑤ Z_{st}^m ——从节点 s 到节点 t 在第 m 号备份拓扑上的最小路径代价（即最短路径长度）， $Z_{st}^m \geq 0$ 。

⑥ $APP_{st}^m(l)$ ——布尔值。为 1：链路 l 在备份拓扑 m 中业务 st 的恢复路径上。为 0：不在恢复路径上。

⑦ $x_{st}^{f,m}(l)$ ——某失效场景 f 下从节点 s 到节点 t 的业务在第 m 个备份拓扑上链路 l 上的分流比， $0 \leq x_{st}^{f,m}(l) \leq 1$ 。

⑧ $K_{st}^{f,m}$ ——用于控制按照 ECMP 路由规则分流， $s \in V$ ， $t \in V$ ， $K_{st}^{f,m} \geq 0$ 。即某失效场景 f 下在第 m 个备份拓扑上，计算从节点 s 到节点 t 的最短路径。以 s 为起点的所有邻接链路中，位于到 t 的最短路径上的那些链路，所承载的到目的节点 t 的业务量均为 $K_{st}^{f,m}$ 。

⑨ $y_l^{f,m}(l)$ ——发生某失效情形 f 后，以节点 t 为目的节点的所有业务在第 m 个备份拓扑上的链路 l 上的分流比之和， $y_l^{f,m}(l) \geq 0$ 。

3. 优化目标

Minimize R

4. 约束条件

(a) 计算所有失效情形下的全网链路最大利用率

$$R \geq U(f), \forall f \in [1, F] \quad (4-1)$$

(b) 计算每种失效情形下的全网链路最大利用率

$$U(f) \geq u_l^f, \forall f \in [1, F], l \in [1, L] \quad (4-2)$$

(c) 若采用 MTR 全局恢复方式时，在每种失效情形下的流量守恒方程

$$\sum_{l \in O(s)} \left\{ \left[x_{st}^0(l) - Move_{st}^f(l) \right] + \sum_{m=1}^M \left[SL_f^m * x_{st}^{f,m}(l) * (1 - SL_l^m) \right] \right\} = 1, \forall f \in [1, F], st \in D \quad (4-3)$$

$$\begin{aligned} & \sum_{l \in O(v)} \left\{ \left[x_{st}^0(l) - Move_{st}^f(l) \right] + \sum_{m=1}^M \left[SL_f^m * x_{st}^{f,m}(l) * (1 - SL_l^m) \right] \right\} - \\ & \sum_{l \in I(v)} \left\{ \left[x_{st}^0(l) - Move_{st}^f(l) \right] + \sum_{m=1}^M \left[SL_f^m * x_{st}^{f,m}(l) * (1 - SL_l^m) \right] \right\} = 0, \quad (4-4) \\ & \forall f \in [1, F], st \in D, v \neq s \neq t \in V \end{aligned}$$

(c') 若采用 MTR 局部恢复方式时，在每种失效情形下的流量守恒方程

$$\begin{aligned} & \sum_{l \in O(F(f))} \left\{ \left[x_{st}^0(l) - Move_{st}^f(l) \right] + \sum_{m=1}^M \left[SL_f^m * x_{st}^{f,m}(l) * (1 - SL_l^m) \right] \right\} = xu_{st}^0(F(f)), \quad (4-5) \\ & \forall f \in [1, F], st \in D \end{aligned}$$

$$\begin{aligned} & \sum_{l \in O(H(f))} \left\{ \left[x_{st}^0(l) - Move_{st}^f(l) \right] + \sum_{m=1}^M \left[SL_f^m * x_{st}^{f,m}(l) * (1 - SL_l^m) \right] \right\} = xu_{st}^0(H(f)), \quad (4-6) \\ & \forall f \in [1, F], st \in D \end{aligned}$$

$$\begin{aligned} & \sum_{l \in O(v)} \left\{ \left[x_{st}^0(l) - Move_{st}^f(l) \right] + \sum_{m=1}^M \left[SL_f^m * x_{st}^{f,m}(l) * (1 - SL_l^m) \right] \right\} - \\ & \sum_{l \in I(v)} \left\{ \left[x_{st}^0(l) - Move_{st}^f(l) \right] + \sum_{m=1}^M \left[SL_f^m * x_{st}^{f,m}(l) * (1 - SL_l^m) \right] \right\} = 0, \quad (4-7) \\ & \forall f \in [1, F], st \in D, v \neq F(f) \neq H(f) \in V, v \neq s \neq t \in V \end{aligned}$$

(d) 在每种失效情形下，不保护此失效链路的所有备份拓扑均不承载任何业务。

$$\sum_{st \in SD} \sum_{l \in [1, L]} \sum_{m=1}^M (1 - SL_f^m) * x_{st}^{f,m}(l) * (1 - SL_l^m) = 0, \quad \forall f \in [1, F] \quad (4-8)$$

(e) 求解最小代价路径。

$$w_l^m - (Z_{F(l)t}^m - Z_{H(l)t}^m) \leq (1 - APP_{F(l)t}^m(l)) * INFINITY, \quad \forall l \in L, m \in [1, M], t \in V \quad (4-9)$$

$$w_l^m - (Z_{F(l)t}^m - Z_{H(l)t}^m) \geq 1 - APP_{F(l)t}^m(l), \quad \forall l \in L, m \in [1, M], t \in V \quad (4-10)$$

(f) 若链路 l 被某备份拓扑保护，则该链路在此备份拓扑上的权重为无穷大。

$$w_l^m \geq INFINITY * SL_l^m, \quad \forall m \in [1, M], l \in L \quad (4-11)$$

(g) 限制每对相邻节点间两个方向上的链路的权重数值相等。

$$w_{uv}^m = w_{vu}^m, \quad \forall m \in [1, M], uv \in [1, L], vu \in [1, L] \quad (4-12)$$

(h) 在每种失效情形下，失效链路的链路利用率为 0。

$$u_l^f = 0, \quad \forall f \in [1, F], l \in [1, L], l = f^+ \text{ or } l = f^- \quad (4-13)$$

(i) 对于每种失效情形，计算此时未发生失效的链路的利用率。

链路利用率=链路负载/链路容量。网络中链路的负载等于：无失效时在原始拓扑中的负载—发生链路失效后需要使用备份拓扑上路径传输的那部分业务量+该链路在备份拓扑中承载的被恢复的业务量。

$$u_l^f = \frac{\sum_{st \in D} \left\{ d_{st} * (x_{st}^0(l) - Move_{st}^f(l)) + d_{st} * \sum_{m=1}^M [SL_f^m * x_{st}^{f,m}(l) * (1 - SL_l^m)] \right\}}{c_l} \quad (4-14)$$

$, \forall f \in [1, F], l \in [1, L], l \neq f, m \in [1, M]$

(j) 限制若备份拓扑 m 保护链路 l ，即 $SL_l^m=1$ 时，此链路 l 不会位于业务在备份拓扑中的最小代价路径上。

$$APP_{st}^m(l) \leq 1 - SL_l^m, \forall st \in D, m \in [1, M], l \in L \quad (4-15)$$

(k) 若某节点有多条邻接链路位于到达同一目的节点的最小代价路径上，则按照 ECMP 路由规则，将需要传输给此目的节点的业务在这些邻接链路上等分传输。

$$\sum_{s \in V, s \neq t} x_{st}^{f,m}(l) = y_t^{f,m}(l), \forall t \in V, m \in [1, M], l \in L, f \in [1, F] \quad (4-16)$$

$$y_t^{f,m}(l) \leq INFINITY * APP_{F(l)t}^m(l), \forall st \in D, m \in [1, M], l \in [1, L], f \in [1, F] \quad (4-17)$$

$$0 \leq K_{F(l)t}^{f,m} - y_t^{f,m}(l) \leq INFINITY * (1 - APP_{F(l)t}^m(l)), \forall st \in D, m \in [1, M], l \in L, f \in [1, F] \quad (4-18)$$

4.4 遗传算法求解多拓扑链路权重

根据实际目标需求，通过某种方法来筛选和排列研究对象的序列值以获得最优的目标值，称为组合最优化问题。可以通过枚举研究对象所有可能的取值，经过对比来计算出组合最优化问题的最优解。然而枚举方法的时间复杂度过大，因此人们提出了一系列用于求解组合最优化问题的算法，称为启发式算法（heuristic algorithms）。

启发式算法是指在可接受的计算费用（如算法运行时间、数据存储空间）范围内寻找组合优化问题的可行解，所求得解不一定是最优解，甚至在多数情况下，无法计算所得解与最优解的偏离程度^[37]。但在实际应用中，由于启发式算法同最优解算法相比可以获得更快的计算速度，因此在求解规模较大复杂度较高的问题时，启发式算法得到了广泛应用。

启发式算法包括遗传算法（Genetic Algorithm, GA）、禁忌搜索（tabu search）、

模拟退火 (simulated annealing)、人工神经网络 (artificial neural networks) 和拉格朗日松弛 (lagrangian relaxation) 等算法, 可应用于运筹学、交通运输、通信网络以及人工智能等诸多领域。

对于最小化目标函数的优化问题, 遗传算法、禁忌搜索、模拟退火、人工神经网络等算法求得的目标函数值都是最优解的上界, 即采用这些算法求得的目标函数值都大于最优值, 而拉格朗日松弛算法则可用于求得最优解的下界。在求解具体问题时无法在理论上断言上述算法的优劣, 只能通过采用不同方法对所得数据进行分析比较, 来判断哪种算法对于求解该问题具有更好的性能^[37]。

在本节中选用遗传算法用于求解 MTR 技术应用于 IP 网络快速恢复后的多拓扑权重设计问题。

4.4.1 遗传算法介绍

遗传算法由美国 Michigan 大学的 J.Holland 教授提出, 是一种基于生物进化理论的高效搜索方法。遗传算法借鉴了生物遗传学的部分原理, 通过自然选择、遗传、变异等机制, 提高各个个体的适应性, 在种群中搜索具有更优适应性的个体。随着对其理论研究和应用研究的逐步深入, 遗传算法的应用范围已从早期的求解组合优化问题拓展到其它更新、更工程化的应用领域。

遗传算法是用于解决搜索问题的算法, 此类算法均具有以下特征:

- ① 首先为待求解的问题构造一组候选解;
- ② 依据特定规则 (例如遗传算法中的适应性条件) 评估这些候选解的优劣;
- ③ 保留部分性能较好的候选解, 丢弃其它候选解;
- ④ 结合所保留的候选解, 按照某种方案继续生成新的候选解, 在解空间中进一步扩大搜索范围。

采用遗传算法解决实际问题时, 为求解某些变量的值从而使目标值趋于最优, 可将此类变量进行编码, 即按照变量的特征将其映射为染色体, 染色体由一定长度的基因链组成。构造候选的染色体, 从对解空间进行并行搜索出发, 统计各染色体的适应性, 由此对染色体种群进行选择, 按照适者生存的思想筛选出可使目标问题具有更优值的染色体, 即保留具有较好适应性的染色体, 以这些染色体作为父代, 通过控制染色体基因的交换和突变, 生成新一代的染色体种群, 使得该染色体种群可向具有更优适应性的方向演化。设定某停止条件用于控制算法在解空间中的搜索时间和范围, 从而使得在有限的迭代次数内可求出具有较好性能的

目标问题的解。

遗传算法具有以下优点：

(1) 许多传统的搜索算法采用单点搜索，即从单个解的初始值开始迭代求解，这一方式使得所求出的解易陷入局部最优。而遗传算法更适用于求解具有多变量和多目标的优化问题，通过从解的串集开始搜索，同时对解空间中的多个解进行筛选，扩大了搜索的覆盖面，减少了陷入局部最优解的风险。

(2) 遗传算法使用适应度函数值来评估染色体的优劣，从而控制染色体的变迁方向。由于可根据实际应用问题来制定适应度函数，使得遗传算法的应用更具普适性和灵活性。

(3) 遗传算法具有自适应和自学习性。遗传算法可利用在解的进化过程中获得的信息进行搜索，可保证较优的解具有更大的生存概率。

4.4.2 求解权重的算法设计

在本节中，针对“求解一组备份拓扑的链路权重，最小化所有单链路失效情形下的网络链路的最大利用率”这一问题需求，结合遗传算法对问题进行具体分析，并设计求解链路权重的实现方案和流程。

4.4.2.1 算法的详细设计

遗传算法的设计主要包括五个步骤：对问题解空间进行编码，创建初始种群，设计适应度函数，遗传繁殖，设置收敛条件。下面依次介绍每个步骤的原理以及设计方案。

(1) 对问题解空间进行编码

遗传算法中的编码是指采用由基因按一定结构组成的染色体（或称为个体）来表示解空间的值。

假定网络原始拓扑的链路集合为 L ，备份拓扑集合 MT ，备份拓扑的总数为 M 。 w_l^m 表示链路 l 在第 m 号备份拓扑上的权重值， $1 \leq w_l^m \leq INFINITY$ ，且 w_l^m 为整数。在 OSPF 协议中，将链路权重取值范围的上界 $INFINITY$ 定义为 65535，在仿真时为了缩小搜索空间的规模，将链路权重的取值范围限定为 $[1, 40]$ 。对于待求解的一组备份拓扑的链路权重，以拓扑中每条链路权重的值作为一个基因，把每个备份拓扑的链路权重集合 w^m 分别映射为一个染色体， w^m 是 L 维向量， $w^m = \langle w_1^m, w_2^m, \dots, w_L^m \rangle$ ， $m \in [1, M]$ 。

(2) 创建初始种群

可以采用随机选取或根据一定准则来选择初始种群，作为对解空间进行搜索的起点。这两种方式各有优缺点：随机选取初始种群使对可行解的遍历更具公平性，但可能会增加迭代的次数；根据一定准则，基于某些预先求得的结论或经验值来生成初始种群，这一方式在某些情况下可以减小对某部分解空间进行不必要的搜索的几率。然而具有争议的是，由于无法评估当前可行解对最优解的偏离程度，一些研究者建议不应在搜索算法开始时，带有一定的偏见来选择初始种群，以免造成所求得解的性能降级。在本节中，采用随机法生成初始种群，初始群体的规模设定为 200，即为每个备份拓扑分别生成 200 套权重方案，链路权重值在 $[1,40]$ 间随机产生。

(3) 设计适应度函数

适应度函数是评估染色体性能优劣的标准，是筛选个体以及在算法迭代中控制个体繁殖演变方向的依据。在遗传算法中，对适应度函数唯一的限制规则是要求其结果为非负值。而适应度函数的计算表达式可根据实际应用问题的目标函数转换而成。

对于备份拓扑的权重设置问题，主要目的是尝试为一组备份拓扑制定相应的链路权重，对于所有可能发生的单链路失效情形，将备份拓扑应用于失效后的业务路径恢复时，可由权重值计算出更优的恢复路径，即采用这些恢复路径传输业务不会导致链路发生拥塞。求解的对象是一组拓扑的链路权重，优化目标是降低所有单链路失效情形下网络链路利用率的最大值。因此将评价染色体（即备份拓扑权重）的适应度函数定义为：对于所有单链路失效情形，采用备份拓扑进行业务路径的恢复后，全网所有链路利用率的最大值。适应度函数值越小，所得到的染色体性能越优，也就是说，使所有失效情形下网络链路最大利用率越小的那套备份拓扑权重的性能越好。

由于在仿真中假定每种单链路失效情形独立发生，且每条链路仅在一个备份拓扑中被保护，因此可将待求解的优化目标进一步简化，把所有单链路失效情形按照链路所对应的备份拓扑编号进行分组。把“在所有单链路失效情形下求解一组备份拓扑的权重”这一问题转化为“依次对于每个备份拓扑可保护的所有单链路失效情形，求解该备份拓扑的权重”。

下面符号的定义同 4.3 节。已知原始拓扑的节点集合 V ，链路集合 L ，单链路失效情形集合 F 。链路视为单向，每条链路的容量为 c_l 。假定若发生失效情形 f ，则与 f 所对应的两个端点间对称方向上的链路 f^* 和 f 同时失效。 d_{st} 表示源宿节点对 st 间的业务量需求值。 u_l^f 表示在失效场景 f 下链路 l 的利用率。链路的利用率等于

链路的负载除以链路容量。布尔值 SL_l^m 为 1 时表示链路 l 被备份拓扑 m 保护, 为 0 时表示链路 l 不被备份拓扑 m 保护。在遗传算法的实现过程中, 将染色体解码为备份拓扑的链路权重值, 使用 ECMP 路由规则根据权重计算路径, 并求解适应度函数的值。 $x_{st}^0(l)$ 表示从 s 到 t 的业务在原始拓扑中链路 l 上的分流比。 $Move_{st}^f(l)$ 表示发生 f 失效后链路 l 上原来承载的 st 间业务量需要切换至备份拓扑上路径传输的百分比。 $x_{st}^{f,m}(l)$ 表示在失效场景 f 下节点对 st 间的业务在第 m 个备份拓扑的链路 l 上的分流比。

优化目标 Φ 表示对于所有单链路失效情形, 采用备份拓扑后全网所有链路利用率的最大值。 Φ^m 表示评价每个备份拓扑权重性能的适应度函数, $m \in [1, M]$ 。如果分别对于每个备份拓扑, 求得使 Φ^m 最小的权重 w^m , 也就等价于求得了可使 Φ 值最小化的一组备份拓扑权重 $\{w^m\}$ 。

适应度函数 Φ^m 定义如下:

$$\min \quad \Phi^m, \quad \forall m \in [1, M] \quad (4-19)$$

$$\Phi^m \geq u_l^f, \quad f \in [1, F] \text{ and } SL_f^m = 1, l \in [1, L] \quad (4-20)$$

$$u_l^f = 0, \quad \forall f \in [1, F] \text{ and } SL_f^m = 1, l \in [1, L], l = f^+ \text{ or } l = f^- \quad (4-21)$$

$$u_l^f = \frac{\sum_{st \in D} d_{st} * x_{st}^0(l) - \sum_{st \in D} d_{st} * Move_{st}^f(l) + \sum_{st \in D} d_{st} * x_{st}^{f,m}(l)}{c_l} \quad (4-22)$$

$$, \quad \forall f \in [1, F] \text{ and } SL_f^m = 1, l \in [1, L], l \neq f$$

式(4-20)定义 Φ^m 为: 对于所有被 m 号备份拓扑保护的失效情形, 采用此备份拓扑进行业务恢复后全网链路利用率的最大值。式(4-21)限定了失效链路的链路利用率为 0。式(4-22)是未发生失效的链路其利用率的计算公式。分母为链路容量。分子在失效场景 f 下链路 l 的负载, 由三部分组成: ①网络未发生失效时链路 l 承载的业务量; ②网络发生失效后链路 l 上因受失效所影响而需重路由的业务量; ③链路 l 作为备份拓扑 m 上链路所承载的被恢复的业务。

在使用备份拓扑计算业务的恢复路径时, 若采用的是全局恢复方案, 则受失效链路所影响的路径是本来应该经过该失效链路传输的业务在原始拓扑中从源点到宿点的工作路径, 在备份拓扑上从该业务源点到宿点计算新的恢复路径; 若采用是局部恢复方案, 则受失效链路所影响的路径是原本应该经过该失效链路传输的业务, 在原始拓扑中从失效链路下游端点到业务宿点之间的这段工作路径。在备份拓扑

上应从失效链路的上游端点计算到业务宿点的恢复路径。

(4) 遗传繁殖

通过繁殖产生新一代种群是遗传算法的核心步骤。为当前种群计算适应度函数值之后，需要设置一系列参数用以控制产生子代种群，以拓展个体对解空间的覆盖范围。遗传算法中涉及到的这些参数称为算子（operator）。

常用的算子有以下几种：

①选择算子(selection)：从种群中按照某一概率选择优秀个体，淘汰劣质个体。最常见的是采用轮盘赌（roulette wheel）模型，使用某个体的适应度值除以种群所有个体的适应度之和来计算该个体被选择的概率。

②交叉算子(crossover)：对于交叉这一过程又可称作杂交或交配。是指将被选中的作为父代的两个个体按照交叉概率 K ，以某种方式相互交换其部分基因，生成新的个体。

③变异算子(mutation)：是指将个体的基因链中各位按照变异概率 p_g 进行调整。变异算子不仅可使遗传算法具有全局的随机搜索能力，还有助于保持种群的多样性^[38]。

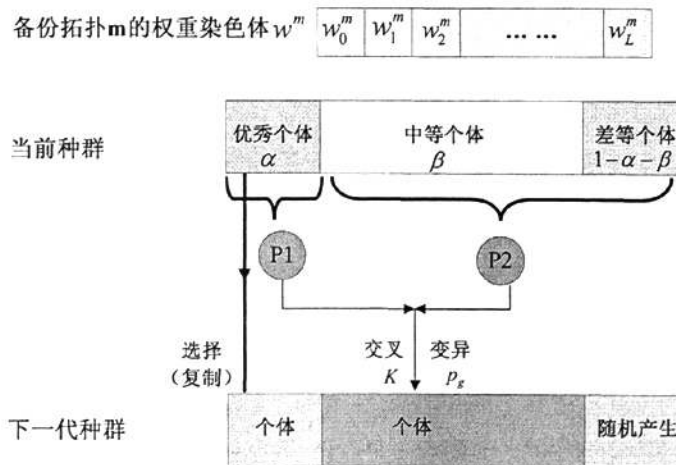


图 4-2 遗传算法生成下一代种群的过程

在本节中，将遗传繁殖这一过程设计为：在根据当前种群产生下一代时，按照个体的适应度值将当前种群分成三类——优秀个体占种群个体数目的比例为 α ，中等个体占种群个体的比例为 β ，差等个体占种群个体的比例为 $1-\alpha-\beta$ 。在实验中 α 取值为 10%， β 为 70%。当前种群中的优秀个体直接晋升为下一代个体，同时随

机产生占种群总数比例为 $1-\alpha-\beta$ 的后代。此外，从当前种群里的优秀个体中选择个体作为父代，记作 P1，从当前种群里的中等个体和差等个体中选择个体作为父代，记作 P2。由父代 P1 和 P2 经过交叉和变异得到占种群总数比例为 β 的下一代个体。

图 4-2 为由当前种群生成下一代种群的过程。在交叉和变异阶段，针对已选中的一对父体 P1 和 P2，交叉概率 K 设置为 0.7，变异概率 p_g 设置为 0.01^[39]。分别随机生成 L 维向量 s 和 r ，向量的取值范围为 0 到 1 之间的实数。由父代进行交叉变异产生子代的伪代码如图 4-3 所示：

```

for all genes  $l=1,2,\dots,L$  do
    if  $s[l] < p_g$ 

         $w^m[l] = \text{random}[1,\dots,\text{INFINITY}]$ 
    else if  $r[l] < K$ 

         $w^m[l] = P1[l]$ 
    else

         $w^m[l] = P2[l]$ 
end

```

图 4-3 父代交叉变异过程的伪码

在根据父代个体生成子代的过程中，对于当前待生成的子代个体的每个基因位，若在向量 s 中对应于此基因位的随机数值小于变异概率 p_g ，则执行变异操作，将子代个体中此基因位的数值赋以 $[1, \text{INFINITY}]$ 之间的某个随机整数；若向量 s 中对应此基因位的数值大于等于变异概率 p_g ，则执行交叉操作——判断若向量 r 中对应于子代个体此基因位的随机数值是否小于交叉概率 K 。若小于，则令子代个体当前基因位的值等于父代 P1 的相应基因位；否则，将父代 P2 的相应基因位的值赋给子代个体。交叉概率 K 的取值越大，子代继承优秀父代的概率就越大。变异概率 p_g 越大，搜索范围也随之扩大。

(5) 设置收敛条件

为了控制遗传算法的运行时间，应根据实际需求设置算法的停止条件。本文根据所求得的备份拓扑权重性能，调整并限制遗传繁殖的迭代次数，即种群的代数 t 。

4.4.2.2 算法的实现流程

图 4-4 为使用遗传算法求解一组备份拓扑权重的流程图。假定已知原始拓扑和备份拓扑的结构，原始拓扑的链路权重值和业务量需求矩阵。备份拓扑总数为 M 。

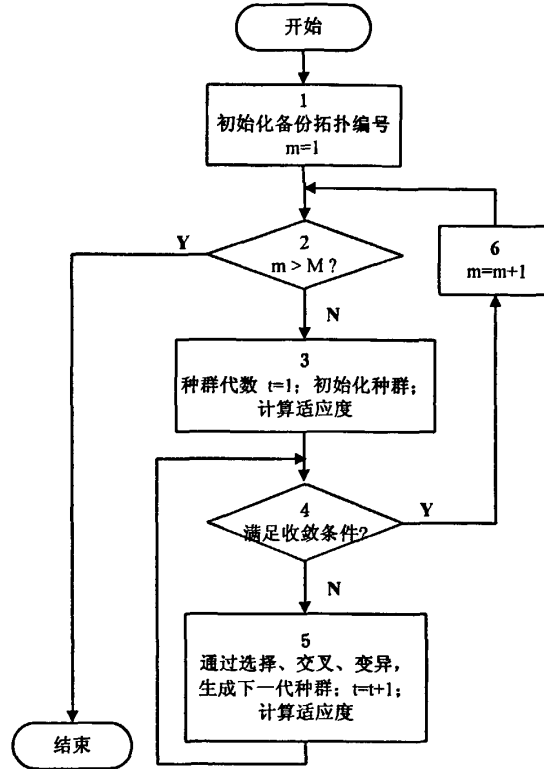


图 4-4 遗传算法求解备份拓扑权重的算法流程图

步骤 1：初始化当前待设权重的备份拓扑编号为 1。将备份拓扑权重按照链路编号进行编码。

步骤 2：判断是否已求解出 M 个备份拓扑的权重，是则程序结束。否则，进入步骤 3。

步骤 3：初始化种群代数 t 为 1 并随机生成初始种群。对于当前待设权重的备份拓扑可保护的所有单链路失效情形，根据指定的全局或局部业务恢复模式，计算当前种群的适应度函数值。

步骤 4：判断如果种群代数 t 已达到所设置的上限值，则转步骤 6；否则进入步骤 5，继续遗传繁殖产生子代。

步骤 5：根据前一代种群的适应度函数值，从种群个体中选择优等个体直接添

加入新一代种群集合，以优等和中、差等个体作为父代，通过变异和交叉产生新一代个体，最后随机产生部分个体。更新种群代数 t ，按照指定的业务恢复模式计算所生成的新一代种群个体的适应度。转步骤 4，判断是否已达到限定的最大迭代次数。

4.5 性能分析

在本节中，对于采用遗传算法所求得的备份拓扑权重的性能加以分析。

原始拓扑采用图 5-1，业务量矩阵按照表 5-1 设置。原始拓扑的链路权重均设置为 1，链路视为双向，链路容量设置为 5Tbit/s，采用 ECMP 路由规则计算链路负载。使用 3.1 节中所介绍的 Minimum 算法和 Rich 算法分别各生成 1 组备份拓扑，备份拓扑数目为 3 和 21。假定每种单链路失效情形独立发生。

为描述方便，下面将链路权重值均为 1 的权重方案简称为原始权重，使用遗传算法求得的新的权重方案称为优化权重。比较的对象是：①采用 OSPF 路由重收敛所得到的拓扑，原始权重；②3 个备份拓扑，原始权重；③3 个备份拓扑，优化权重；④21 个备份拓扑，原始权重；⑤21 个备份拓扑，优化权重。下面所对比的性能参数是：对于所有可能发生的单链路失效情形，使用上述①~⑤的拓扑实现链路失效后业务的恢复时，原始拓扑中所有物理链路的利用率的最大值。

可求得在无失效发生时，原始拓扑链路的最大利用率为 52.2%。

图 4-5 是当业务的恢复方式采用全局恢复时，分别应用拓扑①~⑤，在每种单链路失效情形下，全网链路的最大利用率对比图。横轴为各失效情形对应的链路编号。

当使用的备份拓扑为 21 个（即每个备份拓扑仅保护一种失效情形），此时备份拓扑的构造和采用 OSPF 重收敛后所更新的拓扑结构相同。由于采用全局恢复模式时，受失效影响的业务从业务源点开始计算恢复路径，因此当使用原始权重时，采用 21 个备份拓扑进行业务的全局恢复所求得的链路负载同使用 OSPF 重收敛后所更新的拓扑进行业务恢复相等，故在图 4-5 中，星号和圆点所标注的值完全重合。

由图 4-5 可知，未对权重进行优化时，在所有单链路失效情形下，若采用 3 个备份拓扑，全网最大链路利用率为 76.09%，若采用 OSPF 重收敛所更新后的拓扑，以及使用 MTR 技术应用 21 个备份拓扑分别进行业务的恢复时，全网最大链路利用率峰值均为 77.66%；在使用 4.4 节中介绍的遗传算法进行权重优化后，采用 3 个备份拓扑时最大链路利用率由 76.09%降至 70.99%，采用 21 个备份拓扑时

最大链路利用率可由 77.66%降至 63.63%。

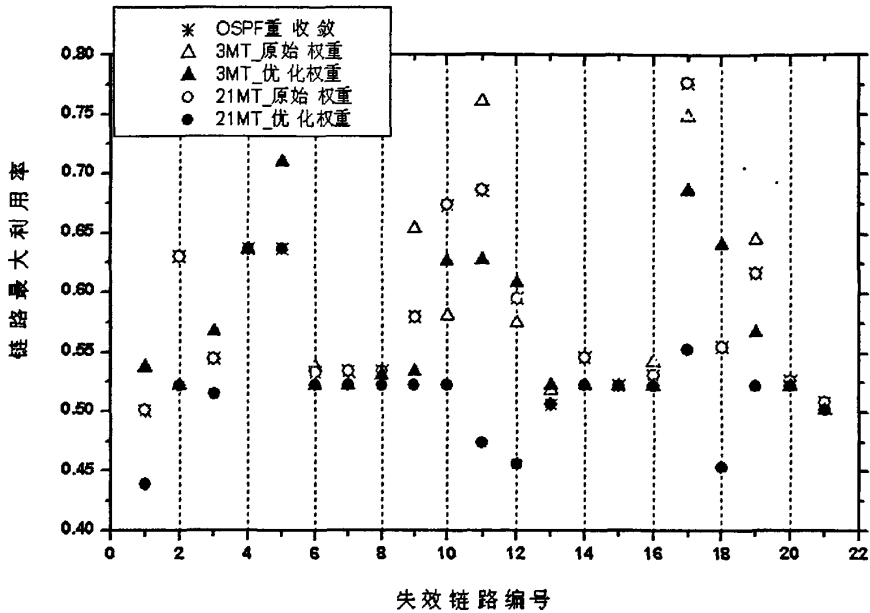


图 4-5 全局恢复模式下优化权重前后的最大链路利用率对比

图 4-6 是当备份拓扑采用局部恢复模式时，分别应用拓扑①~⑤时全网链路最大利用率的对比图。

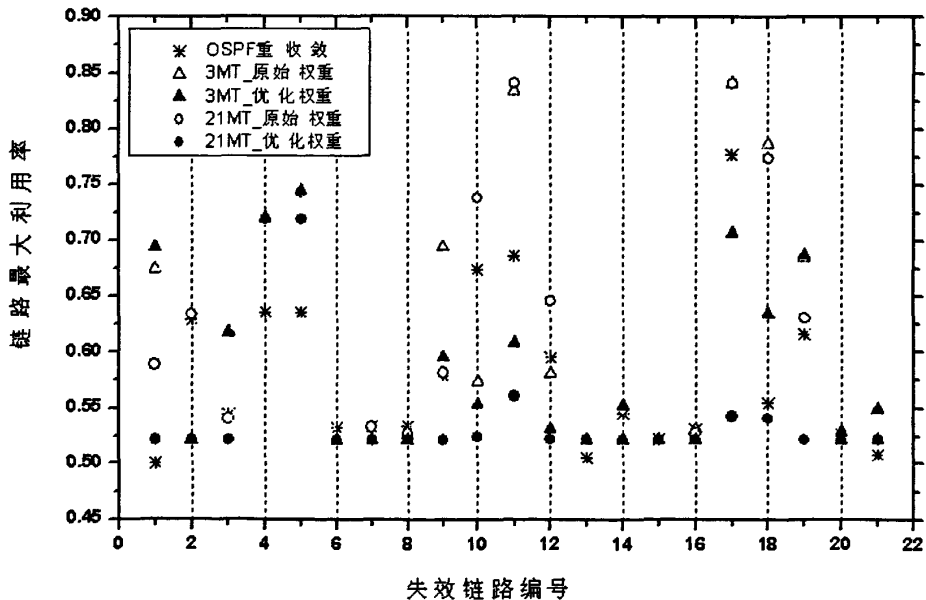


图 4-6 局部恢复模式下优化权重前后的最大链路利用率对比

由图 4-6 可知, 未对权重进行优化时, 在所有单链路失效情形下, 采用 OSPF 重收敛所更新后的拓扑进行业务恢复时, 全网最大链路利用率峰值仍为 77.66%, 分别采用 3 个和 21 个备份拓扑, 使用 MTR 技术进行业务的局部恢复后, 链路利用率峰值均为 84.09%; 使用遗传算法优化备份拓扑权重后, 采用 3 个备份拓扑时最大链路利用率降至 74.47%, 采用 21 个备份拓扑时可降至 71.94%。

下面通过表 4-1 和表 4-2, 对于使用 3 个和 21 个备份拓扑, 在优化权重前后进行业务恢复的效果分别加以对比。表中第二行的“最大值”表示在所有单链路失效情形下全网链路最大利用率的峰值, “平均值”则表示每种失效情形下链路最大利用率的平均值。

表 4-1 在所有单失效情形下使用 3 个 MT 时链路最大利用率对比

	全局恢复		局部恢复	
	最大值	均值	最大值	均值
原始权重	76.09%	58.53%	84.09%	61.85%
优化权重	70.99%	56.89%	74.47%	58.95%
降低	5.10%	1.64%	9.62%	2.9%

表 4-2 在所有单失效情形下使用 21 个 MT 时链路最大利用率对比

	全局恢复		局部恢复	
	最大值	均值	最大值	均值
原始权重	77.66%	57.95%	84.09%	62.03%
优化权重	63.63%	51.96%	71.94%	54.46%
降低	14.03%	5.99%	12.15%	7.57%

由以上数据可知, 将 MTR 技术应用于失效后路径的快速恢复时, 所使用的备份拓扑个数越多, 优化权重后链路的利用率越小, 这是由于备份拓扑的连通度越高, 计算恢复路径时的候选链路越多, 因此增加了负载均衡的几率。同时验证了在 2.2.3 节中得到的结论: 全局恢复方案和局部恢复方案相比, 全局恢复方案在负载均衡方面的性能更好。结果表明, 通过遗传算法设置权重可在一定程度上降低链路的利用率, 从而缓解链路拥塞。

4.4 节中以最小化所有失效场景下全网链路最大利用率为优化目标, 研究求解多拓扑链路权重的方法。对于最小化目标函数的优化问题, 采用遗传算法所求得解是最优解的上界。为了评估所得解与最优解的偏离程度, 可以使用 4.3 节的线性规划模型来求得问题的最优解, 然而对于规模较大的网络, 由于待求解的变量数目巨大, 使用线性规划软件在短时间内求解这一问题极为困难。文献^[32]提出了 ILP 模型, 用于求解所有失效场景下全网链路最大利用率的最小值。此方法是以所

有节点对间的 K 条可行路径作为输入数据, 在这些候选路径中, 求解业务在无失效和单链路失效场景下所使用的传输路径以及在路径上的分流比, 使得采用这些路径按照所求得的分流比承载业务时, 可最小化在无失效和所有失效场景下全网链路最大利用率。此法要求业务可在多条路径上以任意比例分流。使用此方法, 以网络中节点对间所有跳数不超过 7 的路径作为候选路径, 求得在所有失效情形下, 全网链路最大利用率可达到的最小值为 63.63%。可见对于本章所设计的遗传算法求解权重的方案, 与允许业务按任意比例在多路径间分流所求得的最优解还存在一定的差距, 有待于进一步改善算法以提高性能。

4.6 本章小结

本章研究多拓扑的链路权重优化方案, 用以解决当链路发生失效时采用 MTR 技术恢复业务路径后的负载均衡问题。

首先在 4.1 节中介绍了前人已有的研究成果, 然后在 4.2 节中对本章要研究的问题进行分析定义, 接着在 4.3 节中提出了用于求解一组多拓扑权重的 ILP 模型。在 4.4 节中, 将遗传算法应用于多拓扑的权重优化设计问题, 详细介绍了算法所涉及的相关参数设置以及设计流程。仿真结果证明了在将 MTR 技术应用于失效后业务路径快速恢复时, 通过优化多拓扑的权重, 可以降低全网链路的最大利用率。对于本章仿真所使用的拓扑, 可通过优化权重使全网链路的最大利用率降低约 5%~14%。

第五章 仿真平台设计与实现

在本文的第二章中,介绍了在链路失效情形下基于 MTR 技术如何恢复业务传输路径的路由转发方案,第三章中研究了 4 种用于生成多拓扑的算法,并在第四章中进一步讨论多拓扑链路权重的优化问题。为了验证方案的有效性,对不同的业务恢复方案和路径在多拓扑间的切换方案进行性能对比,以及评估分析不同拓扑结构对于性能指标的影响,搭建了 MTR 技术仿真平台。此仿真平台采用 C/C++ 语言编写,开发工具为 OPNET 11.5。仿真平台的调试与运行均在 Windows XP 操作系统下进行。

本章内容安排如下:首先介绍 OPNET 仿真工具,然后介绍仿真采用的拓扑结构以及相关参数,最后介绍 MTR 仿真平台的总体框架,包括仿真方案、节点模型以及进程模型的有限状态机的具体设计。

5.1 OPNET 仿真工具简介

本文使用 OPNET (Optimized Performance Network Engineering Tool) 系列产品中的 Modeler 搭建仿真平台。OPNET 作为一种主流网络仿真软件,为通信网络组网方案和协议配置的建模提供全面的模拟仿真开发环境。OPNET 集成了模型设计工具、数据收集工具和数据分析工具,通过执行离散事件仿真来分析各种模拟系统的行为和性能。OPNET Modeler 具有以下特点:

(1) 具有层次化、模块化的建模机制。使用 OPNET 工具搭建的仿真模型自顶向下可分为网络域、节点域和进程域三层:在网络域中,使用地理场景、子网、节点和链路等设备模型描述网络拓扑,模拟真实网络的物理设备环境;节点域由不同模块组成,模块中包含执行不同功能的进程模型;在进程域中,通过 C 或 C++ 编程语言以及 OPNET 自带的核心函数,驱动有限状态机实现各种协议算法。

(2) 丰富的模型库和灵活的建模机制。OPNET 提供标准模型库,同时可允许用户根据实际需求对现有设备模型的源代码进行修改,结合有限状态机和库函数使设备模型具有用户自定义的功能。

(3) 强大的数据统计能力。OPNET 提供一些典型的统计量可由用户选取收集。

5.2 仿真拓扑及主要参数设置

仿真采用 US 网络^[40]，拓扑结构如图 5-1 所示。该网络由 14 节点和 21 条双向链路构成。节点的最小度数为 2，最大度数为 4，平均度数为 0.23。图中圆圈内数字为节点编号，链路上数字为链路编号。

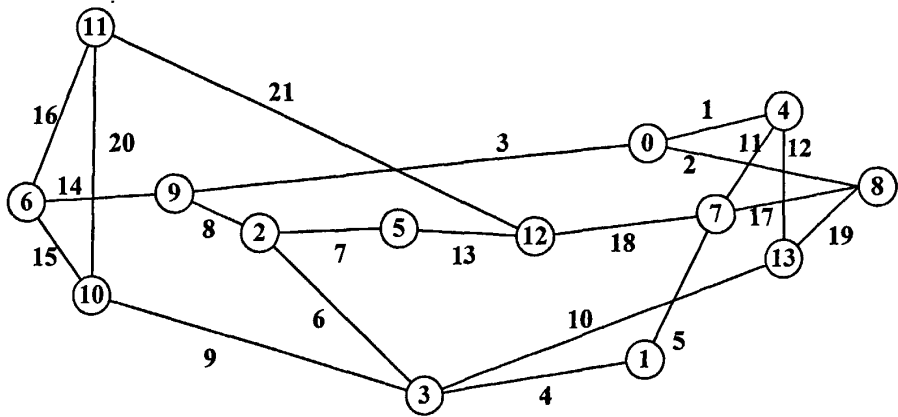


图 5-1 US 网络物理拓扑图

节点间的业务量需求矩阵如表 5-1 所示^[40]。全网所承载的业务量总和为 12102.77Gbit/s，任意节点对间的业务量平均值为 132.997Gbit/s。链路容量设置为 5Tbit/s。

表 5-1 US 网络业务量需求矩阵(Gbit/s)

N \ N	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	.	154. 37	28.7 58	102. 54	135. 41	30.2 04	41.1 08	105. 5	38.9 18	30.1 04	59.6 04	33.9 12	124. 62	86.0 5
1	154. 37	.	54.5 96	213. 5	225. 79	54.6 18	78.5 24	177. 19	65.1 06	57.1 04	114. 59	63.8 9	182. 02	149. 42
2	28.7 58	54.5 96	.	48.2 56	41.9 14	14.2 4	20.1 44	32.1 12	11.8 88	18.8 16	29.9 7	16.1 82	35.4 86	26.6 08
3	102. 54	213. 5	48.2 56	.	149. 31	45.8 06	63.3 82	115. 2	42.4 6	47.6 16	94.3 14	50.6 2	126. 5	95.1 84
4	135. 41	225. 79	41.9 14	149. 31	.	41.1 3	62.0 14	403. 05	208. 66	44.6 76	89.7 92	51.0 46	140. 03	207. 73
5	30.2 04	54.6 18	14.2 4	45.8 06	41.1 3	.	16.8 54	31.5 68	11.6 82	13.1 86	24.6 08	13.8 84	40.9 62	26.2 44
6	41.1 08	78.5 24	20.1 44	63.3 82	62.0 14	16.8 54	.	47.5 82	17.5 24	25.0 5	63.3 5	28.4 7	48.8 88	38.6 9
7	105. 5	177. 19	32.1 12	115. 2	403. 05	31.5 68	47.5 82	.	291. 57	34.2 84	68.9 26	39.0 96	108. 47	240. 14
8	38.9 18	65.1 06	11.8 88	42.4 6	208. 66	11.6 82	17.5 24	291. 57	.	12.6 38	25.3 84	14.4 22	40.0 2	74.5 22
9	30.1 04	57.1 04	18.8 16	47.6 16	44.6 76	13.1 86	25.0 5	34.2 84	12.6 38	.	36.4 36	19.5 46	36.3 3	28.0 84

10	59.6 04	114. 59	29.9 7	94.3 14	89.7 92	24.6 08	63.3 5	68.9 26	25.3 84	36.4 36	.	37.3 86	71.0 48	56.1 46
11	33.9 12	63.8 9	16.1 82	50.6 2	51.0 46	13.8 84	28.4 7	39.0 96	14.4 22	19.5 46	37.3 86	.	40.3 3	31.7 4
12	124. 62	182. 02	35.4 86	126. 5	140. 03	40.9 62	48.8 88	108. 47	40.0 2	36.3 3	71.0 48	40.3 3	.	90.0 4
13	86.0 5	149. 42	26.6 08	95.1 84	207. 73	26.2 44	38.6 9	240. 14	74.5 22	28.0 84	56.1 46	31.7 4	90.0 4	.

节点间物理链路的长度如表 5-2 所示^[40]。链路权重采用“1+链路的物理长度 $\times 10^{-6}$ ”，即设置为整数 1 加上一个正比于传播延迟的小数偏移量。

表 5-2 US 网络链路长度(km)

链路编号	1	2	3	4	5	6	7	8	9	10	11
链路长度	720	942	2820	1350	1008	1746	870	684	2520	2364	438
链路编号	12	13	14	15	16	17	18	19	20	21	
链路长度	468	864	1152	834	1338	540	846	312	2056	3408	

5.3 仿真平台总体框架

在本节所介绍的 MTR 网络仿真平台中，基于 OSPF 路由协议和 MTR 技术，模拟在链路失效场景下路由器转发数据包的过程，用于对比测试前文所研究的算法及方案性能。

5.3.1 仿真方案

该仿真平台包括三种网络设备：链路、节点和中央控制器。节点间、节点与中央控制器之间的链路均为点到点双工。节点负责根据网络中源宿节点间的业务量需求产生数据包，对数据包进行路由转发，并收集统计数据。中央控制器负责控制模拟单、双链路失效场景，以向节点发送报文的方式，通知业务的源节点或失效链路的上游节点开始/停止采用备份拓扑实现业务重路由，用于分别仿真业务路径的全局恢复或局部恢复方案。

链路失效模型设置如下：对于单链路失效，失效情形总数=全网链路总数 $L=21$ 种。双链路失效情形总数 $C_L^2=210$ 种。在仿真时间域内，模拟无链路失效与有链路失效的情形按照等时间间隔交替发生，每次持续时间均为 Δt 秒。每种链路失效情形均出现一次，且不同的失效情形相互独立发生。在仿真中设置统计量，对发生链路失效的时间段内的数据进行采样统计。

使用 C++ 及 AMPL (A Modeling Language for Mathematical Programming) 语言实现第三章中研究的备份拓生成算法, 求解备份拓扑图。按照表 5-1 给出的业务量需求, 根据相应的测试目标, 选择 SPF 或 ECMP 算法计算最小代价路径, 生成与每个备份拓扑相对应的路由表。

以输入文件形式配置以下参数: ①原始拓扑及备份拓扑的路由表; ②备份拓扑与其所保护的链路的编号映射关系; ③失效链路模型, 包括启动/停止恢复操作的节点编号、失效链路编号及失效起止时间; ④转发方案参数, 用于控制调用相关函数, 在对数据包的转发过程中指定选取单次或升序这两种路径在备份拓扑间的切换方案。根据业务量需求计算数据包的产生时间间隔, 作为节点属性中参数预先配置。在仿真中, 为了在相同测试环境下对比转发方案及算法的性能, 设置链路上的传播时延为 0, 调整收发信机的相关参数, 不考虑传输时延。

在仿真过程中收集丢包率、路径跳数、链路负载, 链路利用率等统计量信息, 写入输出文件并生成 Web 报告。

仿真的实验步骤设计如下:

(1) 编码实现节点及中央控制器模型, 搭建网络拓扑。采用业务源进行测试, 分析数据并验证仿真环境的可行性。

(2) 模拟单链路失效场景, 使用 Minimum 算法生成备份拓扑, 从传输路径跳数和链路负载两方面考察 MTR 全局恢复与局部恢复方案的性能, 并将仿真结果同无链路失效以及单链路失效时采用 OSPF 路由重收敛相对比。

(3) 模拟双链路失效场景, 使用 Rich 算法生成备份拓扑, 采用局部恢复方案, 考察单次切换和按拓扑编号升序切换这两种转发方案的丢包率性能。

(4) 模拟单链路失效场景, 使用 Minimum 和 Rich 算法生成备份拓扑, 采用局部恢复方案, 统计在每种失效情形下全网链路负载的峰值与平均值。

(5) 使用 MRPH 和 MLDF 算法生成备份拓扑, 分别采用局部和全局恢复方案。模拟双链路失效场景, 统计在每种失效情形下每个备份拓扑承载的待恢复的业务量; 模拟单链路失效场景, 统计所有失效情形下业务重路由后传输路径的跳数总和。

(6) 模拟单链路失效场景, 使用 Minimum 和 Rich 算法生成备份拓扑, 分别采用局部和全局恢复方案, 考察应用遗传算法优化前后的备份拓扑权重时, 统计在每种失效情形下全网链路负载的峰值与平均值, 并将仿真结果同无链路失效以及单链路失效时采用 OSPF 路由重收敛相对比。

5.3.2 节点模型

节点模型如图 5-2 所示，由数据源模块、转发模块、发射机和接收机组成。数据源用于产生以全网其它节点作为目的地址的数据包，收发信机组负责在节点-节点、节点-中央控制器之间发送和接收数据包。

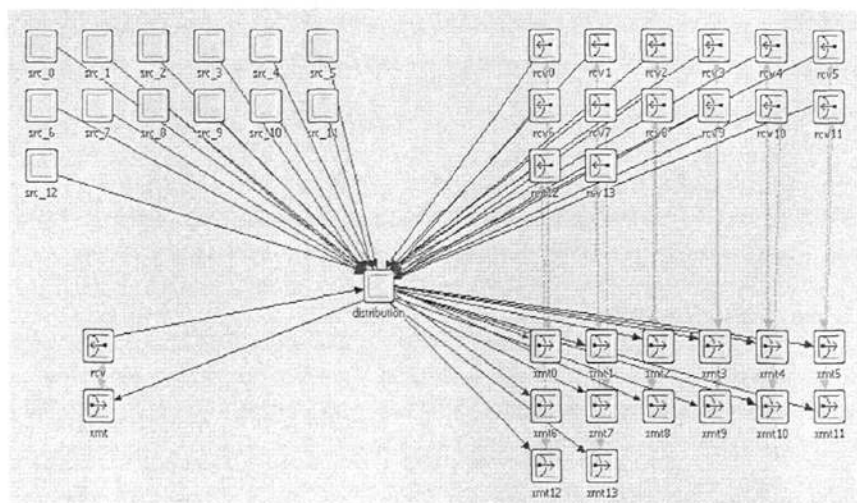


图 5-2 MTR 仿真平台网络节点模型

节点的功能为：

①产生数据包；②接收邻接节点转发的数据包，读取拓扑标识和目的地址，查找对应拓扑的路由表交付给下一跳节点。当数据包的目的地址为本地节点时，记录数据包的传输路径跳数等统计量数据；③接收来自于中央控制器的数据包，实时更新节点邻接链路状态，开始/停止使用备份拓扑重路由。

假定全网节点数为 V ，在仿真所使用的拓扑中，尽管每一节点不与全网其它所有节点相邻，但为了可使度数不同的节点共用相同的节点模型，故在节点模型中配置 V 对收发信机组，其端口号与网络节点编号一一对应。节点-节点的收发信机组名称依次为 rcv0~rcv13 和 xmt0~xmt13，从而使节点可以在数据包转发过程中按照传输路径的下一跳节点编号，通过相同编号的发射机进行转发。此外，使用节点模型中的一对收发信机 rcv 和 xmt 接收来自中央处理器的链路状态通告信息。

下面介绍节点模型中数据源模块和转发模块的状态机设计及其功能。

5.3.2.1 数据源模块

MTR 仿真平台采用泊松源作为产生 IP 数据包的数据源。每个节点内包含 $V-1$

个数据源，用于向全网其它 $V-1$ 个节点发送数据包。数据包的产生时间间隔服从指数分布，其参数根据业务量需求矩阵手动配置。

泊松数据源模块 `src` 的进程模型主要包括三种状态：`init`、`generate` 和 `end`。各状态及相关功能如下。

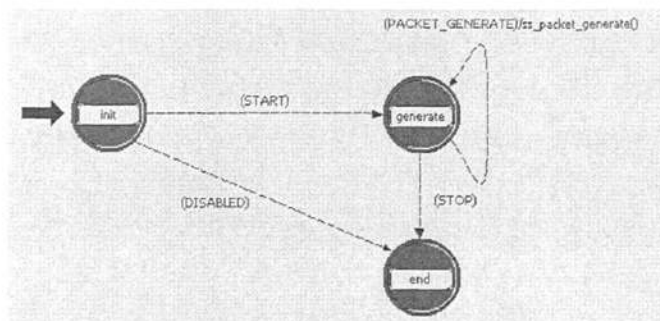


图 5-3 泊松数据源的进程状态机

init 状态：获取本地节点编号以及仿真时间域内发送数据包的起止时间，读取数据包的下列属性参数：产生数据包的间隔时间、数据包长度、数据包格式。获取中断码，进入下一状态。

generate 状态：在每个生成数据包的间隔时间结束时，产生一个自中断 `PACKET_GENERATE`，调用函数 `ss_packet_generate()` 生成数据包。初始化数据包字段中的源节点地址和目的节点地址，并将该数据包传输路径所使用的拓扑的编号初始化为 0。

end 状态：到达仿真时间域内停止发送数据包的时刻后，进入此状态，不作任何调度。

5.3.2.2 转发模块

转发模块是 MTR 仿真平台中网络节点模型的核心模块。该模块的主要功能为初始化路由表、更新链路状态、路由和转发数据包、定时采样统计量等。

转发模块 `distribution` 的进程模型主要包括四种状态：`init`、`idle`、`into_queue` 和 `end`。下面简要介绍各状态及相关功能。

init 状态：仿真开始时初始化路由表，统计量计数器清零，记录当前仿真时刻。根据系统参数定时产生中断以进入下一状态。

idle 状态：负责读取中断码，执行相应函数。当每个发包时刻到达时，产生一个自中断 `SEND_PKT_INTRPT`，执行函数 `send_packet()`，调度队列向下一跳节点

通过发射机转发数据包；仿真运行时每经过一个统计量采样时间间隔，产生中断 STATISTIC_INTRPT，执行函数 write_result()，更新统计量的值；当接收的中断为流中断 PKT_ARRIVE，即有数据包到达，则进入下一状态 into_queue。

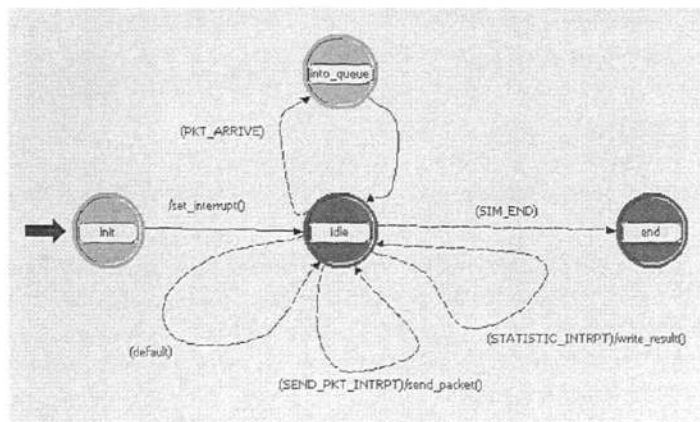


图 5-4 节点核心模块的进程状态机

into_queue 状态：根据接收机数据流线的编号判别数据包来源，执行相应操作。网络节点需处理的数据包有以下三种类型：①自身的源模块产生的数据包；②从邻接节点收到的数据包；③中央控制器发送的链路状态通告报文。

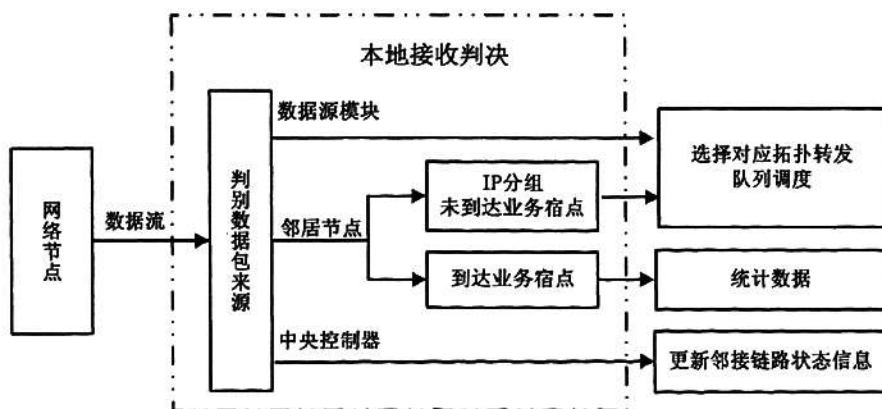


图 5-5 节点核心模块的数据包转发流程

节点根据数据包首部字段解析目的地址、拓扑标识等信息，对数据包的转发判决流程由图 5-5 所示：若待处理的数据包是由自身的源模块产生，或由邻居节点发送的待转发的数据包时，则结合当前邻接链路状态信息以及拓扑标识，查找对应拓扑的转发表进行路由转发，采用第二章介绍的业务恢复方案和路径在拓扑间

的切换方案，查找预配置的与一组备份拓扑相对应的转发表，按照指定模式完成链路失效后业务路径的恢复；当数据包的目的地址为本地节点时，提取数据包所携带的跳数、延迟等统计信息；若接收到中央控制器发来的报文，则更新邻接链路状态，开始/停止将待转发的数据包所携带的拓扑标识更新为备份拓扑编号。

end 状态：仿真停止时进入此状态后，输出全局统计量的值。

5.3.3 中央控制器模型

中央控制器是一个特殊的节点，通过全双工链路与所有网络节点相连。中央控制器不用于数据包的产生及转发，只用于按照输入文件所配置的链路失效模型，以发送数据包的形式将失效链路的状态通告给负责启动/停止恢复操作的节点。网络节点解析链路失效信息后调整路由过程，控制在相应时刻开始或停止按照备份拓扑路由表中的下一跳转发数据包。中央控制器的节点模型如图 5-6 所示，包含 V 对收发信机组，端口号与网络节点编号一一对应。

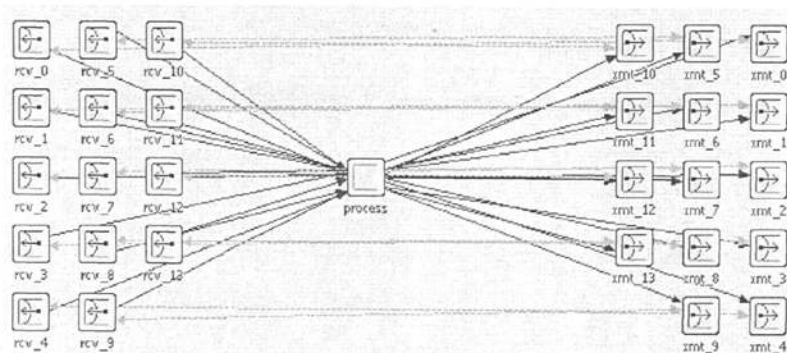


图 5-6 中央控制器的节点模型

图 5-7 为中央控制器节点模型中模块 process 的进程状态图。

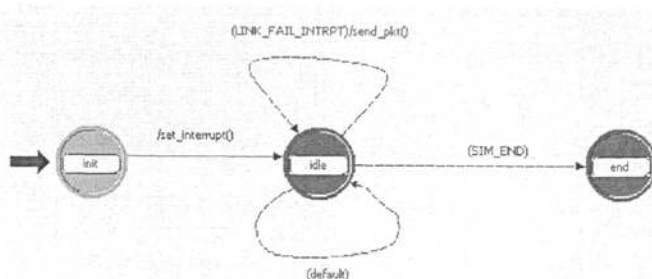


图 5-7 中央控制器的进程状态机

本次仿真设置模拟失效按照等间隔时间发生。此外，也可以调用 OPNET 提供的库函数 `op_dist_load()`，通过设置不同参数，控制产生一系列服从一定概率分布函数的时间间隔，作为自中断 `LINK_FALL_INTRPT` 的发生时刻。当发生链路失效的时刻到达时，执行函数 `send_pkt` 向负责开始/停止启动恢复操作的网络节点发送链路状态报文，通告当前失效链路的编号以及失效的起止时刻。

5.4 本章小结

本章介绍了使用 OPNET 工具搭建的仿真平台。此仿真平台可模拟在单链路或多链路失效场景下，将 MTR 技术应用于失效后重路由时数据包的路由转发过程，用来对第二、三、四章研究的 MTR 相关技术方案以及算法的性能进行定量评估。

首先简要介绍了 OPNET 工具，然后给出仿真所使用的拓扑结构、业务量需求等参数，接着阐述仿真方案及平台的总体框架，分别介绍了各个组成部分的设计思路。

第六章 全文总结

6.1 本文工作总结

IP 网络采用的 OSPF 和 IS-IS 协议可在网络设备发生失效后通过链路状态通告来更新路由器所维护的网络拓扑信息。然而路由重收敛的时间对于一些实时应用而言仍然过长,同时其收敛过程也影响了网络的稳定性。因此为了应对 IP 网络中设备的短时失效,可以将 IETF 提出的多拓扑路由(MTR)技术应用于业务路径的快速恢复。本文在国内外现有研究成果的基础上,对于实现 MTR 技术方案的多个环节进行了研究。完成的工作主要包括:

(1) 对 MTR 基本原理及技术规范的研究与验证。

采用真实网络拓扑结构与业务需求模型,搭建仿真平台,模拟单链路和双链路失效场景,针对业务恢复的局部和全局模式以及路径切换拓扑的单次或升序这两种方案进行动态仿真,对其性能进行了对比及分析。第二章中的仿真结果表明,在大多数失效场景下全局恢复同局部恢复方案相比,在减小路径跳数和降低链路负载方面具有更好性能。此外,通过允许传输路径按照拓扑编号升序切换拓扑以及增加备份拓扑个数,可以降低丢包率。

(2) 多拓扑生成算法的评估与改进。

在第三章中实现了 3 种用于生成备份拓扑的算法,并对第三种算法进行改进,针对不同指标评估所生成的拓扑用于失效后重路由的性能。实验证明通过增加备份拓扑的个数,可以减少业务传输路径的跳数,但当备份拓扑个数增加到一定值之后,传输路径总跳数再无改善。此外,提出了一种新的拓扑生成方案,可分别针对全局或全局恢复方案,生成指定个数的备份拓扑,对于所有备份拓扑可独立保护的双链路失效情形,可以降低在此类失效场景下备份拓扑所需承载的待恢复业务量的最大值。由统计数据可以看出该算法所求得的备份拓扑对于所针对的优化目标具有良好的性能。

(3) 备份拓扑的权重优化。

第四章进一步研究在 MTR 技术应用于失效恢复后缓解链路拥塞这一问题上的解决方案。建立求解一组拓扑权重的 ILP 模型,并将遗传算法应用于备份拓扑的权重设计。使得采用优化的备份拓扑链路权重后,在实现业务路径快速恢复的同

时可有效降低所有失效场景下全网链路的最大利用率。为了进一步衡量遗传算法优化权重后目标值与最优解的差距,将其与“ K 路由间按照任意比例分流”的方案性能进行对比。

6.2 下一步研究方向

对于 MTR 技术应用于网络设备失效后的业务路径恢复,其实现细节可以进一步改进,应用领域也有待于继续探索。总的来说,还可以进行下列工作:

(1) 本文主要对 MTR 技术应用于 IP 网络链路失效后的快速恢复进行研究。下一步可以将实现节点保护的备份拓扑列为研究对象,以光网络等作为新的研究领域,对 MTR 技术的应用范围进行拓展研究。

(2) 文中对于备份拓扑的生成以及备份拓扑权重的优化设计采用两阶段分别求解。可以设计新的算法,在求解拓扑构造的同时联合优化权重,根据 QoS 需求定制完整的 MTR 路由方案。

(3) 本文的研究和仿真均基于已知网络业务量需求这一前提。对于在未知业务量情形下 MTR 相关解决方案的性能稳定性与适应度有待进一步研究。

由于作者水平有限,本文很多工作还有待完善,恳请审阅本文的各位专家不吝指正。

致 谢

本论文的选题和研究工作以及论文的撰写都是在我的导师王晟教授的悉心指导下完成的。王晟教授严谨的治学态度、扎实的理论知识和孜孜不倦的钻研精神都令我佩服和敬仰，是我学习的榜样。感谢老师三年里在学习方面给予我的悉心引导与启迪，感谢老师在我研究中遇到瓶颈时给予的宝贵指点和帮助。在此，我以最诚挚的敬意向我的导师王晟教授表示最由衷的感谢。

衷心感谢教研室的李乐民院士、徐世中老师、虞红芳老师、许都老师、王雄老师、廖丹老师和章小宁老师，在平时的研究工作中都给予我详细指导与热心帮助，使我受益匪浅。感谢老师们为我们建立了研讨班，并每周不辞辛劳地往返于校区之间为我们指导和解惑。

感谢师兄师姐耐心地传授给我很多学习经验。感谢所有同学在学习与生活中对我的热心鼓励和帮助，和你们的交流与协作使我受益良多。

感谢评阅本文的各位专家，是你们帮助我再次审视并修正自己，感谢你们抽出宝贵的时间阅读本文并提出宝贵的意见和建议。

在这里还要特别感谢我的父母，谢谢你们为我的成长所付出的一切，我将珍视并回报你们对我的养育之恩，能够拥有这样的父母是我一生的幸运。

最后要感谢我的母校，感谢所有老师对我的教导。

大学的历程因有你们而变得快乐充实。谢谢你们，与你们相处的点点滴滴，都是对我最为弥足珍贵的馈赠。

借此送上我衷心的祝福，愿所有人永远健康、幸福！

参考文献

- [1] C. Boutremans, G. Iannaccone and C. Diot. Impact of link failures on VoIP performance. Proc. of NOSSDAV Workshop, Miami (May 2002) (ACM Press, New York, 2002).
- [2] A.Markopoulou, G.Iannaccone, S.Bhattacharyya, C.-N.Chuah, and C. Diot. Characterization of failures in an IP backbone network.. Proceedings INFOCOM, Mar. 2004.
- [3] G. Iannaccone, C.-N. Chuah, R. Mortier, et al. Analysis of link failures in an IP backbone. 2nd ACM SIGCOMM Workshop on Internet Measurement, Nov. 2002, pp. 237 - 242.
- [4] James P.G. Sterbenz, David Hutchison, et al. Resilience and Survivability in Communication Networks:Strategies,Principles,and Survey of Disciplines. Computer Networks: Special Issue on Resilient and Survivable Networks (COMNET), Elsevier, 2010.
- [5] G.Iannaccone, C.Chuah, S.Bhattacharyya, and C.Diot. Feasibility of IP Restoration in a Tier-1 Backbone. IEEE Networks, Special Issue, March 2004.
- [6] J.T.Moy. OSPF Version 2. IETF RFC 2328, April 1998.
- [7] A. Basu and J. G. Riecke. Stability issues in OSPF routing. Proceedings of SIGCOMM, San Diego, California, USA, Aug. 2001, pp. 225–236.
- [8] P. Francois, C. Filsfils, et al. Achieving subsecond IGP convergence in large IP networks. ACM SIGCOMM Computer Communication Review, vol. 35, no. 2, pp. 35–44, July 2005.
- [9] C.Alaettinoglu, V.Jacobson, H.Yu. Towards Millisecond IGP Convergence. IETF Draft, 2000.
- [10] S. Rai, B. Mukherjee, and O. Deshpande. IP resilience within an autonomous system: Current approaches, challenges, and future directions. IEEE Communications Magazine, vol. 43, no. 10, pp. 142–149, Oct. 2005.
- [11] G. Schollmeier, J. Charzinski, A. Kirstädter, C. Reichert, et al. Improving the resilience in IP networks. Proceedings of HPSR, Torino, Italy, June 2003, pp. 91–96.
- [12] C. Reichert, Y. Glickmann, and T. Magedanz. Two routing algorithms for failure protection in IP networks. Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC), June 2005, pp. 97–102.

- [13] S.Nelakuditi, S.Lee, Y.Yu, et al. Failure insensitive routing for ensuring service availability. IWQoS'03 Lecture Notes in Computer Science 2707, June 2003.
- [14] Z.Zhong, S.Nelakuditi, Y.Yu, S.Lee, et al. Failure inferencing based fast rerouting for handling transient link and node failures. Proceedings of IEEE Global Internet, Mar. 2005.
- [15] J. Wang, Z. Zhong, and S. Nelakuditi. Handling multiple network failures through interface specific forwarding. Proceedings GLOBECOM, Nov. 2006.
- [16] S. Bryant, M. Shand, and S. Previdi. IP fast reroute using not-via addresses. Internet Draft (work in progress), Oct. 2006, draft-bryantshand-IPFRR-notvia-addresses-03.txt.
- [17] P. Psenak, S. Mirtorabi, A. Roy, L. Nguen, and P. Pillay-Esnault. MT-OSPF: Multi-topology (MT) routing in OSPF. IETF RFC4915, Jun. 2007.
- [18] T. Przygienda, N. Shen, and N. Sheth. M-ISIS: Multi-topology (MT) routing in IS-IS. IETF RFC5120, Feb. 2008.
- [19] Cisco Systems, Multi-Topology Routing, Feb. 2007.
- [20] Yu Guo, Ning Wang, Kin-Hon Ho, et al. Achieving Fast BGP Reroute with Traffic Engineering Using Multiple Routing Planes. Springer Berlin. Volume 5275/2008, Sep. 25, 2008.
- [21] Amund Kvalbein, Olav Lysne. How can Multi-Topology Routing be used for Intradomain Traffic Engineering. INM'07, August 27–31, 2007, Kyoto, Japan.
- [22] Amund Kvalbein, A.F.Hansen, T.Cicic, et al. Fast recovery from link failures using Resilient Routing Layers. ISCC. Pages 554-560. Cartagena, Spain, Jun. 27-30, 2005.
- [23] Amund Kvalbein, A.F.Hansen, T.Cicic, et al. Fast IP Network Recovery using Multiple Routing Configurations. INFOCOM. Barcelona, Spain, Apr. 23-29, 2006.
- [24] T.Cicic, A.F.Hansen, S.Gjessing, O.Lysne. Applicability of Resilient Routing Layers for k-Fault Network Recovery. International Conference on Networking (ICN), Reunion Island, 2005.
- [25] Amund Kvalbein, Olav Lysne. Robust Load Balancing using Multi-Topology Routing. ACM SIGCOMM Workshop on Internet Network Management, Kyoto, Japan, Aug. 27-31, 2007.
- [26] A.F.Hansen, O.Lysne, T.Cicic, and S.Gjessing. Fast proactive recovery from concurrent failures. Proceedings 42nd IEEE International Conference on Communications (ICC 2007), Glasgow, UK, June 2007.

-
- [27] M. Scheffel, C. Gruber, T. Schwabe, and R. Prinz. Optimal multi-topology routing for IP resilience. *AEU International Journal of Electronics and Communications*, 2006.
- [28] Srisuresh, P. and P. Joseph. OSPF-xTE: Experimental Extension to OSPF for Traffic Engineering. IETF RFC 4973, July 2007.
- [29] C. Hopps. Analysis of an equal-cost multi-path algorithm. IETF RFC 2992, 2000.
- [30] D. Thaler and C. Hopps. Multipath Issues in Unicast and Multicast Next-Hop Selection. IETF RFC 2991, 2000.
- [31] B. Fortz and M. Thorup, Internet traffic engineering by optimizing OSPF weights. *Proceedings INFOCOM*, 2000, pp. 519–528.
- [32] A.Nucci, B.Schroeder, S.Bhattacharyya, et al. IGP Link Weight Assignment for Transient Link Failures. 18th International Teletraffic Congress, Berlin, Germany, Aug. 2003.
- [33] B. Fortz and M. Thorup. Robust optimization of OSPF/IS-IS weights. *INOC*, Oct. 2003, pp. 225–230.
- [34] M.Dzida, M.Zagozdzon, M.Piuro, A.Tomaszewski. Optimization of the shortest-path routing with equal-cost multi-path load balancing. *Proceedings of the 2006 International Conference on Transparent Optical Networks*, 2006, pp. 9–12.
- [35] Y. Wang, Z. Wang, L. Zhang. Internet traffic engineering without full mesh overlaying. *Proceedings of IEEE INFOCOM 2001*, Anchorage, Alaska, 2001.
- [36] M. Piuro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publishers, 2004.
- [37] 邢文训,谢金星.现代优化计算方法.清华大学出版社,2007年7月第3次印刷,p16~28.
- [38] 陈国良.遗传算法及其应用.人民邮电出版社,2001年2月北京第3次印刷,p56~60.
- [39] M.Ericsson, M.G.C.Resende, P.M.Pardalos. A Genetic Algorithm for the Weight Setting Problem in OSPF Routing. *Journal of Combinatorial Optimization*. volume 6,Number 3,Sep,2009,p299~333.
- [40] A.Betker, C.Gerlach, R.Hulsermann, M.Jager, et al. Reference transport network scenarios. MultiTeraNet Project, Tech. Rep., 2004.

个人简历

贾冰，女，生于 1983 年，籍贯黑龙江。

[1]. 2002.09 - 2006.06，就读于电子科技大学通信与信息工程学院，获工学学士学位，专业为通信工程（无线电技术）。

[2]. 2007.09 - 至今，就读于电子科技大学通信与信息工程学院，攻读通信与信息系统硕士学位。

攻硕期间取得的研究成果

参加的科研项目：

国家基础课题研究计划 973 项目（2007CB307104 of 2007CB307100）：新网络体系下的可信与移动问题（一体化可信网络与普适服务体系基础研究）。

发表、录用的论文：

[1] 贾冰，王晟，张敏. 数据会议系统中的透明电子白板应用. 计算机应用, 2009. 8.

[2] 贾冰，王晟. 应用于 IP 网络故障恢复的多拓扑生成算法研究. 信息通信, 2010. 6.