中文摘要

摘要：IS-IS 协议是一种基于链路状态的内部网关协议，由于其机制比较简单，常被应用于城域网和大型网络服务提供商。IS-IS 协议扩展支持 IPv6 后存在路径不区分 IPv4 和 IPv6 的问题，可能造成路由黑洞。IS-IS 支持多拓扑功能通过将 IPv4 和 IPv6 分拓扑计算解决这一问题。本文工作一方面是参与 IS-IS 支持多拓扑功能项目的开发，独立设计并实现了链路状态信息更新模块（八项子功能）支持多拓扑和配置/取消多拓扑功能，并利用专门设计的组网和命令配置验证了功能实现；另一方面以 IS-IS 支持多拓扑功能为基础进行了两项扩展研究：

1. 在的大型 IS-IS 路由域中如果存在 IPv6 孤岛，应用隧道技术和无状态 IP、ICMP 翻译技术实现孤岛间通信的最优路径选择问题。本文提出 IPv4 和 IPv6 拓扑组合选路方案以及新的选路标准，更能适应 IPv4/IPv6 混合的组网环境。

2. 提出一种利用多拓扑实现静态流量工程的方案。流量工程（TE）能够合理控制流量转发路径保证链路利用率，避免因链路拥塞造成传输质量降低。静态流量工程不能适应流量不断变化的网络环境，但它能够从全局角度，同时考虑每条链路限制和每条源-目的流的要求，实现网络性能的整体优化。本文提出一种多拓扑静态流量工程方案：基于链路负载自动生成拓扑，以贪婪算法思想分配流量，拓扑生成与流量分配交替进行直到达到预设的链路利用率。利用 MATLAB 针对不同特性的随机连通网络和随机数据流进行仿真研究，证明了本方案可以有效地实现流量工程。

关键词：IS-IS；多拓扑路由；IPv6 孤岛；最短路径；静态流量工程

分类号：TP393

ABSTRACT

ABSTRACT: IS-IS is a link-state based interior gateway protocol; it is often used in metropolitan and large Internet service provider for its relatively simple mechanism. IS-IS link does not distinguish from IPv4 to IPv6 after its supporting of IPv6, which may result in routing black hole. IS-IS MTR (Multi-topology Routing) solves this problem through decision in each topology. In this paper, one part of work is to participate in the project of IS-IS MTR development, design and implementation the MT extention in update modular (8 functions) and the function of configuring and unconfiguring MT, testify the realization by devised network and commands. Another part is two extended research on the basis of IS-IS MTR:

1. If IPv6 Island exists in large scale IS-IS routing domain, analyze how to attain optimum path when tunnels and stateless IP ICMP translation are applied to help communication among islands. We propose solutions of joint topology with IPv4 and IPv6 and new merit, which are more suitable for IPv4/IPv6 hybrid network.
2. Realize static TE (traffic engineering) with MT. TE adjusts the forwarding path of flow to guarantee high level link utilization and avoid congestion. Static TE is not fit for the dynamic traffic, however, it takes consideration of all links and all flows simultaneously and globally, which is more effective in elevating Qos. We propose a solution of static TE based on MT: topology is generated adaptively founded on link load, traffic distributes in greedy way, topology generation and traffic distribution is performed alternately until anticipated link utilization is achieved. The solution is testified to be an effective TE by MATLAB simulation in random networks of different characteristics.

KEYWORDS: IS-IS; Multi-topology Routing; IPv6 islands; Shortest path; Static Traffic Engineering

CLASSNO: **TP393**

致谢

本论文的工作是在我的导师郭宇春教授的悉心指导下完成的，郭宇春教授严谨的治学态度和科学的工作方法给了我极大的帮助和影响。在此衷心感谢三年来郭宇春老师对我的关心和指导。

陈常嘉教授悉心指导我们完成了实验室的科研工作，在学习上和生活上都给予了我很大的关心和帮助，在此向陈常嘉老师表示衷心的谢意。

胡师舜老师，赵永祥副教授对于我的科研工作和论文都提出了许多的宝贵意见，在此表示衷心的感谢。

在实验室工作及撰写论文期间，赵锐敏、颜志杰等同学对我论文中的仿真分析研究工作给予了热情帮助，在此向他们表达我的感激之情。

另外也感谢我的家人，他们的理解和支持使我能够在学校专心完成我的学业。

1 引言

1.1 研究背景

Intermediate System to Intermediate System（IS-IS）协议是一种广泛应用于城域网和大型网络服务提供商的内部网关协议（IGP）。内部网关协议是指应用于自治系统内部的路由协议，常见的内部网关协议包括RIP，OSPF和IS-IS。其中OSPF和IS-IS属于链路状态路由协议，即网络中的每个路由节点了解整个网络的路由信息，节点根据网络拓扑独立计算到达网络中其它节点的最短路径。IS-IS通过Dijkstra算法计算最短路径，可以避免路由环路。

随着Internet以指数级速度增长，越来越多的用户和服务器不断扩充现有网络规模，造成了路由表持续增大，路由器的处理速度不断降低；再加上多媒体网络业务的发展，用户数据业务流量的不断增长，路由器需要处理的数据包越来越多，这使得优化路由性能变得十分重要。现代网络技术的发展对IS-IS路由协议进行了诸多扩充，IS-IS的发展如下：

1990.02，RFC1142，OSI模式下的IS-IS协议。

1990.12，RFC 1195[1]，IS-IS协议适用于TCP/IP网络。

1995，IS-IS协议进入商用。

2000.01，IS-IS支持IPv6互联网草案提出

2002.11，ISO/IEC 10589[2]，面向无连接网络模式的标准IS-IS协议。

2004.06，RFC3784，IS-IS支持流量工程（TE）。

2008.02，RFC5120[3]，IS-IS支持多拓扑路由（MTR）。

2008.10，RFC5305[4]，新的IS-IS流量工程标准。

2008.10，RFC5306[5]，IS-IS支持平滑重启（GR）。

2008.10，RFC5308[6]，IS-IS支持IPv6。

2009.02，RFC5311[7]，IS-IS支持虚拟系统。

1.2 研究意义

为满足对下一代网络的支持，IS-IS 协议扩展支持 IPv6，但支持 IPv6 后存在的问题是：在自治系统中如果存在 IPv4 和 IPv6 混合组网，则 IPv4 和 IPv6 网络拓扑必须相同，即每条链路都支持 IPv4 和 IPv6 双协议。这一限制条件源于 IS-IS 协议

支持 IPv6 技术将路由器的 IPv4 和 IPv6 邻居作为统一的邻居信息进行同步，在此基础上算出的最短路径无法区分 IPv4 和 IPv6。但是在网络中，可能并非所有的路由器都支持 IPv4 和 IPv6 双协议栈，或者一些链路不支持 IPv6，因此 IPv6 和 IPv4 拓扑无法完全一致。此时对于双协议栈路由器是感知不到拓扑中有哪些路由器是不支持 IPv6 的，IPv6 数据流仍会被转发到这些不支持 IPv6 的路由器或链路，从而被丢弃处理，产生路由黑洞。

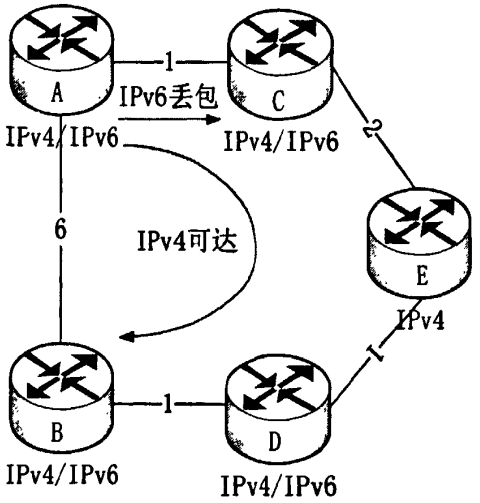

图 1.1 IPv4、IPv6 混合组网时的 IS-IS 转发路径

Figure1.1 Forwarding in IPv4/IPv6 hybrid IS-IS domain

图 1.1 表示在 IPv4、IPv6 混合组网且 IPv4、IPv6 拓扑不相同时的 IS-IS 路由转发路径。链路中的数字代表相应链路开销。Router A、B、C 和 D 均支持 IPv4 和 IPv6，Router E 只支持 IPv4。从 Router A 到 Router B 的最短路径为 A->C->E->D，最小开销为 5。由于 Router E 不支持 IPv6，从 A 到 B 的 IPv6 报文被 Router C 丢弃。

IS-IS MTR 扩展功能能够解决 IS-IS 链路不区分协议造成的路由黑洞问题，将支持 IPv4 的路由和支持 IPv6 的路由划入两个不同拓扑，IPv4 和 IPv6 报文分别根据相应拓扑的路由表转发。

图 1.2 表示 IS-IS 路由配置了 MTR 后的转发路径。IPv4、IPv6 独立计算最短路径，IPv4 拓扑从 Router A 到 Router B 的最短路径仍为 A->C->E->D，IPv6 拓扑的最短路径为 A->B，最小开销为 6。从 A 到 B 的 IPv6 报文不再被丢弃。

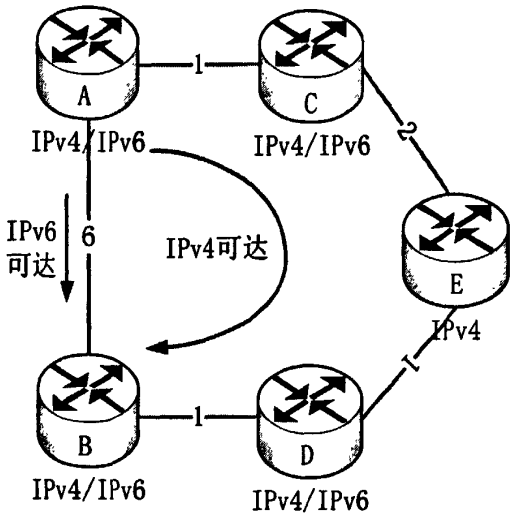

图 1.2 IPv4、IPv6 分拓扑情况下的 IS-IS 转发路径

Figure1.2 Forwarding when IPv4/IPv6 belonging to separate MT

多拓扑功能可以用于优化网络性能。多拓扑功能提供了新的网络资源-拓扑，每个拓扑是整个网络拓扑的子集，不同拓扑的最短路径存在差异，因此在不同拓扑中针对同一目的地址转发路径也不相同，报文转发可以通过选择拓扑确定路径。拓扑资源的有效利用有助于提升路由转发的有效性和可靠性。

1.3 应用与研究现状

IS-IS MTR 扩展功能被应用于 IP 路由转发和一些非转发功能，详见以下拓扑使用规定：

- MT ID #0: 标准拓扑
- MT ID #1: IPv4 网管拓扑
- MT ID #2: IPv6 路由拓扑
- MT ID #3: IPv4 多播路由拓扑
- MT ID #4: IPv6 多播路由拓扑
- MT ID #5: IPv6 网管拓扑
- MT ID #6 - #4095: 用户使用拓扑

MT ID #0 #2 用于区分 IPv4 和 IPv6 协议。MT ID #1 #5 用于网管或组管协议，网管拓扑包含所有需要被管理的路由器。网管路由器分别以不同的地址加入网管拓扑和标准拓扑，这样网管拓扑可以直接利用标准拓扑的最短路径，当路由器配置网管功能后不用再切换到网管拓扑。一些路由器虽然不参与报文转发，但出于网管需求也要加入标准拓扑。MT ID #3 #4 用于多播路由，多播组中的所有路由器

加入同一拓扑，路由器利用此拓扑的路由表进行多播，但多播反向链路的选择却可以利用其它拓扑的路由表。MT ID #6 - #4095 根据用户需求手动分配，例如公司内网用户加入同一拓扑，流量通过内网转发；某种业务用户加入同一拓扑，业务流量在拓扑内转发。

流量工程是 IP 网络的流量控制机制，通过调节流量选路降低网络过载程度。Amund Kvalbein 和 Olav Lysne[9]利用多拓扑实现了在线 TE 和离线 TE。离线 TE 基于静态流量数据，通过最优化链路开销最小化流量转发时的网络过载程度，但其无法有效处理动态的流量变化，每次流量变化都需要重新计算。一些减轻流量变化造成影响的方案已被提出，例如试图找到在链路故障情况下依然有较好负载均衡性能的链路开销设置[10][11][12]，或者提出忘却性路由，即在所有可能的流量情况下在一定范围内都有良好性能[13][14]。离线 TE 的缺陷导致了在线 TE 的产生[15][16]。这些方案都是建立在源节点和目的节点之间存在的多条可选路径的前提下，而这前提正是多拓扑功能能够提供的。

MT 还可以用于保证网络质量，Tarik CiciC[17]提出的 FT-MTR (容错多拓扑路由)是指构建逻辑拓扑作为网络备份，使逻辑拓扑中的某些网络资源不用作报文转发，重传报文选择合适的逻辑拓扑转发，提高重传成功率。FT-MTR 找寻一系列保证网络连通的关键节点，以这些节点为基础进行网络分层并以层级结构作为逻辑拓扑，解决了如何减少逻辑拓扑的数量和出现链路故障后的负载分配的问题。

1.4 论文结构安排

本文作者参与了 IS-IS 支持多拓扑功能项目的开发。功能提供了 IS-IS 路由域内拓扑的手动配置，每个拓扑相当于一个路由域，需要保证路由功能在拓扑内正常执行。项目分为邻居模块，更新模块，路由计算模块和路由扩展功能模块。本文独立设计实现了更新模块支持多拓扑和配置/取消多拓扑功能，编写代码三千行，并设计了专门的组网和命令配置，通过观察路由器的链路状态信息库和路由表的变化验证二部分功能的实现。

更新模块支持多拓扑功能实现的子功能主要包括：多拓扑信息的存储结构设计；收到 LSP 后四种新增多拓扑 TLV 的更新；多拓扑扩展后伪节点报文的更新；节点脱离/进入孤立状态后相应多拓扑路由的更新；LSP 报文失效时多拓扑信息的清除；自身多拓扑链路状态信息变化时多拓扑 TLV 的封装；路由器 OVERLOAD 状态的多拓扑维护；区域地址分拓扑维护和多拓扑 ATT 状态判断。配置/取消多拓扑功能的实现涉及配置的多拓扑无法发布时的处理和各模块响应多拓扑添加删除的处理。

在此基础上本文进行了两项多拓扑选路扩展研究。一是如果 IS-IS 路由域中存在 IPv6 孤岛，应用隧道技术和无状态 IP、ICMP 翻译技术实现孤岛间通信的最优路径选择问题，提出变更选路标准和利用分开的 IPv4 和 IPv6 拓扑组合选路两种方案。二是利用多拓扑能够提供到达同一目的地址的多条路径的特点实现静态流量工程。本文方案的主体思想是：多拓扑生成与流量输入相关，流量在拓扑间分配以降低网络总过载量为目标，生成新拓扑与流量分配交替进行，直到达到预设的链路利用率。最后通过 MATLAB 仿真验证了本方案在不同规模和密集程度随机网络环境下均能良好地实现静态流量工程效果。

本文第二章介绍了 IS-IS 路由协议的基本架构，报文信息和运行机制，以及 MTR 扩展功能的协议要求。第三章描述了 IS-IS MTR 的实现，详述了更新模块和配置/取消多拓扑功能的设计与实现。第四章验证 IS-IS 支持多拓扑功能的实现，主要面向第三章设计的功能。第五章描述多拓扑扩展研究，提出 IPv6‘孤岛’间通信的最短路径选择方案；提出利用多拓扑实现静态流量工程的方案并仿真验证。

2 IS-IS 与 MTR 介绍

2.1 IS-IS 协议

2.1.1 基本概念

IS（intermediate system）：独立进行路由功能的实体，常指一台路由器或路由器的一个 IS-IS 进程。

邻居（neighbor）：两台路由器的报文通过一条链路能够到达对方。

区域（area）：内部路由信息交互并拥有到达外部的出口的路由子域。

LEVEL 1&2：区域的层次，L2 为主干区域，L1 为局部区域。

指定路由器（designated IS）:广播网中选出的执行特殊功能的路由器，负责以广播网中心的名义生成 LSP。

伪节点（pseudonode）：DIS 生成的广播网中的虚拟节点，包含与广播网中的节点有邻居关系。

System ID：每个 IS 的唯一标记。

IIH：IS 到 IS 间 Hello 报文。

LSP：链路状态协议数据单元。

PSNP：部分序列号数据报文。

CSPN：完全序列号数据报文。

LSDB：链路状态数据库。

2.1.2 网络架构

为了支持大型路由域，IS-IS 以分层的形式组织域内路由器。路由域被分割成多个区域，每个 IS 只能属于一个区域。仅在一个区域之内进行的路由活动被称作 L1 路由活动。能够在区域之间进行的路由活动被称为 L2 路由活动。根据其在路由活动中所起到的作用路由域中的 IS 可以划分为如下三类：

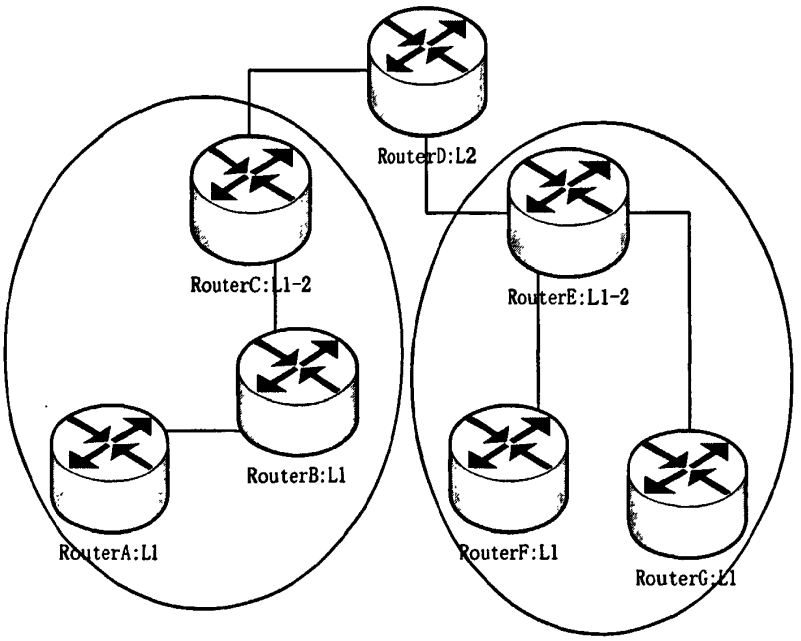

图 2.1 路由域中的 IS 分类

Figure2.1 The style of IS

L1 IS：只能和同一区域内的 IS 进行 L1 路由活动的 IS，使用 L1 LSP 来传递自身的链路状态信息。如果接收到报文的目的地址在本区域内，就直接按地址转发；如果报文的目的地址在本区域外，则将报文转发给离自身最近的一个 L1-2 IS。L1 IS 只能和 L1 IS 相邻，连续的 L1 IS 组成 L1 域。

L2 IS：能够和不同区域的 IS 进行 L2 路由活动的 IS，使用 L2 LSP 来传递自身的链路状态信息，只负责域间路由。L2 IS 只能和 L2 IS 相邻，且所有的 L2 IS 必须是相邻的，共同构成网络的 L2 域（主干域）。

L1-2 IS：既能进行 L1 路由活动又能进行 L2 路由活动的 IS，同时具有 L1 IS 和 L2 IS 双重身份。L1-2 IS 通过 L1 路由活动汇总本区域的 L1 链路状态信息，并通过 L2 LSP 告知 L2 域中的其它 IS。L1-2 IS 发布带有 ATT 标记的 L1 LSP，指导 L1 路由器将通往区域外的流量转发到自身。

IS-IS 协议只支持两种网络类型：广播网和点到点网络，二者采用不同 IIH 建立邻居关系。LSP 报文通过泛洪的方式使整个 LEVEL 内的路由器保持 LSDB 的同步。泛洪（flood）是指当一个路由器向相邻所有路由器报告自己的报文后，相邻路由器再将同样的报文传送到自己相邻的所有路由器，收到过此报文的路由器直接丢弃报文，如此逐级传递报文直至覆盖整个 LEVEL。在广播网中，每台路由器都和网络中其余路由器建立邻居关系，网络逻辑结构是全连通型的，LSP 泛洪将产生巨大的网络流量，且随着广播网的增大，网络流量迅速增加。在星形结构的网络拓扑中，由于节点间的链路很少，节点泛洪 LSP 产生的网络流量很小，因此

IS-IS 在广播网中选出一个指定路由器（DIS），并由 DIS 产生一个伪节点作为网络中心节点与每个路由器相连，使网络逻辑拓扑变为星形。

不同 LEVEL 的广播网中有不同的 DIS，选举优先级最高的路由器当选广播网中的 DIS，如果优先级一样，具有最大的 MAC 地址的路由器将当选 DIS。DIS 的选举是抢占式的，且不存在备份 DIS，当 DIS 与广播网中其它路由器的邻居 Down 或者广播网中出现比 DIS 更高优先级的路由器时，发生 DIS 变更。DIS 发送 IIH 报文的频率是普通路由器的 3 倍，这样可以保证 DIS 失效可以被快速检测到。

2.1.3 协议报文

IS-IS 协议通过九种 PDU（数据链路单元）的传递完成路由功能：L1/L2 LSP 用于传递链路状态信息；L1/L2 CSNP，L1/L2 PSNP 用于路由域中链路状态信息的同步；L1/L2 LAN IIH，Point-to-point IIH 用于 IS 之间建立邻居。

LSP 报文头部如表 2.1 所示。

表 2.1 LSP 报文头部

TABLE2.2 LSP header

<table>
<tr><td colspan="4"></td><td>No. of Octets</td><td></td></tr>
<tr><td colspan="4">Maximum Area Addresses</td><td>1</td><td>每个 IS 最多配置的区域地址数，默认 3</td></tr>
<tr><td colspan="4">PDU Length</td><td>2</td><td>PDU 总长度</td></tr>
<tr><td colspan="4">Remaining Lifetime</td><td>2</td><td>剩余生存时间</td></tr>
<tr><td colspan="4">LSP ID</td><td>ID Length+2</td><td></td></tr>
<tr><td colspan="4">Sequence Number</td><td>4</td><td>序列号</td></tr>
<tr><td colspan="4">Checksum</td><td>2</td><td>校验</td></tr>
<tr><td>P</td><td>ATT</td><td>LSPDBOL</td><td>ISTYPE</td><td>1</td><td>标记位</td></tr>
</table>

其中 LSP ID = System ID + Pseudonode ID （伪节点标记）+ LSP Number（分片号）。

剩余生存时间：用来计算报文失效的时间，LSP 的最大生存时间为 20 分钟，但每隔 15 分钟 LSP 源路由就会重新生成此 LSP，保证正常情况下报文不会超时；剩余时间为 0 的报文是清除报文，收到后找到并清除本地 LSDB 中对应的 LSP 条目。

序列号：表示 LSP 的新旧，每个 LSP 分片的序列号单独进行计算：路由器启动 LSP 分片初次发送时序列号为 1，以后需要重新生成此分片时，新分片序列号为旧分片加 1，序列号更高意味着分片更新。当某分片的序列号达到 0xFFFFFFFF

时，路由器会暂停 IS-IS 进程，等待整个路由域中和此路由器生成的 LSP 全部超时并删除完毕，随后 IS-IS 进程重启，并从序列号 1 开始发送 LSP。当两片 LSP 的序列号相同时，剩余生存时间为 0 的报文比剩余时间非 0 的报文更新，如果报文剩余时间都不为 0，校验值较大者较新。

校验值：提供除剩余生存时间字段外的 LSP 头部校验，当路由器收到校验出错的 LSP，那么它除了清除本地 LSP 条目外，还会向网络其他路由器发送此 LSP 相应的清除报文。

以下标记字段只有在零分片中才有效:

P（区域修复）：目前无厂商支持此功能。

ATT（域间连接）：L1-2 路由器在自身的 L1 LSP 中置位该位，L1 路由器将设置该位的路由器作为通往主干网的出口，设置一条指向该路由器的默认路由。如果 L1 区域中存在多个 L1-2 路由器，则每台 L1 路由器根据开销最小设置默认路由。

LSPDBOL（过载）：如果路由器资源不可用，系统进入过载等待状态，在此状态中路由器只会发布过载标记置位的 LSP 零分片，其它路由器不会以发布过载标记的路由器为转发路径。直到需要的资源或等待定时器超时后系统会退出过载等待状态，此时路由器发布过载标记清零的 LSP 零分片。在过载等待状态中，路由器的报文转发和路由计算功能不会受影响。

ISTYPE：表示 IS 属于 L1 还是 L2。

LSP 变长部分包含多个 TLV，TLV 是一种常用的简单的变长报文内容封装方式，包含 T（Type，TLV 类型），L（Length，TLV 长度），V（Value，封装内容）三部分。除 LSP 报头外，每个 IS 应传递的链路状态信息主要包括所在区域的地址，邻居信息以及 IP/IPv6 路由信息。区域地址是区域的标记，每个 IS 单独配置并发布区域地址，L1-2 IS 在自身 L2 LSP 中还要发布从 L1 LSP 中学到的区域地址。L1-2 IS 通过对比 L1 和 L2 的区域地址判断自身能否通往区域外部。LSP 报文中包含的邻居信息通过 NBR TLV 和 Extended IS TLV 发布，IP 路由信息通过 IP Internal Reachability TLV 和 Extended IP TLV 发布，和扩展邻居 TLV，IPv6 路由信息通过 IPv6 Reachability TLV 发布。

NBR TLV 格式如表 2.2 所示。如果邻居为普通节点，Neighbor ID = System ID + 00；如果邻居为伪节点，Neighbor ID = DIS 的 System ID + 01。虽然 TLV 通过支持四种量度来提供不同服务级别，目前的路由交换设备提供商默认只支持缺省量度，其他量度都是为 QoS 准备的。每种量度只有 6bit 来表示开销值，支持 63 种开销刻度，限制了描述精度，被称作 Narrow 量度模式。Extended IS TLV 利用 3 字节来表示缺省量度的开销值，大幅提升了描述精度，被称为 Wide 量度模式。IP Internal Reachability TLV 属于 Narrow 量度模式，Extended IP TLV 和 IPv6

Reachability TLV 属于 Wide 量度模式。

表 2.2 邻居 TLV

TABLE2.3 NBR TLV

			No. of Octets	
Virtual Flag			1	
0	I/E	Default Metric	1	缺省量度
S	I/E	Delay Metric	1	时延量度
S	I/E	Expense Metric	1	费用量度
S	I/E	Error Metric	1	错误量度
Neighbor ID			ID Length + 1	

2.1.4 运行机制

2.1.4.1 邻居维护

两台 IS-IS 路由器在交互协议报文实现路由功能之前必须首先建立邻接关系。

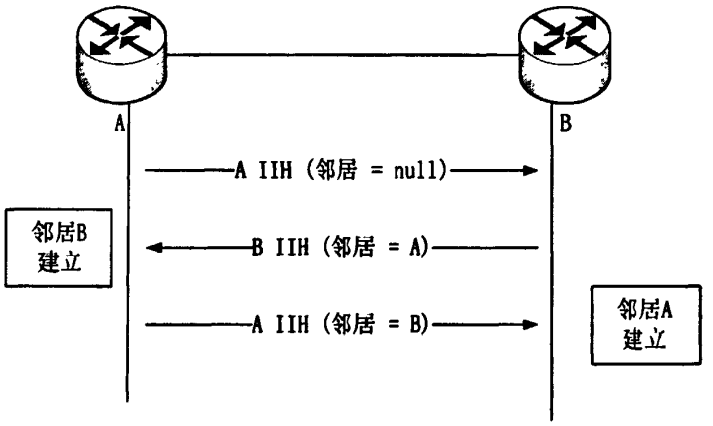

图 2.2 广播网邻居建立流程

Figure2.2 Establish LAN adjacency

IS-IS 在广播网中路由器间通过三次握手流程建立邻居。路由器收到 IIH 报文时需要进行邻居合法性检验：IIH 报文中的链路类型与收到报文的接口相兼容；报文中的 IP/IPv6 地址与收到报文的接口处于同一网段；报文中最多区域地址数目必须与自身一致，如果是 L1 的 IIH 还需要区域地址和自身配置一致；通过 IIH 的总长度了解对端接口的 MTU，确保形成邻接前能处理邻接点最大的 MTU（或 MTU 差别不大）；如果自身配置了认证，报文中的认证字段与自身匹配。

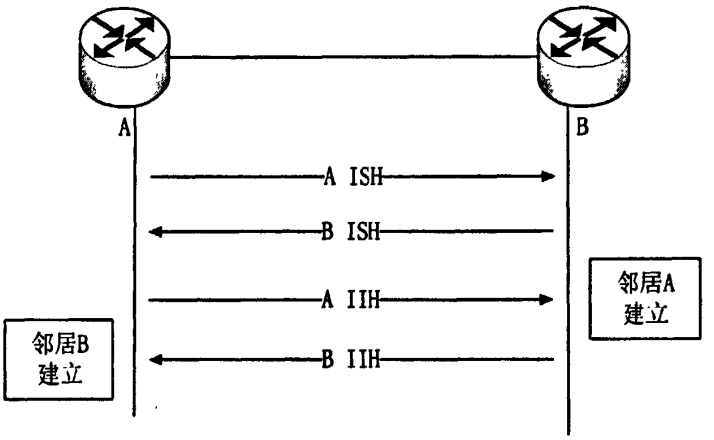

图 2.3 点到点网络邻居建立流程

Figure2.3 Establish Point-to-Point adjacency

IS-IS 点到点邻居关系的建立是通过先交换 ES-IS 协议的 ISH 报文，再交换 Point-to-point IIH 报文实现。每台路由器收到 ISH 包后，路由器先检查与发送者是否邻居关系，如果不是，发送 IIH 报文回应，对端收到 IIH 报文后进行与广播网类似的邻居合法性检验，如果通过则邻居 UP。由于缺少类似三次握手的确认机制，可能造成一端邻居 UP，一端邻居 DOWN 的情况发生。

2.1.4.2 链路状态信息同步

IS-IS 路由域内的所有路由器都会将自身的链路状态信息封装为 LSP 报文并通知其它路由器。在路由域内所有路由器的 LSDB 达到同步后，如果路由器的链路状态发生改变，路由器会重新封装 LSP 并泛洪发布，另外即使 LSP 没有变化，LSP 也会定期重新生成以保证 LSDB 中的 LSP 不会超时。

触发新 LSP 的原因包括：邻居 UP/DOWN；接口 UP/DOWN；接口开销变化；接口配置的 IP/IPv6 地址变化；IS 配置的区域地址变化；IS 的 System ID 变化；IS 的系统类型改变；DIS 变更；系统进入退出过载等待状态；引入、聚合的路由发生变化；验证信息变化；接收到自身生成的剩余时间为 0 的 LSP；LSP 定时刷新。

当路由器收到 LSP 报文时，会判断是否自身生成并与自身 LSDB 中对应的 LSP 进行比较。具体流程如图 2.4 所示。

在点到点网络中，如果路由器收到有效的 LSP 报文，LSP 与自身 LSDB 中的相应 LSP 一致或更新，需要回复 PSNP 报文确认，在路由器启动后邻居路由器会发送 CSNP 报文帮助其快速同步 LSDB。

在广播网中 SNP 报文用于辅助 LSDB 的同步：DIS 定期根据自身 LSDB 生成 CSNP 报文并泛洪，路由器收到 CSNP 后和自身的 LSDB 中的 LSP 进行比对。如果 DIS 拥有的 LSP 较新或者自身 LSDB 中没有，发送 PSNP 报文向 DIS 请求这个

LSP；如果自身 LSDB 中拥有的 LSP 比 DIS 的更新或者 DIS 没有，直接发送自身的 LSP。DIS 收到 PSNP 报文后会根据其中包含的 LSP ID 发送自身 LSDB 中的 LSP。

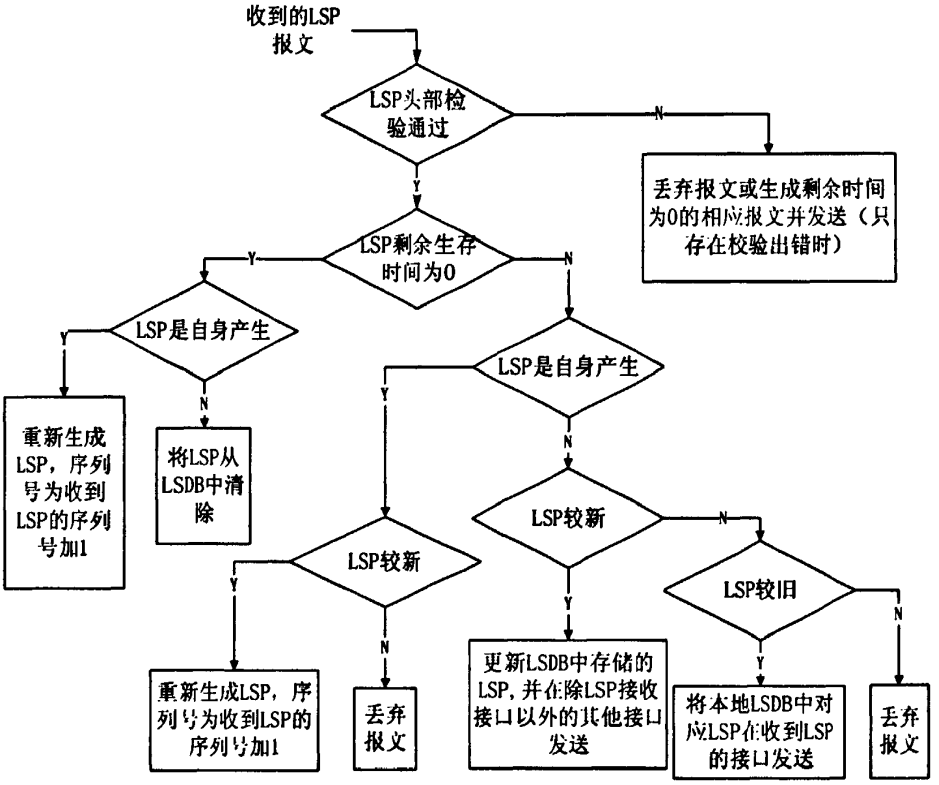

图 2.4 收到 LSP 报文的处理流程

Figure2.4 Receiving LSPs

2.2 MTR 扩展功能

Multi Topology Routing（MTR）多拓扑路由是就是指在自治系统中独立运行多个拓扑，每个拓扑都会单独维护路由表并转发流量，每个路由可以属于多个拓扑。多拓扑路由实现了拓扑级别的流量调度，报文转发时先选择拓扑再选择路径。以下列出多拓扑路由与两种相关技术的比较：

虚拟专用网（VPN）[8]是指私有网络之间通过公用网络（通常是因特网）建立安全稳定的连接，常应用于大企业部门之间传递消息。VPN 主要采用的四项安全保证技术：隧道技术、加解密技术、密钥管理技术和使用者与设备身份认证技术。MTR 和 VPN 都能应用于公司内网，确保流量优先经过内网传输，但 MTR 不具备 VPN 的安全性能，因此不能用在有保密性要求的环境下。

边缘路由最佳化(OER)以路径的效能作为路径选择的依据，针对多条对外路径设计流量负载平衡或路径备份。OER 的路径选择依据包括报文回复时间、丢包率、信号抖动、是否存在可达路径、流量分布和 cost 最小策略。MTR 在报文发端以用户自定规则先选择拓扑，拓扑内以链路开销最小化为标准选路，MTR 和 OER 都能为针对某一目的地址的流量传输提供多条有效路径，但 OER 的选路依据比 MTR 丰富，更利于确保报文传输的有效性和可靠性。

IS-IS MTR 扩展特性将 IS-IS 协议基本功能都进行多拓扑划分，形成了 IS-IS 进程内的多拓扑模型。每个拓扑就像是一个独立的 IS-IS 进程，可以独立地维护邻居关系，独立地维护链路状态信息，独立地计算最短路径并更新 IS-IS 路由表，最终与其它协议发布的路由一起进行优选得到拓扑路由表。

多拓扑功能新增四种 TLV，IIH 和 LSP 报文中新增的 MT TLV 发布配置的多拓扑及标记，LSP 中新增的 MT IS TLV，MT IP TLV 和 MT IPv6 TLV 发布多拓扑链路状态信息。每个 IS 根据这些信息分拓扑实现路由功能。

MT TLV 格式如表 2.7 所示：

表 2.7 多拓扑 TLV

TABLE2.7 MT TLV

					No. of Octets	
O	A	R	R	MT ID	2	

其中 O（OVERLOAD）为过载标记，A（ATT）为域间连接标记。如果路由器需要发布此 TLV，必须包含 MT ID 为 0 的条目。此 TLV 只能出现在 IIH 和 LSP 零分片中，其它 PDU 中出现均视作无效。

MT IS TLV 发布多拓扑邻居信息，格式如表 2.8 所示：

表 2.8 多拓扑邻居 TLV

TABLE2.8 MT IS TLV

					No. of Octets	
R	R	R	R	MT ID	2	MT ID 为 0 则无效
extended IS TLV format					11 - 253	Wide 量度邻居

MT IP TLV 发布多拓扑 IP 路由信息，格式如表 2.9 所示：

表 2.9 多拓扑 IP 路由 TLV

TABLE2.9 MT IP TLV

					No. of Octets	
R	R	R	R	MT ID	2	MT ID 为 0 则无效
extended IP TLV format					5 - 253	Wide 量度 IP 路由

MT IPv6 TLV 发布多拓扑 IPv6 路由信息，格式如表 3.0 所示：

表 2.10 多拓扑 IPv6 路由 TLV

TABLE2.10 MT IPv6 TLV

					No. of Octets	
R	R	R	R	MT ID	2	MT ID 为 0 则无效
IPv6 Reachability format					6 – 253	

可以看出后三种TLV只是在原有Wide量度的TLV基础上额外封装了MT ID，成为分拓扑发布的形式。MTR 扩展不会对现有邻居维护机制（只是增加了邻居合法性检验的内容）和 LSP 同步机制产生影响，DIS，CSNP，PSNP 的功能和报文不变。选举 DIS 仍依据标准拓扑的邻居进行。

3 IS-IS 支持多拓扑的实现

IS-IS 支持多拓扑功能需要 IS 在配置的每个拓扑单独实现 IS-IS 路由功能。未支持多拓扑时，IS-IS 协议主要依靠三大功能保证报文在路由域内按照最优路径转发，即邻居功能（adjacency），更新功能（update）和路由计算功能（decision）。在此基础上，为提高报文转发的质量，IS-IS 协议支持诸如 MPLS[18]，GR，NSR 等路由扩展功能。支持多拓扑后，每个拓扑相当于一个路由域，需要单独维护拓扑邻居，保证拓扑内链路状态信息同步，进行拓扑路由计算，支持路由扩展功能在拓扑内正常执行。因此，IS-IS 支持多拓扑功能的实现可以分为邻居，更新，路由计算和路由扩展功能四大主要模块。

3.1 模块功能与模块间接口

邻居模块负责维护多拓扑邻居关系。对端 IIH 报文的 MT TLV 变化和自身接口状态变化都会触发多拓扑邻居状态变化，接口下有效多拓扑变化时需重新封装自身 IIH 中的 MT TLV。

更新模块负责路由域内多拓扑链路状态信息的同步。收到其它 IS 的 LSP 报文后需要解析多拓扑链路状态信息并存入 LSDB，以及将变化的信息通知路由计算模块。LSP 失效时需要将失效的多拓扑链路状态信息通知路由计算模块，并删除 LSDB 中的结构。配置/取消多拓扑时需要通知激活了更新功能的拓扑。自身多拓扑链路状态变化时需要重新封装 LSP 报文。

路由计算负责计算多拓扑路由表，分为 ISPF 和 PRC 两部分。ISPF 负责分拓扑计算最短路径，邻居和更新模块向 ISPF 传递变化的多拓扑节点和链路信息。ISPF 计算后如果拓扑中节点新加入最短路径树，即从路由自身出发存在链路到达此节点，根据节点 LSP 报文中拓扑对应的路由触发 PRC。PRC 负责根据拓扑最短路径维护拓扑路由表，接口管理和更新模块向 PRC 传递变化的多拓扑路由信息。

路由扩展功能模块负责各扩展功能分拓扑执行。路由引入和聚合是两种简单的路由功能。引入是指将不是由 IS-IS 获取的路由通过 LSP 发布，聚合是指将具有相同前缀的多条路由以一条路由的形式发布到 LSP 中。系统且复杂的路由扩展功能（如 MPLS，GR，NSR）在本文中不做描述。

IS-IS 支持多拓扑的模块功能及模块间接口如图 3.1 所示：

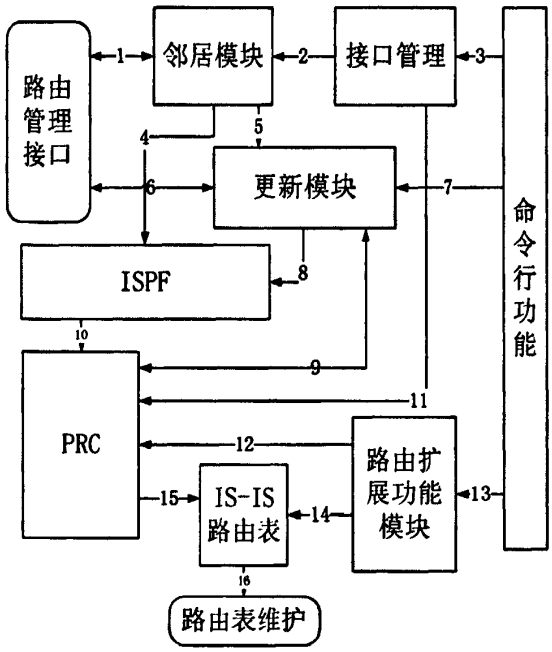

图 3.1 IS-IS 多拓扑扩展的模块分解

Figure3.1 The modular decomposition of IS-IS MTR

模块间接口包括：

1) 邻居模块通过路由管理接口收发 IIH 报文。
2) 接口状态及属性（协议，地址，拓扑）影响邻居状态。
3) 命令行配置接口属性。
4) 通知自身链路状态信息变更（链路）。
5) 触发封装 MT IS TLV。
6) 更新模块通过路由管理接口收发 LSP 和 SNP 报文。
7) 命令行配置/取消拓扑通知更新功能开关，并触发 MT TLV 封装。
8) 更新模块通知其它 IS 链路状态信息变更（节点和链路），ISPF 计算后可能触发 IP/IPv6 路由更新。
9) 更新模块通知其它 IS 链路状态信息变更（路由），PRC 计算后可能触发自身 MT IP TLV 封装。
10) ISPF 结束触发 PRC 生成路由表。
11) 通知自身链路状态信息变更（路由）。
12) 引入聚合路由等功能触发 PRC。
13) 命令行配置多拓扑路由扩展功能开关。
14) 默认路由等功能直接加入 IS-IS 路由表。
15) PRC 计算生成 IS-IS 路由表。

16) IS-IS 路由表通过与其它路由表优选后生成 IP/IPv6 路由表。

3.2 多拓扑邻居与路由计算的设计简述

为表征接口配置的多拓扑，路由器的 IIH 报文中添加了 MT TLV。多拓扑邻居模块负责维护自身产生的 IIH 报文中的 MT TLV，解析对端 IIH 报文中的 MT TLV 并保存在接口结构下。收到新的 IIH 报文后根据 MT TLV 包含的拓扑触发多拓扑邻居更新。

每个拓扑独立维护邻居，但由于邻居建立的三次握手过程与多拓扑功能无关，多拓扑邻居状态与标准拓扑保持一致。当路由器自身多拓扑删除和接口下的多拓扑开销变化事件发生时，触发对应拓扑的邻居维护；当邻居状态变化，接口 down 事件发生时触发接口下所有拓扑邻居维护。

路由计算分为 ISPF 和 PRC 两部分，其中 ISPF 负责响应 SPF 节点和链路变化根据 Dijkstra 算法计算根节点到其它路由节点的最短路径，ISPF 计算后更新路由叶子节点的开销值和下一跳。路由信息是指 IP/IPv6 地址前缀，每条路由信息以树的形式存储，树的每个叶子节点由路由开销及发布源组成。路由的发布源有如下六种：

直连：来自路由自身配置的接口地址。

学到：来自 LSP 中通告的路由信息。

默认：能够匹配所有地址的路由信息， L1 路由器根据 LSP 报文 ATT 标志位生成。

静态：来自命令行配置且不能自动更新。

引入：来自其它发布源，由于路由信息在其他发布源中已存在，不需要在叶子节点优选时考虑。

聚合：来自命令行配置的聚合命令，将自身路由表中具有相同前缀的多个地址以统一前缀形式发布到 LSP 中，由于路由信息已经以聚合前的状态参与优选，同样不需要在叶子节点优选时考虑。

PRC 负责响应路由叶子节点变化进行叶子节点优选并生成路由表。PRC 优选叶子节点先考虑最高优先级的发布源（直连>静态>学到>默认），优先级相同时再考虑最小的开销值，最后根据最优叶子节点刷新在路由表中的路由项。

每个拓扑独立进行 ISPF 和 PRC 计算，且每个 MT 对应一份有关路由计算的数据结构如 SPF 节点，链路，路由叶子节点以及路由表。

3.3 多拓扑更新的设计

多拓扑更新的目标是保证网络中多拓扑链路状态信息的同步，侦知自身以外 IS 的链路状态信息变化并触发多拓扑路由计算。由此更新模块的功能可以分为三部分：发布自身产生的多拓扑链路状态信息，处理非自身产生的多拓扑链路状态信息，保证网络中 IS 的链路状态数据库同步。MTR 对于第三项功能没有扩展，仍然沿用 IS-IS 协议规定的链路状态数据库同步机制。

发布自身产生的多拓扑链路状态信息功能响应四种新增 TLV 的多拓扑信息变更，触发 LSP 重新生成。如图 3.2 所示。

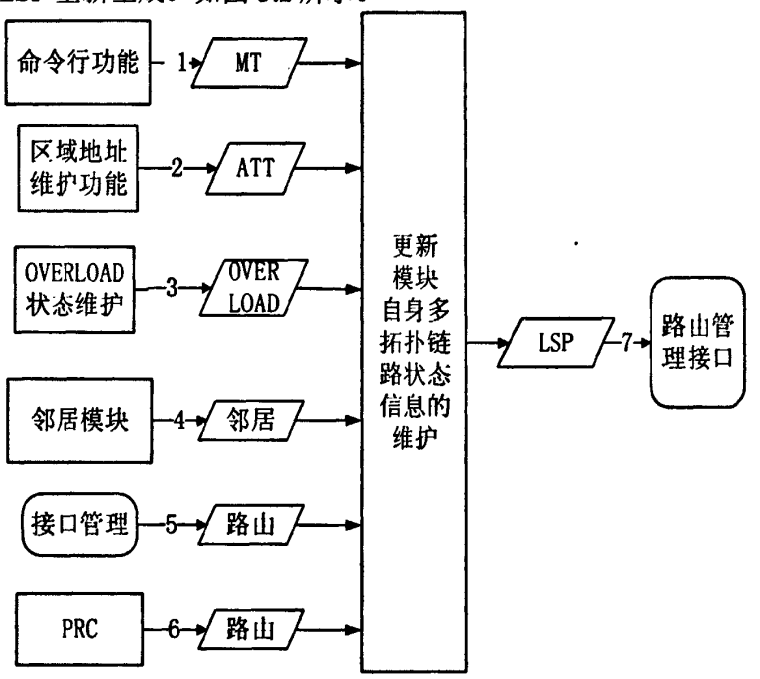

图 3.2 更新模块自身多拓扑链路状态信息的维护

Figure3.2 Update MT link state information of its own

◆ 子功能 1：MT TLV 的封装。

输入：1) 命令行功能通知配置/取消拓扑。

2) 多拓扑 ATT 标记改变。

3) OVERLOAD 状态改变。

处理：封装 MT TLV。

输出：7) LSP 重新生成。

◆ 子功能 2：MT IS/IP/IPv6 TLV 的封装。

输入：4) 邻居模块通知多拓扑邻居改变。

5) 接口管理通知多拓扑直连路由改变。

6) PRC 通知多拓扑学到、引入、聚合路由改变。

处理：封装 MT IS/IP/IPv6 TLV。

输出：7) LSP 重新生成。

◆ 子功能 3：OVERLOAD 状态维护。

输入：命令行配置(undo) set overload。

系统资源不足或恢复。

处理：所有配置的拓扑 OVERLOAD 标记置位/清零，LSP 中取消/重新发布引入和学到的路由。

输出：子功能 1，子功能 2。

◆ 子功能 4：区域地址维护。

输入：自身配置或学到的区域地址变化。

命令行功能通知配置/取消拓扑。

非自身产生 LSP 的 MT TLV 中拓扑变化。

处理：维护区域地址链表，每个区域地址针对每个拓扑都有 L1 和 L2 的引用计数。引用计数的变化可能引起区域地址计算，获取 ATT 标记。

输出：子功能 1。

子功能 1 和子功能 2 比较简单，实现子功能 1 需注意：只有标准拓扑时不需要封装 MT TLV，但封装多拓扑时需要保证标准拓扑在 MT TLV 中。这要求新分配 MT TLV 时添加 2 字节的标准拓扑，删除 MT TLV 中的拓扑后如果 TLV 长度剩下 2 则删除整个 MT TLV。拓扑发布的 OVERLOAD 和 ATT 标记改变时需要触发 MT TLV 拓扑字段重新封装。

实现子功能 2 需注意: MT IS/IP/IPv6 TLV 只是在原有 Extended IS/IP TLV 和 IPv6 Reachability TLV 格式上加封了 2 字节的 MT ID，因此可以复用原有 TLV 的封装流程。查找链路状态信息能够插入的 TLV 时需比对 MT ID，为新的 TLV 分配的最小空间需考虑 2 字节 MT ID。

处理非自身产生的多拓扑链路状态信息功能响应收到 LSP，命令行配置/取消多拓扑和 ISOLATE 状态变化三种事件；触发 ISPF 和 PRC 计算，区域地址维护功能和 LSDB 维护；如图 3.3 所示。

◆ 子功能 1：收到 LSP 报文的更新。

输入：1) 从路由管理接口收到 IS-IS LSP 报文。

处理：解析并找出变化的链路状态信息。

输出：4) 由区域地址触发区域地址维护功能。

5) 6) 多拓扑信息和邻居信息触发 SPF 节点和链路变更并进行 ISPF 计算。

7) 多拓扑路由信息触发叶子节点变更并进行 PRC 计算。

8) 解析出的多拓扑信息存入 LSDB。

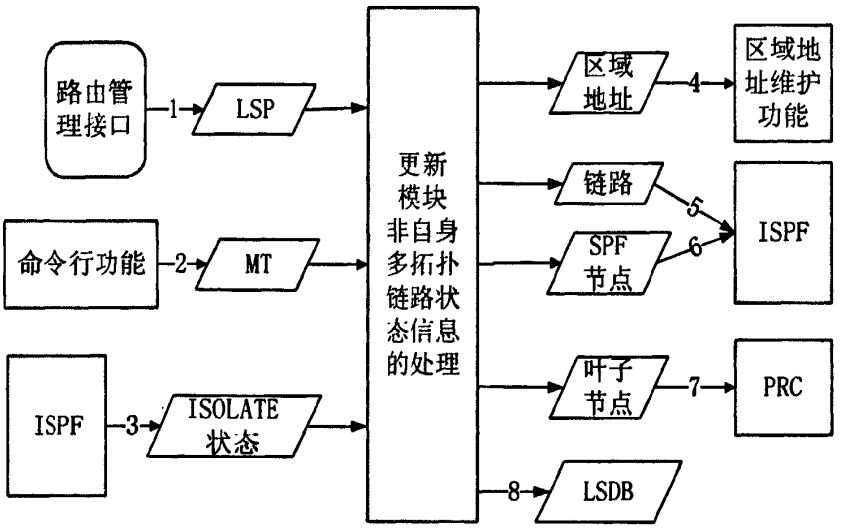

图 3.3 更新模块非自身多拓扑链路状态信息处理

Figure3.3 Update other's MT link state information

◆ 子功能 2：响应配置/取消多拓扑的处理。

输入：2) 命令行功能通知变化的有效拓扑。

处理：开启/关闭拓扑的更新功能。

◆ 子功能 3：响应 ISOLATE 状态变化的处理。

输入：3) 拓扑 ISPF 计算结束后节点 ISOLATE 状态变化。

处理：从 LSDB 中找出节点发布的对应拓扑的路由信息。

输出：7) 触发叶子节点新增并进行 PRC 计算。

◆ 子功能 4：报文清除。

输入：1) 收到剩余时间为 0 的 LSP。

更新模块自身 LSDB 维护 LSP 超时定时器超时。

处理：根据失效 LSP 是否零分片决定需要触发清除的链路状态信息。

输出：4) 由区域地址触发区域地址维护功能。

5) 6) 多拓扑信息和邻居信息触发 SPF 节点和链路删除并进行 ISPF 计算。

7) 多拓扑路由信息触发叶子节点删除并进行 PRC 计算。

8) 删除 LSDB 中失效 LSP 的数据结构。

3.3.1 数据结构设计

在 LSDB 中，每个 IS 对应唯一的 LspSystemInfo 数据结构，保存 IS 的基本信息如 System ID，标记位，区域地址等。每个 LSP 分片对应一个 LspInfo 数据结构，

主要记录 LSP 中携带的邻居信息，IP/IPv6 路由信息和其它 TLV 中包含的信息。LspSystemInfo 的访问主要是按 System ID 查找，因此采用 Hash 表存储；LspInfo 的访问多为遍历，因此采用双链表存储。

由于每个 IS 允许发布 256 片 LSP，每个 LspSystemInfo 最多与 256 个 LspInfo 相对应。为表征这种对应关系，LspSystemInfo 中包含一条以 LspInfo 为节点的链表，LspInfo 中也记录了相应 LspSystemInfo 的地址。LspInfo 在初次收到 LSP 分片时创建，LspSystemInfo 根据其代表的意义应该在初次收到代表整个 LSP 系统有效性的零分片时创建，但这样做使维护二者对应关系变得很复杂。为规避复杂性，LspSystemInfo 在收到 LSP 任意分片时创建，但此时 LSP 零分片是否到达将无法判断，需要添加额外标记记录。

由于新增的 MT TLV 只有包含在 LSP 零分片中才有效，属于路由器基本信息，其包含的拓扑及相关标记应在 LspSystemInfo 中保存。在 LspSystemInfo 构建一条拓扑顺序链表，每个链表节点代表一个拓扑，节点还记录了更新功能需要的标记位：OVERLOAD 和 ATT 来自 LSP 报文零分片，ISOLATE 和 ISOLATE_V6 来自 LSP 系统对应的 SPF 节点的标记位，Zero 代表零分片是否到达。

MT IS TLV，MT IP/IPv6 TLV 中包含的多拓扑邻居和路由信息保存在对应分片的 LspInfo 数据结构中。未进行多拓扑扩展时邻居信息，IPv4 和 IPv6 路由信息分别通过一条链表存储，扩展后多拓扑信息的保存有三种方案：

- 多拓扑信息与标准拓扑信息存在同样的链表中。
- 多拓扑信息存储在额外的三条链表中。
- 多拓扑与标准拓扑信息按双层链表嵌套的形式存储：第一层链表按顺序存储拓扑 ID，第二层链表存储每个拓扑的相关信息。

前两种方案存储时操作比较简单，第二种方案对标准拓扑信息的处理可以复用以往的实现，但鉴于拓扑级的操作（本地或网络中其它路由器的 IS-IS 进程配置/取消拓扑）只需要取出相应拓扑的信息，依这两种方案存储只能遍历所有拓扑的信息查找（第二种方案对多拓扑操作时需遍历所有多拓扑），造成巨大的冗余操作。第三种方案应对拓扑级操作具有很高效率，但是对原有实现的复用率较低，最终被采用。

综上所述，更新模块数据结构如图 3.4 所示。

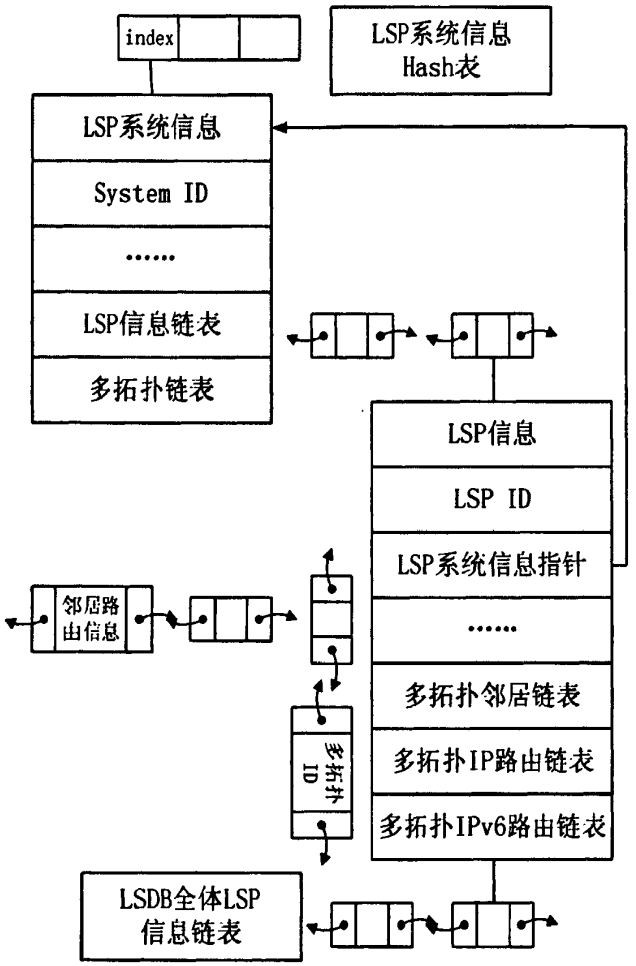

图 3.4 更新模块数据结构

Figure3.4 Data structure of updating

3.3.2 收到 LSP 的处理

3.2.1.1 MT IS/IP/IPv6 TLV 的更新

命令行配置多拓扑后，系统会为新增的拓扑创建保存基本信息（如各种标记位）和路由计算相关的数据结构，因此只有本地 IS-IS 进程配置过的拓扑才会进行路由计算并生成拓扑路由表。当接收到的 LSP 报文中含有本地 IS-IS 进程未配置过的多拓扑信息，则这些信息无效，不会参与路由计算。

由于 IS-IS 进程支持的拓扑 ID 都会保存在 LSP 零分片的 MT TLV 中，如果接收到的 LSP 分片中 MT IS TLV，MT IP TLV 或者 MT IPv6 TLV 包含的拓扑 ID 在对应 LSP 零分片的 MT TLV 中不存在，该 TLV 可以视作无效，不会触发拓扑信息更新和路由计算。

综上所述，接收到的多拓扑信息需要进行更新操作的条件（简称更新条件）包括：本地 IS-IS 进程配置过此拓扑且对应的 LSP 零分片 MT TLV 中包含此拓扑。对于不满足条件的多拓扑信息是否需要存入 LSDB，考虑如下：这些信息的存储占用了额外的空间，但拓扑满足更新条件后拓扑相关邻居和路由信息可以从 LSDB 中直接提取触发更新，便于拓扑链路状态信息快速同步，否则需要等待 20 分钟（LSP 报文发送间隔）以内的一段时间才能同步。由于路由协议的实效性十分重要，且大容量存储介质的成本在不断降低，不满足更新条件的多拓扑信息需要被保存。

MT IP/IPv6 reach TLV 的更新过程需要考虑 LSP 系统标记位的影响，ISOLATE 表示从根节点到 LSP 系统相应 SPF 节点没有下一跳 IPv4 地址，ISOLATE_V6 表示没有下一跳 IPv6 地址。未经路由计算的新增节点都标记为 ISOLATE 和 ISOLATE_V6，路由计算后如果根节点到此节点存在 IP/IPv6 下一跳地址，则取消 ISOLATE/ISOLATE_V6 标记。根节点的 IP/IPv6 报文无法到达 ISOLATE/ISOLATE_V6 状态下的节点，因此这些节点不需要添加 MT IP/IPv6 TLV 中包含的路由信息，直到节点 ISOLATE/ISOLATE_V6 标记取消时系统才会根据 LSDB 中存储的 LSP 向节点添加路由信息。

MT IS TLV 和 MT IP/IPv6 reach TLV 被解析后存储为拓扑链表和拓扑信息链表双层嵌套的形式，更新时需要先比较外层链表再比较内层链表。具体过程如下：

判断外层链表每个节点的新旧：

根据更新条件决定是否处理拓扑。

根据 ISOLATE 条件决定是否处理拓扑。

如果拓扑为新增，以拓扑节点包含的第二层链表的节点触发多拓扑路由计算结构（链路、IPv4/IPv6 叶子节点）创建。

如果拓扑为原有，比较拓扑对应的新旧第二层链表：如果第二层链表节点为新增，触发多拓扑路由计算结构创建；如果节点为原有且节点的参数值改变，触发多拓扑路由计算结构修改；如果节点需删除，触发多拓扑链路删除。

如果拓扑需删除，以拓扑节点包含的第二层链表的节点触发多拓扑路由计算结构删除。

释放原有双层嵌套链表结构，将新的双层链表保存至 LspInfo 数据结构中。

判断链表节点新旧的的常见方案有两种：

- 首先遍历新链表，将每个节点在旧链表中查找对应节点，查找不到就触发新增，查找到且属性改变触发修改，修改以后需要将旧链表中的对应节点删除。然后遍历新链表，以每个节点触发删除。假设新旧链表长度分别为 m 和 n，复杂度为 m*n。

- 将新旧链表按从小到大的顺序排序，然后按照归并排序算法处理节点的顺序判断每个节点应触发的操作。复杂度需要考虑新链表的排序和归并比较，根据解析过程的特点，新链表排序采用插入排序，复杂度为 m*m，归并比较复杂度为 m+n。

两种算法复杂度相当，但由于拓扑级操作需要提取单个拓扑的信息进行更新，拓扑链表排序后查找的平均复杂度降低，拓扑链表都采用顺序表存储，且链表节点新旧判断采用方案二。而其它链表直接采用方案一复用平台代码的部分流程。

3.2.1.2 MT TLV 的解析与更新

MT TLV 只有存在于 LSP 零分片中才有效，解析时读出每个拓扑及其标记位并以链表形式存储。需要考虑的问题包括两点：拓扑重复出现和标准拓扑及标记位的存储。前者属于协议中没有规定的异常情况，鉴于每个拓扑只能与一组标记位对应，重复的拓扑只需存储一次，如果重复拓扑的标记位不同则存储第一次读到此拓扑时的标记位。IS-IS MTR 协议规定如果配置了多拓扑则 MT TLV 中需加入标准拓扑（拓扑 ID 为 0），但由于系统默认标准拓扑存在，不管多拓扑功能是否开启，拓扑链表中的标准拓扑节点必须存在，因此标准拓扑节点的创建需要在 MT TLV 解析功能外部进行，且标准拓扑的标记位以 LSP 零分片头部的信息为准。MT TLV 中即使包含标准拓扑也不需要解析出来。

在进行多拓扑扩展之前，IS-IS 进程接收到新的 LSP 零分片后会在标准拓扑中创建相应 SPF 节点并根据邻居信息添加链路，如果 LSP 其它分片先于 LSP 零分片收到，由于标准拓扑中 SPF 节点不存在，分片中的邻居和路由信息解析后无法进行更新，存储于 LSDB 中。零分片到达后需要从 LSDB 中提取其它分片的邻居信息触发链路添加。MT TLV 的更新与上述过程类似：MT TLV 中新增本地配置的拓扑时，需要在此拓扑中创建相应 SPF 节点并依据所有分片（其它分片从 LSDB 中提取）中此拓扑包含的邻居信息触发多拓扑链路添加。MT TLV 中原有的本地配置的拓扑需删除时，除了拓扑中链路和节点的删除，还要触发区域地址模块删除节点区域地址。

MT TLV 的更新过程还需要考虑多拓扑 ATT 和 OVERLOAD 标记位的处理。某一拓扑的 OVERLOAD 标记位变化需要触发此拓扑中 LSP 对应 SPF 节点状态更改，重新进行 ISPF 计算。从非 OVERLOAD 状态变为 OVERLOAD 状态的节点只能作为根节点最短路径树的叶子节点，也就是说除了节点的直连地址，根节点流量的不会通过此节点转发；从 OVERLOAD 状态恢复后节点恢复转发根节点流量的功能。如果 L1 LSP 中某一拓扑的 ATT 标记位变化，首先需要判断 LspSystemInfo 中拓扑的 ISOLATE（ISOLATE_V6）标记位是否置位，如果未置位才触发此拓扑

中根节点默认路由的变化，根节点重新进行 PRC 计算。如果节点从非 ATT 状态变为 ATT 状态，根节点会生成一条以此节点为目的的默认路由来参与默认路由的优选；如果状态变化相反，根节点会取消以此节点为目的的默认路由并重新进行默认路由的优选。MT TLV 的更新具体过程描述如下：

比较新旧拓扑链表：

如果拓扑为新增且本地 IS-IS 进程配置过，触发 SPF 节点添加，从 LSDB 中提取此 LSP 所有分片中拓扑对应的邻居信息触发链路添加，处理拓扑的 OVERLOAD 标记位。

如果拓扑已存在，处理 OVERLOAD 标记位，如果当前处理的是 L1 LSP 且 LspSystemInfo 中拓扑的 ISOLATE 标记位未置位，处理 ATT 标记位，将旧的拓扑节点的 ISOLATE 标记位同步到新的拓扑节点。

如果拓扑需删除且本地 IS-IS 进程配置过，如果当前处理的是 L1 LSP，LspSystemInfo 中拓扑的 ATT 标记位置位且 ISOLATE 标记位未置位，处理 ATT 标记位产生的默认路由，触发区域地址模块删除拓扑对应的区域地址，删除拓扑中 LSP 所有分片中拓扑的邻居信息对应的链路，删除拓扑中 LSP 对应的 SPF 节点。

释放旧的拓扑链表，将新的拓扑链表保存至 LspSystemInfo 数据结构中。

3.2.1.3 伪节点报文更新

指定中间系统（DIS）负责生成伪节点 LSP 报文，伪节点只是一个虚拟节点，它的功能是将以太网中互相建立邻居的节点以星形的组织形式连接起来。为了保证任意两个互为邻居的以太网节点在其共有拓扑中能够互通，每个节点需要在其配置的所有拓扑中添加伪节点和伪节点与根节点之间的链路。

每个以太网节点独立进行 DIS 选举，选举使用标准拓扑下的邻居，与多拓扑无关。在正确组网情况下所有节点选出的 DIS 一致，选举结束后在本地配置的所有拓扑中添加根节点到选出的 DIS 相应伪节点的链路，链路添加过程与 DIS 是否配置过这些拓扑无关。

DIS 将自身以太网接口下标准拓扑的邻居发布到伪节点 LSP 报文中。IS-IS MTR 协议规定伪节点的 LSP 报文不能包含新增的四种多拓扑相关 TLV，因此伪节点报文包含的邻居信息不区分拓扑。在收到在伪节点的 LSP 报文时需在本地配置的所有拓扑中更新伪节点的 SPF 节点和伪节点到根节点的链路。在错误的组网情况下（以太网所有节点不在同一网段），以太网中存在多个 DIS，每个以太网节点都会收到多个伪节点 LSP 报文。根据链路双向存在才有效的原则，既然根节点到伪节点的链路已采用精确控制，伪节点到根节点的链路可以直接根据每个收到的伪节点 LSP 更新。

总之，对伪节点 LSP 报文的处理方式为：初次收到伪节点的 LSP 报文后在本地 IS-IS 进程配置的所有多拓扑中创建 SPF 节点，本地配置新拓扑时需要在新拓扑中添加伪节点的 SPF 节点。此外收到的伪节点 LSP 中邻居信息改变时需要在本地 IS-IS 进程配置的所有拓扑下触发链路更新。

多拓扑情况下的拓扑结构如图 3.5 所示，其中 B 被选为 DIS。在标准拓扑和 MT ID 为 12 的拓扑中，链路结构如左下图所示。在 MT ID 为 100 的拓扑中，链路结构如右下图所示，虽然 B 未配置此拓扑，伪节点到 D 和 E 的链路依然被添加。

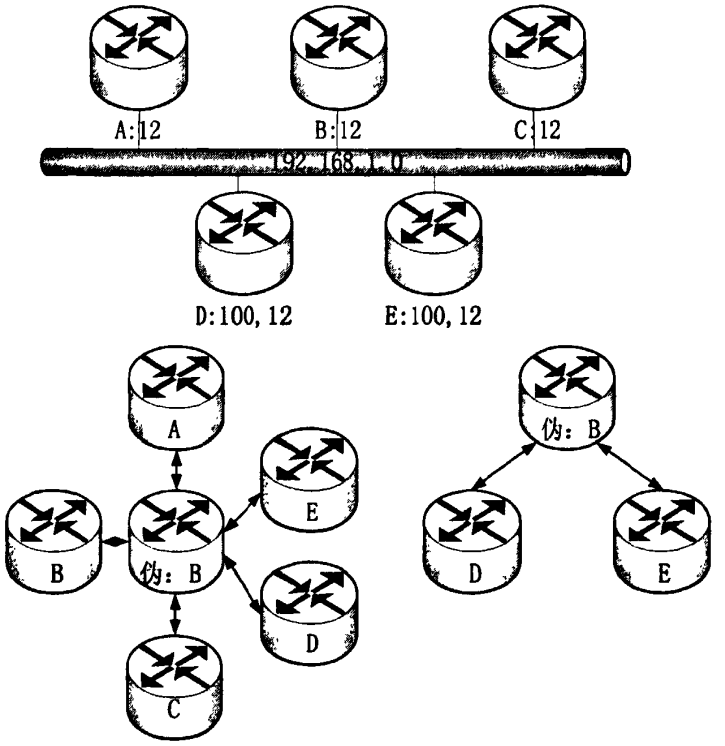

图 3.5 多拓扑情况下的拓扑结构

Figure3.5 Topology structure in MT

错误组网情况下的拓扑结构如图 3.6 所示，其中 A、B、C 属于同一网段，相互建立邻居，选举 B 为 DIS；D、E 属于另一个网段，选举 D 为 DIS。以太网中存在两个伪节点，A、B、C 与 B 的产生的伪节点间存在双向链路，与 D 产生的伪节点间只存在单向链路。

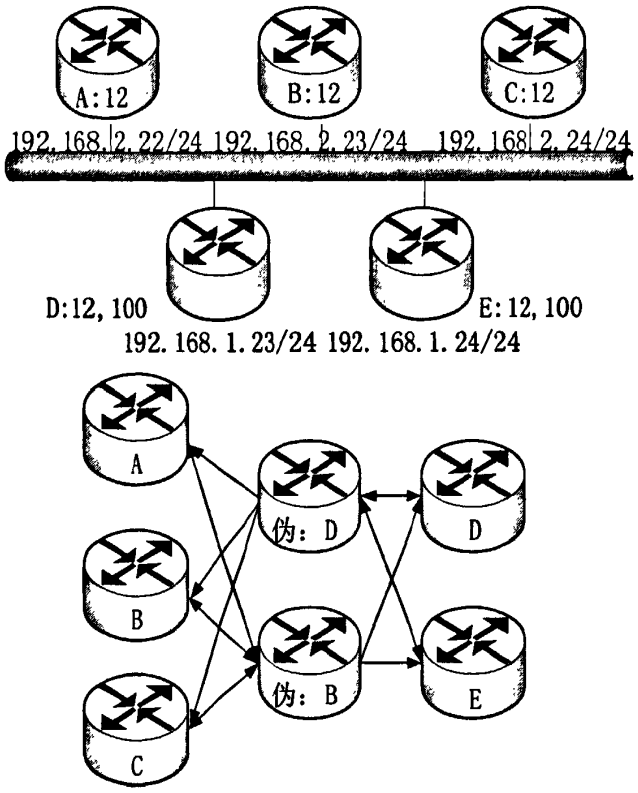

图 3.6 错误组网情况下拓扑结构

Figure3.6 Topology structure in false network

3.3.3 ISOLATE 状态变化的处理

一条完整的路由表项包括目的地址，出接口或下一跳地址以及开销值，拓扑 ISPF 计算完成后如果 SPF 节点的 ISOLATE（ISOLATE_V6）标记位置零，根节点到此节点的 IP（IPv6）下一跳存在，故以此节点配置的 IP（IPv6）地址为目的地址的路由表项可以完成。此时应从 LSDB 中提取此节点对应的所有 LSP 分片，以拓扑对应 IP（IPv6）路由信息触发路由叶子节点创建，根据拓扑对应的 ATT 标记位触发叶子节点（默认路由）的添加，触发区域地址模块区域地址链表更新，并进行此节点的 PRC 计算以生成路由表项。如果 SPF 节点的 ISOLATE（ISOLATE_V6）标记位置位，处理过程相反。

由于 LSP 只能包括 256 片分片，每片分片最大达到链路 MTU（以太网为 1500 字节），总共能传递 4 万条左右的邻居和路由信息，但这一数字有时不足以描述网络现状。按照协议规定，配置了多拓扑功能后，每个拓扑的邻居和路由信息通过多拓扑相关 TLV 额外传输，这无疑增加了 LSP 需要传递的信息量，尤其在每个节

点属于多个拓扑时传递信息的冗余更加明显。一种简单的消除冗余的方式是为标准拓扑的邻居和路由信息附加隶属的拓扑，但这种方式更新过程的效率不如协议规定的方式，拓扑级操作时仍需遍历所有邻居或路由信息节点查找属于本拓扑的内容，鉴于路由器比较强调时效性，协议规定的方式也可以理解。

虚拟系统(RFC 5311)可以解决 LSP 空间不足的问题。虚拟系统是指一个 IS-IS 进程可以发布多个具有不同 System ID（手动配置）的 LSP，虚拟系统的 LSP 中只存放原始系统存储不下的 IP/IPv6 路由信息。虚拟系统技术具有兼容性，在不支持虚拟系统的路由看来，虚拟系统节点只与相应原始系统节点建立开销为 0 的邻居关系，以虚拟系统节点为目的的报文仍然会转发到原始系统节点。由于虚拟系统与原始系统来自同一个 IS-IS 进程，虚拟系统的标记位与原始系统一致。为了表示虚拟系统与原始系统的关系，在 LSDB 中每个原始系统的 LspSystemInfo 中包含一条以虚拟系统的 LspSystemInfo 为节点的链表。

如果节点配置了虚拟系统功能，原始系统 LspSystemInfo 中某一拓扑的 ISOLATE（ISOLATE_V6）标记位置零/置位后，除了原始系统的 LspInfo 外，还需要根据相关虚拟系统所有 LspInfo 中拓扑的 IP（IPv6）路由信息触发路由叶子节点添加/删除。

3.3.4 报文清除

IS-IS 协议规定，当 LSDB 中存储的 LSP 剩余时间为 0 或者收到剩余时间为 0 的 LSP 清除报文时需要触发 LSP 报文清除，先清除报文相关的节点、链路或路由叶子节点，如果路节点和链路存在于最短路径树中，触发 SPF 计算，如果路由叶子节点发布于 IS-IS 路由表中，触发 PRC 计算；再将 LspInfo 标记为 DELETE 状态，在协议规定的清除时间之后从 LSDB 中删除 LspInfo。

由于网络中 LSP 报文校验出错也会造成 LSP 清除报文的发送，此时 LSP 源路由器收到此清除报文后需要重新生成此条 LSP 并泛洪发布。在这种情况下如果清除流程直接从 LSDB 中删除 LSP 的话，网络中的路由器都要在短期内先删除 LSP 再重新添加新的 LSP，造成冗余操作。采用等待清除计时器为 LSP 源路由提供了充分的重新生成报文和泛洪的时间，因此能够避免此种情况。但 LSP 不即时删除会造成 LSP 零分片是否初次收到难以判断。如果 LSP 被清除但等待清除计时器未超时，LspInfo 依然保存在 LSDB 中，LSP 其它分片可以正常更新，但 LSP 零分片是否初次收到决定 LSP 其它分片的更新。采取的解决办法是 LSP 零分片清除时将 LspSystemInfo 中标准拓扑的 ZERO 标记位清零，收到 LSP 零分片时通过 ZERO 标记判断是否初次收到。

原始系统 LSP 非零分片清除时，清除本片相关的多拓扑链路和路由叶子节点：根据分片中符合更新条件的邻居信息触发多拓扑链路删除，如果 LspSystemInfo 中 ISOLATE（ISOLATE_V6）标志位未置位，根据分片中的 IP/IPv6 路由信息触发多拓扑路由叶子节点删除。

原始系统的 LSP 零分片清除时，对本地 IS-IS 进程配置且 LspSystemInfo 中包含的每个拓扑进行以下操作：如果当前处理的是 L1 LSP，拓扑的 ATT 标记位置位且 ISOLATE（ISOLATE_V6）标记位未置位，处理 ATT 标记位产生的默认路由；触发区域地址模块删除拓扑对应的区域地址；将拓扑的 OVERLOAD，ATT 标记位置零，ISOLATE，ISOLATE_V6 标记位置位。根据 LSP 所有分片中多拓扑邻居信息触发链路删除，根据 LspSystemInfo 中每个本地 IS-IS 进程配置的拓扑删除 LSP 对应的 SPF 节点。将 LspSystemInfo 中标准拓扑的 Zero 标记位取消。

虚拟系统的 LSP 非零分片清除时，清除本片相关的路由叶子节点。虚拟系统的 LSP 零分片清除时，清除虚拟系统 LSP 所有分片相关的路由叶子节点。本地 IS-IS 进程若不支持虚拟系统，根据 LspSystemInfo 中每个本地 IS-IS 进程配置的拓扑清除虚拟系统节点及其到原始系统的链路，删除虚拟系统节点。将虚拟系统 LspSystemInfo 中标准拓扑的 Zero 标记位取消。

原始系统的 LspInfo 删除流程：从 LspSystemInfo 的 LspInfo 链表中移除节点，如果 LspInfo 链表变空且虚拟系统链表也为空，删除 LspSystemInfo；从全局 LspInfo 链表中移除节点；删除 LspInfo。

虚拟系统的 LspInfo 删除流程：从全局 LspInfo 链表中移除节点，从虚拟系统 LspSystemInfo 的 LspInfo 链表中移除节点，删除 LspInfo；如果 LspInfo 链表变空，从原始系统 LspSystemInfo 的虚拟系统链表中移除此虚拟系统，从全局 LspSystemInfo Hash 表中移除此虚拟系统，删除 LspSystemInfo；如果原始系统 LspSystemInfo 的虚拟系统链表变空且 LspInfo 链表为空，从全局 LspSystemInfo Hash 表中移除此原始系统，删除 LspSystemInfo。

3.3.5 OVERLOAD 状态的维护

系统进入 OVERLOAD 状态的触发条件包括：手动配置 set overload 命令，IS-IS 更新模块分配不到内存，IS-IS 以外的路由模块导致系统内存不足。三种方式触发系统进入 OVERLOAD 状态后都需要重新封装 LSP 报文，将 LSP 报文头部 OVERLOAD 标记位置位，并在 MT TLV 所有拓扑的 OVERLOAD 标记位置位。手动配置触发时还需要删除 LSP 中发布的多拓扑引入路由和 L2 LSP 中发布的从 L1 LSP 中学到的多拓扑路由，LSP 中只发布多拓扑直连路由。由于内存不足的两种

方式触发时还需要将引入路由的叶子节点删除。

系统退出 OVERLOAD 状态的条件包括：手动配置 undo set overload 命令，内存恢复后启动的定时器超时。前者应对手动配置命令的触发方式，后者应对内存不足的两种触发方式。系统退出内存不足引发的 OVERLOAD 状态后需要重新执行引入流程，如果系统还配置了 set overload 命令，则引入路由不需要发布，否则，重新封装 LSP 报文，将 LSP 报文头部 OVERLOAD 标记位置零，将 MT TLV 所有拓扑的 OVERLOAD 标记位置零，在 LSP 中发布多拓扑引入路由并在 L2 LSP 中发布的从 L1 LSP 中学到的多拓扑路由。

在内存不足触发 OVERLOAD 状态后，系统无法再为重新封装的 LSP 报文分配内存，只能利用已经分配过的内存。解决此问题的方法是预先留出内存空间，在 LSDB 中为每个 LSP 报文分配存储空间并存入报文，空间地址保存在 LSP 对应的 LspInfo 中。内存不足触发 OVERLOAD 时利用 LSDB 中自身生成的 LSP 零分片报文存本，在此基础上将报文头部 OVERLOAD 标记位置位，查找存本中的 MT TLV，将每个拓扑的 OVERLOAD 标记位置位，查找 MT IP/IPv6 TLV，将引入路由和从 L1 学到的路由对应的条目删除。

3.3.6 区域地址更新

MT TLV 中的 ATT 标记指示 L12 路由器是否有通向区域外部的出口，ATT 标记是否置位是根据区域地址计算得到的。本功能应用于 L12 路由器负责记录区域地址并计算自身 ATT 标记位，具体步骤如下：维护 L12 路由器的区域地址链表，链表中存放的区域地址来源有两种：系统自身配置且有效，从接收到的 LSP 中学到，每个区域地址包含 L1 和 L2 引用计数。当配置或学到的区域地址增加时，引用计数加 1，相反则引用计数减 1。判断 ATT 是否置位的办法是：遍历链表中所有的区域地址，如果拓扑中某一个区域地址 L1 的引用计数为 0 而 L2 的引用计数不为 0，则 ATT 置位；如果全部区域地址没有只在 L2 中出现的情况，ATT 清零。

通过统计报文能够转发到的区域，能够得出 L12 路由器能否作为 L1 区域报文向 L2 转发的出口，当路由节点处于 ISOLATE 状态时，报文无法转发到节点所处区域，因此不需要将节点对应 LSP 的区域地址加入到区域地址链表。涉及拓扑中路由节点删除或拓扑整体删除的情况下，虽然路由节点可以通过 ISOLATE 状态变化触发减少区域地址引用计数，但为提高删除处理的效率，区域地址引用计数不通过 ISPF 直接减少。基于此两点，触发区域地址维护的条件包括：IS-IS 进程配置或取消有效的区域地址；接收到不是自身产生的且剩余时间不为零的 LSP 零分片，且 LSP 对应路由节点不处于 ISOLATE 状态，LSP 中区域地址变化或是原有拓扑不

复存在；拓扑某 LEVEL 节点 ISPF 计算后进入或退出 ISOLATE 状态；接收到剩余时间为 0 的 LSP 或 LSP 超时。

由于区域地址 TLV 不分拓扑，LSP 中的区域地址可以视作与 MT TLV 中包含的所有拓扑相关联，链表的维护处理如下：IS-IS 进程配置或取消区域地址需要在自身配置的所有拓扑中添加或减少 L1 和 L2 的引用计数。从 LSP 中学到的区域地址变化则在 LSP 零分片 MT TLV 中包含的拓扑中增减 LSP 对应 LEVEL 的引用计数。LSP 更新处理需删除的拓扑时将 LSP 包含的区域地址在此拓扑中 LSP 对应 LEVEL 的引用计数减少。拓扑节点 ISOLATE 状态改变则节点对应 LSP 包含的区域地址在此拓扑中引用计数变化。LSP 报文清除则在 MT TLV 中所有拓扑中减少区域地址 TLV 中每个地址的引用计数。

将 L1 学到的区域地址发布到 L2 的 LSP 中，超过三个区域地址则不再发布。当拓扑中某区域地址的 L1 或 L2 引用计数出现从 0 到 1 或从 1 到 0 的变化，触发区域地址计算。拓扑状态由 NOT ATT 变为 ATT，触发 LSP 封装，如果拓扑中原先存在到 L12 的默认路由，删除该路由。拓扑状态由 ATT 变为 NOT ATT，封装 LSP 并更新默认路由。

3.4 配置/取消多拓扑的设计

配置/取消多拓扑功能负责管理各路由功能对拓扑的支持，将自身支持的拓扑下发给各模块进行相关处理。四大功能模块均受本功能影响：邻居模块受接口下配置的拓扑影响，接口下的拓扑是本功能配置拓扑的子集；更新模块直接响应本功能，由于系统配置了拓扑是拓扑链路状态信息需要更新的必要条件，当配置的拓扑变化时，LSDB 中存储的一些多拓扑链路状态信息可能变为可更新状态，需要直接触发更新动作；路由计算模块响应本功能创建/删除拓扑的路由计算数据结构；多拓扑路由扩展功能只有在拓扑配置的情况下才有效。

路由器配置多拓扑后既需要对外发布也需要开启各拓扑路由功能。前者通过向 LSP 零分片的 MT TLV 中添加 MT ID 实现并有可能失败。当 LSP 零分片空间已满无法添加 MT ID 时，配置路由器无法将拓扑通告其它路由器。此时配置路由器如果只触发各路由功能支持此拓扑，会造成网络中节点的链路状态不完全同步。配置路由器可以通过该拓扑转发报文，但其它节点利用该拓扑转发的报文不会到达配置路由器。为了避免这种情况，分拓扑计算功能应该先判断拓扑发布是否成功，如果失败将拓扑置为 disable 状态，路由功能不会支持该拓扑。

多拓扑分为 IP 拓扑和 IPv6 拓扑两类，配置/取消多拓扑功能需要根据两类情况分别处理。其中配置/取消 IPv6 单播拓扑（即 IPv4、IPv6 分拓扑功能）比较复

杂，分为两部分处理，分别涉及标准拓扑 IPv6 的去使能/使能和 IPv6 单播拓扑的使能/去使能。具体的配置 IPv6 单播拓扑的功能流程如图 3.7 所示。

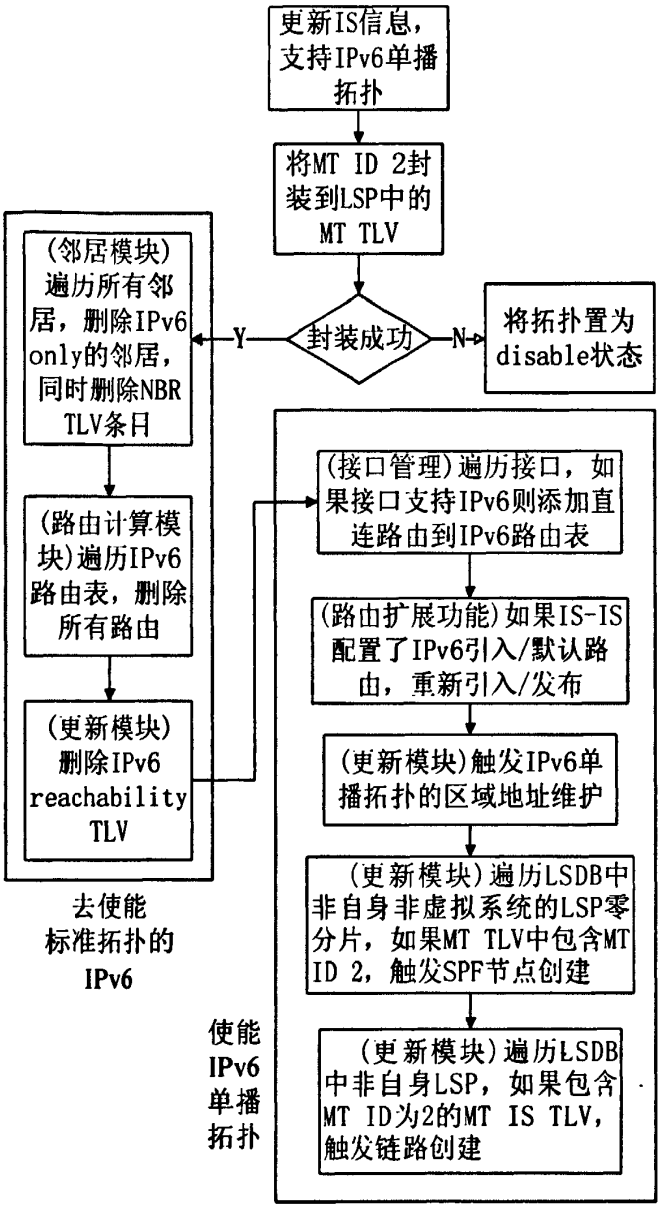

图 3.7 配置 IPv6 单播拓扑的功能流程

Figure3.7 Configuration of IPv6 unicast topology

4 IS-IS 支持多拓扑的验证

本章通过专门设计的组网和命令配置，以路由器 display 命令显示的 LSDB 和路由表验证第三章中更新模块及配置/取消多拓扑功能的实现。

4.1 广播网中 LSP 更新验证

为验证 DIS 支持多拓扑功能，搭建三台路由器通过以太网互联的组网环境，如图 4.1 所示：

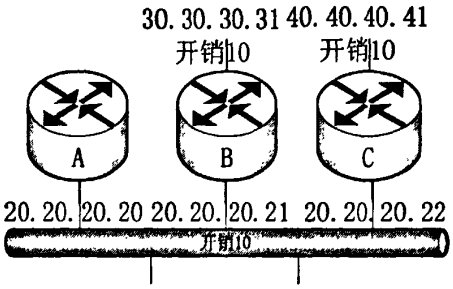

图 4.1 组网图 a

Figure4.1 Network a

A 先配置拓扑 test，然后 B 和 C 再配置拓扑 test，路由器各接口也配置此拓扑。B 和 C 配置拓扑后将 test 拓扑封装到 LSP 的 MT TLV 中，test 拓扑下的邻居封装到 MT IS TLV 中，将 test 拓扑下的 IP 路由封装到 MT IP TLV 中。A 收到 B 和 C 的 LSP 后触发多拓扑节点，链路，叶子节点新增并触发路由计算，如果三种 TLV 更新功能错误，A 在拓扑 test 中无法获得通往 B 和 C 非以太网接口的路由。

由于在广播网中伪节点作为星形结构的中心联系其它节点，如果 DIS 生成的伪节点报文多拓扑更新功能错误，拓扑 test 将呈现不连通状态，A 在拓扑 test 中同样无法获得通往 B 和 C 非以太网接口的路由。

A 的拓扑 test 路由表如图 4.2 所示，路由表中包含了 B 和 C 非以太网接口的路由 30.30.30.0/24 和 40.40.40.0/24。

```
[H3C]display isis route 1 ipv4-unicast test

                  Route information for ISIS(1)
                  -----------------------------

             ISIS(1) IPv4 MT(test-6) Level-1 Forwarding Table
             ------------------------------------------------

IPV4 Destination    IntCost    ExtCost ExitInterface   NextHop         Flags
-------------------------------------------------------------------------------
40.40.40.0/24       20         NULL    Eth0/1/0        20.20.20.22     R/L/-
30.30.30.0/24       20         NULL    Eth0/1/0        20.20.20.21     R/L/-
20.20.20.0/24       10         NULL    Eth0/1/0        Direct          D/L/-

     Flags: D-Direct, R-Added to RM, L-Advertised in LSPs, U-Up/Down Bit Set
```

图 4.2 A 的拓扑 test 路由表（新增）

Figure4.2 MT test RIB of A (add)

在 DIS 节点 B 取消拓扑 test，此时 DIS 节点不会发生变化，伪节点报文不会变化，test 拓扑依然连通，如果功能错误，A 在拓扑 test 中将无法获得通往 C 非以太网接口的路由。

B 取消拓扑 test 后，其 LSP 中的 MT TLV 将删除拓扑 test，并删除拓扑 test 对应的 MT IS TLV 和 MT IP TLV。A 收到 B 的 LSP 后，触发多拓扑节点，链路，叶子节点删除并触发路由计算，如果三种 TLV 更新功能错误，A 在拓扑 test 中仍然保留通往 B 非以太网接口的路由。

A 的拓扑 test 路由表如图 4.3 所示，路由表中包含了 C 非以太网接口的路由 40.40.40.0/24，B 的路由 30.30.30.0/24 消失。

```
[H3C]display isis route 1 ipv4-unicast test

                  Route information for ISIS(1)
                  -----------------------------

             ISIS(1) IPv4 MT(test-6) Level-1 Forwarding Table
             ------------------------------------------------

IPV4 Destination    IntCost    ExtCost ExitInterface   NextHop         Flags
-------------------------------------------------------------------------------
40.40.40.0/24       20         NULL    Eth0/1/0        20.20.20.22     R/L/-
20.20.20.0/24       10         NULL    Eth0/1/0        Direct          D/L/-

     Flags: D-Direct, R-Added to RM, L-Advertised in LSPs, U-Up/Down Bit Set
```

图 4.3 A 的拓扑 test 路由表（删除）

Figure4.3 MT test RIB of A (delete)

A 的标准拓扑的路由表如图 4.4 所示，路由表中依然包含 B 的路由 30.30.30.0/24，说明更新和路由计算是分拓扑进行的，拓扑之间没有干扰。

```
[H3C]display isis route 1

                Route information for ISIS(1)
                -----------------------------

               ISIS(1) IPv4 Level-1 Forwarding Table
               -------------------------------------

IPV4 Destination     IntCost     ExtCost ExitInterface   NextHop         Flags
--------------------------------------------------------------------------------
40.40.40.0/24        20          NULL    Eth0/1/0        20.20.20.22     R/L/-
30.30.30.0/24        20          NULL    Eth0/1/0        20.20.20.21     R/L/-
20.20.20.0/24        10          NULL    Eth0/1/0        Direct          D/L/-

     Flags: D-Direct, R-Added to RM, L-Advertised in LSPs, U-Up/Down Bit Set
```

图 4.4 A 的标准拓扑路由表

Figure4.4 IPv4 RIB of A

4.2 IPv4/IPv6 分拓扑验证

采用第一章中存在路由黑洞问题的 IPv4/IPv6 混合组网，如图 4.5 所示。

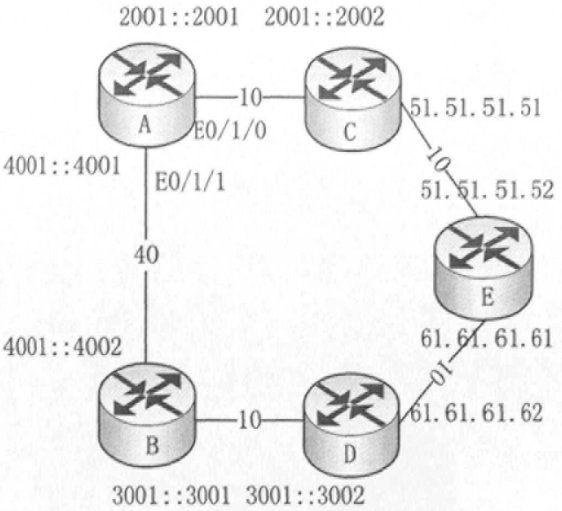

图 4.5 组网图 b

Figure4.5 Network b

未分拓扑情况下 A 的 IPv6 路由表如图 4.6 所示。从 A 到 D 的路径的出接口为 E0/1/0，开销为 40，以 D 为目的的 IPv6 报文将向 C 转发并被 C 丢弃。

```
[H3C]display isis route ipv6

                    Route information for ISIS(1)
                    -----------------------------

                  ISIS(1) IPv6 Level-1 Forwarding Table
                  -------------------------------------
Destination: 4001::                             PrefixLen: 64
Flag       : D/L/-                              Cost     : 10
Next Hop   : Direct                             Interface: Eth0/1/1

Destination: 3001::                             PrefixLen: 64
Flag       : R/L/-                              Cost     : 40
Next Hop   : FE80::200:5EFF:FE71:4204           Interface: Eth0/1/0

Destination: 2001::                             PrefixLen: 64
Flag       : D/L/-                              Cost     : 10
Next Hop   : Direct                             Interface: Eth0/1/0

    Flags: D-Direct, R-Added to RM, L-Advertised in LSPs, U-Up/Down Bit Set
```

图 4.6 A 的 IPv6 拓扑路由表（未配置多拓扑）

Figure4.6 IPv6 RIB of A (original)

分拓扑后 A 的 IPv6 路由表如图 4.7 所示，从 A 到 D 的路径的出接口变为 E0/1/1，开销值变为 50，以 D 为目标的 IPv6 报文向 B 转发最后到达 D，选路正确，避免了路由黑洞。

```
[H3C-Ethernet0/1/0]dis is rou ipv6

                    Route information for ISIS(1)
                    -----------------------------

                  ISIS(1) IPv6 Level-1 Forwarding Table
                  -------------------------------------
Destination: 4001::                             PrefixLen: 64
Flag       : D/L/-                              Cost     : 10
Next Hop   : Direct                             Interface: Eth0/1/1

Destination: 3001::                             PrefixLen: 64
Flag       : R/L/-                              Cost     : 50
Next Hop   : FE80::200:5EFF:FF83:4931           Interface: Eth0/1/1

Destination: 2001::                             PrefixLen: 64
Flag       : D/L/-                              Cost     : 10
Next Hop   : Direct                             Interface: Eth0/1/0

    Flags: D-Direct, R-Added to RM, L-Advertised in LSPs, U-Up/Down Bit Set
```

图 4.7 A 的 IPv6 拓扑路由表（配置多拓扑）

Figure4.7 IPv6 RIB of A (MT)

4.3 Overload 维护验证

OVERLOAD 只与路由器自身情况有关，因此采用简单组网，如 4.8 所示

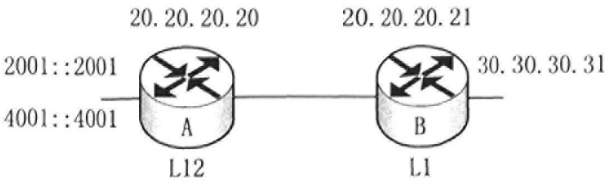

图 4.8 组网图 c

Figure4.8 Network c

路由器未处于 OVERLOAD 状态时 LSP 中发布自身的多拓扑直连路由和引入路由，L1-2 路由器还要在 L2 LSP 中发布从 L1 学到的多拓扑路由。A 配置一条 IP 静态路由 50.50.50.0/24，并配置静态路由引入到 IS-IS；A 和 B 都配置拓扑 test(6)，A 为 L1-2 路由器，A 和 B 建立 L1 邻居，A 能够学到 B 的拓扑 test 路由 30.30.30.0/24；因此 A 的 L2 LSP 中应该包含这两条路由，如图 4.9 所示。

```
0000.0000.0001.00-01* 0x00000037   0x3c9d        1177          97       0/0/0
 SOURCE       0000.0000.0001.00
 +IP-Extended
   20.20.20.0       255.255.255.0   COST: 10
 +IP-Extended
   30.30.30.0       255.255.255.0   COST: 20
 IPV4 UNICAST
   30.30.30.0       255.255.255.0   COST: 20         MTR ID: 6
 IPV4 UNICAST
   20.20.20.0       255.255.255.0   COST: 10         MTR ID: 6
 IPV6
   2001::/64                        COST: 10
 IPV6
   4001::/64                        COST: 10

0000.0000.0001.00-02* 0x00000001   0xb30a        997           37       0/0/0
 SOURCE       0000.0000.0001.00
 +IP-Extended
   50.50.50.0       255.255.255.0   COST: 0

    *-Self LSP, +-Self LSP(Extended), ATT-Attached, P-Partition, OL-Overload
```

图 4.9 未处于 OVERLOAD 状态时 A 的 L2 LSP

Figure4.9 L2 LSP of A (not overloaded)

进入 OVERLOAD 状态后，路由器 LSP 只发布自身多拓扑直连路由，且多拓扑 OVERLOAD 标记需要置位。在 A 配置 set OVERLOAD 命令，A 的 L2 LSP 如图 4.10 所示，引入路由 50.50.50.0/24 和学到路由 30.30.30.0/24 已取消发布，LSP 零分片中 MT TLV 的 OVERLOAD 标记已置位。

```
SOURCE          0000.0000.0001.00
NLPID           IPV4
NLPID           IPV6
AREA ADDR       10
INTF ADDR       20.20.20.20
INTF ADDR V6    2001::2001
INTF ADDR V6    4001::4001
MTR ID          00 0/0/1
MTR ID          06 0/0/1
+NBR  ID
  0000.0000.0002.01                COST: 10
IPV4 UNICAST NBR ID
  0000.0000.0002.01                COST: 10          MTR ID: 6

0000.0000.0001.00-01* 0x00000039   0x1369        1180          79       0/0/0
 SOURCE         0000.0000.0001.00
 +IP-Extended
  *20.20.20.0       255.255.255.0   COST: 10
 IPV4 UNICAST
  20.20.20.0        255.255.255.0   COST: 10          MTR ID: 6
 IPV6
  2001::/64                         COST: 10
 IPV6
  4001::/64                         COST: 10

0000.0000.0001.00-02* 0x00000001   0xe912        0 (1178)      27       0/0/0

   *-Self LSP, +-Self LSP(Extended), ATT-Attached, P-Partition, OL-Overload
```

图 4.10 处于 OVERLOAD 状态时 A 的 L2 LSP

Figure4.10 L2 LSP of A (overload)

在 A 配置 undo set OVERLOAD 命令，A 退出 OVERLOAD 状态后其 L2 LSP 恢复成图 4.9 所示的情况。

4.4 区域地址维护验证

按照图 4.11 组网，A 与 B 都配置拓扑 test(6)，二者之间建立 L2 邻居。

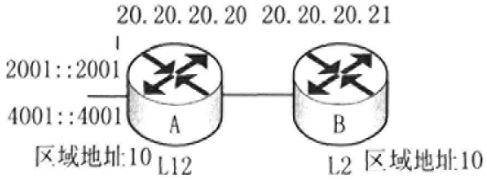

图 4.11 组网图 d

Figure4.11 Network d

B 配置区域地址 20，A 从 B 的 L2 LSP 学到 20 并触发区域地址计算。由于 20 无法通过 L1 获取，表明 A 有区域外邻居，发布带有 ATT 的 L1 LSP。A 的 L1 LSP 如图 4.12 所示，所有拓扑的 ATT 标记已置位。

```
                    Level-1 Link State Database

LSPID                 Seq Num      Checksum      Holdtime      Length  ATT/P/OL
-------------------------------------------------------------------------------

0000.0000.0001.00-00* 0x0000007e   0x511a        1186          81      1/0/0
 SOURCE       0000.0000.0001.00
 NLPID        IPV4
 NLPID        IPV6
 AREA ADDR    10
 INTF ADDR    20.20.20.20
 INTF ADDR V6 2001::2001
 INTF ADDR V6 4001::4001
 MTR ID       00 1/0/0
 MTR ID       06 1/0/0
```

图 4.12 处于 ATT 状态时 A 的 L1 LSP

Figure4.12 L1 LSP of A (attached)

取消 B 的 test 拓扑，B 的 MT TLV 中删除拓扑 test，A 收到 B 的 L2 LSP 后需要在拓扑 test 中排除从 B 学到的区域地址（10、20）并进行区域地址计算。由于不再存在只能从 L2 中获取的地址，A 没有区域外邻居，取消发布 ATT 标记。A 的 L1 LSP 如图 4.13 所示，拓扑 test 的 ATT 标记已清零。

```
                    Level-1 Link State Database

LSPID                 Seq Num      Checksum      Holdtime      Length  ATT/P/OL
-------------------------------------------------------------------------------

0000.0000.0001.00-00* 0x0000007f   0x3133        1028          96      1/0/0
 SOURCE       0000.0000.0001.00
 NLPID        IPV4
 NLPID        IPV6
 AREA ADDR    10
 INTF ADDR    20.20.20.20
 INTF ADDR V6 2001::2001
 INTF ADDR V6 4001::4001
 MTR ID       00 1/0/0
 MTR ID       06 0/0/0
```

图 4.13 拓扑 test 退出 ATT 状态后 A 的 L1 LSP

Figure4.13 L1 LSP of A (not attached in test)

4.5 配置多拓扑验证

按照图 4.8 简单组网，延长 B 的 LSP 生存和重传时间到 60 分钟左右。B 先配置拓扑 test(6)，待网络稳定后，A 再配置拓扑 test。A 直接从 LSDB 中提取 B 的属于拓扑 test 的链路状态信息进行更新，而不是等待 B 重新传递这些信息。A 拓扑 test 的 L1 路由表如图 4.14 所示，从路由 B 学到的路由 30.30.30.0/24 经过一次路由计算的时间（30 秒）后出现在路由表中，而不是几十分钟后。

```
                    Route information for ISIS(1)
                    -----------------------------

                ISIS(1) IPv4 MT(test-6) Level-1 Forwarding Table
                --------------------------------------------------
IPv4 Destination    IntCost    ExtCost ExitInterface   NextHop         Flags
-------------------------------------------------------------------------------
20.20.20.0/24       10         NULL    Eth0/1/0        Direct          D/L/-
30.30.30.0/24       20         NULL    Eth0/1/0        20.20.20.21     D/L/-
```

图 4.14 A 的拓扑 test 比 B 晚配置情况下的 L1 路由表

Figure4.14 MT test L1 RIB of A (A configured MT test later than B)

利用脚本语言编程为 A 添加 400 个回环接口，每个接口配置 IP 地址并配置 IS-IS，此时 LSP 零分片将被接口地址 TLV 装满，配置拓扑 test。此时拓扑 test 处于 disable 状态，不会执行路由功能。A 的的邻居关系如图 4.15 所示，只有邻居 B 的 IIH 报文中包含拓扑 test，A 不包含。

```
[H3C-isis-1]dis is pe ver

                        Peer information for ISIS(1)
                        ----------------------------

 System Id: 0000.0000.0002
 Interface: Eth0/1/0                Circuit Id:  0000.0000.0002.01
 State: Up     HoldTime: 9s         Type: L1(L1L2)      PRI: 64
 Area Address(es):10
 Peer IP Address(es): 20.20.20.21
 Uptime: 00:09:58
 Adj Protocol:  IPv4
 Peer Topology:
 6
```

图 4.15 A 的邻居关系

Figure4.15 The adjacency of A

5 多拓扑扩展研究

5.1 IPv6 孤岛间互通的最短路径选择

在 IPv4、IPv6 混合组网时，可能网络中并非所有路由器都支持 IPv6 协议，利用 IS-IS MTR 扩展功能虽然可以实现 IPv4、IPv6 分拓扑计算，避免出现路由黑洞，但 IPv4 和 IPv6 拓扑之间依然无法互通。如果路由域中任意两台支持 IPv6 协议的路由器之间存在无法利用 IPv6 拓扑转发报文的现象，即 IPv6 拓扑被 IPv4 拓扑分隔为不连通的部分,每个连通区域被称为 IPv6 孤岛。IP/IPv6 双协议栈、隧道[19]和 IPv4/IPv6 地址转换（例如 SIIT[20]和 NAT-PT[21]）都是常用的解决 IPv6 孤岛之间互通的方式。

5.1.1 应用隧道技术的路径选择

隧道技术是通过将 IPv6 报文利用 IP 头部进行二次封装，包括配置隧道和自动隧道两种，配置隧道只能实现点到点的连接，要求隧道两端的节点都支持双协议栈；自动隧道可以实现点到多点的连接，但要求 IPv6 地址必须兼容 IPv4 地址。在每个 IPv6 孤岛中选定一个或多个路由器配置隧道出口，手动建立隧道连接，每条隧道开销可以通过 IPv4 拓扑直接算出。将隧道看做 IPv6 拓扑中的链路，则 IPv6 拓扑连通，通过路由计算可以获取到达每个节点的最短路径。

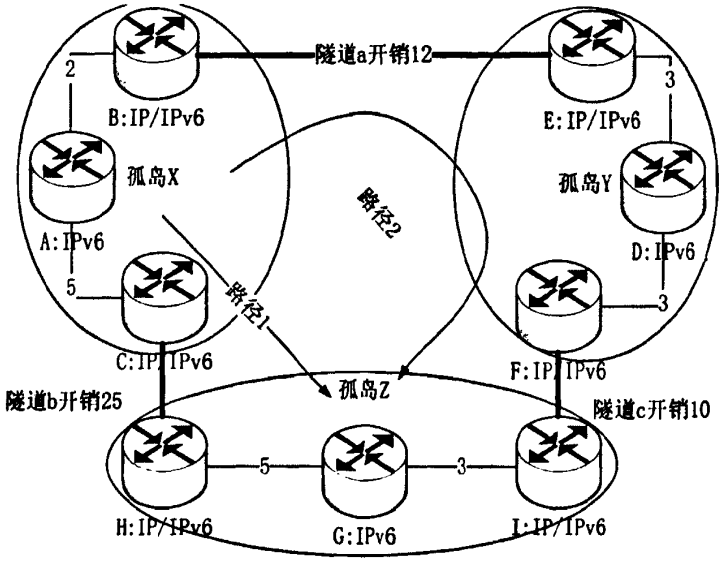

图 5.1 利用隧道解决 IPv6 孤岛间互通的路径选择

Figure5.1 Optimum path in Tunnel solution of IPv6 Island

如图 4.1 所示，A,B 和 C 属于孤岛 X，D,E 和 F 属于孤岛 Y，G,H 和 I 属于孤岛 Z。B 和 E 之间配置隧道 a，C 和 H 之间配置隧道 b，F 和 I 之间配置隧道 c。从 A 到 G 沿路径 2 的开销为 34，沿路径 1 的开销为 35，因此从 A 到 G 的报文通过路径 2 转发。由于隧道封装也需要占用资源（时间，空间），计算路径时可以既考虑开销又考虑通过隧道的次数，即

$$NewCost=Cost+\alpha*Times \qquad ...(5.1)$$

其中 α 为隧道封装解封的次数所占权重。当 α 大于 1 时，路径 1 的开销小于路线 2，报文通过路径 1 转发。设置高 α 值可以保证 IPv6 报文优先根据 IPv6 拓扑转发。

5.1.2 应用 SIIT 的路径选择

SIIT（无状态 IP、ICMP 翻译技术）是一种简单的 IPv4/IPv6 地址转换方式，将节点的 IPv6 地址配置为包含 IPv4 地址的形式，转发报文时如果发现目的地址为 IPv6 节点的 IPv4 地址或是 IPv4 节点的 IPv6 地址，通过加前缀或截短来翻译地址，转换协议报文头。SIIT 适用于除转换路由器外只支持一种协议的路由器占多数的网络环境下 IPv4 拓扑与 IPv6 拓扑之间的互通。

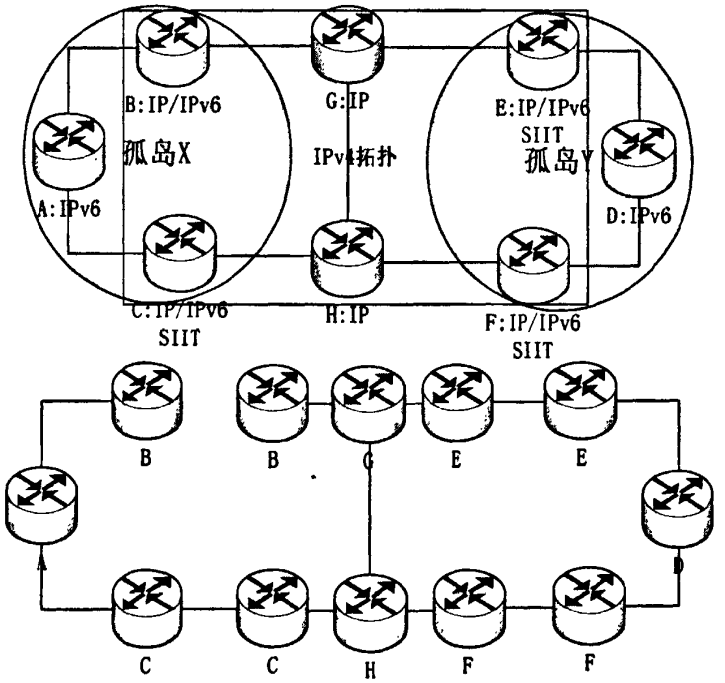

图 5.2 利用 SIIT 解决 IPv6 孤岛间互通的路径选择

Figure5.2 Optimum path in SIIT solution of IPv6 Island

图 4.2 表示的是在配置 SIIT 的网络中，选择 IPv6 路由之间的最短路径应用的

拓扑图。其中 A, B 和 C 属于孤岛 X，D, E 和 F 属于孤岛 Y，二者都包含在 IPv6 拓扑中，B, C, E, F, G 和 H 包含在 IPv4 拓扑中。配置了 SIIT 后，IPv6 报文转发可以通过 IPv4 拓扑，但与隧道不同的是，报文进入 IPv4 拓扑后出口不是唯一的，因此报文通过 IPv4 拓扑的开销无法确定。为了保证报文在 IPv4 拓扑和 IPv6 拓扑中的转发路径开销和最小，将 IPv4 拓扑和 IPv6 拓扑组成一张拓扑，配置了双协议栈的节点在拓扑图中出现了两次。配置了 SIIT 的路由器必须是双协议栈路由器，而且是 IPv4 和 IPv6 拓扑的连接点，因此在拓扑中具有 SIIT 功能的重复节点之间添加链路保证组合拓扑的连通。若要在选路时考虑地址转换次数的因素，只需为这些链路手动配置开销值即可。

5.2 利用多拓扑实现静态流量工程

路由协议提供的路径选择都是基于目的地址和最短路径进行的，没有考虑链路容量。这样做造成流量集中于网络中开销比较低的链路，即便低开销值的链路有较高的带宽，但在流量高峰期这些链路仍然会因负载过重造成时延丢包等问题。而最短路径之外的链路没有用来转发流量，处于带宽利用率不足的情况。随着 IP 网络对服务质量要求的提高，控制流量转发路径保证链路合理利用的机制被引入，即流量工程。流量工程分为两种，动态 TE（在线 TE）和静态 TE（离线 TE）。静态 TE 是指根据一种网络拓扑以及流量分布情况进行流量控制的办法。静态 TE 从全局考虑，能够最大程度降低网络总过载量，适用于流量稳定的网络环境。动态 TE 是指根据网络中不断变化的流量分布情况动态调整流量转发路径的办法。

5.2.1 相关方案

应用最广泛的流量工程技术是 MPLS（多协议标签交换）。MPLS 的主要功能包括：报文转发功能、信息发布功能和路径选择功能。报文转发功能通过封装和处理 MPLS 头部引导 IP 报文按 MPLS 选定路径通过网络，而不受到达目的地址的最短路径影响。信息发布功能定期测量链路利用率，缓冲器占用等链路负载情况并利用路由协议报文发布。路径选择功能分为在线 TE 和离线 TE，在线 TE 基于链路负载信息采用 CSPF（约束最短路径优先）算法在带宽限制情况下选定通往其它节点的路径，但处理目的节点的次序将影响路径的选定，先处理的节点选路时可以享受更多的带宽资源；离线 TE 则从全局角度入手，同时考虑每条链路限制和每条源-目的流的要求，实现网络性能的整体优化。

Amund Kvalbein 和 Olav Lysne[9]提出了利用多拓扑实现流量工程的方案。多拓扑本身可以为同一个目的地址提供多条转发路径，报文在源路由通过选取拓扑确定发送路径，在拓扑内按照目的地址进行转发，因此不需要类似 MPLS 的额外报头信息封装。MPLS 只能应用于 IP 网络中，多拓扑 TE 还能应用于 IPv6 网络环境。多拓扑 TE 也需要定期检测每条链路的负载并发布，但是仅仅以带宽占用情况代表链路负载。

Amund Kvalbein 和 Olav Lysne 的多拓扑 TE 方案的路径选择功能分为两步：构建拓扑和选择转发拓扑。多拓扑构建的原则包括：

- ◆ 每个拓扑必须包括全部所有节点且连通，任意两点间存在路径。
- ◆ 路由域中每条链路只能从一个拓扑中被移除。

在此基础上，离线 TE 的转发拓扑选择方案是：所有流量先依靠标准拓扑转发，将过载最严重的链路上的流量一个接一个移动到不存在此链路的拓扑中转发，直到网络总过载量不再减少。

在线 TE 的转发拓扑选择方案是：所有流量先依靠标准拓扑转发，当某条链路的流量超过预设上限，则将通过此链路的流量一个接一个移动到不存在此链路的拓扑中转发；当某条链路的流量低于下限，则做相反的流量移动。

Amund Kvalbein 和 Olav Lysne 的多拓扑离线 TE 方案存在一些不足：

- ◆ 新拓扑的结构只由网络拓扑决定，无法结合不同的流量输入降低网络过载程度。
- ◆ 转移过载最重链路的流量不一定能达到最大利用率。
- ◆ 无法确定利用几个拓扑实现流量工程能达到最优效果。

5.2.2 方案设计

待均衡网络的节点数为 n，边数为 m，链路容量 Cap_{m*1}，网络拓扑邻接矩阵 $A_{0(n*n)}$，数据流个数 t，数据流流量 $Totalflow_{(t*1)}$。假设以 N 个额外拓扑实现流量工程，每个额外拓扑的连接矩阵 $A_{i(n*n)}$是在网络拓扑基础上重新设置了链路开销。根据 $A_{i(n*n)}$计算每个节点的最短路径树，进而获得每条链路可能经过的源-目的流，以矩阵 $F_{i(m*t)}$记录。拓扑中链路负载情况表示为：

$$F_{i\ (m*t)} * flow_{i((t*1)} = Burden_{i\ (m*1)} \quad ...(5.2)$$

$$\sum_{i=1}^{k} flow_{i((t*1)} = Totalflow_{(t*1)} \quad ...(5.3)$$

其中 $flow_i$ 代表 t 条源-目的流在拓扑 i 中的流量，$Burden_i$ 代表拓扑 i 中 m 条链路的负载。

流量工程的目的是为了降低网络总过载量，因此利用多拓扑实现静态 TE 可表

示为：对于给定的 n，m，Cap，A_0，t，$Totalflow$，获取满足公式 5.2 和 5.3 的 A_l 和 $flow_l$，并保证

$$\min(\sum_{j=1}^{m}(\sum_{i=1}^{k}(Burden_{ij})-Cap_j)) \quad ...(5.4)$$

j 满足 $\sum_{i=1}^{k}(Burden_{ij})-Cap_j>0$。

本文的多拓扑静态 TE 方案需针对 Amund K 方案的三大不足，即确定最优化流量工程效果所需的拓扑数量，根据网络中的流量构建拓扑，提出新的拓扑间流量分配方案。

5.2.2.1 确定流量工程所需拓扑数

鉴于不同的拓扑构建方式有不同的最优流量分配方案，一次性构建所有多拓扑的方式难以确定怎样的拓扑个数及结构能够保证流量工程的效果最佳。因此本文采用每次只构建一个拓扑，将原有拓扑中的流量部分分配到新拓扑中，重复进行构建拓扑和流量分配的过程，以贪婪思想提供近似的流量工程最优解。功能分解如图 5.3 所示。

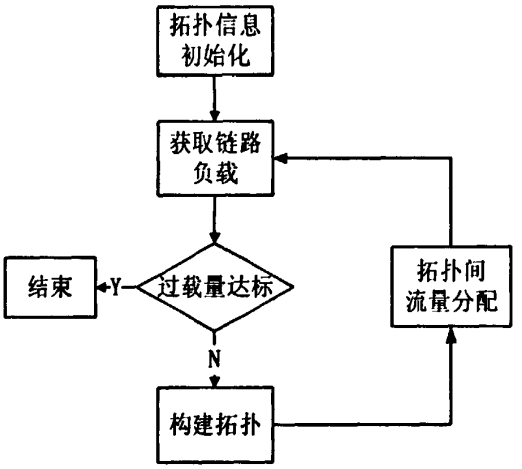

图 5.3 功能分解

Figure5.3 Function decomposition

首先初始化拓扑信息，包括网络拓扑，源-目的流和链路容量。然后进入循环：首先计算当前多拓扑环境下的链路负载及网络总过载量；然后将当前网络总过载量与结束条件相比较，达到条件则程序结束；最后根据当前链路负载情况构建新拓扑，将原有拓扑中的流量部分分到新拓扑中。

结束条件的设置有两种：一种是以过载量为标准，当网络总过载量为 0 或者连续两轮过载量的差值少于预期标准时结束程序；另一种是以链路总利用率为标准，链路利用率达到预期标准时结束程序。链路总利用率是链路有效负载与链路总容量的比值，链路有效负载等于链路总负载减去过载流量。链路总利用率可表

示为：

$$Utility = \frac{\sum_{i=0}^{N}\sum_{k=1}^{m}(Burden_{ik}) - \sum_{j=1}^{m}(\sum_{i=0}^{N}(Burden_{ij}) - Cap_j))}{\sum_{k=1}^{m} Cap_k} \quad ...(5.5)$$

其中j满足$\sum_{i=1}^{k}(Burden_{ij}) - Cap_j > 0$。初次进入循环时根据链路总负载与总容量的比值确定预期的链路总利用率。考虑到不同网络拓扑能够达到的流量工程效果不同，预估的结束条件存在不确定性，设置拓扑总数上限充当额外的结束条件。

网络中每条链路的负载相当于每个拓扑中此链路负载之和，每个拓扑的链路负载可以单独计算。在每个拓扑中报文按照最短路径转发，因此计算链路负载时先要计算每个节点的最短路径树，以此为基础确定每条源-目的数据流的转发路径，并将数据流流量计入转发路径中每条链路的负载。

由于每次流量分配后拓扑的链路负载都会改变，而每次负载计算过程中最短路径都是不变的，只有源-目的流的流量发生变化。因此根据最短路径将拓扑中每条链路可能经过的所有源-目的流保存为中间结构，获取每条链路负载时只需将此结构记录的源-目的流在拓扑中的流量叠加。

5.2.2.2 构建拓扑

构建新拓扑的目的是使当前过载链路尽量少地参与报文转发。新拓扑在标准拓扑的基础上处理过载链路，方案有两种：

- ◆ 保证拓扑连通的前提下断开过载链路。
- ◆ 提高过载链路的开销值。

前者效果很直接，新拓扑中流量必然不会通过原过载链路转发，但存在不少缺点：拓扑构建复杂度较高，每断开一条链路都要检测拓扑是否连通；过载链路完全断开拓扑不连通时，需要一次构建多个拓扑，增大了流量分配的复杂度；极端情况下过载链路存在为保证网络连通的必要条件时，无法处理。后者比较简单灵活，只需要人为定制过载链路的开销值增加规则，还可以根据链路过载程度采用不同的开销值增量确保过载严重的链路更少参与转发，因此被采用。

5.2.2.3 拓扑间流量分配

拓扑间流量分配的目的是将过载链路的部分流量转移到新拓扑中，减轻过载链路的负载。流量的分配方式有两种：

- ◆ 将经过过载链路的流量按比例分配到新拓扑中。
- ◆ 以源-目的流为基本单位分配流量。

前者比较简单，但当拓扑较多时，源-目的流被分割的比较细碎，同一报文流在多个拓扑中通过不同路径转发会产生时延、乱序的问题，对接收缓存大小有要

求，因此不予采用。

后者需确定哪些流被转移到新拓扑，一种方法是以每条过载链路上的过载流量为上限，从经过链路的所有流中挑选一部分转移到新拓扑中。此方法的流分配基于静态的链路过载情况，忽略了流转移本身对其它过载链路负载的影响。挑选待转移的流没有简单有效的办法，具有随机性，难以保证过载链路的负载有效降低。

第二种方法是转移过载最重的链路的数据流，数据流转移根据流量从大到小顺序进行，转移判断条件是链路总利用率的提高，为 Amund Kvalbein 和 Olav Lysne 的多拓扑 TE 方案所采用。此方法忽略了轻度过载链路的流量均衡，且提出过载。

第三种方法也是采用贪婪算法思想分配数据流，但赋予所有经过过载链路的源-目的流尝试转移的机会，且数据流尝试转移按照随机顺序进行，链路总过载量减少则确认转移。此方法复杂度较高，每转移一条流都要计算网络总过载量的变化，但可以保证每次转移都能有效降低网络的总过载量，被本文采用。

应用方法二时如果过载最重链路的所有流量转移时都不会减少总过载量，流量分配过程将过早结束，过载较轻链路的流量得不到均衡，应用方法三可以解决此问题。如图 5.4 上所示，网络中存在四条流，流 A->E 流量为 1 单位，A->B，D->E 流量为 2 单位，B->D 流量为 3 单位。根据最短路径，四条流都在链路 AB，BD 和 DE 上转发，但三条链路容量均只有 2 单位，因此处于过载状态。其中 BD 过载最严重，按照方法二应先转移经过 BD 的流，但显然 BD 为左右两部分网络连通的桥梁，从左部到右部的流量必须经过 BD 转发，由于流量无法转移分配过程结束，网络总过载量为*(1+1+2) =4* 单位。采用方法三时，链路 AB 和 DE 上的流也被考虑进行转移，将流 AE 转移到新拓扑（如图 5.4 下所示）中，按照 A->C->B->D->F->E 的路径转发，此时链路 AB 和 DE 脱离过载状态，网络总过载量为 2 单位。

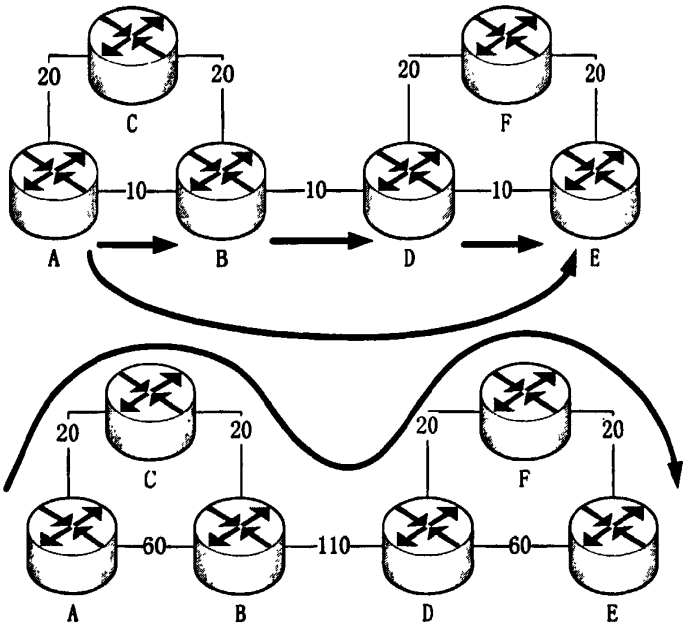

图 5.4 仿真流程图

Figure5.4 The process of Simulation

5.2.2.4 方案效果对比

Amund Kvalbein 和 Olav Lysne 的多拓扑离线 TE 方案根据人为定义的拓扑数一次构建所有拓扑，多拓扑的构建通过移除网络拓扑中的链路实现。针对如图 5.5 左所示的拓扑和初始流量，定义拓扑总数为 4，构建的多拓扑如图 5.5 右所示，与本例无关拓扑未画出。定义链路开销与容量的映射关系为:*容量=4-开销*。

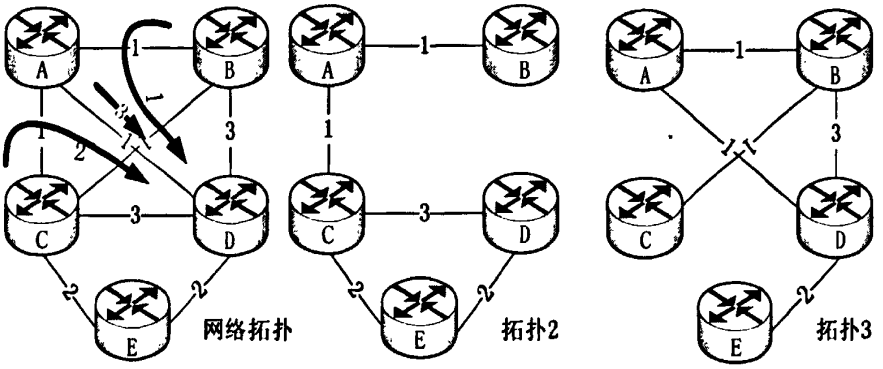

图 5.5 初始状态

Figure5.5 Original state

由于链路 AD 过载 3 单位流量，根据流量从大到小的顺序，先将链路 AD 上的流 AD 转移到拓扑 2 中，按照路径 A->C->D 转发，链路 AC 过载量 *2+3-3=2*，CD 过载量 *3-1=2*，总过载量增大，所以流 AD 取消转移。将流 CD 转移到拓扑 2 中，

按照路径 C->D 转发，链路 AD 过载量 *1+3-3=1*，CD 过载量 *2-1=1*，总过载量减少，流 CD 确认转移，流量分布如图 5.6 所示。

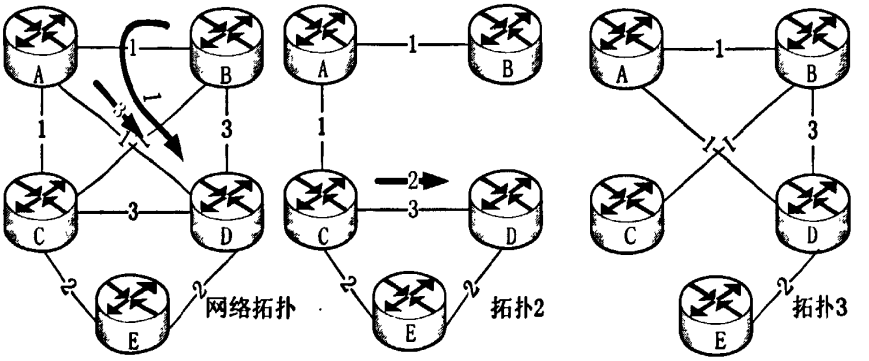

图 5.6 转移第一条流

Figure5.6 Transfer first flow

链路 AD 和链路 CD 过载 1 单位流量，先转移链路 AD 上的流 AD 到拓扑 2，总过载量不变，取消转移；转移流 BD 到拓扑 2，沿路径 B->A->C->D 转发，链路 CD 过载量 *1+2-1=2*，总过载量不变，取消转移；再转移链路 CD 上的流量 CD 到拓扑 3，按照路径 C->B->D 转发，链路 BD 过载量 *2-1=1*，链路总过载量不变，取消转移，均衡过程结束。最终流量分布如图 5.6 所示，总过载量 2 单位。

本文方案每轮构建一个拓扑并分配流量。同样针对如图 5.5 左所示的网络拓扑和初始流量，由于链路 AD 过载 3 单位流量，调整链路 AD 开销生成拓扑 2，如图 5.7 所示。根据随机顺序，假设先转移链路 AD 上的流 CD 到拓扑 2，转发路径为 C->D，链路 AD 过载量 *3+1-3=1*，CD 过载量 *2-1=1*，总过载量减少，确认转移；再转移流 AD，路径 A->B->D，链路 AB 过载量 *3+1-3=1*，BD 过载量 *3-1=2*，CD 过载量为 1，总过载量增大，取消转移；最后转移流 BD，路径 B->D，只剩链路 CD 过载 1 单位流量，因此确认转移。

第一轮流量分配后，过载链路为 CD，构建拓扑 3，如图 5.8 所示。将链路 CD 上的流 CD 转移到拓扑 3 中，转发路径为 C->E->D，总过载量为 0，均衡过程结束。

与 Amund Kvalbein 和 Olav Lysne 方案相比，本文方案通过拓扑逐个构建确定了达到近似最优均衡效果所需拓扑数量，采用了与流量相关的拓扑构建机制以及更完备的流量转移机制。从两种方案针对同一拓扑的负载均衡效果可以看出，本文方案实现的静态 TE 效果更显著。

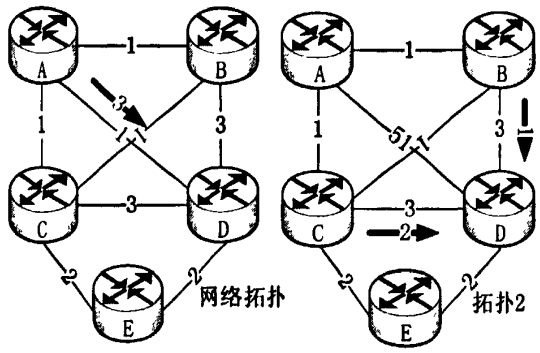

图 5.7 第一轮流量重分配

Figure5.7 Flow redistribution (first)

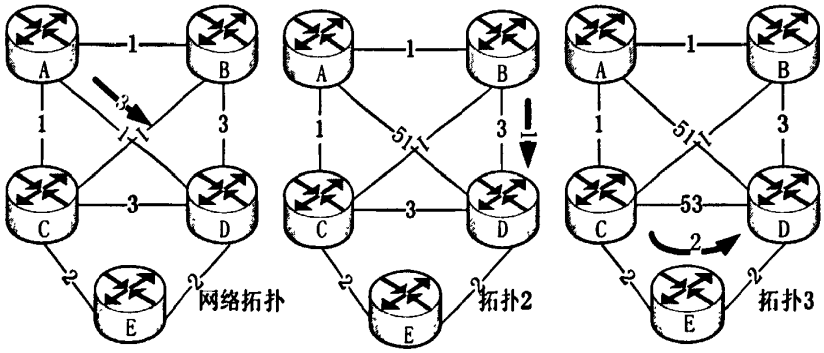

图 5.8 第二轮流量重分配

Figure5.8 Flow redistribution (second)

5.2.3 仿真实现

选择 MATLAB 为仿真工具，输入数据包括：以带权值的连接矩阵表示的自治系统网络拓扑，权值代表链路的开销，如果两个节点之间没有链路，则开销为 Inf；以 N*N 矩阵表示网络中数据流的流量，其中 N 为节点数，矩阵行数为数据流源节点，列数为目的节点。

输出数据：以连接矩阵表示的每个新增拓扑，负载均衡后每个拓扑对应的数据流矩阵，链路负载矩阵，链路总利用率。

设置的参数：构建新拓扑时过载链路开销增加规则，结束条件包括的链路总利用率和拓扑数量。

中间数据：节点到其它所有节点的最短路径树存储为记录每个节点的上一跳的向量，以 cell 矩阵的形式存储拓扑中每条链路上可能经过的数据流，每个 cell 中存储这些数据流的源节点和目的节点。

根据设计方案中的三大功能，封装了五个函数，Init 函数执行初始化操作，CalculateFlow 函数获取拓扑中每条链路上经过流的 cell 矩阵。AquireBurden 函数根据 cell 矩阵计算链路负载。ConstituteMT 函数执行构建拓扑功能，根据过载链路开销提升规则生成新拓扑。TransferFlow 函数执行拓扑间流量分配功能，从原有几个拓扑中基于总过载流量减少准则分配部分流到新拓扑。TransferFlow 函数的流程如表 5.1 所示。

表 5.1 TransferFlow 函数流程

TABLE5.1 The process of TransferFlow

```
For 所有过载链路 e_i
    For 所有已存在拓扑 tp_j
        通过拓扑 cell 矩阵获取流经 e_i 的所有流 f_ij
        将流组 f_ij 按流量由大到小排序
        For 流组 f_ij 中的所有流 f_ijk
            将 f_ijk 移动到新拓扑
            AquireBurden 获取当前链路负载
            If 链路总过载量不变或增大
                将 f_ijk 移回拓扑 tp_j
            Else
                记录链路总过载量
                记录新拓扑的链路负载
            End if
        End for
    End for
End for
```

仿真程序的流程如图 5.9 所示。

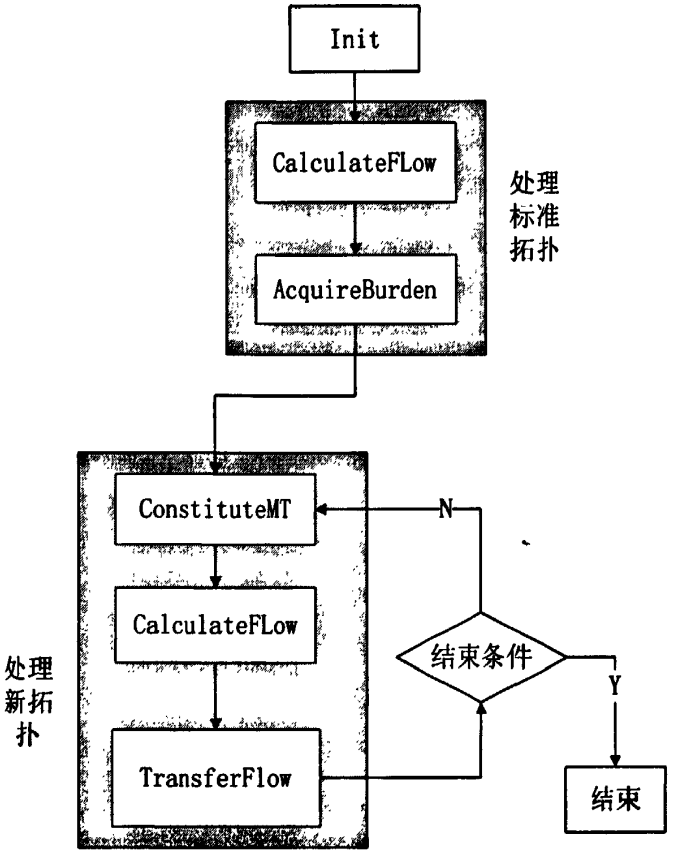

图 5.9 仿真流程图

Figure5.9 The process of Simulation

5.2.4 ER 随机拓扑效果验证

生成 50 个节点 150 条边的 ER 随机拓扑，每条链路的开销设置为 1-6 中的随机值。鉴于链路开销默认情况下与链路带宽相关，带宽越大链路容量越高且开销值越小，因此链路容量与链路开销的映射关系设计如表 5.2 所示：

表 5.2 链路容量与开销映射

TABLE5.2 Mapping between link capacity and cost

开销	1	2	3	4	5	6
容量	50	40	30	20	15	10

设置构建新拓扑时链路开销调整规则：过载量大于 30 开销增加 200，过载量在 10 到 30 之间开销增加 100，过载量小于 10 开销增加 50。随机生成数据流矩阵，根据链路总容量设计矩阵占空比和流量最大值。取占空比为 16/21，流量最大值为 4，启动仿真。图 5.5 为链路总容量为 4025，链路总负载为 4137 时采用两个拓扑

进行负载均衡前后每条链路负载对比。

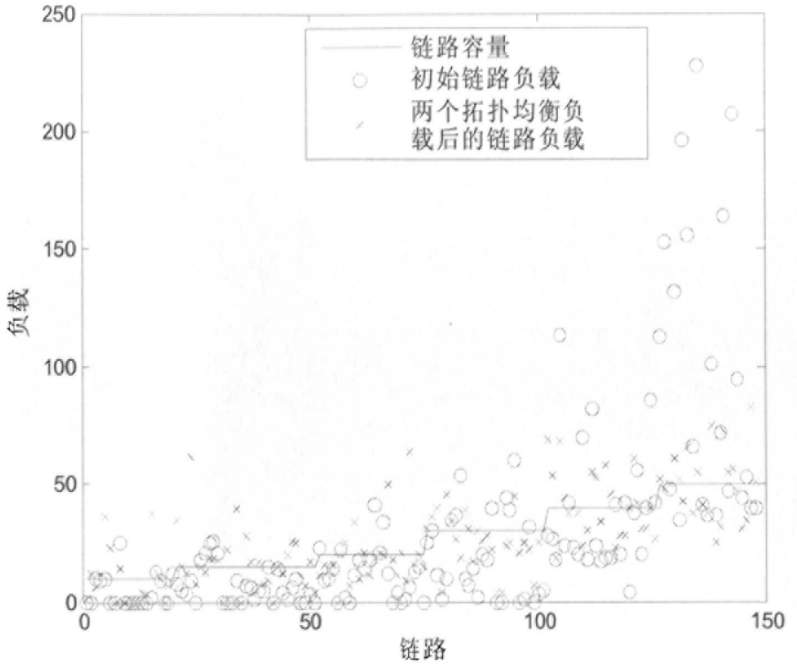

图 5.10 流量工程效果图 a

Figure5.10 The effect of traffic engineering (a)

从图 5.10 中可以看出初始状态下，数据流主要通过链路容量较高也就是开销较小的链路发送，尤其是容量为 60 的链路大部分都处于过载状态，最高负载能达到链路容量的 4 倍。大量容量为 10 和 20 的链路负载为 0，即没有被应用于流量转发。负载平衡主要是要针对这些链路的进行流量调整。利用两个拓扑转发流量后，大容量链路的高过载流量由其它链路分担而大量削减，而小容量链路的起用率提高，说明负载均衡起到了明显作用，但中等容量链路并不严重的过载现象并未减弱，小容量链路的过载情况增加了，说明本方案对控制小容量链路负载的作用有限。图 5.11 通过两个拓扑和三个拓扑进行负载均衡的链路负载对比。

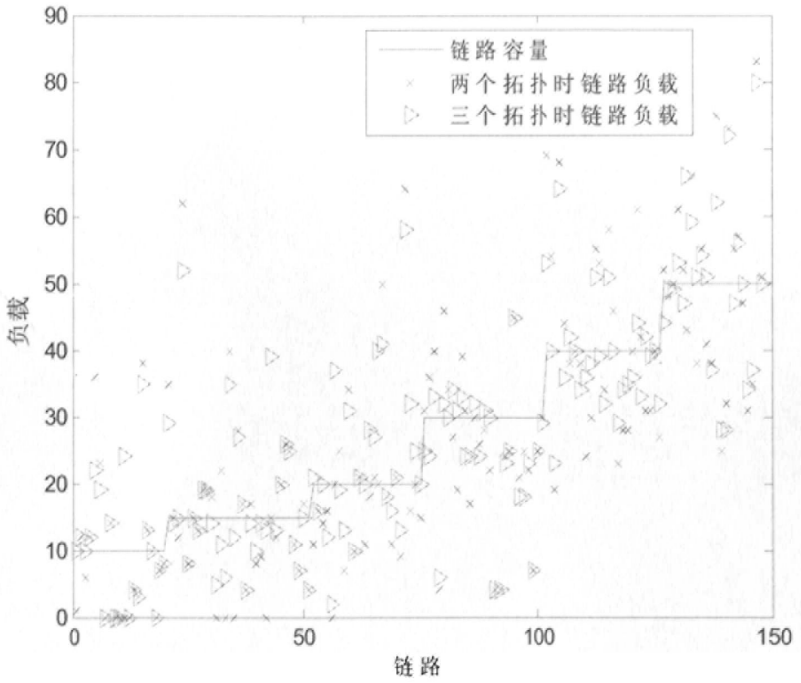

图 5.11 负载均衡效果图 b

Figure5.11 The effect of traffic engineering (b)

三个拓扑作负载均衡时，链路负载比两个拓扑分布的范围稍小，链路负载更集中于链路容量周围。负载低于容量的链路的负载量有所提升，过载状态的节点数变化不大，因此链路总利用率有所提升。三个拓扑时容量为 20 的链路已全部起用，容量为 10 的链路起用率仍十分有限。总体来讲，拓扑由两个增长到三个带来的负载均衡效果远低于从一个到两个。

为了验证本方案对网络的适用性，采用六种具有不同规模和稠密程度的随机网络测试本方案的负载均衡效果，如表 5.3 所示。对于所有的验证网络，每一轮构建拓扑并分配流量过程产生的负载均衡效果都不如前一轮，第一轮过后过载量明显降低，第二轮过后的过载量的减少量少于第一轮的 20%，第三轮过后过载量的减少量相当于第二轮的 20%-85%，第四轮过后过载量已没有明显变化，因此只需三轮均衡，利用三个额外拓扑就能够接近本方案的最优效果。

采用相同网络和流量分布测试 Amund Kvalbein 和 Olav Lysne[9]的方案的负载均衡效果，如表 5.4 所示。在点/边数为 40/80，40/120 和 80/400 的网络中，利用三个额外拓扑和四个额外拓扑实现流量工程的效果差别较明显，其它三种网络中效果差别不大。利用三个额外拓扑和四个额外拓扑的效果优劣与网络拓扑以及额外拓扑构建策略有关，无法根据网络特征提前判断。与表 5.3 相比较，对于除点/边数为 40/80 外的所有的网络，本文方案三轮均衡之后的过载量明显少于 Amund

Kvalbein 和 Olav Lysne 的方案，尤其对于规模较大的网络（80 个节点），验证了本方案对于不同网络结构和流量分布的适用性以及良好的流量工程效果。

表 5.3 针对不同网络的负载均衡效果

TABLE5.3 The effect of load-balance for different network

点/边	40/80	40/120	50/150	50/200	80/400	80/320
总负载/总容量	1230/1845	2413/3435	3935/4000	3296/5470	8881/10385	10629/8455
初始过载量	248	489	1217	479	2267	4844
第一轮后过载量	113	143	608	6	305	1987
第二轮后过载量	91	84	506	1	129	1623
第三轮后过载量	83	69	419	0	81	1502

表 5.4 本方案与 Amund K 的方案的负载均衡效果对比

TABLE5.4 The contrast of load-balance between our solution and Amund K's solution

点/边	40/80	40/120	50/150	50/200	80/400	80/320
初始过载量	248	489	1217	479	2267	4844
三个额外拓扑	50	183	647	17	702	3790
四个额外拓扑	130	132	698	15	1110	3501

6 结论

IS-IS 支持多拓扑使 IS-IS 拓扑可以在 IS-IS 路由域内根据人为意愿划分，每个拓扑与一个路由域功能相当，所有路由功能在拓扑内正常执行。IS-IS 支持多拓扑后通过将 IPv4 和 IPv6 划分为不同拓扑解决了原有的由于路径不区分 IPv4/IPv6 造成的 IPv6 路由黑洞问题。本文作者独立设计实现了更新模块支持多拓扑和配置/取消多拓扑功能，包括完善了多拓扑信息的存储结构；设计并实现了收到 LSP 后多拓扑信息的处理，LSP 报文失效时多拓扑信息的清除，自身多拓扑链路状态信息变化时多拓扑 TLV 的封装；更改了伪节点报文的更新流程，路由器 OVERLOAD 状态的维护流程，区域地址的维护和 ATT 状态判断；尤其是新增了一系列拓扑级别的更新操作。总之，既保障了更新模块支持多拓扑和配置/取消多拓扑功能的完整性，又考虑了处理效率以及和平台代码的兼容性。最后以专门设计的组网和命令行组合，通过路由器的 LSDB 和路由表的变化验证二部分功能的实现。

在此基础本文还进行了两项扩展研究：

1. 在配置 IS-IS 协议的大型自治系统中存在 IPv4/IPv6 混合组网时，针对隧道技术和 SIIT 技术两种实现 IPv6 孤岛间通信方式，提供了利用 IPv4 和 IPv6 拓扑组合选路以及利用新选路标准的方案，从理论上分析了在双协议网络环境中应用这两种方案进行最优路径选择相比以往方案的优势。

2. 利用多拓扑实现静态流量工程。本文先对以往流量工程的实现手段进行了分析，指出一些不足。提出一种基于多拓扑的静态流量工程方案，方案采用贪婪算法思想基于链路总利用率进行多轮的拓扑划分和流量分配，每轮均负责减轻上一轮负载均衡后的过载情况，最终达到近似最佳的流量工程效果。其中拓扑构建采用灵活的过载链路开销增长规则，流量分配则通过过载链路流量的转移尝试实现。利用 MATLAB 针对六种不同规模和密集程度的随机连通网络和随机源-目的流进行仿真研究，对比本方案和 Amund Kvalbein 和 Olav Lysne 方案的负载均衡效果，证明了本方案可以有效地实现流量工程。

本方案只是启发式的，尚有一些值得进一步研究之处：

首先，本方案能够与多路径相结合。以每条流为单元进行流量分配存在局限性，当网络中出现大型数据流时，分割流量并采取多路径传输才可能进一步降低网络总负载量，但需要注意多条路径之间的时延差别以保证接收缓存足够。

其次，为了使新拓扑更有效地完成负载分担的任务，根据链路过载情况调整链路开销的机制有待进一步研究。

参考文献

[1]. R. Callon. RFC1195 - -December 1990. Use of OSI IS-IS for Routing in TCP/IP and Dual Environments

[2]. ISO/IEC 10589 - -2002.11.Information technology-Telecommunications and information exchange between systems-Intermediate System to Intermediate System intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service

[3]. T. Przygienda, Z2 Sagl, N. Shen, Cisco Systems, N. Sheth, Juniper Networks. RFC5120 - -February 2008. M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)

[4]. T. Li, Redback Networks Inc, H. Smit. RFC5305 - -October 2008. IS-IS Extensions for Traffic Engineering

[5]. M. Shand, L. Ginsberg, Cisco Systems.RFC5306 - -October 2008. Restart Signaling for IS-IS

[6]. C. Hopps, Cisco Systems. RFC5308 - - October 2008. Routing IPv6 with IS-IS

[7]. D. McPherson, Ed., Arbor Networks, L. Ginsberg, S. Previdi, M. Shand, Cisco Systems. RFC5311 - -February 2009. Simplified Extension of Link State PDU (LSP) Space for IS-IS

[8]. Callon, R. and M. Suzuki, "A Framework for Layer 3 Provider Provisioned Virtual Private Networks", Work in Progress, July 2003

[9]. Amund Kvalbein, POlav Lysne, How can multi-topology routing be used for intradomain traffic engineering, Proceedings of the 2007 SIGCOMM workshop on Internet network management, Aug.2007

[10]. A. Nucci, B. Schroeder, S. Bhattacharyya, N. Taft, and C. Diot, "IGP Link Weight Assignment for Transient Link Failures," in 18th International Teletraffic Congress, Berlin, Germany, Aug. 2003

[11]. B. Fortz and M. Thorup, "Robust optimization of OSPF/IS-IS weights," in INOC, Oct. 2003, pp. 225–230.

[12]. A. Sridharan and R. Guerin, "Making IGP routing robust to link failures," in Proceedings of Networking, Waterloo, Canada, 2005.

[13]. D. Applegate and E. Cohen, "Making routing robust to changing traffic demands: Algorithms and evaluation," IEEE Transactions on Networking, vol. 14, no. 6, pp. 1193–1206, Dec. 2006.

[14]. H. Wang, H. Xie, L. Qiu, Y. R. Yang, Y. Zhang, and A. Greenberg, "COPE: traffic engineering in dynamic networks," in Proceedings SIGCOMM, 2006, pp. 99–110.

[15]. A. Elwalid, C. Jin, S. H. Low, and I. Widjaja, "MATE: MPLS adaptive traffic engineering," in Proceedings INFOCOM, 2001, pp. 1300–1309.

[16]. S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the tightrope: responsive yet stable traffic engineering," in Proceedings SIGCOMM, 2005, pp. 253–264.

[17]. Tarik CiciC, On basic properties of fault-tolerant multi-topology routing, Proceedings of the Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services, Feb.2006

[18].E. Rosen, Cisco Systems, Inc, A. Viswanathan, Force10 Networks, Inc, R. Callon, Juniper Networks, Inc, RFC3031 - -January 2001, Multiprotocol Label Switching Architecture

[19].R. Gilligan, FreeGate Corp, E. Nordmark, Sun Microsystems, Inc, RFC2893 - -August 2000. Transition Mechanisms for IPv6 Hosts and Routers

[20].E. Nordmark. RFC 2765 - - February 2000. Stateless IP/ICMP Translation Algorithm (SIIT). IETF

[21].G Tsirtsis and P. Srisuresh. RFC2766 - - February 2000. Network Address Translation- Protocol Translation (NAT-PT). IETF

作者简历

魏巍，男，26岁，2003.9-2007.7在北京交通大学获得本科学位，2008.9-2011.3在北京交通大学攻读硕士学位。攻读硕士学位期间曾撰写《因特网拓扑聚合特征分析》，发表在《2010海峡两岸科学与技术交流会论文集》；撰写《基于不同划分优度的因特网拓扑聚合特征研究》，投稿《交大学报》。在华三通信有限公司北研所实习一年零三个月。

学位论文数据集

表 1.1： 数据集页

<table>
<tr><td>关键词*</td><td>密级*</td><td>中图分类号*</td><td>UDC</td><td>论文资助</td></tr>
<tr><td>IS-IS；多拓扑路由；IPv6 孤岛；最短路径；静态流量工程</td><td>保密 公开</td><td>TP393</td><td></td><td></td></tr>
<tr><td colspan="2">学位授予单位名称*</td><td>学位授予单位代码*</td><td>学位类别*</td><td>学位级别*</td></tr>
<tr><td colspan="2">北京交通大学</td><td>10004</td><td>工学</td><td>硕士</td></tr>
<tr><td colspan="2">论文题名*</td><td colspan="2">并列题名</td><td>论文语种*</td></tr>
<tr><td colspan="2">IS-IS 支持多拓扑功能的实现与选路研究</td><td colspan="2"></td><td>中文</td></tr>
<tr><td>作者姓名*</td><td colspan="2">魏巍</td><td>学号*</td><td>08120095</td></tr>
<tr><td colspan="2">培养单位名称*</td><td>培养单位代码*</td><td>培养单位地址</td><td>邮编</td></tr>
<tr><td colspan="2">北京交通大学</td><td>10004</td><td>北京市海淀区西直门外上园村 3 号</td><td>100044</td></tr>
<tr><td colspan="2">学科专业*</td><td>研究方向*</td><td>学制*</td><td>学位授予年*</td></tr>
<tr><td colspan="2">通信与信息系统</td><td>网络理论与研究</td><td>两年半</td><td>2011</td></tr>
<tr><td>论文提交日期*</td><td colspan="4">2010.12.22</td></tr>
<tr><td>导师姓名*</td><td colspan="2">郭宇春</td><td>职称*</td><td>教授</td></tr>
<tr><td>评阅人</td><td colspan="2">答辩委员会主席*</td><td colspan="2">答辩委员会成员</td></tr>
<tr><td>赵永祥
郑宏云</td><td colspan="2">赵永祥</td><td colspan="2">张立军
郑宏云</td></tr>
<tr><td colspan="5">电子版论文提交格式 文本（ ） 图像（ ） 视频（ ） 音频（ ） 多媒体（ ） 其他（ ）
推荐格式：application/msword；application/pdf</td></tr>
<tr><td colspan="2">电子版论文出版（发布）者</td><td colspan="2">电子版论文出版（发布）地</td><td>权限声明</td></tr>
<tr><td colspan="2"></td><td colspan="2"></td><td></td></tr>
<tr><td>论文总页数*</td><td colspan="4">61</td></tr>
<tr><td colspan="5">共 33 项，其中带*为必填数据，为 22 项。</td></tr>
</table>