



(12) 发明专利

(10) 授权公告号 CN 110661700 B

(45) 授权公告日 2021. 10. 15

(21) 申请号 201810703774.9

H04L 12/723 (2013.01)

(22) 申请日 2018.06.30

审查员 高书姣

(65) 同一申请的已公布的文献号  
申请公布号 CN 110661700 A

(43) 申请公布日 2020.01.07

(73) 专利权人 华为技术有限公司  
地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

(72) 发明人 胡志波 姚俊达

(74) 专利代理机构 北京同达信恒知识产权代理有限公司 11291  
代理人 冯艳莲

(51) Int. Cl.

H04L 12/703 (2013.01)

H04L 12/707 (2013.01)

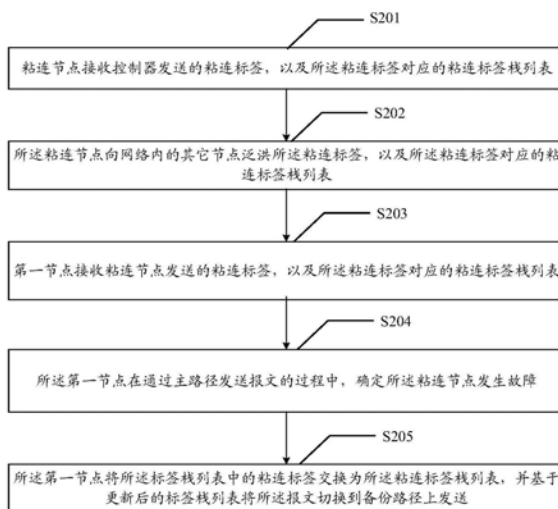
权利要求书3页 说明书18页 附图5页

(54) 发明名称

一种传输路径故障的处理方法、装置及系统

(57) 摘要

本申请公开了一种传输路径故障的处理方法、装置及系统,用以解决现有技术中存在故障节点的上一跳节点无法处理粘连标签而导致的转发报文失败的问题。方法包括:第一网络设备接收粘连网络设备发送的粘连标签,以及所述粘连标签对应的粘连标签栈列表;所述第一网络设备在通过主路径发送报文的过程中,确定所述粘连网络设备发生故障;其中,所述报文中包括用于指导所述报文在所述主路径上转发的标签栈列表,所述粘连网络设备为在所述主路径上所述第一网络设备的下一跳网络设备;所述第一网络设备将所述标签栈列表中的粘连标签交换为所述粘连标签栈列表,并基于更新后的标签栈列表将所述报文切换到备份路径上发送。



1. 一种传输路径故障的处理方法,其特征在于,包括:

第一网络设备接收粘连网络设备发送的粘连标签,以及所述粘连标签对应的粘连标签栈列表,所述粘连网络设备能够将报文中的所述粘连标签交换为所述粘连标签栈列表;

所述第一网络设备在通过主路径发送报文的过程中,确定所述粘连网络设备发生故障;

其中,所述报文中包括用于指导所述报文在所述主路径上转发的标签栈列表,所述粘连网络设备为在所述主路径上所述第一网络设备的下一跳网络设备;

所述第一网络设备将所述标签栈列表中的粘连标签交换为所述粘连标签栈列表,并基于更新后的标签栈列表将所述报文切换到备份路径上发送。

2. 如权利要求1所述的方法,其特征在于,所述第一网络设备接收粘连网络设备发送的粘连标签,以及粘连标签对应的粘连标签栈列表后,还包括:

所述第一网络设备将所述粘连标签以及对应的粘连标签栈列表保存在所述第一网络设备为所述粘连网络设备生成的标签信息表中;

所述第一网络设备将所述标签栈列表中的粘连标签交换为所述粘连标签栈列表,并将更新后的所述报文切换到备份路径上发送,包括:

所述第一网络设备从所述标签栈列表的栈顶获取第一标签,所述第一标签用于指示转发所述报文的下一跳网络设备为所述粘连网络设备;

所述第一网络设备基于备份转发表确定所述第一标签对应的第一转发行为;所述备份转发表用于指示所述备份路径;

所述第一转发行为用于指示查找所述标签信息表;

所述第一网络设备基于所述标签信息表确定位于所述粘连标签栈列表栈顶的第二标签,所述第二标签用于指示在所述主路径上,所述粘连网络设备的下一跳网络设备为第三网络设备;

所述第一网络设备将所述第二标签作为入标签,从所述备份转发表中确定由所述第一网络设备到所述第三网络设备的备份路径的出标签;

所述第一网络设备基于所述出标签向备份路径上所述第一网络设备的下一跳网络设备发送所述报文。

3. 如权利要求1或2所述的方法,其特征在于,第一网络设备接收粘连网络设备发送的粘连标签,以及粘连标签对应的粘连标签栈列表,包括:

所述第一网络设备接收所述粘连网络设备发送的标签长度内容TLV,所述TLV中包括所述粘连标签以及粘连标签对应的粘连标签栈列表。

4. 如权利要求3所述的方法,其特征在于,所述TLV为标签绑定SID/Label Binding TLV。

5. 如权利要求1或2所述的方法,其特征在于,还包括:

所述第一网络设备在接收到所述粘连网络设备在网络内泛洪的第三网络设备的节点段标识之后,向所述网络内的其它网络设备泛洪所述第三网络设备的节点段标识。

6. 如权利要求1或2所述的方法,其特征在于,还包括:

所述第一网络设备在通过备份路径发送报文时,所述第一网络设备接收所述粘连网络设备在故障恢复后发送的指示信息,所述指示信息用于指示所述粘连网络设备无法正确转

发报文,所述第一网络设备继续通过所述备份路径发送报文。

7. 一种传输路径故障的处理方法,其特征在于,包括:

粘连网络设备接收控制器发送的粘连标签,以及所述粘连标签对应的粘连标签栈列表,所述粘连网络设备能够将报文中的所述粘连标签交换为所述粘连标签栈列表;

所述粘连网络设备向网络内的其它网络设备泛洪所述粘连标签,以及所述粘连标签对应的粘连标签栈列表,使得所述网络内的其他网络设备获得所述粘连标签对应的所述粘连标签栈列表。

8. 一种传输路径故障的处理装置,其特征在于,所述装置应用第一网络设备,包括:

接收模块,用于接收粘连网络设备发送的粘连标签,以及粘连标签对应的粘连标签栈列表,所述粘连网络设备能够将报文中的所述粘连标签交换为所述粘连标签栈列表;

处理模块,用于在通过主路径发送报文的过程中,确定所述粘连网络设备发生故障;

其中,所述报文中包括用于指导所述报文在所述主路径上转发的标签栈列表,所述粘连网络设备为在所述主路径上所述第一网络设备的下一跳网络设备;

所述处理模块,还用于将所述标签栈列表中的粘连标签交换为所述粘连标签栈列表;

发送模块,用于基于所述处理模块更新后的标签栈列表将所述报文切换到备份路径上发送。

9. 如权利要求8所述的装置,其特征在于,所述处理模块,还用于在所述接收模块接收粘连网络设备发送的粘连标签,以及粘连标签对应的粘连标签栈列表后,将所述粘连标签以及对应的粘连标签栈列表保存在所述第一网络设备为所述粘连网络设备生成的标签信息表中;

所述处理模块,在将所述标签栈列表中的粘连标签交换为所述粘连标签栈列表时,具体用于从所述标签栈列表的栈顶获取第一标签,所述第一标签用于指示转发所述报文的下一跳网络设备为所述粘连网络设备;基于备份转发表确定所述第一标签对应的第一转发行为;所述备份转发表用于指示所述备份路径;所述第一转发行为用于指示查找所述标签信息表;基于所述标签信息表确定位于所述粘连标签栈列表栈顶的第二标签,所述第二标签用于指示在所述主路径上,所述粘连网络设备的下一跳网络设备为第三网络设备;将所述第二标签作为入标签,从所述备份转发表中确定由所述第一网络设备到所述第三网络设备的备份路径的出标签;

所述发送模块,具体用于基于所述出标签向备份路径上所述第一网络设备的下一跳网络设备发送所述报文。

10. 如权利要求8或9所述的装置,其特征在于,所述接收模块,具体用于接收所述粘连网络设备发送的标签长度内容TLV,所述TLV中包括所述粘连标签以及粘连标签对应的粘连标签栈列表。

11. 如权利要求10所述的装置,其特征在于,所述TLV为标签绑定SID/Label Binding TLV。

12. 如权利要求8或9所述的装置,其特征在于,所述发送模块,还用于在所述接收模块接收到所述粘连网络设备在网络内泛洪的第三网络设备的节点段标识之后,向所述网络内的其它网络设备泛洪所述第三网络设备的节点段标识。

13. 如权利要求8或9所述的装置,其特征在于,所述接收模块,还用于在所述发送模块

通过备份路径发送报文时,接收所述粘连网络设备在故障恢复后发送的指示信息,所述指示信息用于指示所述粘连网络设备无法正确转发报文,所述发送模块,还用于继续通过所述备份路径发送报文。

14. 一种传输路径故障的处理装置,其特征在于,包括:

接收模块,用于接收控制器发送的粘连标签,以及所述粘连标签对应的粘连标签栈列表,所述装置能够将报文中的所述粘连标签交换为所述粘连标签栈列表;

发送模块,用于向网络内的其它网络设备泛洪所述粘连标签,以及所述粘连标签对应的粘连标签栈列表,使得所述网络内的其他网络设备获得所述粘连标签对应的所述粘连标签栈列表。

15. 一种网络设备,其特征在于,包括:

通信接口、处理器和存储器;

所述存储器用于存储软件程序,所述处理器用于读取所述存储器中存储的软件程序,通过所述通信接口收发消息,并实现权利要求1至6任一项所述的方法。

16. 一种网络设备,其特征在于,包括:

通信接口、处理器和存储器;

所述存储器用于存储软件程序,所述处理器用于读取所述存储器中存储的软件程序,通过所述通信接口收发消息,并实现权利要求7所述的方法。

17. 一种网络系统,其特征在于,包括如权利要求8至13任一所述的装置和如权利要求14所述的装置;或者,所述系统包括如权利要求15所述的网络设备和如权利要求16所述的网络设备。

18. 一种计算机可读存储介质,其特征在于,所述计算机可读存储介质中存储软件程序,所述软件程序在被一个或多个处理器读取并执行时可实现权利要求1至7任一项所述的方法。

## 一种传输路径故障的处理方法、装置及系统

### 技术领域

[0001] 本申请涉及通信技术领域,尤其涉及一种传输路径故障的处理方法、装置及系统。

### 背景技术

[0002] 段路由流量工程(segment routing-traffic engineering,SR-TE)是使用内部网关协议(interior gateway protocol,IGP)或者边界网关协议(border gateway protocol,BGP)协议作为控制信令的一种新型的多协议标签交换(multiprotocol label switching,MPLS)TE隧道技术。控制器负责计算隧道的转发路径,并将与路径严格对应的标签栈下发给转发器。在SR-TE隧道的首端节点上,转发器根据标签栈,即可控制报文在网络中的传输路径。

[0003] 当主路径某一个节点发生故障时,目前基于快速重路由(fast reroute,FRR)技术使用备份路径来转发报文。

[0004] 如果故障节点是粘连节点,且标签栈中携带有粘连节点的粘连标签的情况下,在粘连节点的上一跳节点确定粘连节点发生故障时,上一跳节点需要确定绕过粘连节点的备份路径,但是上一跳节点无法处理粘连标签,从而会导致转发报文失败。

### 发明内容

[0005] 本申请提供一种传输路径故障的处理方法、装置及系统,用以解决现有技术中存在的故障节点的上一跳节点无法处理粘连标签而导致的转发报文失败的问题。

[0006] 第一方面,本申请实施例提供了一种传输路径故障的处理方法,包括:第一网络设备接收粘连网络设备发送的粘连标签,以及所述粘连标签对应的粘连标签栈列表;所述第一网络设备在通过主路径发送报文的过程中,确定所述粘连网络设备发生故障;其中,所述报文中包括用于指导所述报文在所述主路径上转发的标签栈列表,所述粘连网络设备为在所述主路径上所述第一网络设备的下一跳网络设备;所述第一网络设备将所述标签栈列表中的粘连标签交换为所述粘连标签栈列表,并基于更新后的标签栈列表将所述报文切换到备份路径上发送。

[0007] 本申请实施例提供的方案中,粘连网络设备将粘连标签以及粘连标签栈发布到全网,使得粘连网络设备的上一跳网络设备获知后,将所述标签栈列表中的粘连标签交换为所述粘连标签栈列表,并基于更新后的标签栈列表将所述报文切换到备份路径上发送,避免上一跳网络设备由于无法处理粘连标签导致无法切换到备份路径,而造成的数据报文转发失败。

[0008] 在一种可能的设计中,所述第一网络设备接收粘连网络设备发送的粘连标签,以及粘连标签对应的粘连标签栈列表后,还包括:所述第一网络设备将所述粘连标签以及对应的粘连标签栈列表保存在所述第一网络设备为所述粘连网络设备生成的标签信息表中;所述第一网络设备将所述标签栈列表中的粘连标签交换为所述粘连标签栈列表,并将更新后的所述报文切换到备份路径上发送,包括:所述第一网络设备从所述标签栈列表的栈顶

获取第一标签,所述第一标签用于指示转发所述报文的下一跳网络设备为所述粘连网络设备;所述第一网络设备基于备份转发表确定所述第一标签对应的第一转发行为;所述备份转发表用于指示所述备份路径;所述第一转发行为用于指示查找所述标签信息表;所述第一网络设备基于所述标签信息表确定位于所述粘连标签栈列表栈顶的第二标签,所述第二标签用于指示在所述主路径上,所述粘连网络设备的下一跳网络设备为第三网络设备;所述第一网络设备将所述第二标签作为入标签,从所述备份转发表中确定由所述第一网络设备到所述第三网络设备的备份路径的出标签;所述第一网络设备基于所述出标签向备份路径上所述第一网络设备的下一跳网络设备发送所述报文。

[0009] 通过上述设计,将所述粘连标签以及对应的粘连标签栈列表保存在所述第一网络设备为所述粘连网络设备生成的标签信息表,为查找提供便利,节省时间。

[0010] 在一种可能的设计中,第一网络设备接收粘连网络设备发送的粘连标签,以及粘连标签对应的粘连标签栈列表,包括:所述第一网络设备接收所述粘连网络设备发送的标签长度内容TLV,所述TLV中包括所述粘连标签以及粘连标签对应的粘连标签栈列表。

[0011] 在一种可能的设计中,所述TLV为标签绑定SID/Label Binding TLV。

[0012] 在一种可能的设计中,还包括:所述第一网络设备在接收到所述粘连网络设备在网络内泛洪的所述第三网络设备的节点段标识之后,向所述网络内的其它网络设备泛洪所述第三网络设备的节点段标识。

[0013] 由于,在转发路径中存在松散路径的情况下,当松散路径段的目的地网络设备(粘连网络设备)发生故障时,该网络设备的故障信息泛洪到整个网络后,全网所有的网络设备都会删除到该粘连网络设备的对应的主备转发表。在松散路径的情况下,主路径和备份路径中,从首网络设备到尾网络设备,都需要通过松散路径段且需要从粘连网络设备的上一跳网络设备绕过粘连网络设备到达尾网络设备,由于该路径段的源网络设备在删除到粘连网络设备的备份转发表后,无法将数据报文转发到粘连网络设备的上一跳网络设备,会导致转发报文失败。因此,通过上述设计,在松散路径段的目的地网络设备的上一跳网络设备收到松散路径段的目的地网络设备的节点段标识后,将该目的地网络设备的节点段标识在全网泛洪,从而松散路径段的源网络设备能够根据目的地网络设备的节点段标识来确定转发路径,转发数据报文。

[0014] 在一种可能的设计中,还包括:所述第一网络设备在通过备份路径发送报文时,所述第一网络设备接收所述粘连网络设备在故障恢复后发送的指示信息,所述指示信息用于指示所述粘连网络设备无法正确转发报文,所述第一网络设备继续通过所述备份路径发送报文。

[0015] 在主路径上的粘连节点发生故障时,网络设备切换到备份路径上转发报文。当粘连网络设备恢复后重启完成时,故障恢复节点会重现将自身的节点(前缀)标签、邻接标签等信息重新泛洪给网络中的其它节点。网络设备接收到故障恢复节点泛洪的信息后,将数据报文重新切回到主路径,即重新通过故障恢复节点传输报文。但是由于故障恢复的粘连节点刚刚重启,本地数据库可能不完整,比如还没有学习到其它节点,比如主路径该粘连节点的下一跳节点的节点标签、邻接标签等信息,在该情况下,RT3接收到报文后,无法转发报文到下一跳节点,导致数据报文丢失。基于此,通过上述设计,粘连网络设备恢复后优先发送指示信息来指示其上一跳网络设备,该恢复后的粘连网络设备还无法正确转发报文,

从而上一跳网络设备依然通过备份路径来转发数据报文。

[0016] 第二方面,本申请实施例提供了一种传输路径故障的处理方法,包括:粘连网络设备接收控制器发送的粘连标签,以及所述粘连标签对应的粘连标签栈列表;所述粘连网络设备向网络内的其它网络设备泛洪所述粘连标签,以及所述粘连标签对应的粘连标签栈列表。

[0017] 本申请实施例提供的方案中,粘连网络设备将粘连标签以及粘连标签栈发布到全网,使得粘连网络设备的上一跳网络设备获知后,将所述标签栈列表中的粘连标签交换为所述粘连标签栈列表,并基于更新后的标签栈列表将所述报文切换到备份路径上发送,避免上一跳网络设备由于无法处理粘连标签导致无法切换到备份路径,而造成的数据报文转发失败。

[0018] 第三方面,基于与第一方面的方法实施例同样的发明构思,本申请实施例提供了一种传输路径故障的处理装置,所述装置应用于第一方面所述的第一网络设备,即该装置可以是第一网络设备,也可以是能够应用于第一网络设备的芯片。该装置具有实现上述第一方面的各实施例的功能。该功能可以通过硬件实现,也可以通过硬件执行相应的软件实现。该硬件或软件包括一个或多个与上述功能相对应的模块。

[0019] 第四方面,基于与第二方面提供的方法实施例同样的发明构思,本申请实施例提供了一种传输路径故障的处理装置,所述装置应用于第二方面所述的网络设备,即该装置可以是网络设备,也可以是能够应用于网络设备的芯片。该装置具有实现上述第二方面的各实施例的功能。该功能可以通过硬件实现,也可以通过硬件执行相应的软件实现。该硬件或软件包括一个或多个与上述功能相对应的模块。

[0020] 第五方面,本申请实施例提供一种传输路径故障的处理装置,包括:处理器和存储器;该存储器用于存储指令,当该装置运行时,该处理器执行该存储器存储的该指令,以使该装置执行上述第一方面或第一方面的任一实现方法中的传输路径故障的处理方法。需要说明的是,该存储器可以集成于处理器中,也可以是独立于处理器之外。装置还可以包括总线。其中,处理器通过总线耦接存储器。其中,存储器可以包括可读存储器以及随机存取存储器。当需要运行该装置时,通过固化在只读存储器中的基本输入输出系统或者嵌入式系统中的bootloader引导系统进行启动,引导装置进入正常运行状态。在装置进入正常运行状态后,在随机存取存储器中运行应用程序和操作系统,使得该处理器执行第一方面或第一方面的任意可能的实现设计中的方法。

[0021] 第六方面,本申请实施例提供一种传输路径故障的处理装置,包括:处理器和存储器;该存储器用于存储指令,当该装置运行时,该处理器执行该存储器存储的该指令,以使该装置执行上述第二方面或第二方面的任一实现方法中的传输路径故障的处理方法。需要说明的是,该存储器可以集成于处理器中,也可以是独立于处理器之外。装置还可以包括总线。其中,处理器通过总线耦接存储器。其中,存储器可以包括可读存储器以及随机存取存储器。当需要运行该装置时,通过固化在只读存储器中的基本输入输出系统或者嵌入式系统中的bootloader引导系统进行启动,引导装置进入正常运行状态。在装置进入正常运行状态后,在随机存取存储器中运行应用程序和操作系统,使得该处理器执行第二方面或第二方面的任意可能的实现设计中的方法。

[0022] 第七方面,本申请实施例还提供了一种系统,包括第三方面或第五方面所述的装

置,以及第四方面或第六方面所述的装置。在一个可能的设计中,该系统还可以包括本申请实施例提供的方案中与该两种网络设备进行交互的其他网络设备,例如传输路径上的其它网络设备。

[0023] 第八方面,本申请实施例还提供一种可读存储介质,所述可读存储介质中存储有程序或指令,当其在计算机上运行时,使得上述各方面的任意的传输路径故障的处理方法被执行。

[0024] 第九方面,本申请实施例还提供一种包含指令的计算机程序产品,当其在计算机上运行时,使得计算机执行上述各方面中的任意的传输路径故障的处理方法。

[0025] 另外,第三方面至第九方面中任一种设计方式所带来的技术效果可参见第一方面至第二方面中不同实现方式所带来的技术效果,此处不再赘述。

## 附图说明

[0026] 图1A为本申请实施例提供的一种网络架构示意图;

[0027] 图1B为本申请实施例提供的另一种网络架构示意图;

[0028] 图2为本申请实施例提供的传输路径故障的处理方法流程图;

[0029] 图3为本申请实施例提供的一种传输路径故障的处理方法流程示意图;

[0030] 图4为本申请实施例提供的Binding TLV格式示意图;

[0031] 图5为本申请实施例提供的另一种传输路径故障的处理方法流程示意图;

[0032] 图6为本申请实施例提供的mapping TLV格式示意图;

[0033] 图7为本申请实施例提供的Flags格式示意图;

[0034] 图8为本申请实施例提供的装置800的结构示意图;

[0035] 图9为本申请实施例提供的网络设备900的结构示意图。

## 具体实施方式

[0036] 本申请应用于支持SR-TE的节点。在本申请实施例中,节点也可以称为网络设备,网络设备是在网络中提供路由转发功能的设备,例如,可以是路由器、交换机、转发器或者标签交换路由器(label switching router,LSR)等,不作限定。在后续描述过程中,以节点为例进行说明。

[0037] 在SR-TE技术中,控制平面采用基于链路状态的IGP协议,或者边界网关协议(border gateway protocol,BGP),来分发节点的MPLS标签,数据平面基于该控制平面分发的标签进行MPLS报文转发。

[0038] SR-TE(Segment Routing-Traffic Engineering)是使用内部网关协议(interior gateway protocol,IGP)或者边界网关协议(border gateway protocol,BGP)协议作为控制信令的一种新型的TE隧道技术。控制器负责计算隧道的转发路径,并将与路径严格对应的标签栈列表下发给转发器。在SR-TE隧道的入节点上,转发器根据标签栈列表,即可控制报文在网络中的传输路径。

[0039] 下面先对申请实施例中涉及到的术语进行解释说明。

[0040] 1)、段路由全局块(Segment Routing Global Block,SRGB):用户指定的为段(segment)预留的标签集合。

[0041] 2)、段(segment),通过段标签(Segment ID,SID)来标识。本申请实施例中的段标签可以称为段标识。SID,在转发层面,可以映射为MPLS标签。

[0042] 段包括三类,分别为前缀段(Prefix Segment)、邻接段(Adjacency Segment)和节点段(Node Segment)。

[0043] 前缀段,用于标识SR网络中的某个目的地址前缀(Prefix)。Prefix Segment可以通过IGP协议扩散到其他网元,全局可见,全局有效。Prefix Segment通过前缀段标识(Prefix Segment ID,Prefix SID)来标记。本申请实施例中将前缀段标签简称为前缀标签。Prefix SID是目的节点发布的SRGB范围内的偏移值,接收端会根据自己的SRGB计算实际标签。

[0044] 邻接段(Adjacency Segment)用于标识网络中的某两个邻接节点的路由链路。Adjacency Segment通过IGP协议扩散到其他网元,全局可见,本地有效。Adjacency Segment通过邻接段标签(Adjacency Segment ID,Adjacency SID)。本申请实施例中将邻接段标签简称为邻接标签。

[0045] 邻接标签具有一定的方向性,用于指导报文转发,比如,邻接标签9003对应链路PE1->P3。Adjacency SID为SRGB范围外的本地SID。

[0046] 节点段(Node Segment)是特殊的Prefix Segment,用于标识特定的节点(Node)。在节点的本地环回(Loopback)接口下配置IP地址作为前缀,这个节点的Prefix SID实际就是节点段标签(Node SID)。本申请实施例中将节点段标签可以简称为节点标签。

[0047] 简单来说,Prefix Segment代表目的地址,Adjacency Segment代表数据包的外发链路,可以分别类似于传统IP转发中的目的IP地址和出接口。在IGP区域内,节点使用扩展IGP协议将自身的Node SID以及Adjacency SID进行泛洪,这样任意一个节点都可以获得其他节点的信息。

[0048] 通过前缀(节点)SID和邻接SID的排序组合,可以构建出网络内的任何路径。网络中的一条路径可以通过多个路由段构成。构建出来的路径通过标签栈列表来表示。标签栈列表包括了转发路径上每个路由段对应的段标签(Segment ID)。

[0049] 目的节点通过IGP协议自己的节点(前缀)SID,转发节点解析节点SID,并根据自己的SRGB计算标签值。此后各节点使用IGP协议收集的拓扑信息,根据最短路径算法计算标签转发路径,并将计算的下一跳及出标签(OuterLabel)信息下发转发表,指导数据报文转发。

[0050] 3) 粘连节点和粘连标签

[0051] 当标签栈列表深度超过转发器所支持的标签栈列表深度时,一个标签栈列表无法携带整条路径的段标签,控制器需要为转发器分配多个标签栈列表,在合适的节点下发标签栈列表的同时分配一种特殊的标签,然后将这些标签栈列表关联起来,实现逐段转发。这种特殊的标签称为粘连标签,这个合适的节点称为粘连节点。控制器为粘连节点分配粘连标签,并将其压在标签栈列表的栈底,当报文转发至粘连节点时,根据粘连标签与标签栈列表的关联关系,将粘连标签更换为新的标签栈列表,继续指导下一段转发。需要说明的是,为了描述方便,本申请实施例中标签栈列表简称为标签栈,粘连标签对应的标签栈称为粘连标签栈。

[0052] 另外,需要说明的是,本申请实施例中的多个指两个或两个以上。“和/或”,描述关联对象的关联关系,表示可以存在三种关系,例如,A和/或B,可以表示:单独存在A,同时存

在A和B,单独存在B这三种情况。另外,需要理解的是,在本申请的描述中,“第一”、“第二”等词汇,仅用于区分描述的目的,而不能理解为指示或暗示相对重要性,也不能理解为指示或暗示顺序。节点1或者节点2等也仅用于区分描述的节点,而不能理解为指示或者暗示相对重要性,也不能理解为指示或者暗示顺序。

[0053] 另外,本申请实施例中涉及到的转发表,一般可以是只包括主转发表,或者可以包括主转发表和备份转发表。本申请实施例中涉及到的主备转发表包括主转发表和备转发表。

[0054] 以图1A为例对前缀段的段路径的创建以及数据报文的转发进行说明。

[0055] 比如待建立的前缀段路径为RT1、RT2、RT3、RT4、RT5。RT1为该前缀段的首节点,RT5为该前缀段的目的节点。

[0056] 在RT5上配置SRGB=[5000-5999],在RT5的Loopback接口配置Prefix (node) SID为5,RT5基于SRGB和Prefix (node) SID生成转发表项并下发。之后RT5将SRGB和Prefix SID封装到控制报文,并将该控制报文通过IGP向全网扩散。网络中其他网络设备接收到控制报文后,解析RT5发布的Prefix SID,根据自己的SRGB计算入标签值,同时根据下一跳节点发布的SRGB计算出标签值(OuterLabel),使用IGP拓扑计算标签转发路径,然后生成该入标签值对应的转发表项。

[0057] RT4解析RT5发布的Prefix SID,根据自己的SRGB=[4000-4999]计算标签值,计算公式是:入标签inLabel=SRGB起始值+Prefix SID值,所以inLabel=4000+5=4005。使用IS-IS拓扑计算出标签(OuterLabel),计算公式是:OuterLabel=下一跳设备发布的SRGB的起始值+Prefix SID值(即该前缀段或者节点段上目的节点的节点标签),如图1A所示,RT4的下一跳设备为RT5,RT5发布的SRGB范围是[5000-5999],所以OuterLabel=5000+5=5005。

[0058] RT3计算过程与RT4类似,inLabel=3000+5=3005,OuterLabel=4000+5=4005。RT2计算过程与RT4类似,inLabel=2000+5=2005,OuterLabel=3000+5=3005。RT1计算过程与RT4类似,inLabel=1000+5=1005,OuterLabel=2000+5=2005。

[0059] 在建立RT1-RT5的段路径后,RT1节点在收到数据报文,添加标签值2005并转发。RT2节点收到携带该标签的数据报文,进行标签交换,将标签2005弹出,换成出标签3005,。RT3节点收到该标签报文,进行标签交换,将标签3005弹出,换成标签4004。RT4节点在收到该数据报文后,将标签4005弹出,换成标签5005。RT5节点在收到该数据报文后,将标签5005弹出,继续查路由转发。

[0060] 下面以图1B为例对严格的段路径(不允许跨节点)的创建以及数据报文的转发进行说明。

[0061] 转发器基于IGP协议分配邻接标签,并将分配的邻接标签上报给控制器,从而控制器基于邻接标签生成标签栈列表。

[0062] 邻接标签通过IGP协议泛洪到整个网络中。如图1B所示,以RT1节点为例,IGP协议分配邻接标签的具体过程如下:

[0063] RT1通过IGP协议为其所有链路申请本地动态标签(例如:RT1为链路RT1->RT2分配邻接标签1031)。RT1通过IGP协议发布邻接标签,泛洪到整个网络。网络中的其它设备通过IGP协议学习到RT1发布的邻接标签。具体的,其它节点按照RT1的方式分配和发布邻接标

签,并泛洪给网络中的其它设备。

[0064] 控制器基于SR-TE的隧道约束属性进行路径计算,计算出来的路径根据拓扑和邻接标签,将整条路径的邻接标签整合在一起,生成一个标签栈列表。比如参见图1B所示,控制器计算出SR-TE隧道路径为RT1->RT2->RT3->RT4->RT5,对应的标签栈列表为{1031, 1032, 1034, 1035},控制器通过计算出的标签栈列表下发给转发器入节点RT1。从而转发器根据控制器下发的标签栈列表获得SR-TE隧道。本申请此处也可以使用节点标签来构成标签栈列表,即标签栈列表为{1002, 2003, 3004, 4005}。

[0065] 转发器上根据SR-TE隧道对应的标签栈列表,对报文进行标签操作,并根据栈顶标签逐跳查找转发出口,指导数据报文转发到隧道目的地址。

[0066] 当报文进入SR-TE隧道时,入节点在报文中插入一个标签栈列表;当报文在SR-TE隧道中转发时,根据栈顶的标签查找转发出口之后,将栈顶的标签剥掉。比如,参见图1B所示,入节点PE1为数据报文添加标签栈列表{1031, 1032, 1034, 1035},然后根据栈顶的标签1031匹配链路,找到对应的转发出口为RT1->RT2链路,之后将标签1031弹出。报文携带标签栈{1032, 1034, 1035},通过RT1->RT2链路向下一跳节点RT2转发。中间节点RT2收到报文后,根据栈顶的标签1032匹配链路,找到对应的转发出口为RT2->RT3链路,之后将标签1032弹出,以此类推,节点RT3收到报文后,以与中间节点RT3相同的方式继续转发。直到节点RT4弹出最后一个标签1035,数据报文转发至节点RT5。出节点RT5收到的报文不带标签,通过查找路由表继续转发。

[0067] 严格的段路径方式,在存在等价路径的情况下,也无法进行负载分担。再该情况下可以根据节点标签,或者节点标签和邻接标签构成的混合标签栈,可指定跨节点的节点标签,控制器将标签栈发给转发器的首节点,转发器基于该标签栈,逐跳查找出口弹出标签,指导数据报文进行转发到隧道目的地址。比如,如图1A所示,RT1-RT5的路径,指定需要经过的节点为:RT3、RT4,此时RT1-RT3之间构成了松散路径,可以通过RT2转发,也可以通过RT6转发。在此情况下,RT1上用于转发报文的标签栈列表为{2003, 3004, 4005}。

[0068] 在SR-TE转发报文的过程中,可能发生链路故障或者节点故障。目前,可以通过FRR机制来对路径进行恢复。比如节点R2节点为故障节点。

[0069] 节点R2未发生故障时,SR-TE的标签栈标识的转发路径为:RT1->RT2->RT3->RT4->RT5。当节点RT2故障时,节点RT2的邻居节点RT1替代节点RT2做标签弹出(POP)动作,并使用到下一层标签作为目的地址,且绕过故障节点RT2的路径作为备份路径进行数据报文的转发。备份路径为:RT1->RT6->RT3->RT4->RT5。

[0070] 比如RT1进行数据报文发送的标签栈列表为{1002, 2003, 3004, 4005},RT1可以通过检查标签栈的栈顶标签1002,确定RT2节点是故障节点,然后通过检查标签栈的栈顶标签1002和第二个标签2003,确定故障节点后面路径所需要到达的节点为RT3。RT1计算出绕过故障节点RT2到达第二个标签2003所标识的节点(即RT3)的路径,作为对下一跳节点RT2的SR-TE FRR的节点保护路径。

[0071] 在RT2节点故障前,每一个节点通过IGP提前计算好备份转发表和标签信息表(Context表)。每一个节点都为其邻居节点生成Context表以及主备转发表。Context表包括了邻居数\*(全网络所有节点的节点(前缀)标签和邻居节点发布的所有的邻接标签),每个邻居节点对应一个Context表项。比如,节点RT1为其邻居节点RT2生成RT2对应的Context表

项,该Context表项中包含了全网所有节点对应节点(前缀)标签信息以及RT2发布的邻接标签。全网所有节点中每个节点对应的节点(前缀)标签信息是该节点的节点标签与RT2节点的SRGB的初始值的和。在路径中存在故障节点时,故障节点的上一跳节点通过Context表以及主备转发表确定备份路径,并通过备份路径转发数据报文。

[0072] 需要说明的是,主路径中某个节点,或者某个节点到下一跳的路径发生故障时,在主路径中该节点的上一跳节点感知到故障后,上一跳节点的会将该节点的故障信息泛洪到网络中其它节点。

[0073] 如果故障节点是粘连节点,且标签栈中携带有粘连节点的粘连标签的情况下,在粘连节点的上一跳节点确定粘连节点发生故障时,上一跳节点需要确定绕过粘连标签的备份路径,但是因为粘连标签对应的粘连标签栈只有粘连节点知道,上一跳节点不知道,因此针对粘连标签,上一跳节点无法处理。

[0074] 基于此,本申请实施例提供了一种传输路径故障的处理方法,参见图2所示,该方法包括:

[0075] S201,粘连节点接收控制器发送的粘连标签,以及所述粘连标签对应的粘连标签栈列表。

[0076] S202,所述粘连节点向网络内的其它节点泛洪所述粘连标签,以及所述粘连标签对应的粘连标签栈列表。

[0077] S203,第一节点接收粘连节点发送的粘连标签,以及所述粘连标签对应的粘连标签栈列表。

[0078] S204,所述第一节点在通过主路径发送报文的过程中,确定所述粘连节点发生故障;

[0079] 其中,所述报文中包括用于指导所述报文在所述主路径上转发的标签栈列表,所述粘连节点为在所述主路径上所述第一节点的下一跳节点;

[0080] S205,所述第一节点将所述标签栈列表中的粘连标签交换为所述粘连标签栈列表,并基于更新后的标签栈列表将所述报文切换到备份路径上发送。

[0081] 在一种可能的实施方式中,所述第一节点接收粘连节点发送的粘连标签,以及粘连标签对应的粘连标签栈列表后,所述第一节点将所述粘连标签以及对应的粘连标签栈列表保存在所述第一节点为所述粘连节点生成的标签信息表中。

[0082] 所述第一节点将所述标签栈列表中的粘连标签交换为所述粘连标签栈列表,并将更新后的所述报文切换到备份路径上发送,可以通过如下方式实现:

[0083] 所述第一节点从所述标签栈列表的栈顶获取第一标签,所述第一标签用于指示转发所述报文的下一跳节点为所述粘连节点;所述第一节点基于备份转发表确定所述第一标签对应的第一转发行为;所述备份转发表用于指示所述备份路径;所述第一转发行为用于指示查找所述标签信息表;所述第一节点基于所述标签信息表确定位于所述粘连标签栈列表栈顶的第二标签,所述第二标签用于指示在所述主路径上,所述粘连节点的下一跳节点为第三节点;所述第一节点将所述第二标签作为入标签,从所述备份转发表中确定由所述第一节点到所述第三节点的备份路径的出标签;所述第一节点基于所述出标签向备份路径上所述第一节点的下一跳节点发送所述报文。

[0084] 在一种可能的实施方式中,第一节点接收粘连节点发送的粘连标签,以及粘连标

签对应的粘连标签栈列表,包括:所述第一节点接收所述粘连节点发送的标签长度内容TLV,所述TLV中包括所述粘连标签以及粘连标签对应的粘连标签栈列表。示例性地,所述TLV可以为标签绑定SID/Label Binding TLV。

[0085] 在一种可能的实施方式中,还包括:所述第一节点在接收到所述粘连节点在网络内泛洪的所述第三节点的节点段标识之后,确定所述粘连节点发生故障时,向所述网络内的其它节点泛洪所述第三节点的节点段标识。

[0086] 在一种可能的设计中,还包括:所述第一节点在通过备份路径发送报文时,所述第一节点接收所述粘连节点在故障恢复后发送的指示信息,所述指示信息用于指示所述粘连节点无法正确转发报文,所述第一节点继续通过所述备份路径发送报文。

[0087] 以图3所示的网络架构为例,首先包括粘连节点的路径的建立过程以及数据报文转发过程进行详细描述。图3所示的网络架构,RT2可以是图2所示的实施例中的第一节点,RT3可以是图2所示的实施例中的粘连节点。

[0088] 控制器基于SR-TE的隧道约束属性计算传输路径,可以是严格路径也可以是松散路径。控制器根据拓扑和邻接标签(和/或节点标签),将传输路径的邻接标签(和/或节点标签)整合在一起,生成一个标签栈发送给传输路径的首节点。当标签栈深度超过转发器所支持的标签栈深度时,一个标签栈无法携带所有的邻接标签(和/或节点标签),则控制器需要将整条路径分为多个标签栈携带。本申请实施例中粘连标签栈采用的可以是邻接标签,当然也可以是节点标签。

[0089] 如图3所示,以粘连标签栈采用邻接标签为例。图3中,RT1-RT5的路径,指定需要经过的节点为:RT3、RT4,此时RT1-RT3之间构成了松散路径,可以通过RT2转发,也可以通过RT6转发。比如,控制器计算出SR-TE隧道路径为RT1-->RT3->RT4->RT5,对应2个标签栈,{1003,100}和{1034,1035},其中100为粘连标签,1034、1035为邻接标签,1003为RT3的节点标签与RT1的SRGB初始值计算得到的。

[0090] 控制器将标签栈下发给相应的转发器。示例性地,控制器将分配粘连标签,并将粘连标签以及粘连标签对应的粘连标签栈{100:1034,1035}下发给粘连节点RT3。控制器将粘连标签100作为RT1上标签栈的栈底。控制器将标签栈{1003,100}下发给首节点RT1。

[0091] 本申请实施例中RT1经过路径计算,确定的主路径为RT1->RT2->RT3->RT4->RT5。

[0092] RT1确定主路径为经过RT2,从而将数据报文中包括的标签栈的外层标签(栈顶标签)1003交换为2003(RT3的节点标签+RT2的SRGB初始值),并将更新后的数据报文发送给RT2。在粘连节点未发生故障的情况下,RT2收到RT1发送的数据报文后,把栈顶标签交换为3003,并发给RT3。RT3收到后弹出栈顶标签,然后把更新后标签栈列表中的栈顶标签100交换为{1034,1045}。根据栈顶标签1034确定下一跳节点为RT4,将栈顶标签1034弹出后,将更新后的数据报文转发给RT4,RT4根据栈顶标签1035确定下一跳节点为RT5,从而将栈顶标签1035弹出后,将更新后的数据报文发送给RT5。

[0093] 在粘连节点RT3发生故障时,其上一跳节点RT2无法针对标签栈中的粘连标签进行处理导致数据报文转发失败。

[0094] 基于此,本申请实施例提供了一种传输路径故障的处理方案,粘连节点在接收到控制器发送的粘连标签以及粘连标签对应的粘连标签栈时,将粘连标签以及粘连标签对应的粘连标签栈发布到全网。

[0095] 示例性的,粘连标签以及粘连标签栈可以包括的绑定(binding)TLV中发布到全网。粘连节点RT3采用Binding TLV,发布粘连标签(Binding Sid)以及对应的粘连标签栈{100:1034,1045},可以在现有的Binding TLV的定义的基础上添加若干个SID/Lable sub TLV,来填入具体的粘连标签以及粘连标签对应的粘连标签栈),具体Binding TLV格式可以参见图4所示。

[0096] 作为一种示例,RT2在接收到RT3发布的粘连标签以及其对应的粘连标签栈后,可以将粘连标签以及其对应的粘连标签栈存储在Context表中。Context表中RT3对应的Context表项的部分内容可以参见表1所示。

[0097] Inlabel,表示入标签,为本节点的SRGB初始值+邻居节点的节点Sid,即RT2的SRGB初始值2000+RT3的节点标签3=2003。入标签作为Context表查表的关键(KEY)值字段。

[0098] 表1中,本地sid包括RT3发布的邻接标签(Adj Sid)以及粘连标签。

[0099] Adj Sid,表示邻居节点发布的邻接标签,RT3发布的邻接标签分别为RT3与RT4之间链路的邻接标签1034,RT3与RT7之间链路的邻接标签1037,以及RT3与RT6之间链路的邻接标签1036。

[0100] Adj节点Sid,表示邻居节点发布的邻接标签对应的目的节点,使用本节点的SRGB初始值+该目的节点发布的节点Sid。比如,RT3与RT4之间链路的邻接标签1034,目的节点是RT4,则1034对应的Adj节点Sid为RT2的SRGB初始值+RT4的节点标签=2004。

[0101] Action,表示Local sid或者Adj节点SID对应的转发动作。

[0102] 表1

Inlabel	本地(Local) Sid (Adj sid 和)	Action	Adj 节点 SID
[0103]	1034	Fwd to RT4	2004
	1037	Fwd to RT7	2007
	1036	Fwd to RT6	2006
	100	交换为 1034,1035	

[0104] 在图3所示的网络架构中,RT1-RT5的路径,指定需要经过的节点为:RT3、RT4,此时RT1-RT3之间构成了松散路径,可以通过RT2转发,也可以通过RT6转发,本申请实施例中RT1经过路径计算,确定的主路径为RT1->RT2->RT3->RT4->RT5。在此情况下,RT1接收到控制器发送的标签栈为{1003,100}。比如,在RT3故障的情况下备份路径为RT1->RT2->RT7->RT4->RT5。同时,RT2会生成全网的节点Sid的主备转发表。主备转发表的部分内容可以参见表2所示。表2仅示出了入标签为2003和2004的两条表项的部分内容。

[0105] 表2

Inlabel	Action	Outlabel	TYPE	
[0106]	2003	Fwd to RT3	3003	主
		Pop, 查找 Conext 表		备
	2004	Fwd to RT3	3004	主
[0107]		Fwd to RT7	7004	备

[0108] RT1将确定RT1的下一跳节点为RT2,从而RT1将标签栈的栈顶标签1003交换为2003后添加在数据报文中,发送给RT2。则RT2在接收到RT1的数据报文后,根据数据报文的标签栈的栈顶标签2003,确定栈顶标签指示的下一跳节点RT3故障时,RT2是要找到绕过RT3到RT3的下一跳节点RT4的转发路径。从而查找备份转发表(表2),查找该入标签为2003对应的这条转发表项,该条转发表项中指示的转发行为是弹出栈顶标签2003,并查找RT2为邻居节点RT3生成的Context表。RT2弹出栈顶标签2003后,确定标签栈列表的栈顶标签为粘连标签,根据采用粘连标签100查找Context表的Local Sid,查找到入标签为2003对应的这条转发表项,该条表项中指示的转发行为是交换为{3004,4005}。从而RT2将粘连标签100交换为{1034,1035}。RT2获取RT3到RT4的邻接标签为1034,继续根据栈顶标签1034查找Context表的本地sid记录,查找到的转发行为是转发到RT4,并确定RT4对应的adj节点标签为2004,即获取以RT2为入节点到以RT4出节点的路由段,在RT2上的入标签。RT2节点将1034交换为2004,使用2004作为入标签查找主转发表,查找到的下一跳节点为RT3,由于到RT3的接口故障,于是切换到备份路径,查找备份转发表,将2004作为入标签查找到的转发行为是转发到RT7,出标签为7004,从而RT2节点将栈顶标签交换为查找到的出标签,即将栈顶标签2004弹出,并交换为7004,此处的标签栈为{7004,1035},将更新后的数据报文转发给RT7。

[0109] RT7接收到数据报文后,根据栈顶标签7004确定下一跳节点为RT4,从而将栈顶标签7004交换为4004,并将更新后的数据报文转发给RT4,同理RT4接收到数据报文后,确定栈顶标签指示的目的节点为自身,将栈顶标签弹出,确定更新后的标签栈的栈顶标签1035指示的下一跳节点为RT5,并将更新后的数据报文发送给RT5。

[0110] 本申请实施例提供的方案中,粘连节点将粘连标签以及粘连标签栈发布到全网,使得粘连节点的上一跳节点获知后,将所述标签栈列表中的粘连标签交换为所述粘连标签栈列表,并基于更新后的标签栈列表将所述报文切换到备份路径上发送,避免上一跳节点由于无法处理粘连标签导致无法切换到备份路径,而造成的数据报文转发失败。

[0111] 另外,由于Context表是预先生成的,SR-TE FRR的方案是节点为该节点的每个邻居都生成Context表,因此Context表的规格为全网的节点数+邻居节点的邻居数,所以每个节点需要生成邻居数\*(全网节点数+邻居节点的邻居数)这样规模的Context表。

[0112] 当网络包括节点数较多,且有些节点所支持的Context表的规格不足以支持邻居数\*(全网节点数+邻居节点的邻居数)这样的规格时,会导致这些节点生成的Context表中内容不全,在网络中出现故障节点,可能会导致无法切换到正确的备份路径来转发报文。

[0113] 基于此,本申请实施例中提出了一种配置Context表的方式。当节点通过IGP获取到邻居节点发布的邻接标签以及邻居节点其邻接的节点的节点标签,并在学习到邻居节点

的SRGB后,生成Context表.Context表中包括邻居节点发布的邻接标签,以及邻居节点其邻接的节点的节点标签信息,以及本节点的SRGB初始值与邻居节点的SRGB初始值的差值,此处节点标签信息是邻居节点的SRGB初始值与所述邻居节点其邻接的节点的节点标签的和。

[0114] 以图5所示为例,RT2为邻居节点RT3生成的Context表如表3所示。RT3邻接的节点包括RT2、RT7、RT6以及RT4。其中,RT2的SRGB初始值为2000,RT2的邻居节点RT3的SRGB初始值为3000,因此,RT2的SRGB初始值与RT3的SRGB初始值的差值为-1000。本实施例中以粘连标签栈采用节点标签为例进行说明。RT3为粘连节点,粘连标签以及粘连标签对应的粘连标签栈为{100:3004,4005}。

[0115] 作为一种示例,RT2在接收到RT3发布的粘连标签以及其对应的粘连标签栈后,可以将粘连标签以及其对应的粘连标签栈存储在Context表中RT3节点对应的表项中。Context表的部分内容可以参见表3所示,表3中示出了Context表中RT3对应的表项中的部分内容,即入标签为2003的表项中的部分内容。表3中,SRGB差:为本节点的SRGB初始值-邻居节点的SRGB初始值。

[0116] 表3

Inlabel	SRGB 差	本地 (Local) Sid (Adi sid 和)	Action	Adj 节点 SID
[0117] 2003	-1000	1034	Fwd to RT4	2004
		1037	Fwd to RT7	2007
		1036	Fwd to RT6	2006
		100	交换为 3004,4005	

[0118] 下面以图5为例,对数据报文的转发过程进行详细描述。

[0119] 图5中,首节点RT1-尾节点RT5的路径,指定需要经过的节点为:RT3、RT4,此时RT1-RT3之间构成了松散路径,可以通过RT2转发,也可以通过RT6转发,本申请实施例中RT1经过路径计算,确定的主路径为RT1->RT2->RT3->RT4->RT5。RT1上用于转发报文的标签栈列表为{1003,100}。比如,在RT3故障的情况下备份路径为RT1->RT2->RT7->RT4->RT5。同时,RT2会生成全网的节点Sid的主备转发表。主备转发表的部分内容可以参见表4所示,表4仅示出了入标签为2003和2004的两条表项的部分内容。

[0120] 表4

Inlabel	Action (动作)	Outlabel	TYPE (类型)
[0121] 2003	Fwd to (转发到) RT3	3003	主
	Pop, 查找 Conext 表		备
2004	Fwd to (转发到) RT3	3004	主
	Fwd to (转发到) RT7	7004	备

[0122] 在RT3未发生故障时,如图5所示,RT1确定RT1的下一跳节点为RT2时,将标签栈的

栈顶标签1003交换为指示RT2的标签2003后,将标签栈携带在数据报文中发送给RT2。RT2接收到RT1发送的数据报文,数据报文中的标签栈为{2003,100},根据标签栈中的栈顶标签查找主转发表,比如,查找表4,栈顶标签为2003,将2003作为inlabel,查找到的转发行为转发到RT3,从而RT2将栈顶标签交换为3003,即将栈顶标签交换为基于下一跳节点的SRGB初始值(RT3的SRGB初始值)以及目的节点的节点标签(RT3的节点标签)确定的标签值,然后转发更新后的数据报文到RT3。RT3接收到RT2发送的数据报文后,数据报文中的标签栈为{3003,100},确定栈顶标签指示的接收节点为自身,因此,将栈顶标签3003弹出后,将栈顶标签100交换为粘连标签栈列表{3004,4005},确定下一跳节点为RT4,将栈顶标签3004交换为4004,并将更新后数据报文转发到RT4,同理RT4接收到RT3发送的数据报文后,数据报文的标签栈为{3004,4005},确定栈顶标签指示接收节点为自身,因此将栈顶标签弹出,根据栈顶标签4005确定下一跳节点为RT5,因此将栈顶标签交换为5005,并将更新后的数据报文转发到RT5。RT5接收到RT4发送的数据报文后,数据报文的标签栈为{5005},确定栈顶标签指示接收节点为自身,因此将栈顶标签弹出,继续查找转发表继续转发。

[0123] 在粘连节点RT3发生故障时,RT2在接收到RT1的数据报文后,根据数据报文的标签栈的栈顶标签2003,确定栈顶标签指示的下一跳节点RT3故障时,RT2是要找到绕过RT3到RT3的下一跳节点RT4的转发路径。从而查找备份转发表(表3),备份转发表中入标签2003对应的转发行为是弹出栈顶标签2003,并查找RT2为邻居节点RT3生成的Context表。RT2弹出栈顶标签2003后,确定标签栈列表的栈顶标签为粘连标签,根据Context表,将粘连标签100交换为{3004,4005}。根据表3的Context表获取到SRGB差为-1000,RT2获取从RT3到下一跳节点RT4的标签,即栈顶标签3004,并使用栈顶标签与SRGB差作和,即标签(3004)+SRGB差(-1000)=2004,即获取以RT2为入节点到以RT4出节点的路由段,在RT2上的入标签。使用2004作为入标签查找主转发表,查找到的下一跳节点为RT3,由于到RT3的接口故障,于是切换到到备份路径,查找备份转发表,将2004作为入标签查找到的转发行为是转发到RT7,出标签为7004,从而RT2节点将栈顶标签交换为查找到的出标签,即将栈顶标签3004弹出,并交换为7004,此处的标签栈为{7004,4005},将更新后的数据报文转发给RT7。

[0124] RT7接收到数据报文后,根据栈顶标签7004确定下一跳节点为RT4,从而将栈顶标签7004交换为4004,并将更新后的数据报文转发给RT4,同理RT4接收到数据报文后,确定栈顶标签指示的目的节点为自身,将栈顶标签弹出,确定更新后的标签栈的栈顶标签4005指示的下一跳节点为RT5,从而将栈顶标签4005交换为5005,并将更新后的数据报文发送给RT5。

[0125] 现有在主路径中某个节点出现故障的情况下,该故障节点的上一跳节点通过为该故障节点生成Context表以及主备转发表由主路径切换到备份路径上时,通过根据标签栈中的栈顶标签查找备转发表,确定备转发表的转发行为是查找Context表后,根据Context表来确定备份路径上需要转发报文的下一跳节点,也就是现有仅查询一次备转发表,确定备份路径是Context表来实现的,而本申请实施例提供的方案中,通过给转发面节点下发SRGB的差值的情况,仅需要使用Context表确定故障节点的下一跳节点的节点标签,而确定备份路径的转发报文的下一跳节点时,是通过故障节点的下一跳节点的节点标签再查询一次备转发表,因此Context表中不需要包括所有节点对应的节点标签,仅需要包括邻居节点其邻接的节点对应的标签信息即可,从而避免表项容量不足导致的转发报文失败,并且可以节省存储资源。

[0126] 在转发路径中存在松散路径的情况下,松散路径的目的节点为粘连节点,当松散路径段的节点(粘连节点)发生故障时,该节点的故障信息泛洪到整个网络后,全网所有的节点都会删除到该故障节点的对应的主备转发表。在松散路径的情况下,主路径和备份路径中,从首节点到尾节点,都需要通过松散路径段且需要从故障节点的上一跳节点绕过故障节点到达尾节点,由于该路径段的源节点在删除到故障节点的主备转发表后,无法将数据报文转发到故障节点的上一跳节点,会导致转发报文失败。

[0127] 以图5所示为例,RT1指定的SR-TE路径经过的节点为RT3,RT4,RT5,当RT3故障的时候,RT2感知到RT3故障时,通过IGP泛洪(Flooding)该RT3故障的故障信息,并触发SR-TE FRR切换,切换到备份路径RT7,RT4,RT5,绕过故障点RT3,当RT3故障信息泛洪到全网的节点后,RT1开始收敛,会删除到RT3的主备转发表,从而RT1无法转发数据报文到RT2。

[0128] 基于此,本申请实施例提供了一种传输路径故障的处理方法。粘连节点的上一跳节点感知到该粘连节点发生故障时,采用IGP协议向全网的所有节点泛洪该粘连节点的前缀(节点)标签。上一跳节点在泛洪该粘连节点的前缀标签时,可以将粘连节点的前缀标签携带在Mapping TLV(Tag (Type) -Length-Value)。

[0129] 需要说明的是,本申请实施例中有关Mapping TLV的细节,请参考国际互联网工程任务组(Internet Engineering Task Force,IETF)发布的文稿draft-ietf-isis-segment-routing-extensions,该文档与此相关部分的内容好像整体复制一般以引入的方式并入本文本中,此处为了简洁,不再赘述。

[0130] 在TLV的定义中,包括三个域,分别为:标签域(Type),长度域(Length),内容域(Value)。Value中包含了粘连节点的前缀标签。Mapping TLV也可以称为标签绑定TLV(SID/Label Binding TLV)。示例性地,以图6所示为Mapping TLV格式为例。其中,Type表示该TLV的类型,Length表示控制报文的长度,Flags表示标志位,RESERVED表示预留位,当前未用到。Range字段提供了指定的一个地址段与Prefix SID段关联的能力,Prefix length表示前缀的长度,prefix在通告路径的尾节点表示转发等价类,SubTLV表示MappingTLV中用于携带Prefix-SID的sub-TLV。

[0131] 示例性地,本申请实施例中,上一跳节点可以在Flags中增加一个R字段,用于表示重发布标志位,即针对故障节点的前缀标签由该上一跳节点重新发布给全网其它节点。

[0132] Flags的格式可以参见图7所示。其中,F表示地址族标志。如果未置位,代表IPv4前缀;如果置位,代表IPv6前缀。M表示镜像环境标记,如果通告的SID关联一个镜像环境则设置该标记。S表示SID/Label Binding TLV是否可在整个路由域内泛洪,如果置位,表示在整个路由域内泛洪;如果未置位,SID/Label Binding TLV一定不能在level间渗透。D表示当SID/Label Binding TLV从level-2向level-1渗透时,必须设置该标记。否则该标记必须被清除。A表示附属标记。其中,Level是中间系统到中间系统(Intermediate System-to-Intermediate System,IS-IS)路由协议中对网络分层的概念。普通区域称为Level-1,骨干区称为Level-2。

[0133] 以图5所示的网络架构为例,RT2在确定粘连节点RT3故障时,向全网泛洪RT3的节点标签。RT1在收到RT2泛洪的RT3的节点标签后,根据粘连节点RT3的节点标签确定向RT2节点发送数据报文的出标签,即RT2的SRGB初始值+RT3的节点标签=2000+3=2003。从而RT2在接收到数据报文后切换到备份路径上继续转发。RT2切换备份路径转发报文的方式可以

采用现有context表的形式,当然还可以采用上述配置context表的方式,此处对此不作限定。具体的发送过程可以参见图4对应的实施例的转发方式,此处不再赘述。

[0134] 另外,在粘连节点发生故障时,上一跳节点切换到备份路径上转发报文。当粘连节点恢复后重启完成时,粘连节点会重新将自身的节点(前缀)标签、邻接标签等信息重新泛洪给网络中的其它节点。上一跳节点接收到粘连节点泛洪的信息后,将数据报文重新切回到主路径,即重新通过粘连节点传输报文。但是由于粘连节点刚刚重启,本地数据库可能不完整,比如还没有学习到其它节点的相关信息,比如主路径上该粘连节点的下一跳节点的节点标签、邻接标签等信息,在该情况下,RT3接收到报文后,无法转发报文到下一跳节点,导致数据报文丢失。

[0135] 以图5所示,RT1指定的SR-TE路径为RT3,RT4,RT5,当RT3故障的时候,RT2通过感知到RT3故障后,触发SR-TE FRR切换,切换到备份路径RT7,RT4,RT5,因此绕过故障节点RT3。但当RT3故障恢复,并且重启完成时,RT3将Node Sid重新发布到全网,RT2将数据报文重新切回到RT3。但是由于RT3刚刚重启完成,它本地数据库可能还不全,比如还没有学到RT4和RT5的node Sid,这样就导致数据报文在RT3无法转发到下一跳节点RT4,丢弃报文。

[0136] 基于此,本申请实施例提供了一种传输路径故障的处理方法。在粘连节点恢复,且重启完成时,为防止恢复的粘连节点的上一跳节点在感知到粘连节点恢复时,恢复的粘连节点未学习到其它节点的前缀SID,导致数据报文丢失。在粘连节点恢复后首先在全网发布指示信息,指示信息用于指示恢复的粘连节点无法正确转发报文;因此上一跳节点在接收到待发送给恢复的粘连节点的数据报文时,依然从备份路径转发数据报文。

[0137] 示例性地,指示信息可以携带在Overload bit。Overload bit是IS-IS路由协议中规定的。如果一个节点不能完成全部的连接状态数据(LSDB),即本地存储用于存储全网所有节点的拓扑信息,它将在自己发布的链路状态数据包的头部用一个bit来标识其未完成LSDB,无法正确转发报文,这一个bit的标识位叫做Overload (OL) bit。通过该粘连节点发布的链路状态数据包中的Overload bit,其他节点得知该粘连节点可能无法做出正确的路由决定,即无法正确转发数据报文,因为它的LSDB没有完成。其他节点在计算最短路径时不会通过这个节点来转发数据报文。

[0138] 以图5所示的网络架构为例,粘连节点RT3在故障恢复,且重启完成后,为防止RT2首先感知到RT3故障恢复,且此时RT3还未学习到RT4和RT5的前缀Sid,导致数据报文丢失。RT3在故障重启完成后在发布链路状态数据包的头部添加Overload bit标识位。

[0139] 节点RT2收到RT3发布链路状态数据包中的Overload bit,感知到其下一跳节点RT3处于Overload状态,Over状态表示恢复的粘连节点无法正确转发报文。因此,节点RT2继续走故障恢复前的SR-TE FRR转发路径,RT2采用备份路径转发数据报文的方式可以采用现有context表的形式,当然还可以采用上述配置context表的方式,此处对此不作限定。

[0140] 粘连节点完成LSDB的时间,即Overload时间,一般比较长,比如是60s。因此在该时长内RT2可以采用查Context表按着SR-TE FRR路径(备份路径)转发数据报文。示例性地,可以给配置Overload时长,RT2在该时长内采用备份路径转发数据报文。当然,RT3在完成LSDB时,可以再次在全网发布链路状态数据包,且头部不再包括overload bit标志位。RT2在收到该不包括overload bit标志位的链路状态数据包后,再切回到该RT3转发数据报文。

[0141] 基于与方法实施例同样的发明构思,本申请实施例还提供了一种装置,参见图8所

示,该装置800应用于网络设备。该装置800可以包括接收模块801、处理模块802、发送模块803。该装置800具体可以是网络设备中的处理器,或者芯片或者芯片系统,或者是一个功能模块等。处理模块802用于对装置800的动作进行控制管理,接收模块801用于接收信息或者报文,发送模块803用于发送信息或者发送报文,处理模块802用于对接收模块801接收到的信息或者报文进行处理。处理模块802还可以用于指示上述任意实施例中涉及网络设备(比如第一节点或者RT2或者粘连节点或者RT3)的处理过程和/或本申请所描述的技术方案的其他过程。该装置还可以包括存储模块(图8中未示出),存储模块用于存储Context表与主备转发表。

[0142] 在一种可能的实施方式中,该装置800可以应用于粘连节点的上一跳节点,比如图2对应的实施例中的第一节点或者图3、图5对应的实施例中的RT2。

[0143] 以应用到第一节点为例,接收模块801,用于接收粘连网络设备发送的粘连标签,以及粘连标签对应的粘连标签栈列表;

[0144] 处理模块802,用于在通过主路径发送报文的过程中,确定所述粘连节点发生故障;

[0145] 其中,所述报文中包括用于指导所述报文在所述主路径上转发的标签栈列表,所述粘连节点为在所述主路径上所述第一节点的下一跳节点。

[0146] 所述处理模块802,还用于将所述标签栈列表中的粘连标签交换为所述粘连标签栈列表;

[0147] 发送模块803,用于基于所述处理模块802更新后的标签栈列表将所述报文切换到备份路径上发送。

[0148] 在一种可能的示例中,所述处理模块802,还用于在所述接收模块801接收粘连节点发送的粘连标签,以及粘连标签对应的粘连标签栈列表后,将所述粘连标签以及对应的粘连标签栈列表保存在所述第一节点为所述粘连节点生成的标签信息表中;

[0149] 所述处理模块802,在将所述标签栈列表中的粘连标签交换为所述粘连标签栈列表时,具体用于从所述标签栈列表的栈顶获取第一标签,所述第一标签用于指示转发所述报文的下一跳节点为所述粘连节点;基于备份转发表确定所述第一标签对应的第一转发行为;所述备份转发表用于指示所述备份路径;所述第一转发行为用于指示查找所述标签信息表;基于所述标签信息表确定位于所述粘连标签栈列表栈顶的第二标签,所述第二标签用于指示在所述主路径上,所述粘连节点的下一跳节点为第三节点;将所述第二标签作为入标签,从所述备份转发表中确定由所述第一节点到所述第三节点的备份路径的出标签;

[0150] 所述发送模块803,具体用于基于所述出标签向备份路径上所述第一节点的下一跳节点发送所述报文。

[0151] 在一种可能的示例中,,所述发送模块803,还用于在所述接收模块801接收到所述粘连节点在网络内泛洪的所述第三节点的节点段标识之后,向所述网络内的其它节点泛洪所述第三节点的节点段标识。

[0152] 在一种可能的示例中,所述接收模块801,还用于在所述发送模块803通过备份路径发送报文时,接收所述粘连节点在故障恢复后发送的指示信息,所述指示信息用于指示所述粘连节点无法正确转发报文,所述发送模块803,还用于继续通过所述备份路径发送报文。

[0153] 在一种可能的实施方式中,该装置800可以应用于粘连节点,比如图2对应的实施例中的粘连节点或者图3、图5对应的实施例中的RT3。

[0154] 以应用到粘连节点为例,接收模块801,用于接收控制器发送的粘连标签,以及所述粘连标签对应的粘连标签栈列表;发送模块803,用于向网络内的其它网络设备泛洪所述粘连标签,以及所述粘连标签对应的粘连标签栈列表。可选地,处理模块802用于将粘连标签,以及所述粘连标签对应的粘连标签栈列表存储在存储模块中(图8中未示出)。

[0155] 本申请实施例还提供一种网络设备的结构,如图9所示,网络设备900中可以包括通信接口910、处理器920。可选的,网络设备900中还可以包括存储器930。其中,存储器930可以设置于网络设备内部,还可以设置于网络设备外部。上述图8中所示的、处理模块802均可以由处理器920实现。接收模块801、发送模块803可以由通信接口910实现。

[0156] 在一种可能的实施方式中,处理器920通过通信接口910收发报文或者消息,并用于实现图2~图8中所述的节点(第一节点或者RT2)所执行的任一方法。在实现过程中,处理流程的各步骤可以通过处理器920中的硬件的集成逻辑电路或者软件形式的指令完成图2~图5中所述的第一节点或者RT2所执行的方法。为了简洁,在此不再赘述。处理器920用于实现上述方法所执行的程序代码可以存储在存储器930中。存储器930和处理器920耦合。

[0157] 在一种可能的实施方式中,处理器920可以通过通信接口910收发报文或者消息,并用于实现图2~图5中所述的节点(粘连节点或者RT3)所执行的任一方法。在实现过程中,处理流程的各步骤可以通过处理器920中的硬件的集成逻辑电路或者软件形式的指令完成图2~图5中所述的粘连节点或者RT3所执行的方法。为了简洁,在此不再赘述。处理器920用于实现上述方法所执行的程序代码可以存储在存储器930中。存储器930和处理器920耦合。

[0158] 本申请实施例中涉及到的任一通信接口可以是电路、总线、收发器或者其它任意可以用于进行信息交互的装置。比如网络设备900中的通信接口910,示例性地,该其它装置可以是与该网络设备900相连的设备,比如,该其它装置可以是该网络设备900的上一跳节点或者下一跳节点等。

[0159] 本申请实施例中涉及的处理器可以是通用处理器、数字信号处理器、专用集成电路、现场可编程门阵列或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件,可以实现或者执行本申请实施例中的公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者任何常规的处理器等。结合本申请实施例所公开的方法的步骤可以直接体现为硬件处理器执行完成,或者用处理器中的硬件及软件模块组合执行完成。

[0160] 本申请实施例中的耦合是装置、模块或模块之间的间接耦合或通信连接,可以是电性,机械或其它的形式,用于装置、模块或模块之间的信息交互。

[0161] 处理器920可能和存储器930协同操作。存储器930可以是非易失性存储器,比如硬盘(hard disk drive,HDD)或固态硬盘(solid-state drive,SSD)等,还可以是易失性存储器(volatile memory),例如随机存取存储器(random-access memory,RAM)。存储器930是能够用于携带或存储具有指令或数据结构形式的期望的程序代码并能够由计算机存取的任何其他介质,但不限于此。

[0162] 本申请实施例中不限定上述通信接口910、处理器920以及存储器930之间的具体连接介质。本申请实施例在图9中以存储器930、处理器920以及通信接口910之间通过总线连接,总线在图9中以粗线表示,其它部件之间的连接方式,仅是进行示意性说明,并不引以

为限。所述总线可以分为地址总线、数据总线、控制总线等。为便于表示,图9中仅用一条粗线表示,但并不表示仅有一根总线或一种类型的总线。

[0163] 基于以上实施例,本申请实施例还提供了一种计算机存储介质,该存储介质中存储软件程序,该软件程序在被一个或多个处理器读取并执行时可实现上述任意一个或多个实施例提供的方法。所述计算机存储介质可以包括:U盘、移动硬盘、只读存储器、随机存取存储器、磁碟或者光盘等各种可以存储程序代码的介质。

[0164] 基于以上实施例,本申请实施例还提供了一种芯片,该芯片包括处理器,用于实现上述任意一个或多个实施例所涉及的功能,例如用于实现图2-图5中第一节点或者RT2所执行的方法,或者用于实现图2-图5中粘连节点或者RT3所执行的方法。可选地,所述芯片还包括存储器,所述存储器,用于处理器所执行必要的程序指令和数据。该芯片,可以由芯片构成,也可以包含芯片和其他分立器件。

[0165] 本领域内的技术人员应明白,本申请的实施例可提供为方法、系统、或计算机程序产品。因此,本申请可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且,本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[0166] 本申请是参照根据本申请实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0167] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制品,该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0168] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0169] 显然,本领域的技术人员可以对本申请进行各种改动和变型而不脱离本申请的范围。这样,倘若本申请的这些修改和变型属于本申请权利要求及其等同技术的范围之内,则本申请也意图包含这些改动和变型在内。

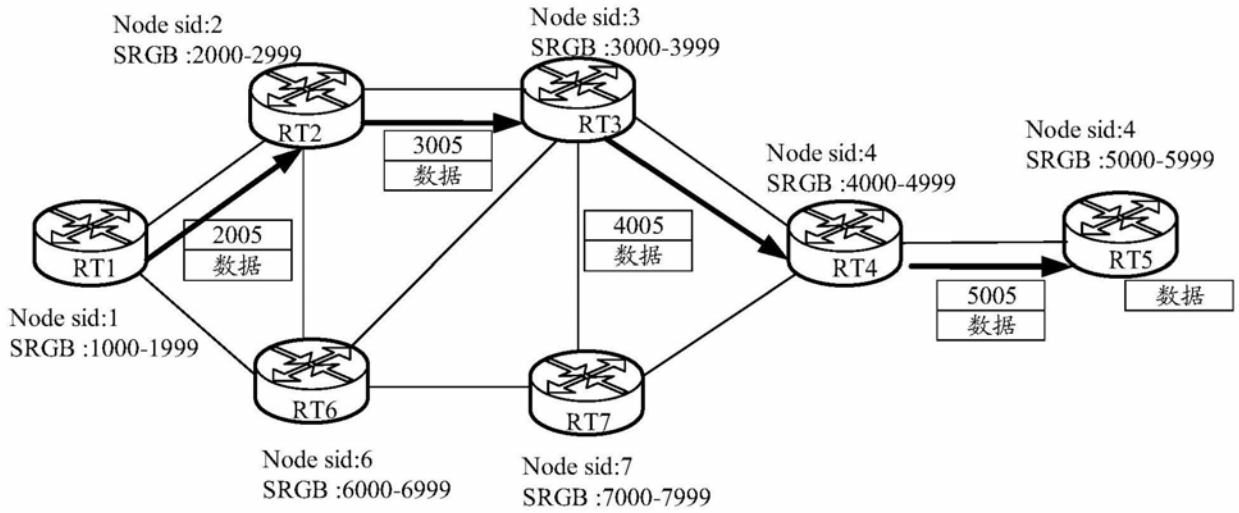


图1A

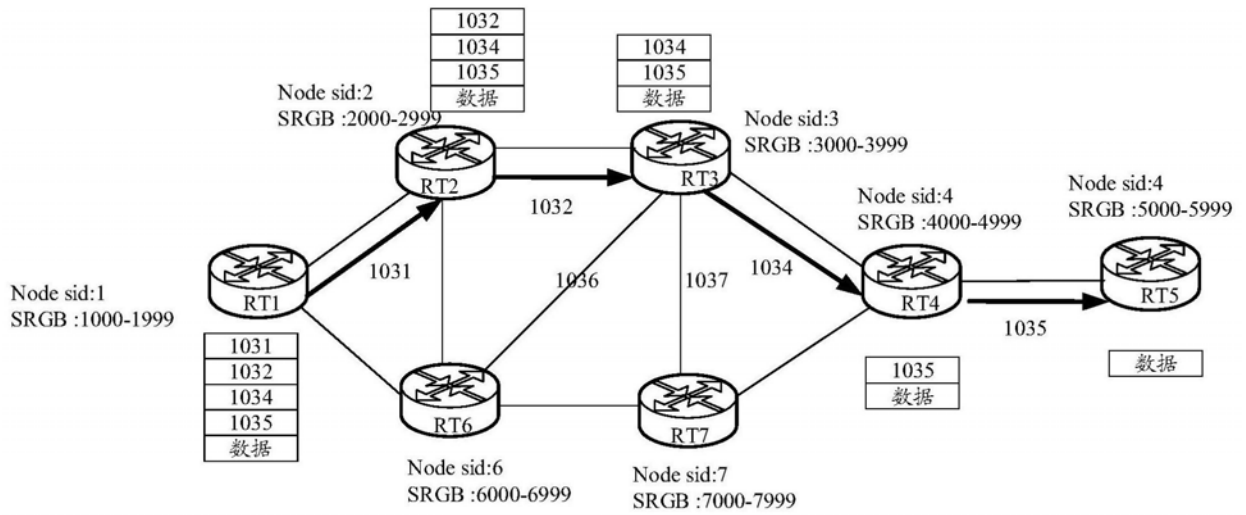


图1B

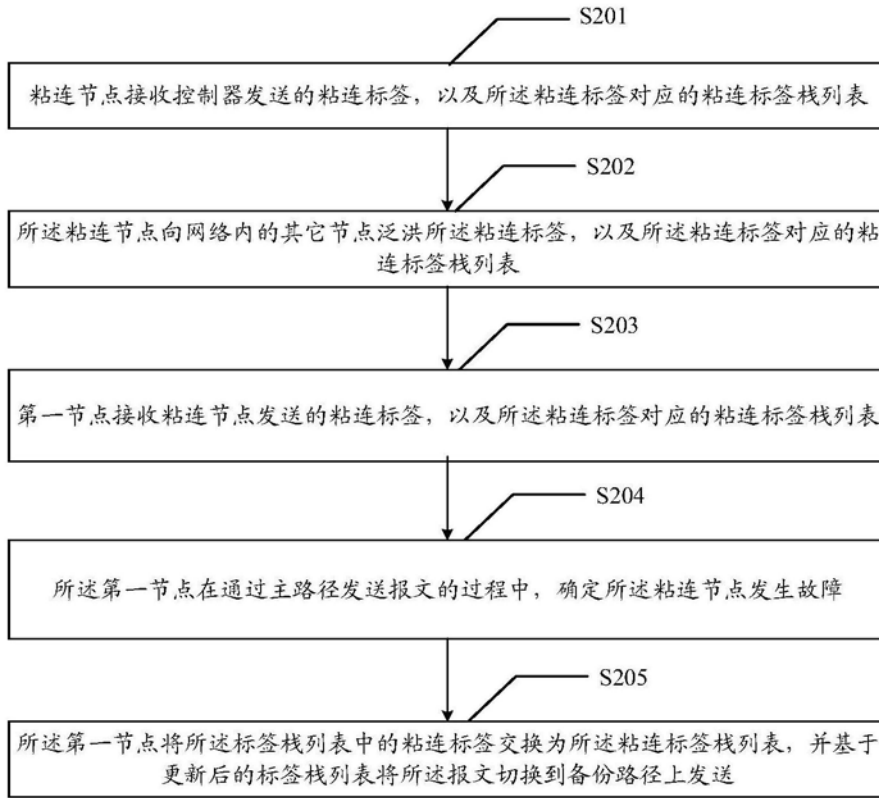


图2

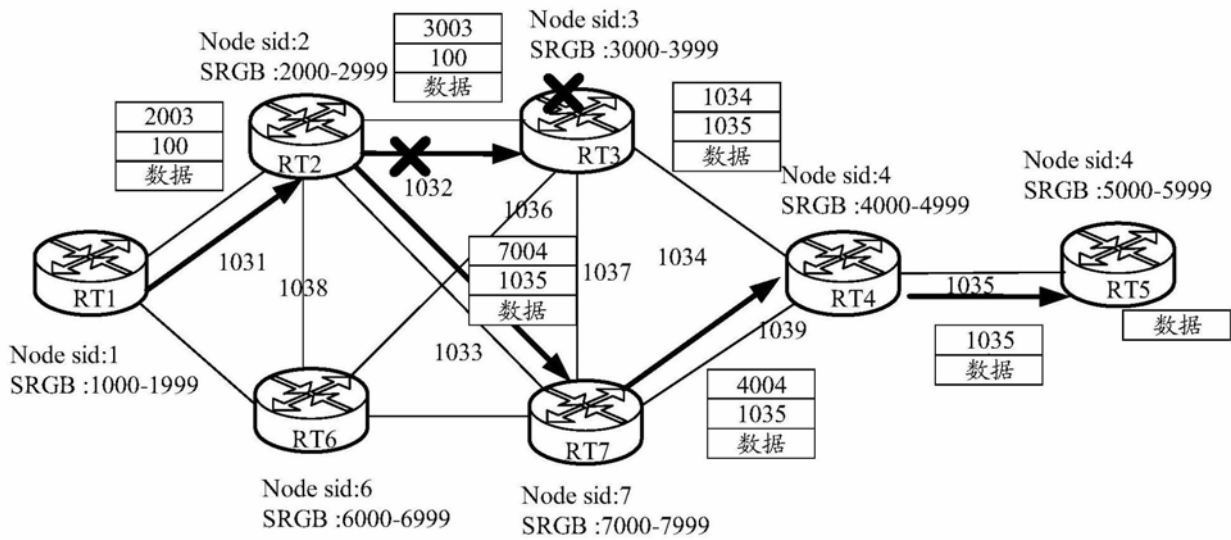


图3

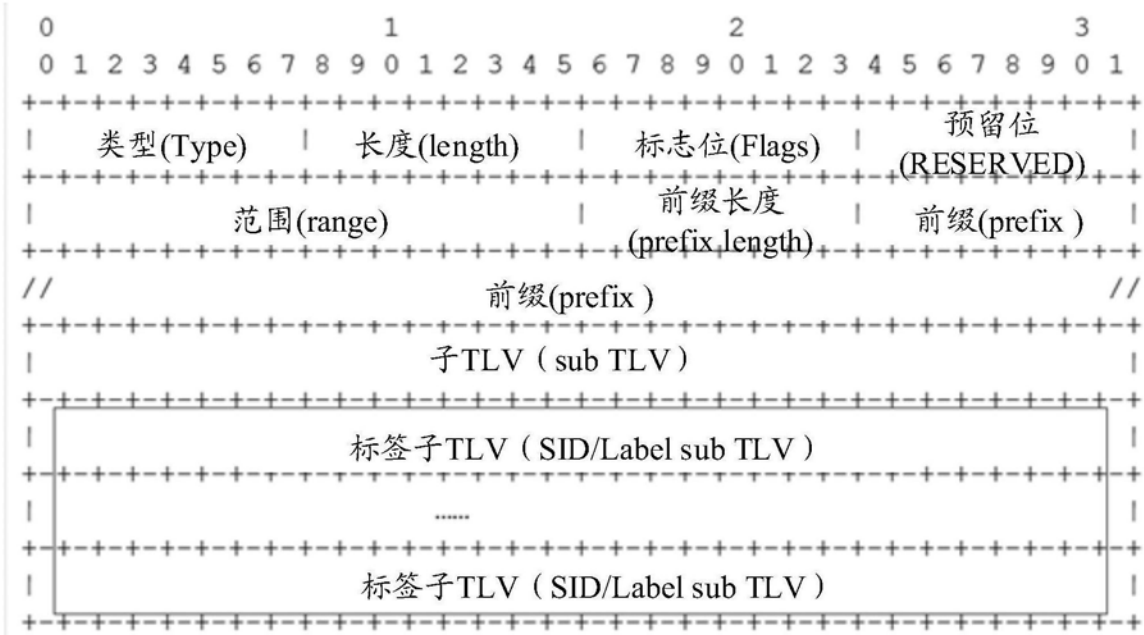


图4

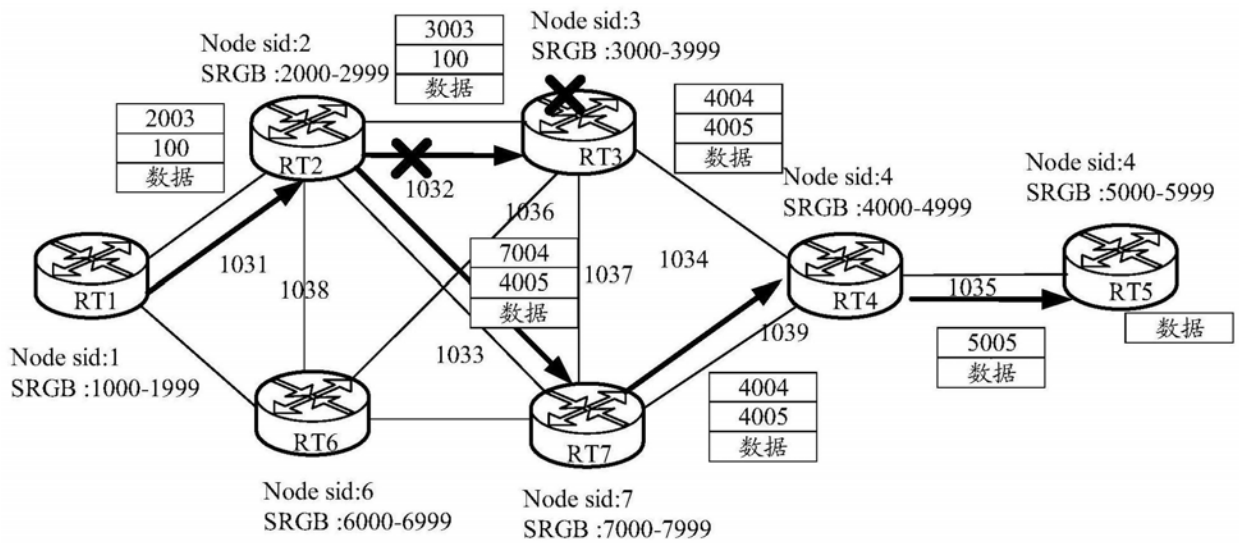


图5

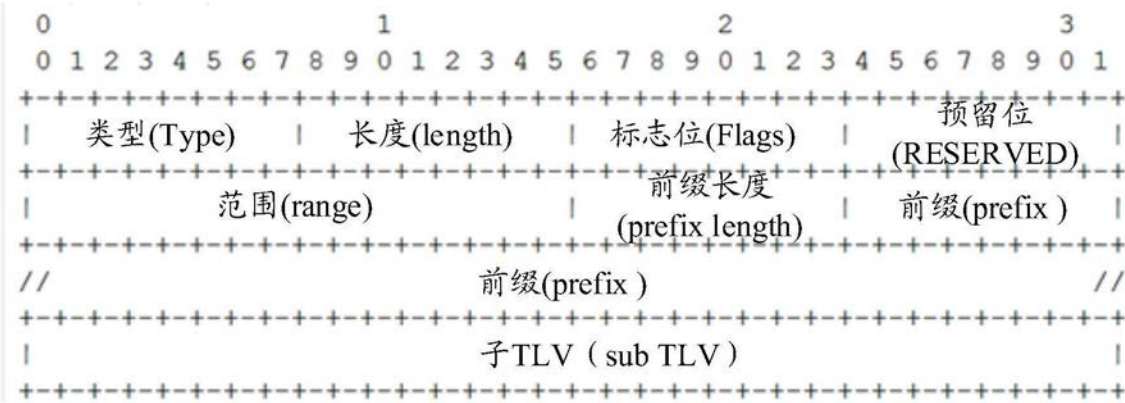


图6

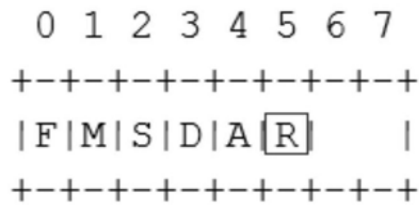


图7

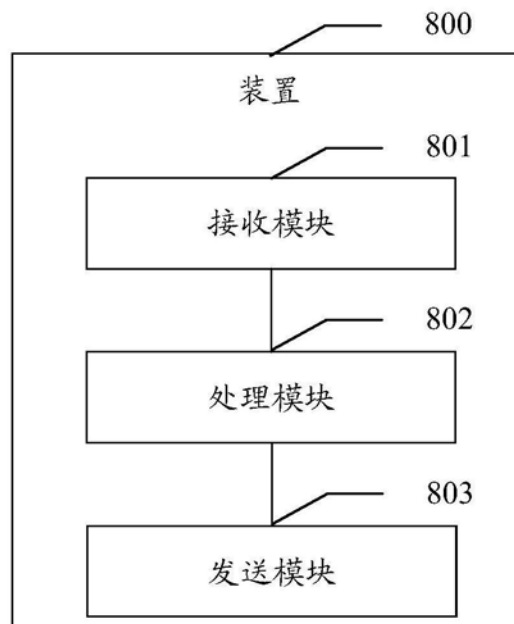


图8

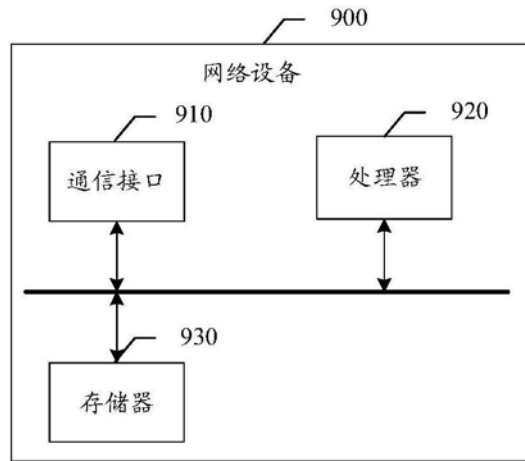


图9