H3C网络之路
2013年第一期
总第十九期

HA专题

高可用性

H3C

目录

Content

网络高可用性概述 陈旭盛

链路级HA

005 链路聚合技术 吴 帅
013 PVST技术 王茂葳
023 RPR技术 涂勇军
036 HDLC捆绑 李 冰

路由HA相关协议剖析

044 ISIS GR 张 岩
051 LDP GR 乔 辉
062 RSVP GR 苏艳梅
067 NSR/SSO 李 军

链路及路径检测技术

072 CFD 李连华
089 BFD原理与应用 曹佐清
098 MPLS OAM 陈锦军

网络级HA

108 VRRP/VRRPE协议简介 李朝辉
121 IRF MAD应用模型及技术分析 叶玉其
130 IP FRR实现概述 丁 炎
134 MPLS TE FRR技术分析 苏艳梅
144 VPN FRR实现概述 蒋益群
148 MPLS TP 贾欣武

系统级HA

163 ISSU 蒋茂勇
176 ComwareV7 HA实现综述 吴 玲

网络高可用性概述

文/陈旭盛

引言

随着互联网的爆炸式发展，越来越多行业的核心业务严重依赖网络开展。作为网络应用发源地的ICT行业自不消说，证券、金融行业现在离了网络几乎寸步难行，零售行业更是通过网络引发了电商革命。互联网大潮之下，几乎没有哪个行业不被波及，触网成了不得不面对的选择。

大量事关行业生死的业务在网络的广泛部署，同时网络带宽也在呈指数级增长，这使得基础网络的短时间中断就会影响大量业务，造成重大损失。据媒体报道，亚马逊13年2月1日网络故障50分钟，直接导致了500万美金的经济损失；而加拿大安大略市社会援助网络04年7月份的一次持续16天的故障，则导致了政府接近5亿美金的损失。

在这种背景下，高可用性（High Availability，HA）日益成为构建网络的关键目标。上至运营商，下到大中型企业客户，构建基础网络时，5个9（99.999%）的网络可用性成为必然的选择，这相当于一年中，各种原因造成的网络中断大约为5.256分钟。对于H3C而言，能否提供完善的高可用产品和解决方案，直接关系到核心竞争力，关系到在未来激烈的竞争中能否生存。

可用性度量

在讨论网络高可用性之前，我们有必要先搞清楚到底什么是网络可用性。所谓网络可用性，顾名思义，就是网络的可用程度， 网络无故障运营的时间占总运营时间的百分比可以用来比较好的反映网络的可用性。如果要以数学方式严谨定义，需要引入两个统计指标：

◆ 平均无故障时间（Mean Time Between Failures，MTBF）。 即两次故障之间正常运行的平均时间。MTBF越大，表明越不容易出故障，可用性自然高。MTBF反映的是网络的可靠性（reliability）。

◆ 平均修复时间（Mean Time To Repair，MTTR）。即出现故障后修复故障的平均时间。MTTR越小，表示故障时间越短，可用性也就越高。MTTR反映的是网络的容错能力（fault-tolerant capability）。

有了这两个指标，可用性可以如下计算：

$$Availability = \frac{MTBF}{MTBF+MTTR} \quad ———（1）$$

我们看一个例子可以更好的理解这个定义：一个网络在一年365天中，出现了4次故障，故障总时间是1个小时，计算这个网络的可用性。

首先，计算网络的1年中的可用时间，即总的时间减去故障时间，为：365*24 – 1 = 8759个小时。

接下来计算MTBF和MTTR：

平均无故障时间，就是总的可用时间除以故障次数：

MTBF = 8759/4 = 2189.75 小时；

平均修复时间，就是故障时间除以故障次数：

MTTR = 1/4 = 0.25 小时

最后，Availability =MTBF/（MTBF+MTTR）= 99.988%

即这个网络的可用性为99.988%，接近4个9的水平。

下面这个表格揭示了n个9的可用性的情况下，对应到一年中中断服务的时间：

表1 n个9的可用性对应的1年中中断服务时间

The nines	Off–service time
90%	36.5天
99%	3.65天
99.9%	8.76小时
99.99%	52.56分钟
99.999%	5.256分钟
99.9999%	31.536秒

你会发现，和主观想象不一样，99%的可用性，其实很差，一年有3.65天网络不可用。即使是4个9的水平，其可用性也并不高，一年中断服务的时间达到了53分钟。只有到5个9的水平，一年只中断服务5分钟，可用性可以接受。这也是为什么很多网络建设的可用性目标定为5个9的原因。

提高网络可用性

我们对公式（1）作一下变化：

$$\frac{1}{Availability} = 1 + \frac{MTTR}{MTBF} \qquad ———（2）$$

从（2）可以直观的看出，提高MTBF，降低MTTR，都可以提高网络的可用性。

先说提高MTBF，也就是避免网络发生故障。在实践中，导致网络故障的因素很多，大致可以分为不可控因素和可控因素。不可控因素主要指战争，自然灾害，大范围停电，人为破坏等，我们不予讨论。可控因素则大致可以归纳如下：

- 设备软硬件故障
- DOS/DDOS攻击
- 线路故障
- 网络拥塞
- 维护升级
- 操作失误

一个很自然的想法，就是把上述各种可控因素对网络可用性的影响降到最小，有针对性的对上述因素一一采取措施：加大测试投入，提高设备软件版本的质量；构建安全网络避免DOS/DDOS攻击；提高

线路质量，减少线路故障；优化网络设计或升级网络，避免拥塞；尽量使用在线升级技术或避开用户高峰进行网络升级，减少维护带来的网络中断；加大对客户网管人员的培训，避免误操作等。通过这些措施，使网络尽量不出故障，事实上就延长了MTBF，可以达到提高网络可用性的目的。

但是，延长MTBF的方法都有其局限。以加大测试投入以提高软硬件版本的质量为例：确实，高质量的软硬件版本可以提高MTBF，是构建高可用网络的基础，也是包括H3C在内的设备商一贯追求的目标，设备的硬件部件的MTBF通常已经在30年以上。但是，对于软件来说，几十年的软件工程实践已经给出答案，再严格的测试也不能穷尽所有问题，软件测试只能尽可能多的暴露问题，避免遗留对客户应用有重大影响的问题。

其他技术，如在线升级技术，比如ISSU（In-Service Software Upgrade）也只能在特定条件下才能保证对业务没有影响。而线路故障，网络拥塞，维护升级，操作失误，虽然可以尽量减少，但难以完全避免。

毫不夸张的说，在实际的网络运营中，出现技术或非技术因素导致的故障是必然的。这是客观事实，我们必须面对，因此，开发能使网络从故障中快速恢复的技术，对提高网络可用性就显得尤其重要。这对应的就是缩小MTTR。事实上，如果网络总是能在不中断（绝大部分）业务的情况下恢复，对多数用户，就其业务体验来说，甚至可以认为是无故障的。

网络高可用性技术

当前的高可用性技术，绝大部分聚焦于如何使网络从故障中快速恢复。这些技术也是本次期刊的重点。

首先，要想让网络从故障中快速恢复，必须能快速自动发现故障。如果出现故障不能尽快发现，或者还需要手工干预，怎么都说不上高可用。快速检测节点及链路的故障，可以通过硬件心跳，也可以通过软件协议，或者协议以专有硬件方式实现。BFD就是很典型的快速故障检测协议，独立于上层协议，但和上层协议联动，一旦检测到故障立即上报促成上层协议的快速收敛。虽然上层协议本身一般都有保活机制，可以通过调整保活报文发送频率来加快故障检测，但上层协议的保活报文一般除了保活之外，还兼其他功能，不够简洁，处理效率不高，所以用于快速故障检测会有很多负面作用（比如路由协议的快速hello等）。专有的BFD要简洁高效得多,也便于通过专有芯片实现。

其次，应该有备用的网络部件，一旦发现故障，备份部件能迅速接替工作。在设备层面，备用的网络部件可以是链路、单板以及整个设备。在网络层面，则需要考虑转发路径的备份，一旦先前的转发路径出现故障，可以把业务流量快速切换到备份路径。

对于备份路径，可以是故障前已经事先构建好的，也可以是在故障后通过计算获得，当然，前者的故障恢复时间要短得多。链路聚合、Smart Link、RRPP、RPR、ECMP等技术可以认为是在局部（一跳范围内）建立了备份链路；而MPLS TE/FRR则是在网络全局建立好了备用转发路径；路由快速收敛、STP/RSTP等技术则是在出现故障后，在全局范围内重新计算选路。

并不是所有故障，都需要备份路径或链路。事实上，在主控冗余的情况下，借助于控制与转发分离，如果只是控制平面出现故障，转发平面正常的情况下，可以通过一些技术处理在业务几乎不中断的情况下，恢复网络。主用控制平面出现故障时，转发平面保持不变，备用控制平面接替工作并重新构建控制平面的各种状态，然后启动平滑进程，使控制和转发平面数据一致，从而完成网络的完全恢复。在这期间，数据转发没有间断，客户业务没有感知，网络始终是可用的。NSR/NSF/SSO/GR就是这类技

术，一些4-7层的备份技术也可以归入这类。 当然，不同的技术，重新构建控制平面状态的过程不同，比如NSR（Non-Stop Routing）较为完全的备份了路由协议的状态，重新构建控制平面状态这个步骤基本可免，不会引起控制平面的震荡，实现了网络可用性的最大化。而NSF（Non-Stop Forwarding）备份的协议状态则很有限，需要协议重新收敛以构建控制平面状态。

对目前常用的高可用性技术，根据前面的描述，可以作一个简单的归类：

◆ 快速故障检测技术：BFD，CFD；

◆ 链路捆绑技术：如以太网链路聚合、MP、MFR等；

◆ 环网技术：RPR、RRPP；

◆ 其他二层冗余技术：STP、Smart Link；

◆ 网关冗余：VRRP（E）、HSRP、GLBP；

◆ 不间断转发技术：NSR/NSF/SSO/GR；

◆ 综合性HA技术：IRF，MPLS FRR；

在本期刊物接下来的文章中，会对上述技术一一详细介绍。

构建高可用网络

虽然高可用性技术的使用可以较大的提高网络的可用性，但构建高可用性网络，从来不是一个单纯的技术问题，而是一个系统工程，需要通盘考虑。

◆ 在网络规划设计阶段，需要细致分析客户业务模型，确定基础网络拓扑，对影响网络可用性的关键节点和链路必须做充分的冗余设计，并根据网络结构，在各个层次、各个节点部署合适的高可用性技术。

◆ 持续进行网络维护和优化。合理的网络维护和优化，可以避免90%以上的网络故障。对全网的业务流量需要利用高效的网管工具持续监控、分析、预测、优化，最大限度的规避网络拥塞。在进行软硬件版本升级或新设备，新业务上线时，需要事先详细规划，制定应急预案。

◆ 高度关注网络安全问题。网络安全问题已经成为影响网络可用性的重要因素，定期对主机系统和网络系统进行安全评估，构建一个多层次的安全防御和预警系统对高可用性网络非常有必要。

◆ 做好业务系统和网络系统的协调。业务和网络的关系，类似于车和路。同样宽的路，控制得当的话，可以跑更多的车流。不合理的业务处理模型，会人为造成网络拥塞点。积极分析业务模型，并适当的进行调整，可以让网络更通畅。

链路聚合技术

文/吴 帅

链路聚合概述

链路聚合（Link Aggregation）由IEEE802.3ad定义，包括链路聚合目标及实现、聚合子层的功能及操作、链路聚合控制及LACP等。链路聚合将多条物理以太链路聚合形成为一个逻辑链路聚合组，这种聚合对于上层协议和应用是透明的，上层协议和应用将同一聚合组内的多条物理链路视为单一链路，无需改变即可在其上运行。

图1 链路聚合示意图

链路聚合的主要优点

◆ 增加带宽：聚合链路的带宽最大为聚合组中所有成员链路的带宽和，目前我司的交换机产品支持最多32条物理链路聚合，极大的拓展了链路带宽；

◆ 增加可靠性：聚合组存在多条成员链路的情况下，单条成员链路故障不会引起聚合链路传输失

败，故障链路承载的业务流量可自动切换到其他成员链路进行传输；

◆ 可负载分担：业务流量按照一定的规则被分配到多条成员链路进行传输，提高了链路使用率；

◆ 可动态配置：缺少人工配置的情况下，链路聚合组能够根据对端和本端的信息灵活调整聚合成员端口的选中/非选中状态。

正是由于链路聚合所具有的链路冗余性和可分担性，使其在很多实际应用场景中，成为提供链路级HA的不二之选，与此同时，其所具有的可动态配置性也保证了不需人为干预便可达到预期的高可用性效果。

链路聚合分类

当前主流厂商的链路聚合实现方式不尽相同，我司按照是否与对端交互LACP协议，链路聚合可分为静态聚合和动态聚合，静态聚合模式下的成员端口选中状态不受网络环境的影响，稳定性较高；动态聚合模式下的成员端口可根据对端相应成员端口的状态自动调整本端的选中状态，灵活性较高。在Cisco设备上支持LACP和PAgP（Port Aggregation Protocol，Cisco私有协议）两种聚合模式，可配置端口为强制加入，或者主动发送协议报文（LACP或PAgP）协商，或者被动处理协议报文，来与对端设备形成链路聚合。

另外，当前主流设备厂商都支持按照聚合成员端口的类型不同，将链路聚合分为二层聚合和三层聚合，二层聚合（bridge-aggregation）的成员端口全部为二层以太网接口，三层聚合（route-aggregation）的成员端口全部为三层以太网接口。

用户可根据不同的需求选择不同的聚合模式，我司设备的链路聚合可与使用IEEE802.3ad标准LACP协议的友商设备完美互通。

链路聚合工作机制

链路聚合所提供的链路级HA与其工作机制息息相关，下面针对我司实现对链路聚合工作机制进行简要介绍。

操作Key

操作Key是用来表示端口的聚合能力的一个数值，它是根据端口信息的组合自动计算而来，组合中任何一项的变化都会引起操作Key的重新计算，从而引起端口的选中状态发生改变，影响操作Key计算的端口信息包括：

◆ 聚合组ID

◆ 端口速率

◆ 端口双工模式

◆ 端口硬件限制值

◆ 端口聚合相关配置

在同一聚合组中，所有的选中端口的操作Key一定相同；在同一设备上，不同聚合组的操作Key一定不同；操作Key只是本端端口属性，聚合链路两端成员端口的操作Key可能不同。

聚合相关配置

端口的聚合相关类配置会影响操作Key计算，只有与聚合口的聚合相关类配置相同的成员端口才有可能成为选中端口，聚合相关类配置包括：

◆ 端口类型配置（Access/Trunk/Hybrid）

◆ 端口隔离配置

◆ VLAN相关配置

◆ QinQ /VLAN mapping

在聚合口上进行的聚合相关类配置，会同步到所有成员端口，在成员端口上进行聚合相关类配置，只会影响该成员端口的操作Key计算，另外需要注意的是，只有二层聚合成员端口的Key值计算才关心聚合相关类配置。

另外，有些端口配置不会影响操作Key计算，但这些配置只会在端口退出聚合组后生效，此类配置称为非聚合相关类配置，比如STP、GVRP、MAC地址学习等。

静态聚合

静态聚合组中的成员端口不会和对端交互LACP报文，所以系统不会自动根据对端信息调整本端成员端口的选中状态，系统将按照以下原则确定端口为选中还是非选中：

首先选择参考端口，聚合组内所有up的成员端口按照端口的高端口优先级->全双工/高速率->全双工/低速率->半双工/高速率->半双工/低速率的优先次序，选择优先次序最高、且聚合相关类配置与对应聚合接口相同的端口作为参考端口；如果优先次序相同，则选择其中端口号最小的端口作为参考端口。

选出参考端口后，系统将下面几种端口置为非选中状态：存在硬件限制的端口、down掉的端口、与参考端口Key值不同的端口以及按端口号排序超出选中端口数量上限的端口。除此之外的端口则置为选中状态，承担转发数据流量的任务。

动态聚合

动态聚合组内的成员端口通过收发LACP协议报文来将本端和对端的端口信息进行比较，以选择能够处于选中状态的成员端口，使双方可以对各自端口的选中/非选中状态达成一致。

在动态聚合组中，参考端口的选择是设备通过交互信息，首先比较设备ID（系统优先级+系统MAC地址），先比较系统优先级，如果相同再比较系统MAC地址，设备ID小的一端为优。然后，对于设备ID为优的一端，比较聚合组内成员端口的端口ID（端口优先级+端口号），端口ID最小且聚合相关类配置与聚合口一致的端口做为参考端口。

在动态聚合组中确定端口的选中/非选中状态时，除了和静态聚合相同的判断条件之外，还需要判断本端端口的对端端口的操作Key和聚合相关类配置是否与参考端口的对端端口相同，如果不同，是不能成为选中端口的。另外，设备上支持的选中端口数量有限，如果后加入的成员端口一旦比当前选中端口更优，则会立刻取代其选中端口的位置。

链路聚合负载分担

链路聚合的可负载分担特性是其能够提供链路级HA的关键所在，目前负载分担主要有两种方式：逐包和逐流负载分担。逐包负载分担虽然能够达到更加均匀的分担效果，但这种方式容易产生报文乱序，所以当前主流厂商所采用的负载分担方式都是逐流负载分担，使用Hash算法决定使用具体哪条聚合成员链路转发流量。

聚合负载分担算法简述

聚合链路虽然被上层协议和应用视为单一链路，但是实际的业务流量转发任务必须要由聚合组中的物理成员链路来承担，为了提高聚合带宽及物理链路使用率，链路聚合使用负载分担策略将业务流量近似均匀的分配到各条成员链路，我司交换机采用hash算法来实现链路聚合的负载分担。

聚合环境下，在入方向，聚合选中端口接收报文，进行地址学习，但是和普通的物理端口不同，报文进入选中端口，根据内部端口ID查询此端口为哪个聚合组的成员端口，将MAC、ARP学习在对应的聚合接口而不是学在接收报文的物理成员端口。比如我司S5800交换机，其端口的内部端口ID由端口所在芯片号和内部端口号两部分组成，通过端口ID进行查询，如果返回信息中相应的字段为真则说明此端口为聚合成员端口，并且从返回信息的其他字段也能够得知此端口为哪一个聚合组的成员端口。

在出方向，如果转发表指示出端口为聚合接口，则需要根据聚合组号查询选中端口信息。以聚合组号为索引，可以查到聚合组选中端口的端口ID，以及Hash算法所使用的报文字段指示，然后根据Hash的输出结果决定报文最终的出端口。

另外，对于已知单播和组播或者广播，在出方向的处理是不同的。

以我司S5800为例，对于已知单播，Hash算法对报文相关字段取最后3bit进行异或运算，然后与当前聚合组中的选中端口数量进行求模运算，Hash的结果决定了报文最终的出端口。对于组播流量，如果聚合接口是已知组播的出端口，则聚合组中所有选中端口在组播表项里都是出端口，组播流量“泛洪”到各芯片，有出端口的芯片需要进一步查看位图信息，决定是否分担该流量。这需要以Hash值为索引，查询端口阻塞列表，该表置1的位所对应的端口将被阻塞。广播报文的处理和组播类似，所不同的是参与Hash计算的报文字段不同。

在Cisco交换机上，早期的低端Cisco交换机只允许绑定2或者4这样偶数数量的成员链路到同一聚合组中，Catalyst 6000或更高端设备则无此限制，其设备在负载分担方面的处理和我司设备原理基本相同。比如根据报文源MAC进行负载分担，如果聚合组内有两条成员链路，那么取各条流源MAC的最后1bit进行异或，如果成员链路增加到四条，则取各条流源MAC的最后2bit进行异或，然后根据异或的结果再决定由哪一条成员链路来承担转发任务。

虽然当前主流厂商都采用逐流，基于Hash算法的负载分担，但因为在逐流分担中会产生分担不均的问题，尤其在成员链路的数量不为2的指数倍时，将来应该可以实现在现有基础上，根据当前成员链路所剩余带宽进行二次Hash，从而实现更加均匀的负载分担效果。

负载分担方式

当前主流厂商都支持使用比如报文源MAC、目的MAC、源IP、目的IP、源端口、目的端口或者它们的组合等做为Hash算法的输入字段，来实现不同的负载分担方式。以我司S5800交换机为例来说，在系统视图下进行全局聚合负载分担类型配置，其支持如下负载分担方式：

◆ 根据报文类型自动匹配负载分担类型；

◆ 根据源IP地址进行聚合负载分担；
◆ 根据目的IP地址进行聚合负载分担；
◆ 根据源MAC地址进行聚合负载分担；
◆ 根据目的MAC地址进行聚合负载分担；
◆ 根据源IP地址与目的IP地址进行聚合负载分担；
◆ 根据源IP地址与源端口进行聚合负载分担；
◆ 根据目的IP地址与目的端口进行聚合负载分担；
◆ 根据源IP地址、源端口、目的IP地址与目的端口进行聚合负载分担；
◆ 根据报文入端口、源MAC地址、目的MAC地址之间不同的组合进行聚合负载分担。

在聚合接口视图下进行聚合组的聚合负载分担模式配置，S5800交换机支持：

◆ 根据源IP地址进行聚合负载分担；
◆ 根据目的IP地址进行聚合负载分担；
◆ 根据源MAC地址进行聚合负载分担；
◆ 根据目的MAC地址进行聚合负载分担；
◆ 根据目的IP地址与源IP地址进行聚合负载分担；
◆ 根据目的MAC地址与源MAC地址进行聚合负载分担。

在我司IRF环境下，存在跨设备链路聚合时，默认采用本地转发优先的负载分担方式，即从本设备接收的已知单播出接口若为聚合接口，优先从本设备上的选中端口发送出去，本设备上有多个选中端口时，在本单板的选中端口之间负载分担，如果本设备上没有选中端口，才在其他成员设备的所有选中端口间负载分担，这样做的好处是避免了框间流量对于IRF堆叠口带宽的消耗，增强设备稳定性。

聚合流量重定向

目前我司主流交换机产品都支持动态聚合组流量重定向功能，从而能够提供更加细致的HA特性。

在动态聚合链路两端都使能了聚合流量重定向功能后，对于集中式设备，当关闭聚合组内某选中端口或者IRF环境下重启某台有聚合组选中端口的成员设备时，系统可以将这些端口上的流量重定向到其他选中端口上，从而实现聚合链路上流量的不中断。对于分布式设备，当重启某块有聚合选中端口的单板或者IRF环境下重启某个有聚合选中端口的成员设备或单板时，同样能够实现聚合链路上流量不中断。

需要注意的是，如果同时使能聚合流量重定向功能和生成树功能，因为聚合口参与生成树计算，在重启单板、设备时会出现少量的丢包，因此不建议同时使能上述两个功能。

附录：LACP协议简介

LACP概述

LACP(Link Aggregation Control Protocol)基于IEEE802.3ad标准，是一种实现链路动态聚合的协议，其通过与对端交互LACPDU实现链路聚合控制。在LACP协议中，链路本端称为Actor，对端称为Partner，本端通过向对端发送LACPDU，向对端通告自己的系统优先级、桥MAC、端口优先级、端口号和操作Key，对端将这些信息与其他端口所保存的信息比较以选择能够聚合的端口，两端可以对端口加

入或退出某个动态聚合组达成一致，从而决定哪些链路可以加入同一动态聚合组以及某条链路何时能够加入聚合组。

LACP报文

LACP报文格式如下所示：

LACPDU Structure	Octets
Destination Address	6
Source Address	6
Length/Type	6
Subtype = LACP	1
Version Number	1
TLV_type = Actor Information	1
Actor_Information_Length = 20	1
Actor_System_Priority	2
Actor_System	6
Actor_Key	2
Actor_Port_Priority	2
Actor_Port	2
Actor_State	1
Reserved	3
TLV_type = Partner Information	1
Partner_Information_Length = 20	1
Partner_System_Priority	2
Partner_System	6
Partner_Key	2
Partner_Port_Priority	2
Partner_Port	2
Partner_State	1
Reserved	3
TLV_type = Collector Information	1
Collector_Information_Length = 16	1
CollectorMaxDelay	2
Reserved	12
TLV_type = Terminator	1
Terminator_Length = 0	1
Reserved	50
FCS	4

图2 LACP报文结构

报文各字段含义如下：

◆ 目的地址：协议规定为组播MAC地址：01-80-C2-00-00-02；

◆ 源地址：协议规定为发送LACPDU端口的MAC地址；

◆ 协议类型：0x8809；

◆ 子协议类型：0x01；

◆ LACP版本号：0x01；

Actor端与Partner端均包括以下详细信息：

◆ 端口信息：0x01；

◆ 端口信息长度：0x04；

◆ 系统优先级：默认情况下为0x8000，即32768；

◆ 系统MAC地址：交换机桥MAC地址；

◆ 端口操作key：系统根据端口的配置组合生成的一个key值；

◆ 端口优先级：默认情况下为0x8000；

◆ 端口号：链路聚合组中链路两端的端口号；

◆ 端口状态：占一个字节，详细记录了本端与对端目前的状态信息，Actor_state和Partner_state里的八位都包括以下信息：

LACP_Activity	LACP_timeout	Aggregation	Synchronization	Collecting	Distributing	Defaulted	Expried

◆ LACP_Activity：协议规定，主动发送LACP协议的报文的端口该bit位置1，被动发送或不发送LACP报文的端口该bit置0，我司设备都为主动发送；

◆ LACP_timeout：这个bit位为LACP是长超时还是短超时，1代表短超时，0代表长超时；

◆ Aggregation：为1时表明此链路是可聚合的，否则此链路是不可聚合的；

◆ Synchronization：表示链路是否处于同步状态，从对端发送过来的LACPDU中提取Partner的信息，查看是否与本端一致；

◆ Collecting：为1则表示可以接收数据帧，否则为0；

◆ Distributing：为1则表示可以发送数据帧，否则为0；

◆ Defaulted：发送端的接收状态机是否处于默认状态，为1标识接收状态机使用缺省操作信息；若为0，使用接收到的LACPDU中的信息；

◆ Expired：发送端的接收状态机是否处于超时状态，为1则表示接收状态机处于超时状态。

◆ 保留字段：占3个字节；

◆ Collector与Terminator相关字段标识了在TLV_type中包含的信息类型、长度；

◆ FCS:帧校验序列，由MAC生成。

LACP状态机

LACPDU协议运行机制受五个状态机的控制，分别为Receive machine、Periodic Transmission machine、Selection Logic、Mux machine、Transmit machine。LACP状态机的描述基于端口，聚合链路本端端口或者对端端口的关键信息发生了变化，都会引起状态机迁移，状态机间的关系如下图所示：

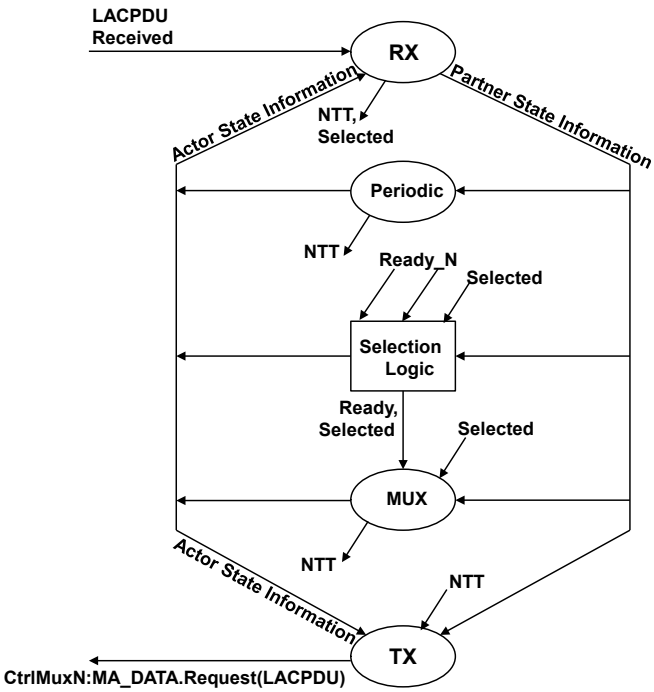


图3 LACP状态机关系图

◆ 当端口收到的LACPDU中的Partner信息，或者是系统默认的Partner信息，通过Receive machine反馈给其他状态机。标记Actor State Information的箭头表示的是状态机更新Actor信息的过程。当Periodic Transmission machine 控制报文的发送周期到了，或者修改Actor信息需要传输LACPDU给对端，这时通过NTT（need to transmit flag）信号把需要传递LACPDU报文通知给Transmit machine，Transmit machine发送LACPDU报文。

◆ Receive machine接收对端发送来的LACPDU，并记录对端信息。根据对端信息判断对端和本端的协议信息是否已经达成了一致，是否都处于可用状态。如果从对端发送过来的协议信息过期，则Receive machine设置为缺省参数，供其它状态机使用。

◆ Periodic Transmission machine状态机用来决定本端与对端需不需要定期发送LACPDU消息来判断是否继续维持一个聚合组（当对端与本端任意一个被配置为主动LACP，则会定时发送LACPDU消息，若两个都为被动LACP时，则不会传送LACPDU消息）。

◆ Selection Logic负责为相关的端口选择聚合组。

◆ Mux machine负责决定端口在指定聚合组中是选中还是非选中，以及根据协议信息，打开或关闭端口的收集和发送状态。

◆ Transmit machine控制LACPDU的传送（对端要求或者定时发送），当某端的状态信息发生改变或者定时器超时，则会发送LACPDU。通过发送NTT告知Transmit machine需要发送LACPDU信息。

PVST 技术

文/王茂葳

PVST技术介绍

PVST产生及衍变

最初生成树协议802.1D–1998 STP，只支持一棵树，并不支持快速迁移。后来出现的802.1w RSTP协议在此基础上增加了快速迁移特性，但也只支持一棵生成树。在VLAN技术出现后，网络中单生成树虽然能做到消除冗余链路，但是无法实现在不同的VLAN上负载分担。为了能够在不同的VLAN上实现负载分担，提高链路带宽的利用率，很多厂商将802.1D –1998 STP和802.1w RSTP放到VLAN上来运行，较为有名的就是Cisco提出来的PVST协议（Per VLAN Spanning Tree，按VLAN的生成树）。

由于最初的PVST协议无法与STP/RSTP互通，所以PVST并没有得到广泛的应用。为了解决这个问题Cisco提出了PVST+（Per VLAN Spanning Tree Plus，增强的按VLAN的生成树）的概念，可以很好的完成与STP/RSTP互通，PVST+在VLAN内实现的其实是STP协议，同样存在收敛速度慢的问题。Rapid PVST是Cisco提出的又一新概念，针对PVST+协议做的改进，这个改进类似于RSTP相对STP的改进，可以做到状态快速迁移，加快收敛速度。

说明：STP/RSTP/MSTP技术介绍，请参看网络之路第六期---交换专题。

基本原理

PVST

PVST（Per VLAN Spanning Tree，按VLAN的生成树）为每个VLAN维护一个单独的生成树实

例。默认情况下，每个VLAN都将运行单个生成树，允许以每个VLAN为基础启用或禁用STP。每个VLAN内的生成树实例都有单独的网络拓扑结构，相互之间没有影响。这样即可消除了VLAN内的冗余环路，还可以实现不同VLAN间负载分担。

VLAN内生成树实例，就相当于把STP协议分别用在每个VLAN上，和普通的STP实例相似，有着根桥、根端口、指定端口和指定桥等相同的概念，端口状态也是沿用了STP的5种基本状态。不同的是PVST是通过在VLAN内传递PVST BPDU报文来确定网络拓扑结构的。PVST BPDU是PVST的协议报文，报文中不仅携带了Cisco独有标志，还携带有STP标准字段。PVST BPDU目的MAC换成了Cisco保留地址01-00-0C- CC-CC-CD。

PVST+

PVST+（Per VLAN Spanning Tree Plus，增强的按VLAN的生成树）是在PVST的基础上实现的，也是为每个VLAN内维护一棵单独的生成树实例。PVST+中的“+”号表示由Cisco通过专有特性来增强STP802.1D。PVST+在增强STP802.1D协议的同时，也解决了PVST与STP /RSTP协议无法互通的缺陷，PVST+也因此得到了广泛的认同。

PVST+对802.1D-1998 STP的增强，主要是增加了Cisco开发的几种新特性：

◆ PortFast：生成树PortFast能够使得端口跳过监听和学习状态，直接进入转发状态；

◆ UplinkFast：作用是能够在直接的链路故障之后提供快速的收敛；

◆ BackboneFast：是对UplinkFast的一种补充，降低默认收敛时间；

这几种特性都是为了加快802.1D生成树的收敛速度。

PVST+在VLAN 1上运行STP协议，通过STP BPDU报文来确定网络拓扑结构，这样的带来的好处是实现了与STP/RSTP协议互通。其他VLAN内运行PVST协议，通过PVST BPDU协议报文确定网络拓扑结构，每个VLAN都会发送带有VLAN Tag的PVST BPDU报文，拓扑运算独立进行。

Rapid PVST

Rapid PVST相对PVST+做的改进主要是加快收敛速度。Rapid PVST在VLAN上使用标准的802.1w RSTP状态机支持生成树，协议报文仍然是PVST BPDU格式，但报文中携带RSTP的相应信息。在VLAN 1上也是运行的RSTP协议，其他VLAN上形成与RSTP类似的状态机，加快收敛速度。

简单说明一下，Cisco和我司设备现在对PVST协议配置实现情况。由于原来的PVST无法与STP/RSTP互通已经被弃用，Cisco设备现在支持的PVST命令有两个：PVST和Rapid PVST，PVST命令对应的是PVST+协议，Rapid PVST命令对应的是Rapid PVST协议。我司设备只有一个PVST命令，实现了Cisco Rapid PVST功能，并且能够兼容Cisco PVST+。文中后面提到的PVST都是PVST+协议。

PVST协议报文

PVST的协议报文确切说包含两种，在VLAN 1内协议报文是STP/RSTP BPDU，在其他VLAN内 协议报文是PVST BPDU。选举过程与STP基本相同，在网络初始化时，所有设备都把自己当作根桥，生成各自的配置BPDU并周期性（2S）向外发送，当网络拓扑稳定以后，只有根桥交换机周期性向外发送配置BDPU，维护网络拓扑稳定。

对于不同端口类型，PVST发送的BPDU格式也有所差别：

◆ Access端口，PVST将根据该VLAN的状态发送STP格式的BPDU；

◆ Trunk端口，PVST将在VLAN 1内根据该VLAN的状态发送STP格式的 BPDU，而对于其他本端口允许通过的VLAN，则发送PVST格式的BPDU。

STP/RSTP BPDU报文格式在这里不再过多介绍，主要介绍下PVST BPDU报文格式。在除VLAN 1之外的其他VLAN内，设备之间传递的是PVST BPDU报文，该报文有两种类型：配置BPDU和TCN BPDU：

◆ 配置BPDU：用于搜集设备信息，完成生成树计算，拓扑稳定情况时周期性的通告生成树信息；

◆ TCN BPDU：类似STP TCN报文，当网络拓扑结构发生变化时，设备会发送TCN BPDU，通知网络中相关设备网络拓扑结构已发生变化。

下面针对STP/RSTP BPDU、PVST BPDU、Rapid PVST BPDU及PVST TCN报文格式对比，如表1所示：

表 1 BPDU报文格式对比

	Bytes	STP	PVST	Rapid PVST	PVST TCN
Dest MAC	6	01-80-c2-00-00-00	01-00-0c-cc-cc-cd	01-00-0c-cc-cc-cd	01-00-0c-cc-cc-cd
Source MAC	6	端口MAC	端口MAC	端口MAC	端口MAC
Length	2	0x20	0x32	0x32	0x2b
DSAP	1	0xAA	0xAA	0xAA	0xAA
SSAP	1	0xAA	0xAA	0xAA	0xAA
Control	1	0x03	0x03	0x03	0x03
OUI	3		0x00000c	0x00000c	0x00000c
PID	2		0x010b	0x010b	0x010b
标准的STP Config	35	标准的STP Config	标准的STP Config	标准的STP Config 36Bytes	标准的STP TCN
PAD	1		0x00		UNKNOWN
VLAN TLV Type	2		0x00	0x00	
VLAN TLV Length	2		0x02	0x02	
VLAN TLV Value	2		VLAN ID	VLAN ID	

从报文结构对比来看，PVST BPDU报文相对STP BPDU报文改变在于以下几点：

◆ 报文的目的MAC地址改变，变成Cisco的私有MAC地址01-00-0c-cc-cc-cd；

◆ 报文中的OUI和PID字段填充固定值，OUI为0x00000c，PID为0x010b，作为Cisco的私有标志。报文中分析显示为Cisco和PVST+；

◆ 报文中携带标准的STP configuration字段，包含了根桥ID、根路径开销、指定桥ID等拓扑运算必要的信息；

◆ 报文携带802.1Q VLAN标签，确定该协议报文归属VLAN。

PVST端口角色及状态

前面已经阐述过PVST在VLAN 1内运行的是STP协议，端口角色及状态自然也没有变化。其他VLAN内虽然是按照PVST BDPU报文进行拓扑运算，但是端口角色及状态还是由报文中携带标准的STP

configuration字段来决定，因此其端口角色及状态和STP相同。

基本特性

InconsistentPortType-Protected特性

InconsistentPortType-Protected，端口类型保护，通过设备发出的BPDU报文，检测链路两端端口配置是否正确，如果配置不一致，端口收到不正常的PVST BPDU报文，就会触发该特性。使能PVST之后，该功能默认开启，不能配置或删除。

触发该特性典型情况有以下两种：

◆ 链路两端端口类型不同，例如一端是Trunk端口，另一端是Access端口，会触发该特性；

◆ PVST设备收到PVST报文之后，检查报文VLAN Tag头和TLV字段中的Value值（即VLAN_ID），如果不一致时会触发该特性。

触发该特性时，不管端口处于什么角色，都会自动转为discarding状态，直到端口收到正常的PVST BPDU报文，才会正常计算生成树和端口角色状态。

Loopback-Protected特性

在PVST模式下，端口的任意实例收到自己的BPDU报文，将阻塞该端口的所有实例。在收到自己的报文阻塞端口后，等待2倍的Forward Delay时间，该时间内任意实例都收不到自己的BPDU报文，则放开该阻塞端口。

PVST对STP的增强特性

PortFast特性

生成树PortFast特性能够使链路层的接入端口跳过监听和学习状态，直接进入转发状态。其目的就是为了尽量缩短接入端口等待STP收敛时间，类似于RSTP中的边缘端口。启用PortFast特性的STP状态机，如图1所示：

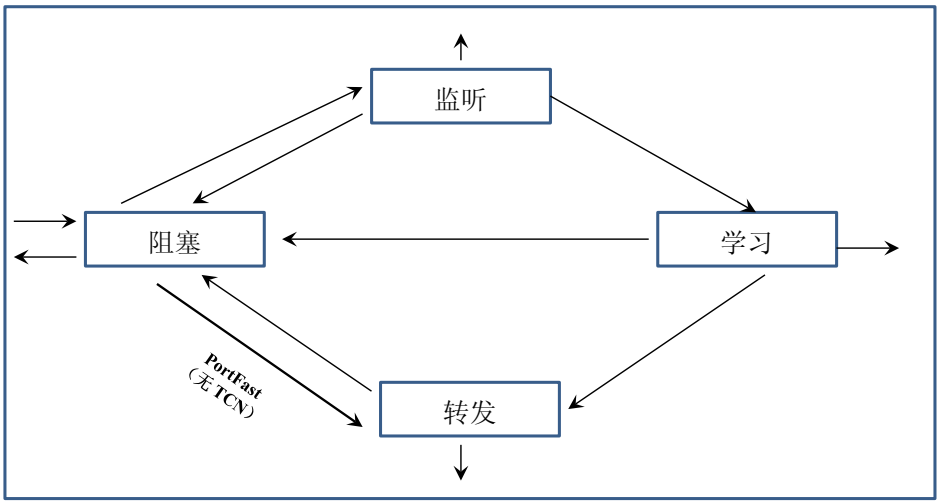


图 1 PortFast特性的STP状态机

由图1状态机变化可以看出，STP将跳过监听和学习状态，直接从阻塞状态过渡到转发状态。此外，该特性还能够抑制拓扑变更通告。

该特性用在与服务器或用户等终端设备相连，可以使得这些设备立即连接到网络中，而不用等待生成树收敛，缩短端口接入时间。

注意：在连接到其他交换机的端口上启用该特性时，可能会形成环路，避免这种配置。

UplinkFast特性

当网络中存在多个上行链路时，如果直连链路出现故障，那么生成树UplinkFast能够使另外一条链路快速收敛，达到快速转发目的。上行链路组中包括根端口和一组阻塞端口，初始状态根端口处于转发状态，某个时段根端口网段发生故障，UplinkFast将解除阻塞端口，并将其直接过渡到转发状态，不需要经过监听和学习状态，该切换过程基本上在5S内可以完成，类似于RSTP中Alternated端口快速切换。

应用场景：如果到根桥存在一条或更多条冗余链路时，使用该特性能够在直连的链路故障之后提供快速收敛。

BackboneFast特性

PVST模式时，影响生成树端口状态及网络收敛的计时器有3个：Hello Timer（默认值2s）、Forwarding Delay Timer （默认值15s）、Max Age（默认值20s）。Max Age是用于确定BPDU是否超时，如果端口等待Max Age时间内没有收到BPDU更新消息，就会改变端口状态重新计算。BackboneFast特性目的就是为了节省Max Age这个最大老化时间。

BackboneFast实现过程

正常情况下，在Max Age时间内如果收到次优BPDU，会忽略此报文。使能了Backbone –Fast的设备，在接收到次优BPDU之后，接收交换机将会尝试确定是否具有到达根交换机的替代路径。

◆ 如果接收次优BPDU的端口已经处于阻塞状态，那么交换机上的根端口和其他阻塞端口将成为到达根交换机的替代路径。

◆ 如果交换机在根端口上接收次优BPDU，那么当前所有阻塞端口都将成为到达根交换机的替代路径。如果不存在其他阻塞端口，则接收交换机将认为没有到达根交换机链路。

如果接收次优BPDU交换机有到达根交换机的替代路径，就会用这些替代路径发送RLQ BPDU（root link query BPDU，根链路查询BPDU）。一定时间内能收到根交换机回复的RLQ响应报文，就认为到达根交换机的替代路径处于活跃状态。对于接收到次优BPDU的端口，不用等待Max Age计时器到期，立即跳跃到监听状态，参与重新计算，从而加快收敛速度。

下面举例说明一下具体过程：

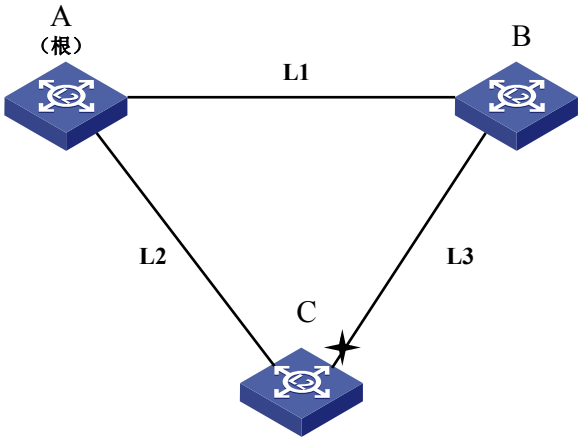


图 2 BackboneFast特性实现过程举例

初始网络如图2所示，交换机A作为根桥，优先级A、B、C依次降低，网络中阻塞链路是L3，交换机C上端口处于阻塞状态。

如果链路L1出现了故障，对于直连的交换机A和交换机B，都会立即检测到链路故障，然而交换机C没有直连L1链路，需要等待Max Age计时器到期，才会将阻塞端口转换为转发状态。

使能BackboneFast功能之后处理过程：

1. 交换机B在检测到链路L1故障之后，把自己选作 根桥，向交换机C发送配置BPDU。

2. 交换机C从B收到次优BPDU的时候，将推断出到根桥A的间接链路已经发生故障，此刻交换机C会向根桥A发送RLQ报文。

3. 交换机A收到之后，回应RLQ响应，并再次将自己列为根桥。

4. 当交换机C收到RLQ回应之后，知道存在到达根桥A的稳定连接，BackboneFast特性会触使交换机C上的阻塞端口立即进入监听状态，无需等待Max Age计时器到期。随后端口由监听状态进一步转换至转发状态，确保交换机B快速加入网络。

BackboneFast特性是PVST模式独有的特性，在Rapid PVST模式中不起作用。为使二者互通不存在问题，在Rapid PVST模式时，收到RLQ请求报文时会默认回传一个RLQ响应报文。

RLQ报文结构如表2所示：

表 2 RLQ报文结构

DMAC（6字节） 01-80-C2-00-00-00(vlan 1) 01-00-0C-CC-CC-CD(其他VLAN)
SMAC（6字节） 端口MAC
Length（2字节） 60
DSAP（1字节） 0xAA
SSAP（1字节） 0xAA
Control（1字节） 0x03
OUI（3字节） 0x00000C
PID（2字节） 0x 0108（请求） 0x 0109（响应）
标准config报文部分（35字节）

RLQ请求报文填充规则：将标准Configuration报文的Bridge Identifier字段值改为本桥的桥ID，其他值同端口优先级向量值。PID字段为0x0108。

RLQ响应报文填充规则（Rapid PVST模式下）：将标准config报文的Bridge Identifier字段值改为收到的RLQ请求报文中的Bridge Identifier字段值，其他值同端口优先级向量值。PID字段为0x0109。

我司设备PVST实现

我司PVST实现

我司PVST实现了Cisco Rapid PVST的功能，并可以很好兼容Cisco PVST协议。实现过程和Cisco Rapid PVST相似，允许以每个VLAN为基础启用或禁用生成树功能，每个VLAN独立运行RSTP状态

机，独立运算。在VLAN 1上生成树根据RSTP BPDU报文进行拓扑运算，其他VLAN上通过带VLAN Tag的PVST报文进行拓扑运算。

不同点在于Cisco Rapid PVST是在每个VLAN上RSTP功能，而我司的PVST是借助MSTP现有的实例和VLAN映射关系模型，通过修改MSTP状态机，将MSTP每个实例映射一个VLAN来实现的，修改后的状态机和RSTP类似。Cisco PVST/Rapid PVST没有实例的概念，而我司的PVST，配置的VLAN会和实例形成一一映射关系，映射方式包括静态和动态两种。

动态映射

实现VLAN与实例的动态映射关系。在设备上使能PVST功能后，创建一个VLAN，该VLAN就会自动映射到一个 非零实例。映射的方式是根据VLAN的创建顺序，按最小的空闲实例映射。非零实例与VLAN是一一对应关系，没创建的VLAN都映射至实例零。

在动态映射下，没法改变VLAN和实例的映射关系。VLAN删除后，动态映射关系自动解除，删除的VLAN映射至实例零。

```
[S5800A]dis stp region-configuration
 Oper configuration
   Format selector      :0
   Region name        :0001
   Revision level       :0
   Configuration digest :0xac36177f50283cd4b83821d8ab26de62

   Instance   Mode      Vlans Mapped
     0       default   2, 4 to 999, 1001 to 4094
     1       dynamic   1
     2       dynamic   1000
     3       dynamic   3
```

静态映射

在域中配置VLAN与实例的静态映射关系，可以将动态映射转化为静态映射，也可以手动配置新VLAN和实例的映射关系。VLAN删除后，静态映射关系不变。

手动配置时，需要注意以下几点：

◆ 已存在的实例x与VLAN y的对应关系，可以手动将动态映射关系改为静态映射，也可以换成新实例与该VLAN映射，替换原有的动态映射；

◆ 如果实例x已经映射到VLAN y，则不允许手动配置该实例静态映射到别的VLAN。

```
[S5800A]stp region-configuration
[S5800A-mst-region]instance 3 vlan 3
[S5800A-mst-region]active region-configuration
[S5800A]dis stp region-configuration
 Oper configuration
   Format selector      :0
   Region name        :0001
   Revision level       :0
   Configuration digest :0x898f576fd7f19cd17cb7e1d9312e6c68

   Instance    Mode      Vlans Mapped
     0       default     2, 4 to 999, 1001 to 4094
     1       dynamic    1
     2       dynamic    1000
     3       static      3
[S5800A]
```

特点

保护处理

支持PVID Inconsistent和Port Type Inconsistent保护，和Cisco PVST的InconsistentPortType–Protected特性实现一样，起保护作用，避免STP会计算混乱，产生环路。

◆ PVID Inconsistent：链路两端PVID不一致，会将链路两端PVID的两个VLAN都阻塞

◆ Port Type Inconsistent：链路两端配置Access口和Trunk/Hybrid口，Access口收到PVST格式报文，会阻塞端口。

发送报文

发送端口增加了Hybrid端口，不同端口类型，PVST发送的BPDU格式：

◆ Access端口，PVST将根据该VLAN的状态发送STP格式的BPDU；

◆ Trunk端口和Hybrid端口，PVST将在VLAN 1内根据该VLAN的状态发送STP格式的BPDU，而对于其他本端口允许通过的VLAN，则发送PVST格式的BPDU。

RLQ报文处理

我司设备在接收PVST报文的端口上默认运行UplinkFast特性，不再增加命令行配置。另外为了支持和Cisco BackboneFast特性互通，收到RLQ请求报文，发送RLQ应答报文，该报文的目的MAC和PVST相同。

互通情况

STP/RSTP与PVST互通

由于STP/RSTP都是单生成树，所有VLAN都映射在实例0上，在VLAN 1内与外界进行BPDU报文交互。在与PVST设备互通时，STP/PVST设备只发送STP/RSTP报文，而PVST设备则在VLAN 1内发送STP报文及不带Tag的PVST报文，其他VLAN内发送带Tag的PVST报文。以两台设备直接交互为例

表 3 STP/RSTP与PVST互通举例

	STP交换机		PVST交换机	
实验 1	非根桥		根桥	
	G1/0/9	G1/0/10	F1/0/9	F1/0/10
VLAN 1	ROOT	ALTE	DESI	DESI
VLAN 2			DESI	DESI
实验 2	根桥		非根桥	
	G1/0/9	G1/0/10	F1/0/9	F1/0/10
VLAN 1	DESI	DESI	ROOT	ALTE
VLAN 2			DESI	BACK

由上面两个实验可以看出，PVST设备和STP设备互通时，交互主要在VLAN 1内完成，其端口角色及状态和STP/RSTP选举相同。在实验 2中可以看到其他VLAN处理情况，STP设备收到带有 VLAN 2 tag的PVST报文，并未处理，直接在VLAN 2内透传，因此Cisco设备的F1/0/10端口角色是BACK。

MSTP与PVST互通

MSTP在域边界端口上以IST的端口状态为准，所有MSTI的端口状态和IST保持一致，只有IST和外界交互报文。MSTP和PVST互通时，MSTP只发送一个报文，PVST每个VLAN都发送报文。考虑如下两种场景：

所有VLAN根都在MSTP域内

MSTP发送802.1D STP报文给PVST设备，PVST设备认为该报文是VLAN 1的报文，将VLAN 1的一个端口计算为Alternate的阻塞端口（如下图的黑色报文和阻塞位置）；PVST设备按照VLAN模式发送PVST报文，该报文在MSTP域内被当作组播数据在相应的VLAN转发，PVST设备收到自己的报文后将端口计算为Backup的阻塞端口（如下图的绿色报文和阻塞位置）。

其特点是：

◆ 可以通过调节PVST参数实现不同VLAN的负载分担。

◆ 链路切换时，Backup端口不能实现快速迁移，依赖与报文老化迁移到转发状态。

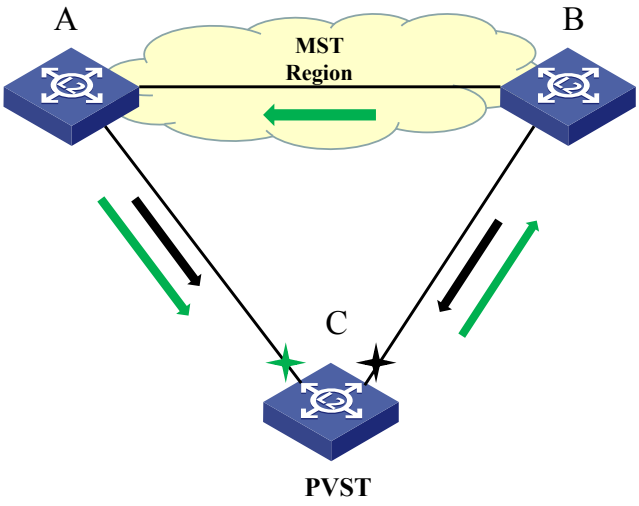


图 3 所有VLAN根都在MSTP域内

所有VLAN的根都在PVST设备上

PVST设备在VLAN 1发送802.1D STP报文，在其他VLAN发送PVST报文。我司MSTP设备只识别802.1D STP报文，认为接收端口为域边界端口，所以将一个端口计算为Alternate的阻塞端口，并且是阻塞所有实例（如下图的黑色报文和阻塞位置）。因该端口阻塞了所有VLAN，导致PVST报文（如下图的绿色报文）在该端口都无法转发，进而PVST设备因收不到自己的PVST报文，而将非VLAN 1的端口放开。

其特点是：

◆ 不能实现负载分担。

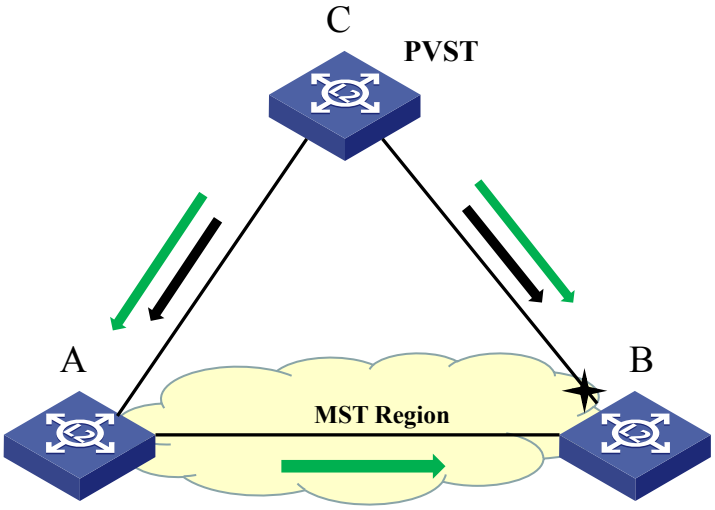


图 4 所有VLAN的根都在PVST设备上

在该互通方案下，因MSTP在域边界端口上要保持所有VLAN状态一致，所以只支持上述的两种配置方式。

MSTP与Rapid PVST互通方式

MSTP与Rapid PVST互通和MSTP与PVST互通类似，只不过在VLAN内运行RSTP机制，加快收敛速度。

结束语

PVST最早是Cisco提出的概念，在消除冗余链路的同时还可以基于不同的VLAN实现负载分担，让链路得以充分利用。从起初不成熟的PVST到得到广泛应用的Rapid PVST，以及我司PVST技术实现，逐步走上了成熟，已经成为HA技术中二层环网的重要协议。

RPR 技术

文/涂勇军

RPR基本原理

相关概念

RPR（Resilient Packet Ring，弹性分组环网）是IEEE 802.17工作组标准化的一种新的MAC层技术，是工作在OSI协议栈第二层的介质访问控制协议，可运行于Ethernet、SONET（Synchronous Optical Network，同步光网络）/SDH（Synchronous Digital Hierarchy，同步数字系列）和DWDM（Dense Wavelength Division Multiplexing，密级波分复用）之上。RPR技术融合了SDH故障自愈的高可靠性与以太网的经济性、高带宽、灵活性、可扩展能力等优势，RPR基于环行拓扑提供数据优化的带宽管理、高性能多业务传输解决方案。

RPR采用逆向双环结构，分为内外两个环，外环称为0环，数据在其上沿顺时针发送，内环称为1环，数据在其上沿逆时针发送。1环发出方向为西，0环发出方向为东，物理接口绑定为0环则表示接收数据在0环，即为西向端口。如下图：

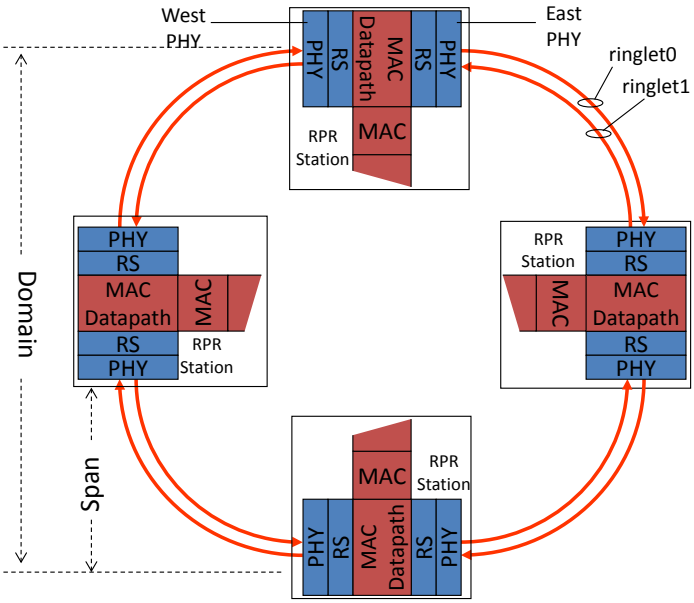


图 1 RPR环网结构示意图

几个概念：

– 0环：RPR双环中，数据帧发送方向为顺时针的称为0环，也称Outer Ring（外环）；

– 1环：RPR双环中，数据帧发送方向为逆时针的称为1环，也称Inner Ring（内环）；

– 节点（Station）：RPR环网上的设备，负责接收和转发数据帧；

– 段（Span）：RPR环网上两个相邻节点之间的链路，由方向相反的两条链路组成；

– 域（Domain）：多个连续的段和这些段上的节点构成了域；

– 西向端口：在0环上接收数据帧、在1环上发送数据帧的物理端口；

– 东向端口：在0环上发送数据帧、在1环上接收数据帧的物理端口；

– 边（Edge）：当段或和段相邻的节点出现故障时，段不能转发数据就成为边；

– 环状态：分为闭环和开环。不存在边的环为闭环，存在边的环为开环。

每个RPR节点采用了一个以太网中的48位MAC地址标识，从设备链路层来看，RPR两个节点之间的物理接口只是一个链路层接口，从网络层来看，也仅需要分配一个IP地址而已。

MAC实体结构

MAC实体是RPR子卡中最为关键的一部分，一方面与上层进行控制和数据的交换，同时需要与各种物理接口协同工作。MAC实体包含一个控制子层和两个MAC数据通道子层，MAC控制实体从两个数据通道中接收、发送数据帧，并通过MAC业务接口与MAC客户端进行控制和数据的交换。如下图：

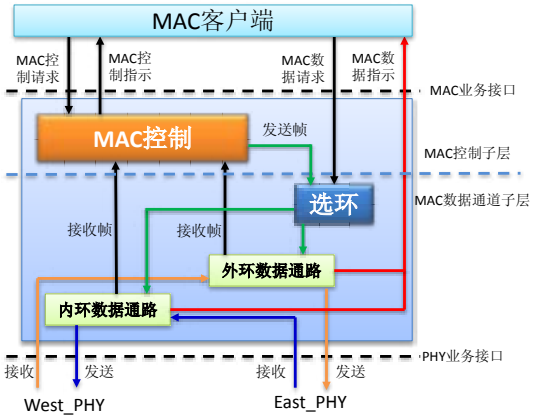


图 2 站点MAC结构

MAC控制子层：控制活动用来维护MAC状态和数据活动，控制实体通过数据通道子层和其他控制实体交互，控制子层的动作主要有：控制服务接口、公平算法和协议处理、保护数据库、拓扑数据库、OAM功能、控制帧的收发。

MAC数据通道子层：包括选环实体和两个不同的环路通道。选环实体利用Dmac、Smac和拓扑数据库来决定哪个环传输报文、如何泛洪、使用基本帧还是扩展帧；各通道主要实现功能有：封装/解封装客户端数据帧、收发帧、流量整形、队列处理、错误帧\超时帧剥离等。

基于这种结构，多个节点可以相互连接起来形成一个完整的端到端的MAC业务处理过程。

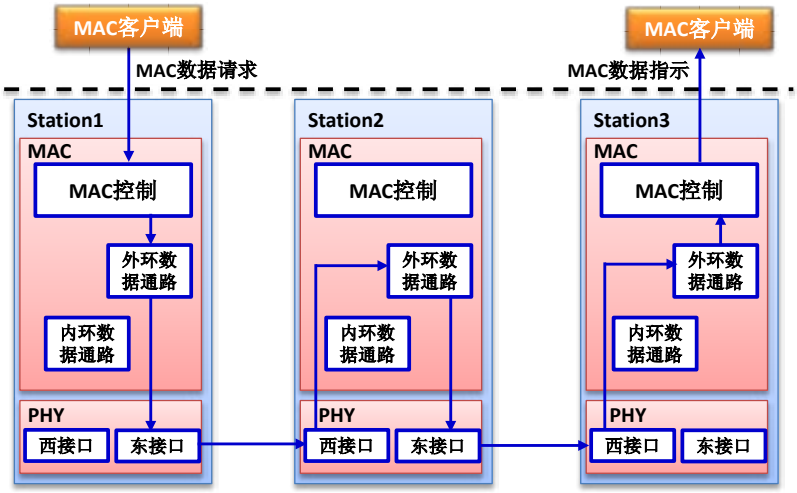


图 3 数据流示意图

如上图，假设流量从station1进入，经过station2从station3出，其数据流程图所示，可以看出，MAC控制实体与MAC客户端交互的时候才起作用，对中间节点而言，MAC实体基本不用处理。对单播数据流来说，仅源\目的节点需要启用MAC控制实体；正常情况下，对同一个数据流，各节点启用相同的数据通道来进行连接（都外环或都内环），从而更好地保证业务连贯性。

RPR数据操作

在RPR环网中，节点与环配合完成数据操作，主要有如下4种操作：

◆ 上环（Insert）：节点设备将来自RPR环网外的数据帧插入到RPR环的数据流中

◆ 过环（Transit）：节点将经过本节点的数据帧继续转发到下一个节点

◆ 下环（Copy）：节点从RPR环网的数据流中将数据帧复制一份并交给本节点的上层进行处理，该操作不会终止数据帧在RPR环网上的转发（环上数据还在）

◆ 剥离（Strip）：使经过本节点的数据不再往下转发，将其从RPR环网上剥离下来，即数据从环上彻底消失

其中最值得关注的是过环，过环操作是不需要设备上层处理的，可以大大提高设备转发的处理性能，也是RPR最大的特性。

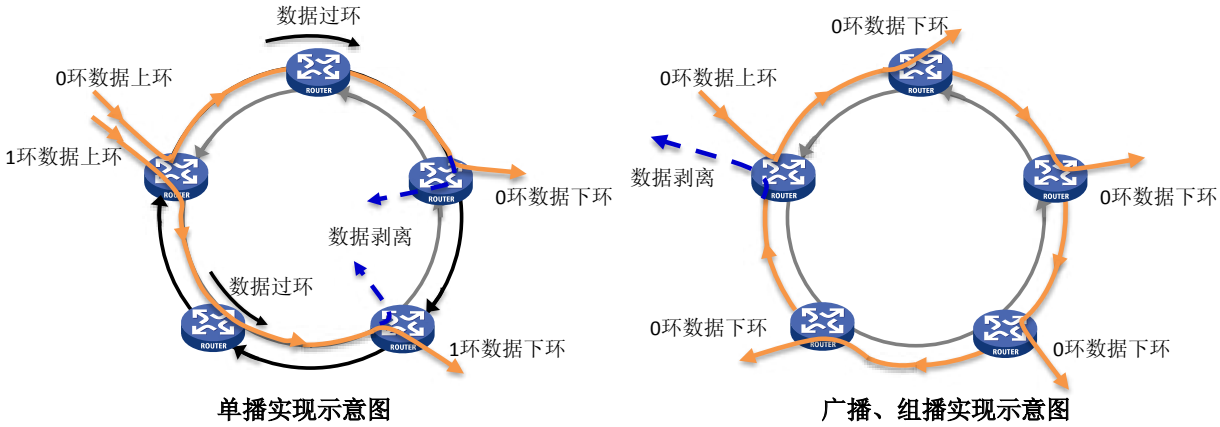


图 4 各类型数据操作示意图

不同于传统环网技术的源节点剥离，RPR单播采用的是目的节点剥离方式，在RPR数据的传输过程中，单播在源节点处执行上环操作，使数据承载到外环或内环中，在目的节点或当数据帧的TTL值为0时执行数据下环和数据剥离操作，而中间节点只执行过环操作。目的节点剥离能够有效提高带宽的利用率，使得带宽的空间重用技术更高效。

对于组播、广播、未知单播报文实行源节点剥离，数据上环后要过环所有RPR节点，只要TTL不为0，就对其执行数据过环和下环操作，当该报文回到源节点或TTL为0时，执行源地址剥离。

报文类型

数据报文（Data Frame）

数据帧是将来自RPR环网外的数据进行格式的统一，RPR数据帧是将外部数据直接放到payload字段中，再加上RPR报文头的封装，形成统一的帧格式。协议中规定数据帧包含两种帧格式：基本帧和扩展帧（报文具体格式可参看802.17_2004）。

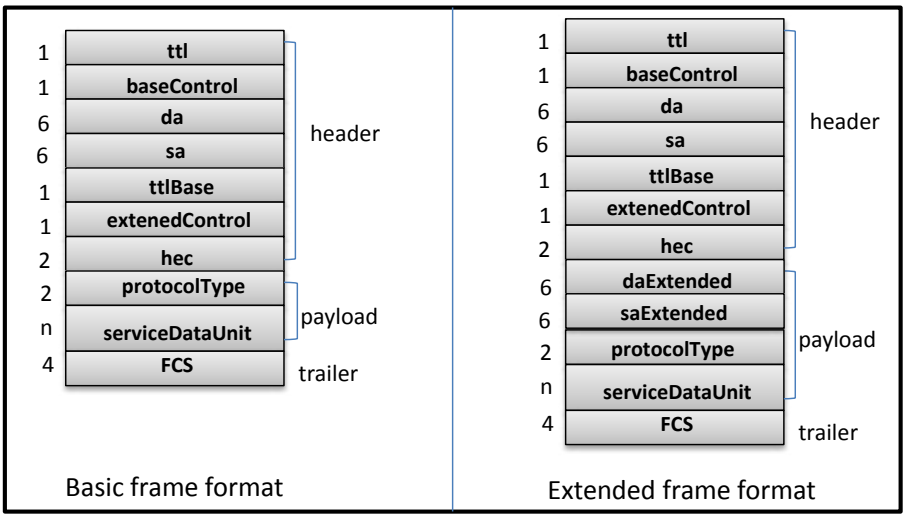


图 5 数据帧格式

扩展帧和基本帧相比，主要是增加了daExtended和saExtended字段，内容是目的MAC和源MAC，这是因为三层单播报文在RPR环上是目的节点剥离，RPR节点在环上的唯一标识就是设备的虚接口mac地址，而三层单播报文的源MAC和目的MAC都是虚接口MAC地址，即都是环上节点mac地址，可以采用基本帧。对于组播和广播报文，目的MAC都是自己的MAC，不是环上节点的MAC，在环上

采用源节点剥离；对于二层报文，目的MAC也是自己的MAC，也采用源节点剥离。为了正常转发这些报文，组播、广播、二层报文，必须采用扩展RPR帧封装，把整个以太帧当作净荷放在自己的报文头后面，这样一来，daExtended和saExtended字段就是原报文的Dmac和Smac字段。对于RPR帧中的Sa，它是上环节点的MAC，当RPR对报文进行源节点剥离的时候，就是根据该MAC进行判断的；对于RPR帧中的Da，由于该帧要在环上跑一圈，所以不能是环上节点的MAC，目前使用原报文的目的MAC，也就是daExtended。

1.4.2 控制报文（Control Frame）

控制报文是为了对RPR环网拓扑、RPR节点的状态进行统一而由各个节点在环上进行发送的报文。

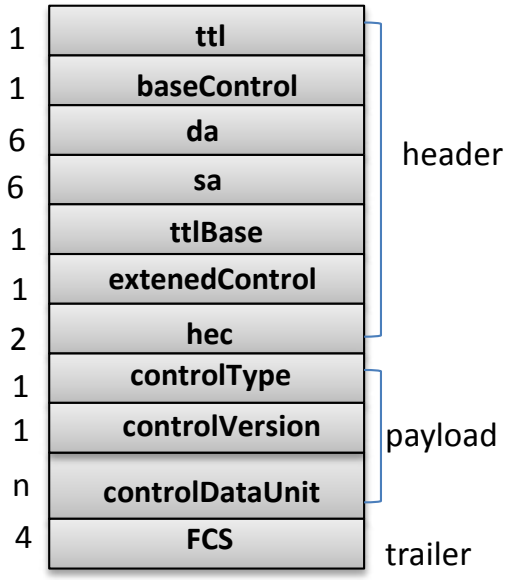


图 6 控制报文帧格式

Value	Name	Description
01_{16}	CT_STATION_ATD	Station attribute discovery frame
02_{16}	CT_TOPO_PROT	Topology and protection protocol frame
03_{16}	CT_TOPO_CHKSUM	Topology and protection checksum frame
04_{16}	CT_LRTT_REQ	Link round trip time measurement request frame
05_{16}	CT_LRTT_RSP	Link round trip time measurement response frame
06_{16}	CT_FDD	Fairness differential delay frame
07_{16}	CT_OAM_ECHO_REQ	OAM echo request frame
08_{16}	CT_OAM_ECHO_RSP	OAM echo response frame
09_{16}	CT_OAM_FLUSH	OAM flush frame
$0A_{16}$	CT_OAM_ORG	OAM organization specific frame
—	—	Reserved

图 7 控制帧类型

控制报文的最小长度为24字节，由header（18Bytes）+payload（2Bytes）+FCS（4Bytes）组成，其controlDataUnit为0Byte；最大长度为1616字节，其controlDataUnit为1500Bytes，另外还预留92Bytes，加上基本的24Bytes，其最大值为1616Bytes。根据控制帧类型，控制报文分为以下几种：TP、TC、ATD、LRTT、FDD、OAM等。

● TP(Topology and Protection)报文

TP对应的控制帧类型为CT_TOPO_PORT，以广播方式来进行拓扑发现，节点保护模式、保护状态等信息的传输。TP报文长度为26Bytes，TTL=255，A0类型，不能Wrap；TP报文的发送分为快周期（1ms~20ms）和慢周期（100ms~1s），当TP信息变化时，先以快周期发送8次，再以慢周期发送。

由于TP报文是广播发送，且为ms级发送，TP报文的流量会很大，如果芯片能在上报TP报文前进行比较，将大大减少TP报文的数量。我司设备采用TP帧序列号来识别不同的TP帧，如果前后序列号相等，则丢弃后收到的TP帧，否则上送CPU。在拓扑变化时会快发TP，快发TP报文的序列号会递增，当拓扑稳定后快发变成慢发此时的TP报文序列号不变（最后一个快发TP报文的序列号）。

● TC(Topology Checksum)报文

TC对应的控制帧类型为CT_TOPO_CHECKSUM，用于校验邻居节点是否和本节点TOP表匹配，用以判断环网是否稳定。TC报文长度为29Bytes，TTL=1，A0类型，不能Wrap；TC报文的发送也分为快慢周期，基本原理和TP一样，当Topo数据库信息变化时，先以快周期发送4个，再以慢周期发送。

LRTT（Loop Round Trip Time）报文

LRTT包括Request和Response两种，Request对应的控制帧类型为CT_LRTT_REQ，Response对应的控制帧类型为CT_LRTT_RSP，用来测量所有节点之间高优先级控制帧的传输时延。当环网拓扑稳定后LRTT时间是固定的，单位为ms。LRTT报文为单播，固定长度为36Bytes，TTL=255，A0类型，不能Wrap。

● ATD（Attribute discovery）报文

ATD报文对应的控制帧类型为CT_STATION_ATD，用来传递节点间的属性信息。和TP报文相比，ATD所携带的信息对时间要求不高，ATD报文发送的频率也不是很高。ATD为广播报文，可以同时广播本节点的多个属性，其报文长度不固定，TTL=255，A0类型，不能Wrap。ATD报文可以封装多种属性，如节点权重、站点配置、带宽预留、节点名称、IP地址、用户自定义ATD帧信息等。

● FDD（Fairness Differential Delay）报文

FDD报文对应的控制帧类型为CT_FDD，FDD用来测量低优先级帧从拥塞结束节点到拥塞起始节点的传输时延。FFD为单播报文，固定长度为30Bytes，TTL = 255，A0/C类型，不能Wrap。测量FDD时，拥塞尾节点向拥塞头节点同时发送A0和C类控制报文（FDD帧是成对发送的，当只收到其中一个时，则FDD帧无效），拥塞头节点信息由Fairness帧来决定，通过比较A0和C类控制报文的时延，就可以测量出低优先级帧从拥塞尾节点到拥塞头节点的传输时间，该信息用于实现Conservative公平算法。

● OAM（Operations、Administration、Maintenance）报文

不同的接口支持的OAM功能是不一样的，基于SONET/SDH的物理层支持的OAM功能更广泛，基于PacketPHY的物理层，OAM支持的级别更低一些；对于RPR来说，主要是在环上传输控制帧来检测各站点的状态，主要完成报警监视、故障定位、故障修复、测试等。

RPR中，OAM帧主要的报文类型有：ECHO、FLUSH和用户自定义类型。

◆ ECHO操作：通过Echo request/response操作来检测某一站点的可达性；Echo操作允许报文在环上的某一节点输入，而响应报文可以在本环或另外一个反向环传输，Echo帧可以指定其服务类型并且帧长可变。

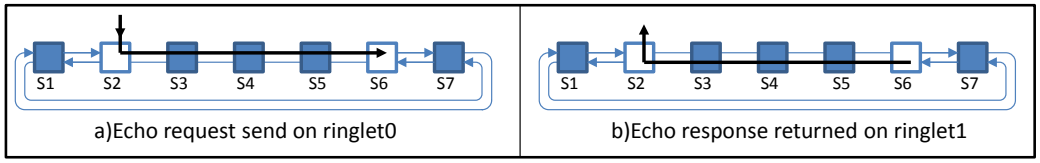


图 8 Echo请求/响应操作

◆ FLUSH操作：Flush是一个特殊的控制帧，节点通过环路发往自身的一个报文，用来测试环路的完整性和连通性。当选环算法发生变化时，通过Flush报文来通告所有节点进行Steer或Wrap保护，从而保证环中先前流量完全传输。Flush帧可以使用任意一种服务类型，用来区分不同动作的结果。

– ClassA：仅在先前流量的主过环队列（PTQ–Primary transit queue）中发送Flush帧，对于仅有单个过环队列的节点，ClassB和ClassC也发送Flush帧；

– ClassB、ClassC：PTQ和STQ（secondary transit queue）都发送Flush帧。

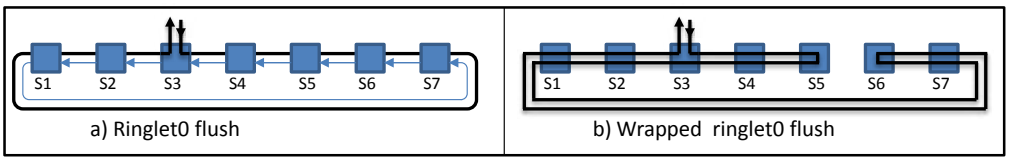


图 9 无保护环和Wrap环中的Flush报文传输

公平报文（Fairness Frame）

公平帧是在环路进行公平算法计算时用的帧，不受公平算法影响，属于A0类报文；其帧格式和数据报文、控制报文稍有不同，帧长度为16Bytes，目的是为了减少带宽消耗和阻塞，提升处理速度；公平帧没有目的地址，通常是发送给最邻近节点或全环广播，从报文格式可以看出，没有目的MAC，saCompact可以是发出此报文的站点MAC或上游站点MAC。

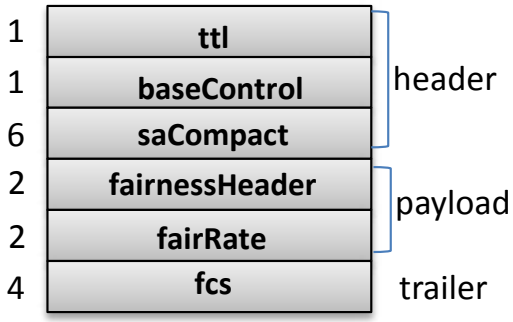


图 10 公平帧格式

空闲报文（Idle Frame）

空闲报文和公平报文相似，不受公平算法影响，属于A0类报文，帧长16Bytes，无目的MAC，帧发送给最邻近节点。空闲帧是可选的，当环网节点之间的报文发送速率有差异时，由空闲帧来调整速率。为了保证每个节点发送相同的流量，发送速率快的节点，插入较多空闲报文；发送速率慢的节点，插入较少空闲报文，从而使整个环网的速率相同。

空闲帧的发送有两种频率：Fast和Slow。在PTQ中设置了IDLE_THRESH门限，当PTQ的深度超过该门限时，按照低频率发送空闲报文；当PTQ的深度低于该门限时，就按照高频率发送空闲报文。

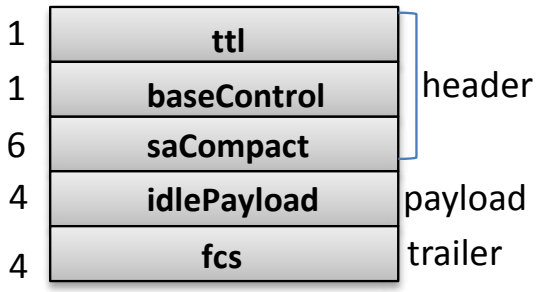


图 11 空闲帧格式

拓扑数据库（Topology Database）

RPR的拓扑发现机制可以使环上的每个节点都能快速、准确的计算出环路的拓扑和环路的变化。所有信息都存放在一个共享的拓扑数据库上，主要包括环路基本信息、站点信息以及TP、TC、LRTT、ATD帧所携带的信息，通过这些报文中参数的变化来进行数据库的增删改并维护状态机。一个站点的数据库包括以下几个部分：

a）环路的基本信息(ringInfo)

从TP帧得到的信息：JumboType、MTU、环路的站点数、环路topo类型（开环或闭环）、在0/1子环上到各站点的（传输\接收）跳数等。

从ATD帧得到的信息：错误帧校验、环路是否有多点阻塞、ringlet0/1可得到的带宽等。

b）本地站点topo信息(myTopoInfo)

从TP帧得到的信息：本地站点Mac地址、站点保护配置（wrap或steer）、东西向的保护状态、东西向最后获取的邻居Mac地址、topo的校验和等。

从TC帧得到的信息：东西向邻居的topo校验和，及其有效标志。

c）各站点得到的TP/TC/LRTT/ATD帧信息

TP帧用来更新各环站点MAC、跳数、边缘状态、保护状态、站点能力等信息；TC用来更新拓扑校验信息；LRTT用来更新LRTT时间信息；ATD用来更新其他属性信息。

报文传输

在1.3节中已经介绍了单播、广播、组播等数据在RPR环上才数据操作，本节主要介绍不同来源流量在RPR环上传输时帧的变化。

● 本地源/本地目的站点单播传输

本地MAC数据请求帧上环后，被封装成RPR帧，帧中增加了RPR特有的帧字段，报文从S1发往S5，当目的MAC和S5的MAC相同的时候，进行剥离或下环操作，当TTL=1且目的MAC不匹配时或者源MAC与S5的MAC相同时，进行报文丢弃。报文字段的变化如下图所示。

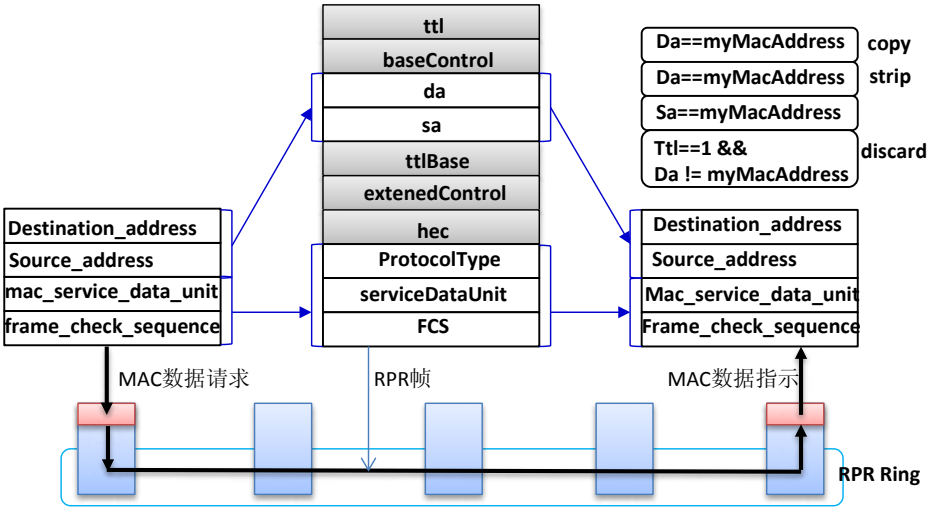


图 12 本地源/本地目的站点单播传输示意图

● 本地源站点广播、组播传输

当站点插入广播或组播报文后，报文将泛洪到所有站点，当TTL超期或者报文回到源站点时进行剥离操作。当RPR环中进行双向泛洪时，TTL的设置必须正确，以保证传输完整且不重复；当RPR环中进行单向泛洪时，TTL需要大于所有邻居个数，以便可以在源站点进行剥离操作；未知单播也以泛洪方式传播，但会将da与本地站点MAC比较，过滤一些不必要的报文转发。如下单向泛洪示意图：

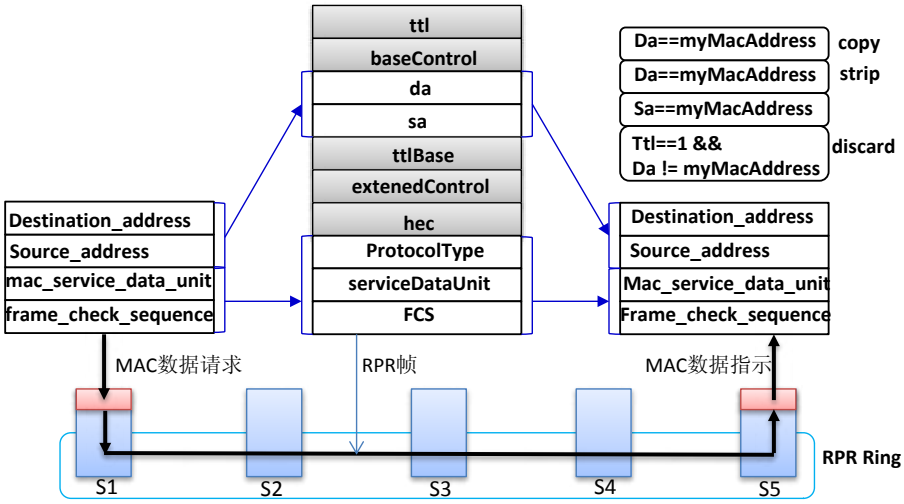


图 13 本地源站点广播、组播示意图

● 远程源站点单播传输

客户端提供了一个远程单播帧从S2上环，报文被RPR封装后在环中泛洪，所有站点将学习到远程站点的地址，虽然泛洪到所有本地站点，但是站点会通过da的比较是否过滤客户端请求帧，当报文回到本地源站点时将被剥离。DestinationAddress和sourceAddress在转发过程中没有被修改，可以用来识别远程源/远程目的站点，并且可以用来进行端到端错误检测。

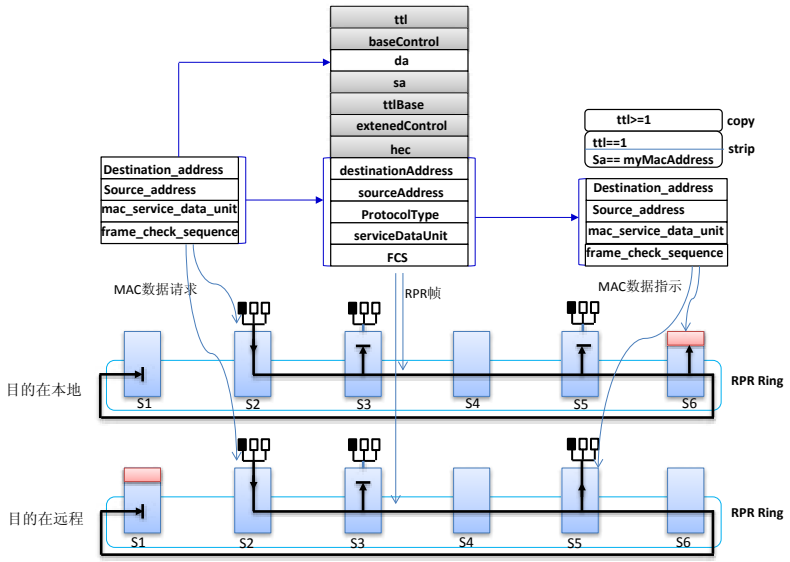


图 14 远程源站点传播示意图

● 远程源站点广播、组播传输

远程源站点广播、组播传输除了报文的da和单播的不一样外，其转发过程是一样的，这里就不再复述。

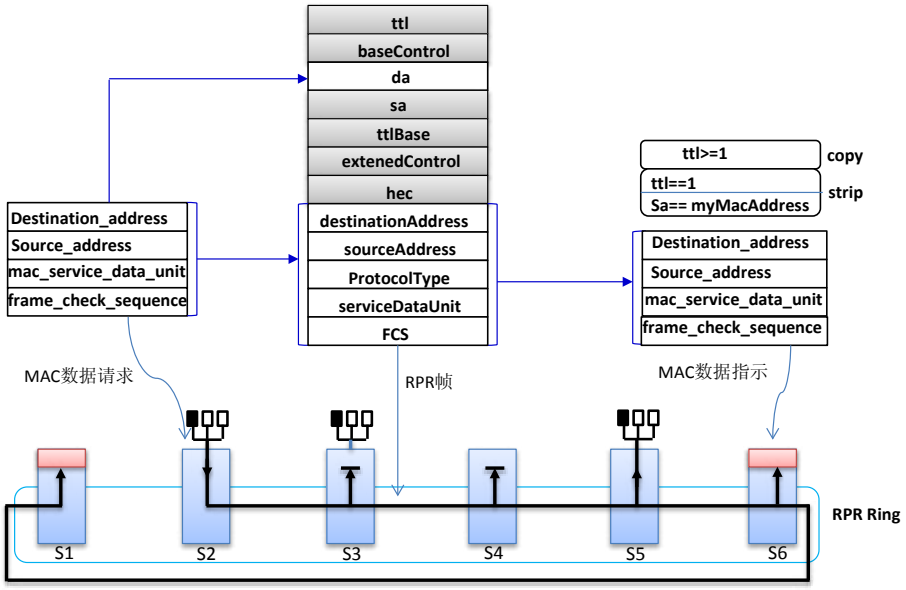


图 15 远程源站点广播、多播传输

故障响应

RPR的双环结构有很强的故障自愈能力，故障响应方式有保护模式和passthrough模式，保护模式包括wrap和steer，能够实现50ms时间内的故障保护切换，passthrough模式是可选的，我司设备没有这个模式。

● assthrough

当节点逻辑故障时（逻辑down、物理up）就会出现passthrough模式，数据仍可以从RPR的物理口传输过去。当站点进入passthrought模式，站点恢复机制能够将相邻站点与故障站点隔离，不会引起SF事件，环路状态保持不变，不会触发保护切换。TP帧是周期性发送的，当一个站点收到一个TP帧时，发现从帧中源Mac地址过来TP帧的TTL值和自己的拓扑数据库的TTL值不相等，则此站点会认为

环路的拓扑发生改变，出现passthrought事件，向两个子环发送自己的TP帧，通告拓扑变化。环路对passthrought事件的收敛时间决定于TP帧的慢周期大小，与50ms切换时间的无关。

● Wrap和Steer模式

当节点不再具有转发数据帧能力时，如掉电、光纤断开等原因造成的故障，节点就需要进入保护倒换方式，即warp或steer。

Steer方式带宽利用率高，但由于需要拓扑重新收敛，可能会造成在响应时间内发送的报文丢失。Steer模式是协议要求必须支持的，Wrap是可选的，只有环上所有节点都支持Wrap倒换模式的时候，环网才使用Wrap倒换模式，Wrap模式数据帧不会丢失，但带宽比较浪费。我司实现中，先Wrap再Steer的方式进行保护倒换。

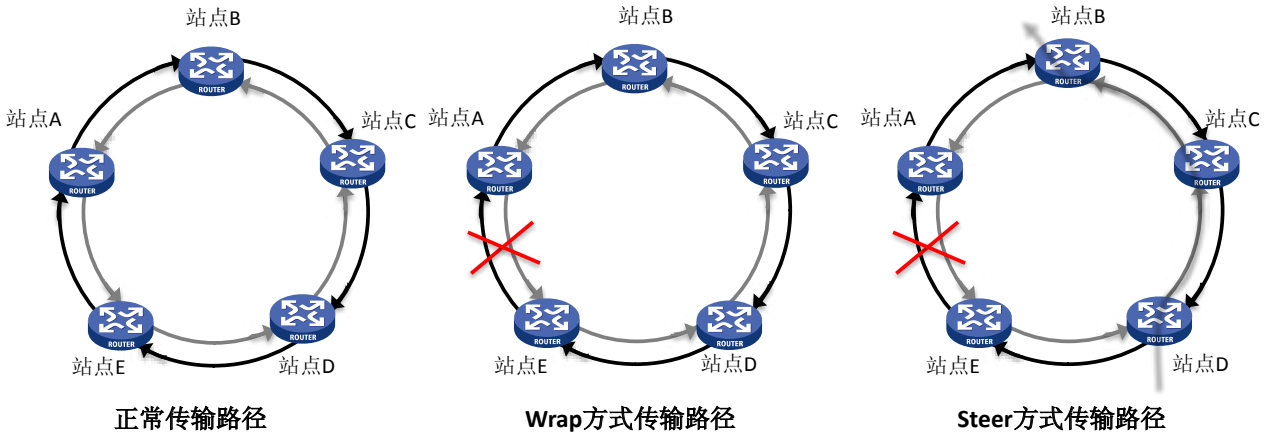


图 16 故障响应处理报文传输路径

如上图，正常传输路径为：D–E–A–B，走0环；当故障发生在A和E之间的时候，wrap和steer保护模式的转发路径为：

Wrap模式：单播报文走0环，但如果报文没到目的节点前遇到了边缘节点，报文从1环走，路径为：D–E–D–C–B–A–B(下环\剔除)；组播报文走0环，但在目的节点B不剔除，回到源节点D才剔除，其路径为：D–E–D–C–B–A–B(下环)–C–D(剔除)。

Steer模式：单播走最短路径，路径为：D–C–B(下环\剔除)；组播报文双向泛洪，路径为：D–E(剔除)，D–C–B(下环)–A(剔除)。

● 保护倒换

当接口的协议信息等发生变化时，会导致接口状态发生变化，处理这种状态变化叫保护倒换。保护倒换发生的条件是保护请求，即只有环上节点发出保护请求时，RPR环才会进行保护倒换。保护请求的取值和优先级与保护倒换一致，其中，FS和MS是手工配置的保护请求，SF、SD和WTR是自动保护请求。另外，SD是由驱动平台上报给软件平台的一种保护状态，我司设备当前不支持SD自动保护请求。

RPR将保护倒换包括6个不同优先级事件，FS、SF、SD、IDLE能在环上的不同节点同时出现，MS、WTR只能在环上的一个节点出现，【注】有一种情况比较特殊：MS、WTR可以同时存在于本站点和邻居站点，当MS、WTR的保护状态被抢占后，相应的edge不可恢复，也就是说MS和WTR会被远端的MS、WTR抢占，但不会被同一个span上的邻居的MS、WTR抢占；若多个保护请求同时发生，优先级较高的将被优先处理，从高到低分别是：

FS（Forced Switch）：强制倒换；

SF（Signal Fail）：信号失效，与当前物理状态相关，Los告警；

SD（Signal Degrade）：信号衰减，与当前物理状态相关；

MS（Manual Switch）：手工倒换；

WTR（Wait to Restore）：等待恢复（只有恢复闭环才会有的状态）；

IDLE：空闲。

生成SD、MS、WTR的保护事件是为了能在各种不稳定情况下，能让环路得到稳定的拓扑状态，这样环上的数据与协议报文就能正确传递。若一个低级别的保护事件已在环上出现，此时又触发了一个更高级别的保护事件，则低级别的保护事件消失（FS和SF除外）。例如，当节点发出MS保护请求时，若环上存在优先级更高的保护请求，MS保护请求将会被清除。当FS、SF发生在同一个节点的同一方向上时，由于FS保护请求的优先级比SF高，则先处理FS保护请求，SF的保护请求不能被立即执行，只有在FS保护请求被清除时，SF保护请求才能被处理。

- KeepAlive的检测

KeepAlive是依靠Fairness报文来实现的，当连续一段时间没有收到Fairness报文，节点将上报KeepAlive中断。由于Fairness报文是定时发送的，如果上报KeepAlive，说明物理链路已经断了，可视为SF进行处理。

- Miscabling的检测

Miscabling用于内外环光纤插反时的错误检测，是依靠TP报文来实现的，如果接收到的TTL=255且ri和本节点不一致的TP报文，则进入Miscabling状态，节点将不再进行协议处理；如果再接收到TTL=255且ri和本节点一致的TP报文时，则退出Miscabling状态。

公平算法

RPR采用共享带宽方式实现各节点对带宽资源的利用，当数据流量较小的情况下，RPR可以满足所有节点流量传输需求；但当流量较大时，往往会出现链路过载、流量拥塞的情况，流量对链路带宽占用需求不能得到完全满足。为了保证在拥塞或过载等情况下各节点能够公平地享用带宽，RPR为此提供了专门的公平算法，实现带宽的公平共享和调度。

基本原理

当某个节点发生拥塞，它将通过向传送数据相反方向的子环发送拥塞公告（Advertise），告知一个Advertise速率，上游节点利用Advertise速率，利用反压信号通知Host来调整其允许上环的速率，使得不超过拥塞节点Advertise速率，如果该节点也发生了拥塞，就以同样方式向其上游节点发送Advertise。

根据协议规定，要解决环路拥塞，必须满足两个条件：

1）bi / wi <= fairRate，其中fairRate公平帧传递的参数，bi为收到公平帧的站点发送数据的速率，wi为此站点在环路上的带宽权重；

2）∑bi<= LINK_RATE*(1 – fa)，其中表达式左边为所有站点的上环速度之和，LINK_RATE为整个带宽大小，fa为站点预留的带宽百分比。

拥塞域

每一段拥塞域都包括拥塞起始节点（Head）和拥塞结束节点（Tail），在每一段拥塞域内，有不同的fairRate，每个节点都遵循bi / wi≤fairRate的规则。

拥塞尾节点是能够收到non–full通告但是不继续往上发送通告的站点，拥塞节点只要满足两个条件之一，就能成为尾节点：1）上游站点收到下游站点发出公平帧的fairRate和自己发出公平帧的fairRate

不相同 2）上游站点向上发出公平帧的fairRate等于FULL_RATE。在评估站点发送流量时，需要考虑站点本身和拥塞点的情况，保证流量不超过允许的带宽。拥塞的头节点就是下游发生拥塞的第一个节点。

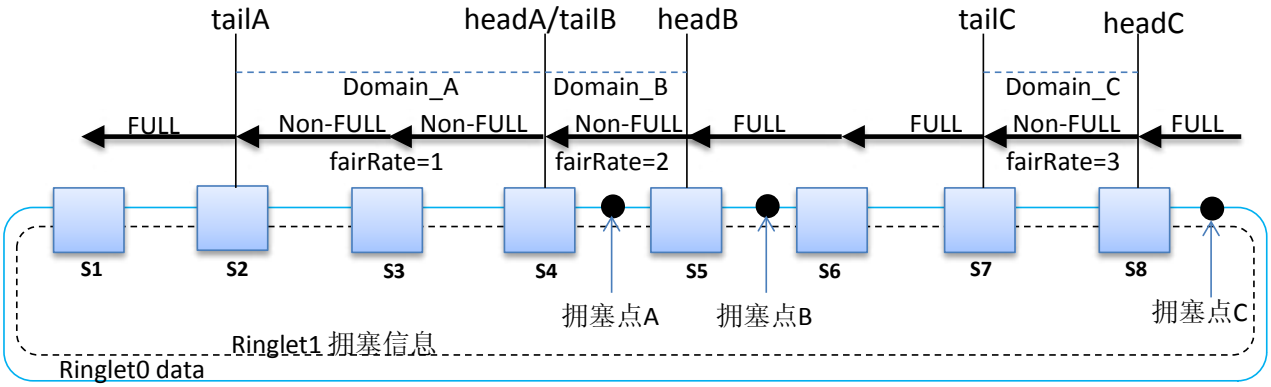


图 17 拥塞域

公平算法特点

◆ 各环独立支持带宽公平操作，互不干扰；

◆ 发生拥塞时，控制信息需要利用另一个子环传输公平帧，帧信息包括允许速率，建议速率，策略指示等；

◆ 规范的只是ClassC和ClassB–EIR流量，上文中fa表示的是ClassA和ClassB–CIR所占百分比；

◆ 根据不同服务级别对环路带宽提供访问控制，采用等值或加权的公平算法来控制环路总带宽的使用；

◆ 每个节点可以根据服务等级、环路带宽的使用情况控制向环上转发数据包的速度，保证每个节点都获得公平的环路带宽分配；

◆ 公平算法包括流量统计和策略处理以及处理中的多个阶段，需要确定三个参数：节点拥塞的门限、节点向上游节点广播的速率、节点本地能允许上环的速率；

◆ 可配置权重，当分配的带宽比实际使用的带宽要大时，会释放自己多余的带宽。

参考文献

[1] IEEE Std 802.17TM–2004

[2] IEEE Std 802.17aTM–2004

[3] IEEE Std 802.17bTM–2007

[4] IEEE Std 802.17cTM–2010

[5] http://press、http://tech相关资料

HDLC 捆绑

文 / 李冰

HDLC概述

综述

HDLC（High-level Data Link Control，高级数据链路控制）是一种面向比特的链路层协议，其最大特点是对任何一种比特流，均可以实现透明的传输。在HDLC中，链路的监控功能是通过一定的比特组合所表示的命令和响应来实现的，这些命令和响应可以与信息报文一起传送。

HDLC是一套完善的数据链路控制体系，采用它的子集衍生出来了很多种链路层协议，其中包括：X.25协议的链路层所使用的LAPB协议，ISDN所采用的LAPD协议，以及现在比较常用的PPP协议和FR协议等。

HDLC的特点

HDLC协议作为一种简单高效的数据链路层控制协议，具有如下一些特点：

（1）透明传输，对要传输的信息报文比特结构无任何限制，信息报文可以是任意字符码集或比特串，它不会影响链路的监控操作。

（2）可靠性高，在所有的HDLC帧中，都采用了循环冗余校验，并且将信息帧按顺序编号，以防信息报文漏收和重收。

（3）传输效率高，在通信中不必等待对端的响应就可以发送下一帧，可以进行双向同时通信。

在H3C设备上实现的HDLC则具有如下一些特殊限定：

（1）HDLC协议只支持点到点链路，不支持点到多点，也不存在广播网络的概念。

（2）HDLC不支持IP地址协商，不支持认证。协议内部通过发送以0x8F为目的地址的Keepalive报

文来检测链路状态。

（3）HDLC协议只能封装在同步链路上。目前在H3C设备的接口类型中只有工作在同步模式下的Serial接口和POS接口可以支持HDLC封装的链路层。

HDLC帧类型和帧结构

HDLC帧结构概述

HDLC作为一个链路层协议，将上层协议送来的报文按一定规则封装起来，形成了HDLC结构的链路帧。

HDLC协议中包含三种类型的帧：信息帧（I帧）、监控帧（S帧）以及无编号帧（U帧）；

◆ 信息帧用于传送有效信息或数据，通常简称为I帧。

◆ 监控帧用于差错控制和流量控制，通常称为S帧；S帧有RR（接收准备好）、RNR（接收未准备好）、REJ（拒绝）和SREJ（选择拒绝）四种类型的命令响应。

◆ 无编号帧用于提供对链路的建立、拆除以及多种控制功能，简称U帧。

HDLC的通用帧结构如下：

图 1 HDLC帧结构

◆ 标志字段Flag：标志字段为固定的值01111110即0x7E，标志一个HDLC帧的开始和结束，所有的帧必须以7E开头，并以7E结束；在邻近两帧之间的7E，既作为前面帧的结束，又作为后续帧的开头；Flag也可作为同步字符和帧间填充字符；不包括Flag字段在内，一个HDLC帧至少应为32bit，若少于32bit则认为该帧无效。

◆ “0比特插入法”：由于01111110比特序列是用于Flag字段，那么就应当避免Address、Control、Information以及FCS字段中出现01111110这样的序列，于是采用0比特填充法，在发送端对帧的内容进行控制，如发现连续5个“1”比特出现，就插入一个“0”比特，在接收端检查帧的内容，如发现连续5个“1”比特，就将其后的一个“0”比特删去，从而恢复了原始帧的内容。

◆ 地址字段Address：Address字段长度为8bit，用于标识接收或发送HDLC帧的地址；在H3C设备实现的HDLC中，地址字段可能被填充为如下一些数值：

通用的地址：维持HDLC链路UP的Keepalive报文为0x8F，若承载的是ISIS报文则Address也是0x8F，其他各类数据报文均为0x0F；

STAC压缩后的报文Address为0x2F或0xAF。

下图所示分别为Keepalive报文和ICMP报文，可看到所谓协议报文的Keepalive其目的地址为0x8F，而作为数据报文的ICMP其目的地址为0x0F；

```
*May 23 19:05:51:212 2000 SR6602-2 DRVDBG/7/debugging:
(Serial1/0)PHY/PKT:
   Packet Input, Packet Len = 22, Partial data as follows:
   8F 00 80 35 00 00 00 02 00 00 03 A1 00 03 76 22
   FF FF 00 91 44 55
```

```
*May 25 18:47:44:104 2000 SR6602-2 DRVDBG/7/debugging:
(Serial1/0)PHY/PKT:
   Packet Input, Packet Len = 128, Partial data as follows:
   0F 00 08 00 4F 00 00 7C 4E D1 00 00 FF 01 37 85
   0A 01 01 01 14 01 01 01 07 27 04 00 00 00 00 00
   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

◆ 控制字段Control：Control字段长度为8bit，用来实现HDLC的各种控制信息，并标识是否为数据；Control字段表示命令和响应的种类，并对帧进行编号；由于H3C实现的是HDLC的一个简化版本，未用到HDLC中规定的控制信息，Control字段固定填为0x00，如上面的图中所示。

◆ 信息字段Information：Information字段可以是任意的二进制比特串，长度未作限定，其上限由FCS（帧校验序列）字段或通讯节点的缓冲容量来决定，目前国际上用得较多的是1000–2000字节，而下限可以是0，即无信息字段。

◆ 帧校验字段FCS：FCS字段采用16bit或32bit的CRC，在H3C的HDLC实现中该字段采用16bit的CRC，对两个Flag字段之间的整个帧的内容进行循环冗余校验。

关于H3C设备的HDLC实现

HDLC协议本身是由ITU–T制定的一套完整的数据链路层控制协议，其内容相当复杂，包括多种监控帧、无编号帧，这两种类型的帧中又包含了多种命令、响应报文，从而实现完整的对数据链路层通信的控制。

此外，完整的HDLC协议还支持点到点、点到多点，支持全双工、半双工链路上的通信，包含NRM、ARM、NBM等多种操作方式以及异常状态的报告与恢复机制。

但这种复杂的协议细节对于设备的实现是有一定障碍的，并且现代通信系统中通常也已不需要如此复杂的链路层控制协议。

因此H3C的HDLC实现是参照CHDLC（即Cisco HDLC）的方式，实现了HDLC中的基本功能，以保证在点到点链路上与友商的设备以HDLC的方式互通。该实现简化到仅保留一种控制类的协议报文，即Keepalive帧，以保证点到点的HDLC链路在物理层和链路层上的UP状态；另一种报文就是承载了各种上层协议的Information帧。

HDLC捆绑

由于支持HDLC封装的接口类型（SAE接口、POS接口），其单接口的数据传输速率都不是很高，逐渐已不能适应当代通信网络高吞吐量、高可靠性的要求。在实际组网中为满足增加带宽、提高可靠性的要求，需要将多个封装了HDLC协议的接口捆绑在一起使用，形成一条逻辑上的数据链路，以提高链路的带宽、实现负载分担，以及增加链路的冗余保护。

HDLC捆绑与以太网接口的静态链路聚合有些相似，其基本目的都是增加通信带宽、实现链路备份和数据流的负载分担，甚至包括通过哈希算法将不同的数据流引向不同的物理接口发送，可以说是静态链路聚合在WAN接口上的一种应用。

HDLC捆绑的几个基本概念

1、HDLC捆绑接口：

HDLC捆绑接口是一个逻辑接口。一个HDLC捆绑接口对应一个HDLC捆绑。

2、HDLC捆绑：

一个HDLC捆绑与一个HDLC捆绑接口相对应，是一组HDLC接口的集合。HDLC捆绑是随着HDLC捆绑接口的创建而自动生成的，其编号与HDLC捆绑接口编号相同。

3、成员接口

加入HDLC捆绑后的实际物理接口称为成员接口；总体来讲，成员接口可以分为选中和非选中两种状态、处于选中状态的成员接口才能参与数据报文的接收和发送，而非选中状态的接口则只能收发Keepalive报文，不参与数据报文的转发。

本文中关于HDLC捆绑的内容，均是基于如下的组网进行验证的，

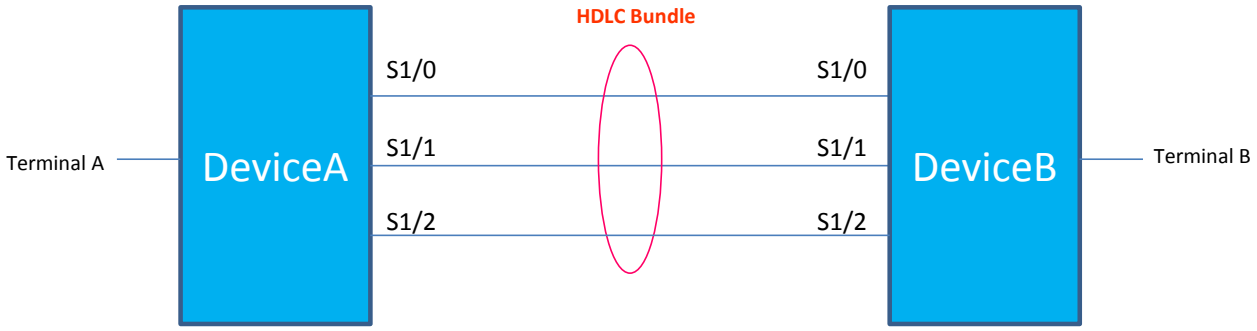


图 2 实验组网拓扑

HDLC捆绑选中接口的确定

HDLC成员接口的状态机

HDLC捆绑中的每个成员接口，无论是从初始状态变为选中状态还是从选中状态变为非选中状态，都遵循一个状态机。HDLC捆绑中的成员接口存在如下几种状态：

◆ 初始化状态Initial：成员接口的链路层协议处于down状态。

◆ 协商状态Negotiated：成员接口的链路层协议处于up状态，但是成员接口不满足选中条件。

◆ 就绪状态Ready：成员接口的链路层协议处于up状态，且成员接口满足选中条件，但由于最多选中成员接口数目/最少选中成员接口数目/最小激活带宽的限制，使得该成员接口没有被选中，那么该成员接口将处于就绪状态。

◆ 选中状态Selected：成员接口的链路层协议处于up状态，且成员接口满足选中条件，处于选中状态。只有处于此状态的成员接口才能转发流量。

表 1 HDLC成员接口状态表

	初始化	协商	就绪	选中
链路可用	协商			
具备选中条件		就绪		
不具备选中条件			协商	协商
被选中			选中	
去选中				就绪
链路不可用		初始化	初始化	初始化

上面的表格描述了成员接口状态的迁移关系，基于这种状态机，当某一成员接口处于协商状态时，若系统判断其具备选中条件时，状态即迁移到就绪状态；处于就绪状态时，系统进一步判断其可以被选中，参与HDLC捆绑的数据转发，状态即迁移到选中。

选中接口确定的原则

无论是处于Initial状态的成员接口还是处于Negotiated、Ready状态的成员接口，都可以统称为非选中状态，它们都不能参加流量的转发，只有处于Selected状态的选中接口才能参与数据报文的转发，因此如何从Ready状态进入Selected状态是关键的一步。需要明确的一点是，HDLC选中端口的确定是基于本地参数判断选择的，而非交互协商的结果，因此如果配置不当可能会出现两端选中结果不一致的情况。

影响成员接口成为Selected状态的因素有接口的速率、捆绑优先级、接口索引号以及配置的最大/最小选中接口数和最小激活带宽。

处于Ready状态的成员接口有M个，设备限制Selected状态的接口数目为N；

◆ 当M<=N时，所有的M个成员接口都可以变为Selected状态；

◆ 当M>N时，则按照接口速率（速率越大越好）、捆绑优先级（数值越小越好）、接口索引号（数值越小越好）的顺序决定哪些接口可以变为Selected状态；

下面的实验将按顺序验证上面的原则：

实验一：首要决定因素是接口速率

如图2中所示的三个Serial接口加入HDLC捆绑，配置Serial1/0接口的速率为64Kbps，接口的捆绑优先级为最高的1；

```
<SR6602-1>dis cur int Serial 1/0
#
interface Serial1/0
 link-protocol hdlc
 bundle id 1
 bundle member-priority 1
#
return
```

配置Serial1/1和Serial1/2接口的速率为2.048Mbps，接口的捆绑优先级为最低的65535

```
#
interface Serial1/1
 baudrate 2048000
 link-protocol hdlc
 bundle id 1
 bundle member-priority 65535
#
return
```

由于在HDLC捆绑接口上配置了最大选中接口数目bundle max-active links为2，因此不会出现三个成员接口都为Selected的情况，必有一个接口不能变为Selected，结果如下所示：

```
<SR6602-1>dis bundle member hdlc-bundle
 bundle is Hdlc-bundle1, slot 0
 max-active links:2, min-active links:1
  Selected members: 2, total bandwidth: 4096 kbps
  State: S -- Selected, R -- Ready, N -- Negotiated, I -- Initial
  Member        State        Bandwidth(kbps)    Priority
  S1/1            S            2048               65535
  S1/2            S            2048               65535
  S1/0            R             64                1
```

由上述结果可以看出，Serial1/1和Serial1/2变为了Selected状态，而接口捆绑优先级比较高、并且接口索引号比较小的Serial1/0仍然处于Ready状态，说明了接口的速率比捆绑优先级和接口索引号的重要性要高，它是首先需要被考虑的因素。

实验二：次要的决定因素是捆绑优先级

接下来验证接口的捆绑优先级和接口索引号对于接口成为Selected状态的决定作用。仍然利用上面的验证环境，将三个成员接口的速率都配置为2.048Mbps，从而排除了速率对于Selected接口的影响；将Serial1/0接口的捆绑优先级调整为最低的65535，而将Serial1/1和Serial1/2的捆绑优先级调整为最高的1；依然由于配置了bundle max-active links为2，因此不会出现三个成员接口都为Selected的情况，必有一个接口不能变为Selected，结果如下所示：

```
<SR6602-2>dis bundle member hdlc-bundle
 bundle is Hdlc-bundle1, slot 0
 max-active links:2, min-active links:1
  Selected members: 2, total bandwidth: 4096 kbps
  State: S -- Selected, R -- Ready, N -- Negotiated, I -- Initial
  Member        State        Bandwidth(kbps)    Priority
  S1/1            S            2048               1
  S1/2            S            2048               1
  S1/0            R            2048               65535
```

由上述结果可看出，接口索引号比较小的Serial1/0仍处于Ready状态，而接口索引号较大的Serial1/1等成为Selected接口，说明捆绑优先级比接口索引号的重要性要高。

实验三：当捆绑优先级一致时，接口索引号小的接口优先被选中

在上一个实验的基础上，将Serial1/1接口的捆绑优先级降低为与Serial1/0相同的65535，而Serial1/2的捆绑优先级依然保持1，得出如下的结果：

```
<SR6602-1>dis bundle member hdlc-bundle
 bundle is Hdlc-bundle1, slot 0
 max-active links:2, min-active links:1
  Selected members: 2, total bandwidth: 4096 kbps
  State: S -- Selected, R -- Ready, N -- Negotiated, I -- Initial
  Member        State        Bandwidth(kbps)    Priority
  S1/2            S            2048               1
  S1/0            S            2048               65535
  S1/1            R            2048               65535
```

可以看到，当接口速率、捆绑优先级都一样时，接口索引号就成为了决定接口能否成为Selected状态的条件，索引号越小的接口具有越高的成为Selected状态的优先权。

由上面的几个实验，验证了决定Ready状态的成员接口成为Selected状态的因素，即接口速率->接口捆绑优先级->接口索引号。

除了上面这些决定因素外，可以影响成员接口是否能由Ready转为Selected的因素还包括配置的最大选中接口数目bundle max-active links，最小选中接口数目bundle min-active links，以及最小激活带宽bundle min-active bandwidth，设备上处于Ready的M个接口按照前面所述的原则排好序后，还要满足下面的条件才能成为Selected接口：

◆ 如果M>bundle max-active links，那么按照前面的顺序确定的前bundle max-active links个成员接口变为Selected状态，其余接口保持Ready状态；

◆ 如果M<bundle min-active links，那么所有接口都仍将处于Ready状态；

◆ 如果M条链路的带宽之和仍小于bundle min-active bandwidth的数值，那么也达不到选中条件，所有接口仍保持Ready状态；

HDLC捆绑的负载分担

HDLC捆绑的一个重要目的就是将数据流量在Selected的成员接口之间进行负载分担，从而达到增大带宽和链路冗余备份的功能。

如图2中所示的情况，Terminal A向Terminal B发送数据，经由HDLC捆绑接口进行转发；HDLC捆绑模块会将报文从一条实际的HDLC物理接口（Selected成员接口）发送，并通过哈希算法将属于同一条流的报文引向同一个物理接口，以保证发送的报文不会乱序。

在H3C设备实现中，决定报文从哪一个物理接口发送取决于三个因素：源IP地址、目的IP地址以及当前处于Selected状态的接口数目。

第一步，将源IP地址每8bit为一组，分为4组，将这4组8bit序列进行异或（XOR）运算，得到异或运算后的二进制序列；

$$XOR_{SIP}=SIP_{Addr1}\ xor\ SIP_{Addr2}\ xor\ SIP_{Addr3}\ xor\ SIP_{Addr4}$$

第二步，将目的IP地址也按照源IP地址一样的方式进行处理，得到一个异或运算后的二进制序列；

$$XOR_{DIP}=DIP_{Addr1}\ xor\ DIP_{Addr2}\ xor\ DIP_{Addr3}\ xor\ DIP_{Addr4}$$

第三步，将上两步的运算出来的二进制序列再进行一次异或运算，得出最终的二进制序列，并将其转为十进制数字；

$$XOR_{Final}=XOR_{SIP}\ xor\ XOR_{DIP}$$

$$T=(XOR_{Final})_{10}$$

第四步，确定HDLC桶深；假设Selected接口数目为M，取一个最接近32（但小于32）并且是M的整数倍的数，作为HDLC桶深，记为bucketLength；并建立一个数组，长度为bucketLength，其中循环填充Selected接口的M个编号；

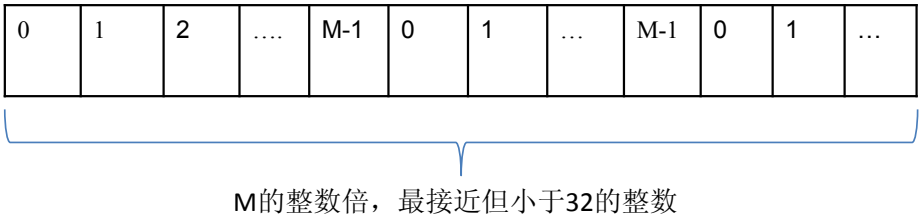


图 3 HDLC桶深示意图

第五步，将第三步得出的T与第四步得出的bucketLength求模，T%bucketLength=Z，在第四步得到的数组中的第Z个元素（从0开始数）所代表的Selected接口即为发送数据的接口。

从HDLC捆绑负载分担的原理来看，它是按照逐流的方式进行的，也就是说只有源或目的IP变化的情况下，流量才能从不同的成员接口进行发送，否则将一直只从一个Selected物理接口进行转发。

HDLC捆绑的收敛效率

HDLC捆绑的一个重要作用就是链路备份，也就是说当捆绑中的一条链路失效时，设备可以将该条链路上的流量转移到其它处于Active状态的链路上去，因此HDLC捆绑发生链路故障时的收敛效率指标是非常重要的。

为了验证该项指标，依然采用如图2所示的组网环境，图中的设备采用SR6602型路由器，HDLC捆绑的三个物理接口均是工作在2.048Mbps速率下的SAE接口（采用V.35电缆直连）。

由Terminal A向Terminal B发送数据流量，该流量初始时从Device A的Serial 1/1接口发出。发送报文长度为64bytes，以2Mbps的速率发送40000个包。

在流量持续发送过程中，使Serial 1/1接口所在链路失效（例如拔掉电缆），在设备将流量切换到其它正常链路上之前，共丢失了4个报文，约合1.34ms；也就是说在链路失效情况下，HDLC捆绑大约可在1.34ms左右的时间收敛，将流量切换到其它正常链路上。

上述测试是在设备链路直连的情况下进行的，而实际中的组网多为穿越传输网的长途线路，若另一端的接口down掉，本端不能马上检测到，这时候就要通过Keepalive接收超时从而触发链路的切换。设备可配置的Keepalive最小间隔是1秒，而连续5个时间间隔内收不到对端发送的Keepalive报文的话会使该接口链路层down掉，也就是说在穿越长途传输网的情况下，至少需要5秒钟时间设备才能检测到链路失效，从而将流量切换到正常状态的链路上。

ISIS GR

文 / 张 岩

简介

GR（Graceful Restart，平滑重启）是一种在主备切换或协议重启时保证转发业务不中断的机制。

能够平滑重启的主要原因在于路由器技术基本上都实现了控制平面与业务平面的分离，在路由器平滑重启发生的时候虽然相应的控制平面的内存信息全部被清空了，但是重启之前的业务平面的信息仍然保留了下来，从而能够有效的实现对数据包的转发。

ISIS GR [RFC3847：Restart Signaling for Intermediate System to Intermediate System (IS–IS)] 是指针对ISIS路由协议的平滑重启的实现。基本的工作过程是设备进行协议重启时，能够通知其周边设备，使到该设备的邻居关系和路由在一定时间内保持稳定。在协议重启完毕后，周边设备协助其进行信息同步，其中包括支持GR的相关协议所维护的各种拓扑、路由和会话信息等，在尽量短的时间内恢复到重启前的状态。在协议重启过程中不会产生路由振荡。

ISIS GR 的路由器能力划分

GR Capable路由器

有GR功能的路由器：通常配备双主控，当进行主备切换的时候，能够通告邻居，同时保持自己的转发表，切换后重建路由表。

GR Aware路由器

GR感知路由器：能识别GR路由器的信令，可以配合在GR路由器重启期间保持邻接关系和维持GR路由器的转发表项。GR Aware设备只需要识别GR信令，不需本身具备主备GR Capable的能力。有GR Capable的设备当然也是GR Aware设备。

GR Unaware路由器

不能感知GR的路由器：无法识别GR信令，不能协助GR路由器在主备切换时不间断转发，也不能执行GR。一般来说是系统软件没有GR特性或者GR特性被关闭。

ISIS GR 的路由器角色划分

从GR中完成的任务来看，分为GR Restarter（协议重启的设备）和GR Helper（协助完成协议重启的设备）两个角色。

对于GR Capable的设备而言，可以做GR Restarter和GR Helper的角色。

GR Aware设备通常只做GR Helper的角色。

GR Unaware设备没有能力参与GR的过程。

ISIS GR的两种不同的类型

在ISIS中定义了两种GR过程类型：

Restarting：

主备倒换、保留FIB的过程，为了保证主备倒换时的转发不中断；

Starting：

对于无法保存FIB的设备发生重启后，由于网络中会存在重启前生成的LSP，序列号较之重启后生成的LSP序列号大，会被作为最新的LSP进行SPF计算，导致路由走到Starting设备，由于Starting设备没有保存FIB，会出现短暂的路由黑洞。

Starting类型的GR就是针对不保留FIB的主备倒换或者设备重启的过程而设计的，该过程是ISIS GR中特有，目的是为了减少路由黑洞。

GR机制中使用的3个标志位

RR位：RR位标识用于restart路由器向邻居发送一个信号表明自己正处于restart的状态，而相应的邻居接收到RR位标识的HELLO后，会保持邻接关系不变，刷新邻接关系保持时间，并将开始Restarting/Starting过程的邻居标识为restart mode。

RA位：RA位标识用于restart路由器的邻居向restart路由器发送一个信号表明自己已经收到restart路由器发送的RR位的HELLO报文。

SA位：SA位标识是一个可选项，只用在前面提过的starting类型的GR过程中。用于start路由器向邻居发送一个信号，使邻居在自己的LSP报文中抑制到start路由器邻居内容。其目的是为了避免出现路由黑洞，starting方式不保留FIB，所以需要通知邻居自己的LSP暂时失效，需要被抑制，在这种情况下GR Restarter发送的Hello报文中必须将SA位置1，而GR Helper接收到这种SA位被置1的Hello报文后就不会将发送该Hello报文的GR Restarter放入LSP扩散出去，即所有设备都不会将报文送到GR Restarter上来进行转发，这样就可以有效的避免路由黑洞的问题。

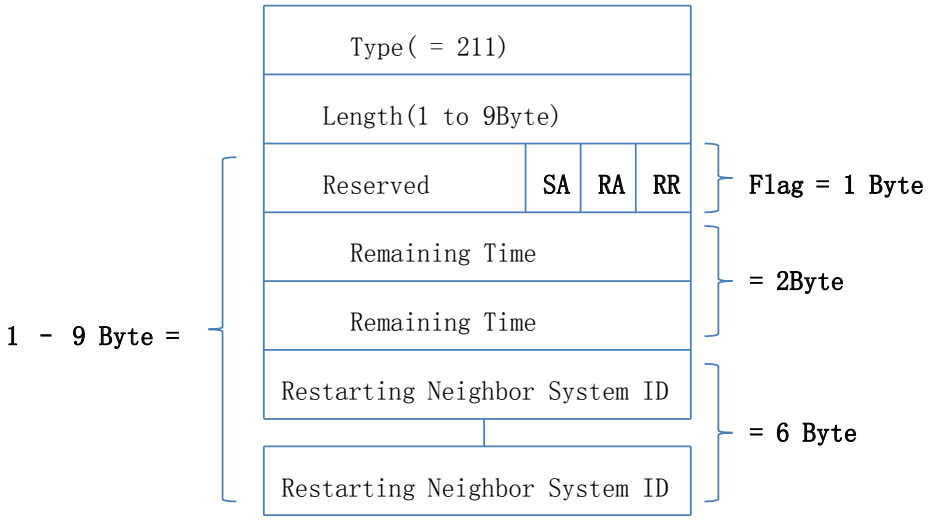


图1 ISIS GR 字段结构图

Type = 211：表示ISIS GR的TLV

Remaining Time ：在RA置位的ISIS HELLO报文中携带邻居关系保持的时间，这个时间也就是邻居能维护GR状态最大的等待时间。

邻居在收到RR置位的ISIS HELLO后，会重置邻居关系保持时间，并将该值减去当前的邻接消耗时间置入Remaining Time；默认的时间是300秒。

Restarting Neighbor System ID：在LAN网络中，RA置位的ISIS HELLO报文中携带，用于对RR置位的ISIS HELLO报文的IS系统进行标识。这是为了在广播网络上多台重启设备之间进行区分，当LAN网络中同时发生多台设备重启时，RA置位的响应报文可以通过此携带此信息明确是对哪个RR置位的ISIS HELLO报文的响应。

GR机制中使用的3个定时器

T1定时器：

为每一个接口所维护。该定时器说明当该定时器时间到后，将重复发送一个未收到应答的restarting或starting信号，一般设置为3秒。

路由器启动的时候只要支持GR则都会在所有UP的接口上启动T1定时器。T1超时且未达到最大定时器次数时则需要重启。当T1超时且达到最大定时器重启次数时，或接收到对方的RA=1的HELLO报文且该接口上接收到所有的对端helper路由器发送的CSNP报文时，才会删除T1定时器；

T2定时器：

T2定时器为每一个LSP数据库所维护。该定时器是系统等待LSP数据库达到同步的最大等待时间，如果同时属于L1和L2，则需要为L1和L2数据库分别维护一个定时器，一般设置为60秒。当所有接口的T1定时器删除后才能删除T2定时器。T2的删除表示此Level的LSP已经同步，进行SPF计算，但不会更新转发

表项。

如果T2定时器超时后LSDB 还没有同步完成，则撤销T2和T3定时器，GR失败。

T3定时器：

T3定时器为每一个ISIS实例所维护。该定时器表明路由器LSP数据库同步失败的时间（通过在自己生成的LSP中设置OL位），这个时间初始化为65535，在接收到含有GR TLV字段[参数为211]且RA位设置为1的HELLO报文后，通过对比当前的T3定时器的时长和HELLO报文中remaining TIME的时间，更新为其中较小的值。

所有的T2定时器被删除时，才能删除T3定时器。T3删除时才会更新转发表项。

当T3定时器超时的时候，LSDB仍然未完成同步，则意味着GR失败。

Restarting的过程

如图2，中间为GR的设备进行主备切换的操作。与左侧设备建立ISIS_LEVEL2的邻居关系，与右侧设备建立ISIS_LEVEL1的邻居。

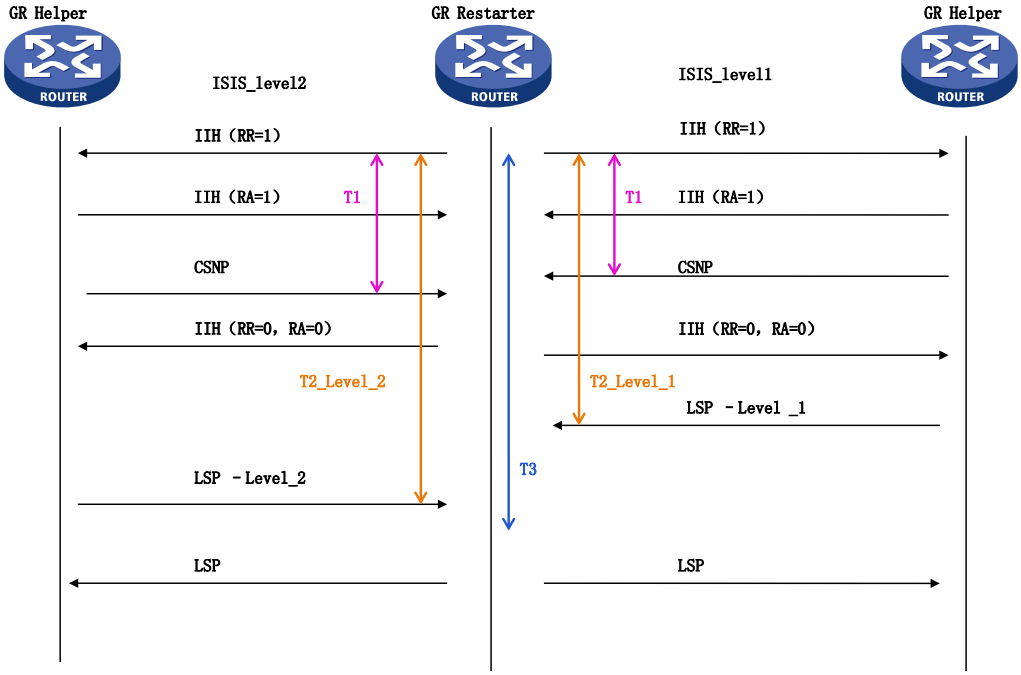


图2 ISIS GR Restarting的过程

当设备进行主备切换时，中间GR的设备首先启动3个定时器。T1、T2、T3，向外发送RR置位的ISIS HELLO报文通知邻居进行GR的restarting的过程。T1取代了发送HELLO报文的时间间隔。

邻居路由器收到RR的HELLO报文后，通过System ID和源MAC（LAN网络）判断和邻居是否为UP，如果是则保持邻接关系不变，刷新holdtime时间。GR_helper响应RA置位的HELLO报文。

对于P2P网络而言，GR helper发送CSNP和所有LSP。对于LAN网络，当前DIS发送CSNP和所有LSP。此DIS是由在LAN内所有支持GR的设备中选出的临时的DIS，仅用于GR的过程，其中选举不会包括GR Restarter设备。ISIS设备在收到LAN环境中携带211 TLV的HELLO报文中，会记录邻居的GR支持能力。

当GR Restarter收到RA IIH和完整的CSNP，则取消T1，否则会周期T1发送RR置位的HELLO报文直至重发最大次数。T1取消后，发送RR和RA清零的HELLO报文。

当LSP有较多分片时，如何判断收到完整的CNSP呢？CSNP中会有StartLspId以及EndLspId两个字段分别记录了该CSNP报文携带的LSP序列的起始ID和结束ID，当LSP都包含在一个CSNP报文中时，StartLspId以及EndLspId分别为0和全F。当LSP过多导致CSNP报文分片时，那么这些分片报文的StartLspId以及EndLspId首尾相连，最终通过判断StartLspId以及EndLspId是否连续以及是否有0和全F决定CSNP报文是否完整。只有接收到完整CSNP报文后才会后续的ISIS GR的过程。

T1取消只是说明收到了完整CSNP，当根据CNSP得到LSP时才能删除对应Level的T2定时器。如图2中从右侧的GR helper获得到了完整的level_1的LSP后，则完成LSDB的同步，取消T2_Level_1。从右侧的GR helper获得到了完整的level_2的LSP后，Level2的LSDB完成同步，取消T2_Level_2。

当所有的T2都被取消后，表示完成了全部的LSDB同步，则取消T3进入标准的ISIS协议流程，重新生成各层的LSP，清除重启前生成的LSP以及更新FIB表，完成GR restarting的过程。

在GR的过程中，可能会出现没能收到RA、没能收到CSNP、或T1超时超过最大重传次数等情况，此时设备会退出GR过程，发送不带GR TLV的普通ISIS HELLO报文，进入普通的ISIS流程。

具体过程可以参考表格1

表 1 ISIS GR__restarting 流程

事件	restarting	收到RA	收到CSNP	等待SPF计算
重启发生GR	启动T1,T2,T3定时器 发送RR置位的ISIS HELLO；			
收到RR	回应RA置位的ISIS HELLO；			
收到RA	根据携带Remaining Time调整T3定时器； 跳转到“收到RA”的状态；		根据携带Remaining Time调整T3定时器； 取消T1；	
收到CSNP	跳转到“收到CSNP”的状态；	取消T1；		
收到不支持GR的HELLO	在点到点的方式下，取消GR过程，发送普通ISIS HELLO			
T1 超时	重启T1； 发送RR置位的ISIS HELLO；	重启T1； 发送RR置位的ISIS HELLO；	重启T1； 发送RR置位的ISIS HELLO；	
T1 超过了最大重试次数	发送不带GR TLV的普通ISIS HELLO；	发送不带GR TLV的普通ISIS HELLO；	发送不带GR TLV的普通ISIS HELLO；	
T2 取消	触发SPF计算； 跳转到“等待SPF计算”的状态；			
T3 超时	为LSP设置OL位； 泛洪LSP update转发表			
完成LSDB的同步	取消T2和T3 触发SPF计算 跳转到“等待SPF计算”的状态；			
全部的SPF计算完成				清除LSP的OL位； 泛洪LSP；完成GR过程

Starting的过程

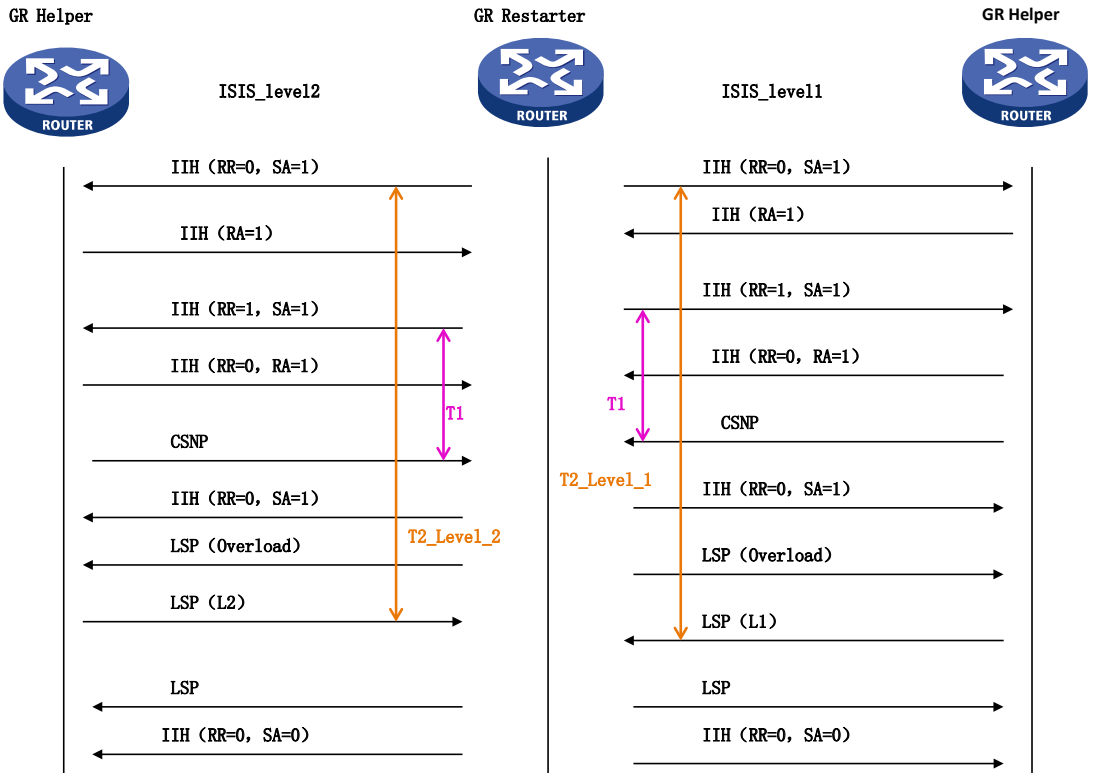


图3 ISIS GR Staring的过程

Starting设备在启动后,ISIS模块完成初始化时，就会启动T2定时器，决定每Level的LSDB同步的时限。Starting的过程不涉及到T3，而只有T1和T2定时器。

系统启动后会主动发送RR清零，SA置位的ISIS Hello 报文。当邻居收到SA置位的报文后，会抑制与GR Restarter的邻居关系。在SA位清零的Hello报文收到之前，邻接关系会一直抑制。 GR helper会响应RA置位的Hello报文。

GR Restarter设备的接口UP后会启动T1定时器，发送RR与SA置位的ISIS Hello报文。RR置位是通知邻居，该接口已经准备好了进行GR的过程，但是该设备还未完成数据库的同步。此过程与restarting的过程类似，希望获取对应CSNP。到T1超时的时候会重发RR与SA置位的ISIS Hello报文，直到发送的次数超过了设置的最大发送次数。对于GR Helper而言，收到RR和SA置位的报文，发送响应报文时Remaining time的设置和Restarting是相同的。

GR Helper会先发送RA置位的ISIS HELLO确认报文，然后会发送CSNP。当GR Restarting设备接收到完整的CNSP后，就会取消T1，并开始发送RR清零但SA仍然置位的ISIS Hello报文.

GR Restarter设备在邻接关系UP，LSDB同步完成之前，会向邻居发送自己的设置OL位的LSP信息，表示自己还未完成数据库的同步，本端设备的LSP暂时不可用。

等到GR Restarter设备获得了相应的LSP完成LSDB同步之后，取消T2定时器并重新生成LSP（OL位均清零）泛洪。之后发送的IIH中RR、RA、SA均清零，完成GR Starting的过程。

具体过程可以参考的表格2

表 2 ISIS GR__starting 流程

事件	Starting	收到RA	收到CSNP
重启发生GR	启动T1,T2定时器 发送SA置位的ISIS HELLO;		
收到RR	回应RA置位的ISIS HELLO;		
收到RA	跳转到“收到RA”的状态;		取消T1;
收到CSNP	跳转到“收到CSNP”的状态;	取消T1;	
收到不支持GR的HELLO	在点到点的方式下，取消T1		
ADJ邻居UP	启动T1; 发送的本地LSP携带OL标记		
T1 超时	重启T1; 发送RR和SA置位的ISIS HELLO;	重启T1; 发送RR和SA置位的ISIS HELLO;	重启T1; 发送RR和SA置位的ISIS HELLO;
T1 超过了最大重试次数	发送SA置位的ISIS HELLO;	发送SA置位的ISIS HELLO;	发送SA置位的ISIS HELLO;
T2 超时	发送的LSP清除OL标记; 发送正常的ISIS HELLO报文; 转入正常运行的状态		
LSDP同步完成	取消T2 发送的LSP清除OL标记; 发送正常的ISIS HELLO报文;		

小结

ISIS通过Starting和Restarting两个不同类型的GR过程，达到了既能协助ISIS路由的快速收敛，又可以避免路由黑洞的产生的目的，增强了ISIS协议的健壮性。

LDP GR

文 / 乔辉

LDP GR概述

GR简介

Graceful Restart简称GR，即常说的平滑重启，主要实现的功能是在协议层重启的时候保证数据层转发的正常，以此来保证关键业务的不中断。

GR技术属于高可靠性（HA，High Availability）技术的一种。HA是一整套综合技术，主要包括冗余容错、链路保证、节点故障修复及流量控制。GR是一种冗余容错技术，目前已被广泛使用在主备切换和系统升级方面。

随着网络设备普遍采用了控制和转发分离的技术，GR技术就成为了可能。为了提高设备的转发性能和可靠性，目前中高端设备中普遍采用多处理器的结构。负责协议层等控制模块的处理器一般位于主控板上，而负责转发的处理器则一般处于线卡板上。这样在发生主备倒换，或协议重启的时候才有可能不影响线卡板上的数据转发。基于这个原因，目前实现GR的设备都需要具备以下特征：双主控结构，并且线卡板具有自己独立的处理器和内存。

LDP GR产生背景

MPLS节点（LSR）由控制平面和转发平面两部分组成：

◆ 控制平面（Control Plane）：负责利用LDP等信令协议分发标签、建立标签交换路径、建立标签转发表等

◆ 转发平面（Forwarding Plane）：依据标签转发表对收到的报文进行转发

在MPLS网络中，某个LSR由于信令协议或者控制平面异常时，将会导致该LSR上所有由信令协议产生的标签转发表项删除，所有经过本LSR的LSP消失，通过LSP转发的业务将会中断；该LSR恢复正常后，数据流量还需要待LSP重新建立后才能恢复。

实际上，利用控制平面和转发平面分离的特点，可以实现在控制平面异常的时候，保证转发平面的正常工作，从而使得数据流量的转发不受影响，提高网络的可靠性。LDP GR正是基于这种思想的一种可靠性机制。LDP GR在信令协议或者控制平面出现异常时，保留标签转发表项，LSR依然根据该表项转发数据，从而使得数据流量的转发不会中断。同时，信令协议或者控制平面恢复后，LSR在邻居的协助下恢复到异常之前的状态。

导致信令协议或者控制平面异常的情况主要有：

◆ 在指定时间内未收到LDP对等体的Hello或Keepalive消息

◆ 承载LDP会话的TCP连接中断

◆ LDP协议异常重启

◆ 主备倒换

部署LDP GR带来的好处

在没有LDP GR的时候，因为各种原因出现的LDP协议重启，都会造成短时间的流量转发中断，甚至会引起MPLS域内的网络震荡，通过图1可以看到问题所在。对于一个大型网络，尤其是运营商网络，网络震荡和业务中断是不可容忍的。使用LDP GR技术则可以解决这些问题。如图2所示，在R1与其他邻居建立邻居关系的时候，就会进行GR能力的协商。这样在R1进行协议重启的时候，它的邻居会维护邻居关系不变，并保持标签映射关系的稳定和数据流量的正常转发。除了R1的邻居，网络中的其它设备并不知道R1进行了协议的重启。

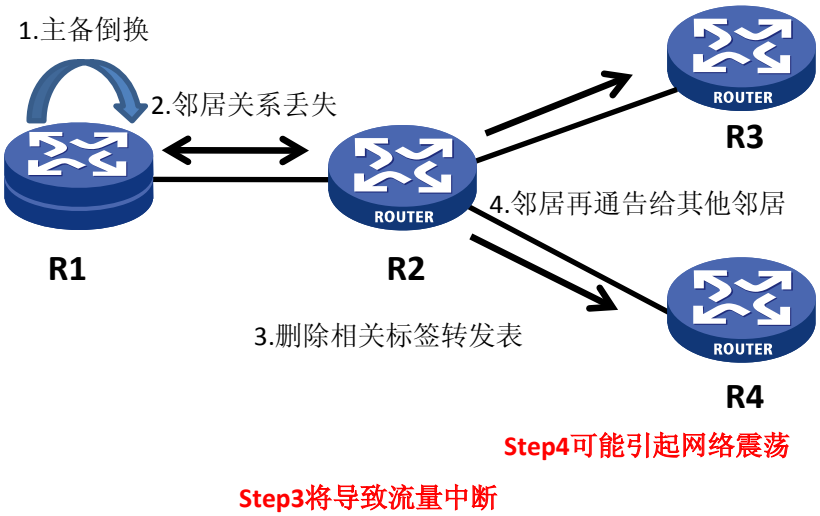


图 1 传统主备切换时造成的网络问题

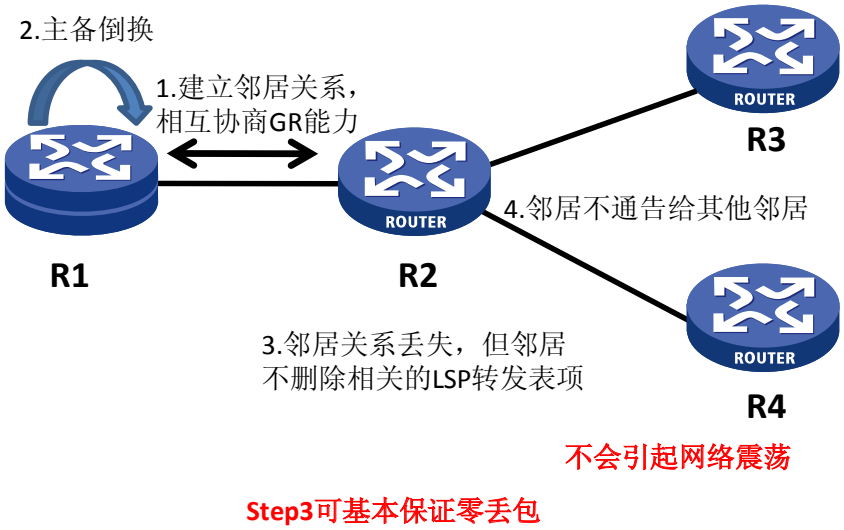


图 2 有LDP GR能力的网络情况

相比较传统LDP协议重启对网络的影响，LDP GR技术具有以下技术优点：

◆ 保证LDP协议重启或主备倒换等过程中转发业务不中断。

◆ 减少单点故障对网络的影响，提高整个网络的可靠性。

◆ 实现简单，对LDP协议进行简单扩展，便可实现LDP GR功能。

◆ 具有良好的兼容性。如果LSR对等体之间有一方不支持LDP GR能力，则具有LDP GR能力的LSR协议重启时，将按照普通的方式重启协议，即删除LDP会话上通告的标签转发表项。

LDP GR技术实现

概念介绍

根据GR过程中的任务划分，参与LDP GR过程的设备分为以下两种：

◆ GR Restarter：GR重启路由器，由管理员手工触发或控制平面异常而重启协议的设备，必须具备GR能力。

◆ GR Helper：GR Restarter的邻居，与重启的GR Restarter保持邻居关系，并协助其恢复重启前的转发状态，也必须具备GR能力。

GR Restarter和GR Helper的角色不是固定的，两者可以互换，具体由该设备在LDP GR过程中的作用决定。

LDP GR工作机制

LDP GR机制的核心在于：当某个LSR（GR Restarter）进行协议重启时，该LSR和它的LDP对等体（GR Helper）在一定时间内保持从对端获取的标签来转发表项。在协议重启完成后，GR Helper协助其在尽量短的时间内恢复重启前的标签映射信息。在协议重启过程中分组转发路径没有任何改变，整个系统可以不间断地转发数据。

LDP扩展

LSR通过在发送的LDP初始化消息（Initialization）中携带FT Session TLV(FT：Fault Tolerance)，并将网络学习标记位（L）置为1，用来标识其LDP GR能力。

0			15				16	31
U	F	FT Session TLV (0x0503)					Length(0x000C)	
R	Reserved		S	A	C	L	Reserved	
FT Reconnect Timeout (ms)								
FT Recovery Time(ms)								

图 3 FT Session TLV格式

FT Session TLV格式如图3所示，其类型值为0x0503，与LDP GR相关的主要字段为：

L：网络学习标识位（Learn from network），取值为1，表示LSR协议重启后，从LDP对等体重新学习标签转发状态。

FT Reconnect Timeout：FT重连时间。FT Session TLV的发送者期望接收者在检测到LDP对等体失效后等待LDP会话重建的时间。

Recovery Time：LDP恢复时间。LDP会话重建后，FT Session TLV的发送者期待接收者在LDP恢复时间内协助其完成标签转发表项的更新。

图4是LDP Initialization报文的格式：

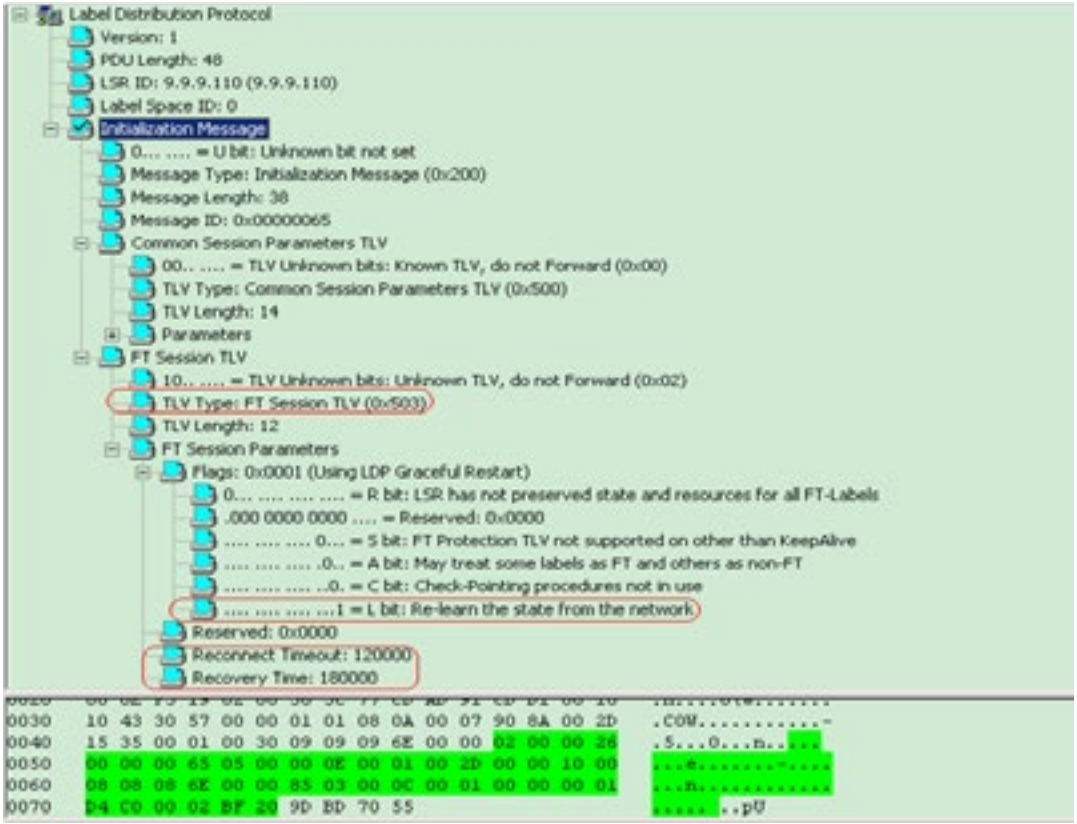


图 4 LDP Initialization报文格式

LDP GR定时器

RFC3478定义了LDP GR的几个定时器：

◆ MPLS Forwarding State Holding Timer：MPLS转发状态保持定时器，GR Restarter协议重启时，保留转发表项的时间。GR Restarter协议重启时，将转发表项置为Stale状态，并启动该定时器。该定时器超时后，删除状态依旧为Stale的转发表项。

◆ Reconnect Timer：重连定时器，GR Helper等待LDP会话重建的时间。GR Helper发现与GR

Restarter之间的LDP会话down后，保留LDP会话上建立的FEC-标签映射，将其置为Stale状态，并启动LDP重连定时器。如果该定时器超时仍未建立LDP会话，则删除状态为Stale的FEC-标签映射及对应的标签转发表项。GR Helper收到FT Session TLV后，将该TLV中的Reconnect Timeout与本地设置的邻居存活时间（Neighbor Liveness Timer：GR Helper上设置的值，用来标识本地设备期望等待LDP会话重建的时间，其实就是本地设置的重连时间）进行比较，选择其中的最小者作为重连定时器的值。

◆ LDP Recovery Timer：LDP恢复定时器。LDP会话重建后，GR Helper协助GR Restarter更新标签转发表项的时间。LDP会话重建后，启动该定时器。该定时器超时后，删除状态依旧为Stale的FEC-标签映射及对应的标签转发表项。LDP恢复定时器的值为GR Restarter发送的FT Session TLV中携带的Recovery Time与GR Helper上设置的Recovery Time的最小值。

LDP GR过程中，几个时间点的对应关系如图5所示：

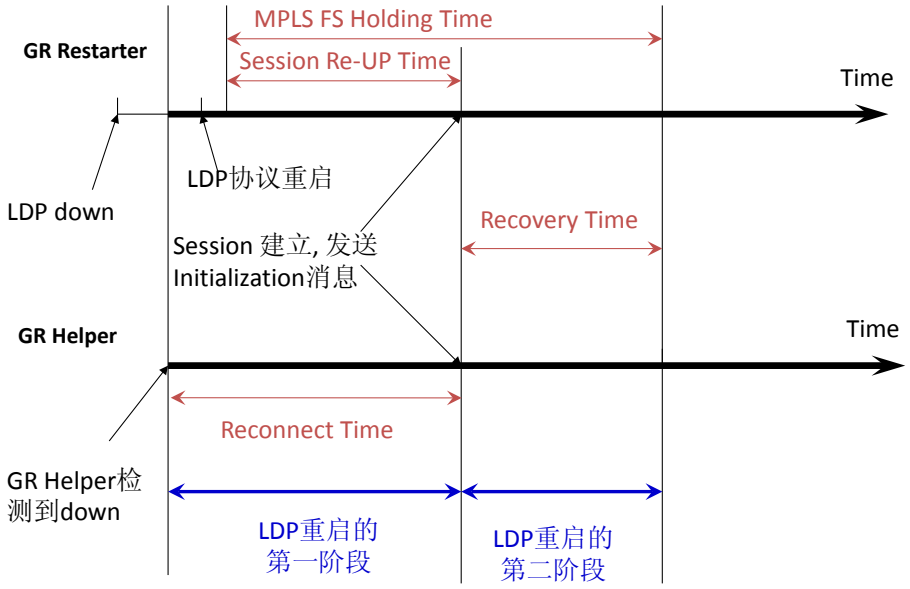


图 5 LDP GR过程中几个时间点的对应关系

LDP GR工作过程

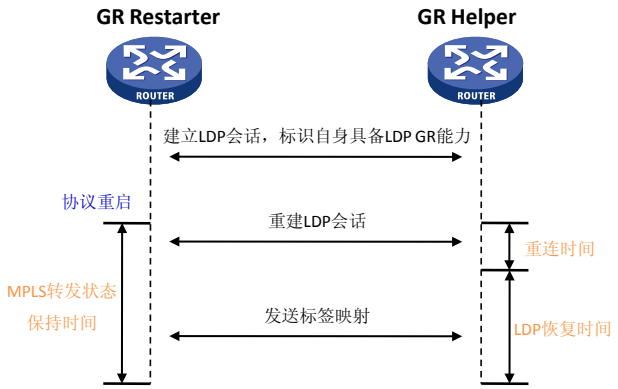


图 6 LDP GR工作过程示意图

如图6所示，LDP GR的过程为：

（1）LSR之间建立LDP邻居，通过LDP的Initialization Message消息中携带FT（Fault Tolerance，容错）Session TLV，且L位置为1，标识它们具有LDP GR能力；

（2）GR Restarter进行协议重启时，启动MPLS转发状态保持定时器，并将标签转发表项置为Stale状态。GR Helper发现与GR Restarter之间的LDP会话down后，将通过该LDP会话接收的FEC—标签映射置为Stale状态，并启动重连定时器；

（3）GR Restarter协议重启后，重新建立与GR Helper的LDP会话。如果在重连定时器超时前，没有建立LDP会话，则GR Helper删除标记为Stale的FEC—标签映射及对应的标签转发表项；

（4）GR Restarter和GR Helper之间重新建立LDP会话后，GR Helper启动LDP恢复定时器；

（5）GR Restarter和GR Helper在新建立的LDP会话上交互标签映射，更新标签转发表。GR Restarter接收到标签映射后，与标签转发表进行比较，如果标签转发表中存在与标签映射一致的Stale表项，则删除该表项的Stale标记；否则，按照正常的LDP处理流程，添加新的标签转发表项。GR Helper接收到标签映射后，与本地保存的FEC－标签映射进行比较，如果存在一致的标签映射，则删除该FEC－标签映射的Stale标记；否则，按照正常的LDP处理流程，添加新的FEC－标签映射及对应的标签转发表项；

（6）MPLS转发状态保持定时器超时后，GR Restarter删除标记为Stale的标签转发表项；

（7）LDP恢复定时器超时后，GR Helper删除标记为Stale的FEC—标签映射。

LDP GR应用限制

◆ LDP对等体必须同时支持LDP GR功能，其中一台LSR协议重启时才能实现平滑重启。

◆ LDP GR通常与IGP路由协议的GR功能（如OSPF GR、ISIS GR等）配合使用，保证主备倒换过程中路由及标签转发路径不受影响。

LDP GR典型组网应用

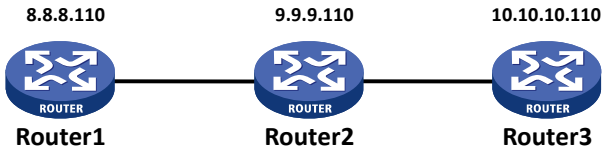


图 7 LDP GR典型组网

如图7所示，在MPLS域内，Router1、Router2和Router3之间建立LDP邻居，且均使能LDP GR功能。当Router2发生主备倒换时，私网数据依然沿着Router1——Router2——Router3的路径正常转发。Router2主备倒换完成后，与Router1和Router3重新建立LDP会话，并在该会话上交互标签映射，以便及时恢复标签转发表。

Router1上的LDP会话信息：

```
[Router1]display mpls ldp peer 9.9.9.110 verbose
Peer LDP ID      : 9.9.9.110:0        Local LDP ID     : 8.8.8.110:0
 TCP Connection    : 9.9.9.110:40578 –> 8.8.8.110:646
 Session State     : Operational        Session Role     : Passive
```

```
 Session Up Time  : 0000:00:03 (DD:HH:MM)
 Max PDU Length   : 4096 bytes (Local: 4096 bytes, Peer: 4096 bytes)
 Keepalive Time   : 45 sec (Local: 45 sec, Peer: 45 sec)
 Keepalive Interval : 15 sec
 Msgs Sent/Rcvd: 18/23            KA Sent/Rcvd   : 14/14
 Label Adv Mode   : DU            Graceful Restart   : On
 Reconnect Time   : 120 sec       Recovery Time   : 180 sec
 Loop Detection   : Off           Path Vector Limit: 0
 Discovery Sources:
   Ten-GigabitEthernet1/5/0/11
     Hello Hold Time: 15 sec      Hello Interval   : 5000 ms
 Addresses received from peer:
   20.20.0.1      9.9.9.110
```

Router2上的LDP会话信息：

```
[Router2]display mpls ldp peer 8.8.8.110  verbose
Peer LDP ID      : 8.8.8.110:0      Local LDP ID     : 9.9.9.110:0
 TCP Connection   : 9.9.9.110:40578 -> 8.8.8.110:646
 Session State    : Operational      Session Role     : Active
 Session Up Time  : 0000:00:02 (DD:HH:MM)
 Max PDU Length   : 4096 bytes (Local: 4096 bytes, Peer: 4096 bytes)
 Keepalive Time   : 45 sec (Local: 45 sec, Peer: 45 sec)
 Keepalive Interval : 15 sec
 Msgs Sent/Rcvd: 20/15            KA Sent/Rcvd   : 11/11
 Label Adv Mode   : DU            Graceful Restart   : On
 Reconnect Time   : 120 sec       Recovery Time   : 180 sec
 Loop Detection   : Off           Path Vector Limit: 0
 Discovery Sources:
   Ten-GigabitEthernet4/4/0/11
     Hello Hold Time: 15 sec      Hello Interval   : 5000 ms
 Addresses received from peer:
   20.20.0.2      8.8.8.110
```

Router2主备倒换时，Router1上的FEC-标签映射信息：

```
[Router1]display mpls ldp fec 9.9.9.110 32
 FEC: 9.9.9.110/32
   In Label: 1136
   Upstream Info:
     Peer: 9.9.9.110:0          State: Established (Stale)
   Downstream Info:
     Peer: 9.9.9.110:0
       Out Label: 3           State: Established (Stale)
       Next Hops: 20.20.0.1          XGE1/5/0/11
```

Router1上的标签转发表项：

[Router1]display mpls ldp lsp 9.9.9.110 32

```
Status Flags: * - stale, L - liberal
Statistics:
  FECs: 1      Ingress LSPs: 1     Transit LSPs: 1     Egress LSPs: 0

FEC              In/Out Label    Nexthop        OutInterface
9.9.9.110/32      -/3(*)         20.20.0.1        XGE1/5/0/11
                  1136/3(*)      20.20.0.1        XGE1/5/0/11
```

LDP GR过程中GR Restarter和GR Helper的处理过程：

```
GR Helper（Router1）：
%Dec 27 09:41:57:241 2012 Router1 LDP/5/LDP_SESSION_CHG: -MDC=1; Session (
9.9.9.110:0, public instance) is down (TCP connection down).
%Dec 27 09:41:57:241 2012 Router1 LDP/5/LDP_SESSION_GR: -MDC=1; Session (
9.9.9.110:0, public instance): Start reconnection.
%Dec 27 09:41:57:919 2012 Router1 LDP/5/LDP_SESSION_CHG: -MDC=1; Session
(9.9.9.110:0, public instance) is up.
%Dec 27 09:41:57:919 2012 Router1 LDP/5/LDP_SESSION_GR: -MDC=1; Session (
9.9.9.110:0, public instance): Start recovery.
%Dec 27 09:44:53:016 2012 Router1 LDP/5/LDP_SESSION_GR: -MDC=1; Session (
9.9.9.110:0, public instance): Recovery completed.        恢复时间约为175s，与下文一致
GR Restarter（Router2）：
%Dec 27 09:41:57:529 2012 Router2 LDP/5/LDP_SESSION_CHG: -MDC=1; Session
(8.8.8.110:0, public instance) is up.
```

此时，Router1端LDP会话的详细信息：

```
[Router1]display mpls ldp peer verbose
 Peer LDP ID      : 9.9.9.110:0        Local LDP ID     : 8.8.8.110:0
 TCP Connection   : 9.9.9.110:40576 -> 8.8.8.110:646
 Session State    : Operational        Session Role     : Passive
 Session Up Time  : 0000:00:04 (DD:HH:MM)
 Max PDU Length   : 4096 bytes (Local: 4096 bytes, Peer: 4096 bytes)
 Keepalive Time   : 45 sec (Local: 45 sec, Peer: 45 sec)
 Keepalive Interval : 15 sec
 Msgs Sent/Rcvd: 23/28              KA Sent/Rcvd     : 19/19
 Label Adv Mode   : DU              Graceful Restart : On
 Reconnect Time   : 120 sec         Recovery Time    : 175 sec
 Loop Detection   : Off             Path Vector Limit: 0
 Discovery Sources:
   Ten-GigabitEthernet1/5/0/11
     Hello Hold Time: 15 sec        Hello Interval   : 5000 ms
 Addresses received from peer:
   20.20.0.1        9.9.9.110
```

在GR Restarter端，通过以下命令可以查看是否进入了GR：

```
[Router2]display mpls summary
Memory State     : Normal
MPLS LSR ID      : 9.9.9.110
Egress Label Type: Implicit-null
Labels:
  Range          Idle
  16-1023        1008
  1024-13311     12282
  65536-69631    4096
  131072-139263  8192
Protocols:
  Type           State
  LDP            Recover
  BGP            Recover
  Static         Normal
```

GR Helper端恢复完成后，查看该状态，变为Normal：

```
[Router2]display mpls summary
Memory State     : Normal
MPLS LSR ID      : 9.9.9.110
Egress Label Type: Implicit-null
Labels:
  Range          Idle
  16-1023        1008
  1024-13311     12282
  65536-69631    4096
  131072-139263  8192
Protocols:
  Type           State
  LDP            Normal
  BGP            Normal
  Static         Normal
```

Router2主备倒换完成后，Router1上的FEC-标签映射信息和标签转发表项：

```
[Router1]display mpls ldp fec 9.9.9.110 32
 FEC: 9.9.9.110/32
   In Label: 1136
   Upstream Info:
     Peer: 9.9.9.110:0          State: Established
   Downstream Info:
     Peer: 9.9.9.110:0
       Out Label: 3             State: Established
       Next Hops: 20.20.0.1             XGE1/5/0/11
```

```
[Router1]display mpls ldp lsp 9.9.9.110 32
Status Flags: * - stale, L - liberal
Statistics:
FECs: 1     Ingress LSPs: 1     Transit LSPs: 1     Egress LSPs: 0

FEC             In/Out Label    Nexthop        OutInterface
9.9.9.110/32        -/3         20.20.0.1        XGE1/5/0/11
                  1136/3        20.20.0.1        XGE1/5/0/11
```

如果Reconnection Timer超时后，仍然未重新建立LDP会话，则提示失败，同时GR Helper删除Stale状态的FEC-标签映射信息：

```
%Dec 26 09:08:44:559 2012 Router1 LDP/5/LDP_SESSION_CHG: -MDC=1; Session
(9.9.9.110:0, public instance) is down (interface not operational).
%Dec 26 09:08:44:559 2012 Router1 LDP/5/LDP_SESSION_GR: -MDC=1; Session (
9.9.9.110:0, public instance): Start reconnection.
%Dec 26 09:10:44:616 2012 Router1 LDP/5/LDP_SESSION_GR: -MDC=1; Session (
9.9.9.110:0, public instance): Reconnection failed
（120s后超时，重连失败，删除FEC-标签映射）
[Router1]display mpls ldp fec 9.9.9.110 32      GR失败，删除该标签映射信息
```

如图7所示，Router1（Router3）和Router2可能会因为Keepalive超时，均进入到GR Helper角色：

```
Router1端:
Dec 26 10:54:18:016 2012 Router1 LDP/5/LDP_SESSION_CHG: -MDC=1; Session
(9.9.9.110:0, public instance) is down (keepalive hold timer expired).
%Dec 26 10:54:18:016 2012 Router1 LDP/5/LDP_SESSION_GR: -MDC=1; Session
(9.9.9.110:0, public instance): Start reconnection.
%Dec 26 10:55:00:976 2012 Router1 LDP/5/LDP_SESSION_CHG: -MDC=1; Session
(9.9.9.110:0, public instance) is up.
%Dec 26 10:55:00:976 2012 Router1 LDP/5/LDP_SESSION_GR: -MDC=1; Session (
9.9.9.110:0, public instance): Start recovery.
%Dec 26 10:58:01:016 2012 Router1 LDP/5/LDP_SESSION_GR: -MDC=1; Session
(9.9.9.110:0, public instance): Recovery completed.
Router2端:
%Dec 26 10:54:47:168 2012 Router2 LDP/5/LDP_SESSION_CHG: -MDC=1; Session
(8.8.8.110:0, public instance) is down (keepalive hold timer expired).
%Dec 26 10:54:47:168 2012 Router2 LDP/5/LDP_SESSION_GR: -MDC=1; Session
(8.8.8.110:0, public instance): Start reconnection.
%Dec 26 10:55:00:070 2012 Router2 LDP/5/LDP_SESSION_CHG: -MDC=1; Session
(8.8.8.110:0, public instance) is up.
%Dec 26 10:55:00:070 2012 Router2 LDP/5/LDP_SESSION_GR: -MDC=1; Session
(8.8.8.110:0, public instance): Start recovery.
%Dec 26 10:58:00:168 2012 Router2 LDP/5/LDP_SESSION_GR: -MDC=1; Session
(8.8.8.110:0, public instance): Recovery completed.
```

LDP GR安全考虑

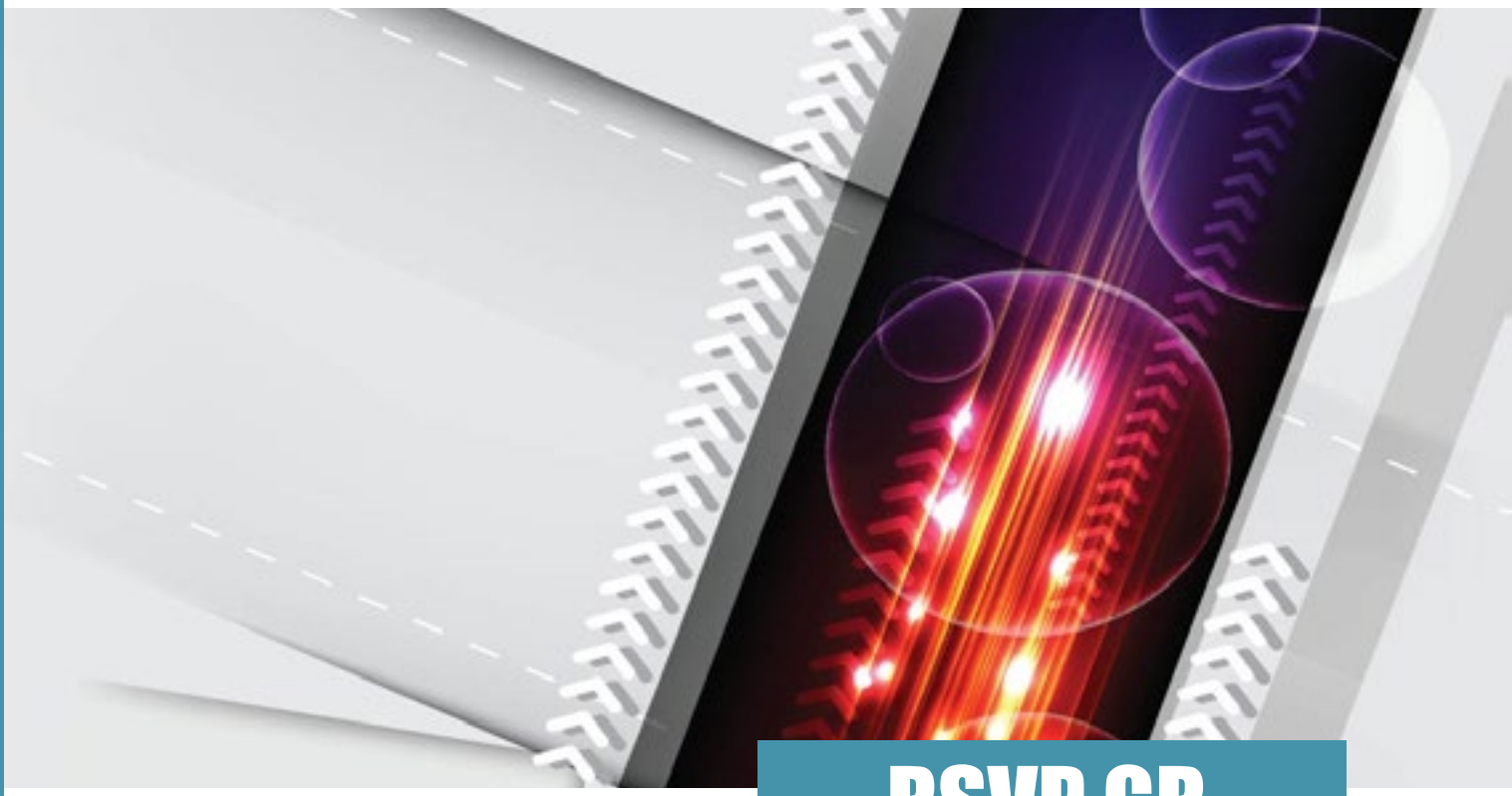
LDP GR面临的安全隐患：

◆ 入侵者通过扮演LSR的LDP邻居，然后强制中断LDP的TCP连接，同时通告LDP恢复时间为0。

◆ 入侵者截取LDP邻居间交互的信息，并将LDP恢复时间篡改为0通告给合法的LSR。

以上两种攻击均使得合法LSR收到的LDP初始化消息中，LDP恢复时间为0，一旦进入GR流程，该LSR将立即删除从邻居收到的标签，从而导致流量转发的中断。

解决这些安全隐患的方法是在LDP邻居间配置MD5认证来提高LDP会话的安全性。



RSVP GR

文 / 苏艳梅

RSVP-TE GR概述

RSVP-TE GR（Graceful Restart，平滑重启）功能是指在信令协议或控制平面出现异常时，保留软状态信息和转发表项信息，以保证数据转发不中断。

在讲解RSVP-TE GR之前，先了解两个基本概念：

GR Restarter：GR重启路由器，指由管理员手工或设备故障触发而重启协议的设备，它必须具备GR能力。

GR Helper：GR Restarter的邻居，与重启的GR Restarter保持邻居关系，并协助其恢复重启前的转发状态，它也必须具备GR能力。

RFC3473对RSVP-TE GR进行了详细描述 。

RSVP-TE GR实现原理

RSVP-TE Hello扩展

RSVP_TE Hello扩展用于在两个直连的RSVP邻居之间维持邻居关系，同时也是检测RSVP邻居关系失效的一种动态机制。如果节点连续发送的Hello消息都得不到邻居的Hello消息确认时，节点则认为RSVP邻居关系失效。

RSVP_TE Hello扩展能力通过RSVP_TE Hello消息实现，Hello消息包括HELLO REQUEST、HELLO ACK 两种对象类型。直连的两个邻居节点之间有HELLO REQUEST、HELLO ACK两种消息类型，HELLO ACK消息是对HELLO REQUEST消息进行确认。一个节点不能同时发送HELLO

REQUEST消息和HELLO ACK消息，只能是发送其中一种消息。如果节点发送HELLO REQUEST消息，对端节点只能发送HELLO ACK消息，来对HELLO REQUEST消息进行确认，否则相反。

Hello消息报文格式：

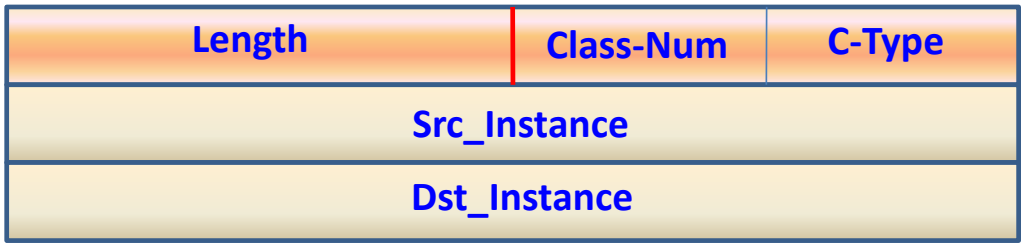


图1 Hello消息报文格式

Hello消息中，Class_Num字段固定值为22，C-Type字段值为1时，是HELLO REQUEST对象，C-Type字段值为2时，是HELLO ACK对象。

Src_Instance： 是一个32位整数值，由节点为每个RSVP接口分配的唯一代表发送节点的标示值，该字段不能为零。

Dst_Instance：根据收到的Src_Instance值来填写本字段值，如果没有收到对端节点的Src_Instance值，该字段值为零。

RSVP协议通过HELLO REQUEST消息，HELLO ACK消息中Src_Instance字段值、Dst_Instance字段值相互对应来确认一个RSVP邻居关系。

举例，DUT1和DUT2建立RSVP邻居，DUT1发送HELLO REQUEST消息，DUT2发送HELLO ACK消息，具体信息如下：

图2 举例组网图

HELLO REQUEST消息报文如下：

```
Internet Protocol, Src: 20.1.1.1 (20.1.1.1), Dst: 20.1.1.2 (20.1.1.2)
Resource Reservation Protocol (RSVP): HELLO Message.
⊞ RSVP Header. HELLO Message.
⊟ HELLO Request/Ack: REQUEST. Src Instance: 0x14f5b. Dest Instance: 0xa89.
    Length: 12
    Object class: HELLO object (22)
    C-Type: 1 - HELLO REQUEST object
    Source Instance: 0x14f5b
    Destination Instance: 0xa89
```

图3 HELLO REQUEST消息报文

HELLO ACK消息报文如下：

```
Internet Protocol, Src: 20.1.1.2 (20.1.1.2), Dst: 20.1.1.1 (20.1.1.1)
Resource Reservation Protocol (RSVP): HELLO Message.
⊞ RSVP Header. HELLO Message.
⊟ HELLO Request/Ack: ACK. Src Instance: 0xa89. Dest Instance: 0x14f5b.
    Length: 12
    Object class: HELLO object (22)
    C-Type: 2 - HELLO ACK object
    Source Instance: 0xa89
    Destination Instance: 0x14f5b
```

图4 HELLO ACK消息报文

从上述两个报文可知，HELLO REQUEST消息中的Src_Instance字段值和HELLO ACK消息中的Dst_Instance字段值相同，而HELLO REQUEST消息中的Dst_Instance字段值和HELLO ACK消息中的Src_Instance字段值相同，说明了DUT1和DUT2之间建立了RSVP邻居关系。

RSVP-TE GR原理

RSVP-TE GR能力要依赖RSVP-TE Hello扩展能力，RSVP-TE Hello能力不是RSVP-TE协议必选特性，如果要使能RSVP-TE GR能力必须先使能RSVP-TE Hello扩展能力，设备上使能RSVP-TE GR能力后通过扩展的RSVP Hello报文向邻居通告自己的GR能力和相关时间参数。

Restart_Cap Object对象用来描述RSVP-TE GR能力以及GR相关时间参数。当节点使能RSVP-TE GR能力之后，节点会在其发送的Hello消息中携带Restart_Cap Object对象来通知对端节点，其具有了RSVP-TE GR能力。

Restart_Cap Object对象格式如下：

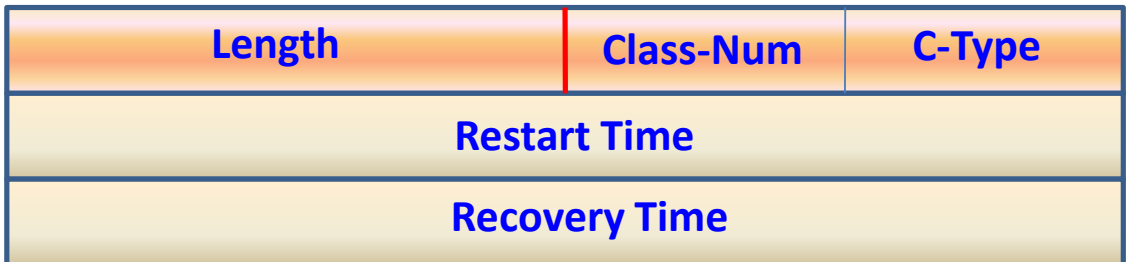


图5 Restart_Cap Object对象格式

在Restart_Cap Object对象中，Class_Num字段值是131；C_Type字段值是1。

其中Restart Time、Recovery Time字段值根据具体配置设置，其含义是：

Restart Time：重启RSVP-TE组件以及重新建立RSVP通信通道的所有时间之和。

Recovery Time：Restarter节点与Helper节点重新建立RSVP邻居关系之后，用于同步RSVP状态以及MPLS转发状态的所有时间之和。

RSVP Hello消息中，Restart_Cap Object对象具体报文内容如下：

```
RESTART CAPABILITY: Restart Time: 120000 ms. Recovery Time: 300000 ms.
  Length: 12
  Object class: RESTART-CAPABILITY object (131)
  C-type: 1
  Restart Time: 120000 ms
  Recovery Time: 300000 ms
```

图6 Restart_Cap Object对象

RSVP-TE GR协议流程

GR Helper节点如果连续几次（具体次数决定于具体配置）发送的Hello报文都没有收到Restarter节点的确认Hello报文，会由此判定GR Restarter节点发生了故障。Restarter节点故障之后，Helper节点启动Restart定时器，进入GR流程，同时会持续向Restarter节点周期性发送Hello报文，其Dst_Instance值置零，Src_Instance值和先前保持一致。同时为了避免RSVP软状态信息因为刷新时间超时而被清除，节点需要抑制RSVP和Path消息刷新机制。后续Helper节点再次收到Restarter节点的Hello报文，要检查Hello报文中的Src_Instance字段，如果该字段值和故障之前相同，则是控制通道故障，否则

是控制平面故障，比如主备倒换。如果 Helper节点和Restarter节点在Restart定时器超时前，重新建立了RSVP邻居关系，Helper节点会终止Restart定时器，启动Recovery定时器，并触发信令报文交互，帮助Restarter节点恢复Rsvp软状态以及重新下刷MPLS转发表项。当Recovery定时器超时后，Helper节点正常退出GR流程。

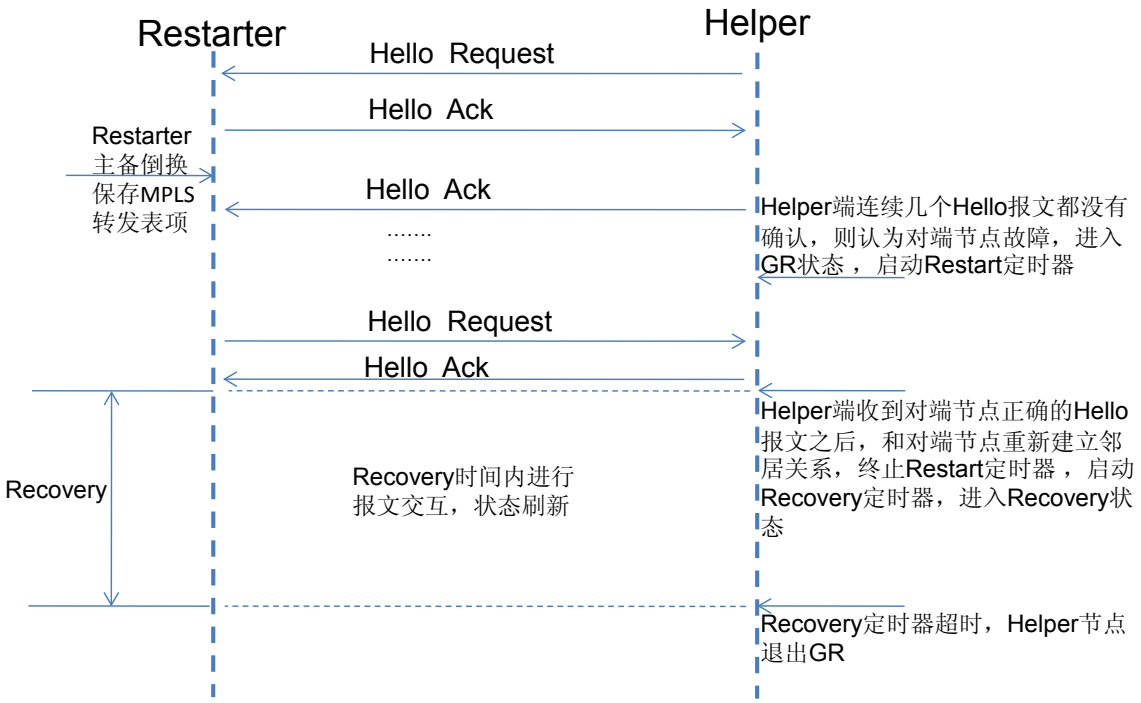


图7 RSVP GR流程图

Restarter节点GR流程

在GR过程中，Restarter节点

首先，Restarter节点在重启其控制协议平面之后，要对MPLS转发表项进行检查，如果没有需要保存的MPLS转发表项，Restarter节点重启之后，发送给Helper节点的Hello报文中的Recovery Time字段值置零，来通知Helper节点不需要进入GR状态。如果有需要保存的MPLS转发表项，Restarter节点重启之后，发送给Helper节点的Hello报文中的Recovery Time字段值为配置值，且在Recovery时间内需要支持RSVP状态恢复过程。 如果Restarter节点在Recovery时间内，其上还有RSVP状态没有完成同步，则需要删除这些RSVP状态，退出GR状态。

其次，Restarter节点重启之后需要和Helper节点协商同步，在此阶段，Restarter节点需要判断Helper是否支持GR能力，如果Helper节点支持GR能力，Restarter节点启动Recovery定时器，进入Recovery恢复状态；果Helper节点不支持GR能力，Restarter节点结束GR流程，删除先前保存的MPLS转发表项，进入正常的协议处理流程。

在GR过程中，Restarter节点如何处理Path、Resv消息

Restarter节点收到Path Message消息之后，要对相应的RSVP软状态进行检查，如果有相应的RSVP软状态，则刷新相应的RSVP软状态；如果没有相应的RSVP软状态，则要检查Path消息中是否含有Recovery_label对象。Restarter节点如果没有Recovery_label对象，会把该Path消息当成一条新LSP

的建立，按照正常流程处理。

Restarter节点如果含有Recovery_label对象，会进一步查找MPLS转发表项，查看MPLS转发表项的入接口以及入标签是否和Path消息相对应，如果没有对应的MPLS转发表项，Restarter节点会把此Path消息当成一条新LSP的建立，按照正常流程处理。如果有相应的MPLS转发表项，Restarter节点需要创建Path消息对应的RSVP状态，相应的MPLS转发表项会被刷新，同时向下游发送相应的Path消息时，需要携带Suggested_label对象，Suggested_label值应该和MPLS转发表项中的出标签值相同。

Restarter节点在GR期间会正常处理Resv消息。但是在处理Resv消息时，如果没有新标签可被分配，或者系统没有足够资源时，Resv消息会被丢弃。当Restarter节点收到的Resv消息没有相应的Path状态时，需要产生ResvErr消息时，Resv消息会被丢弃。

Helper节点GR流程

Helper节点在确认Restarter节点控制协议平面启动之后，须保存RSVP状态信息以及MPLS转发表项。当Helper节点和Restarter节点重新建立通信之后，Helper节点需要刷新和Restarter节点之间的所有Path状态，并向Restarter节点发送相应的Path消息，消息中必须包含Recovery_label对象，对象中Label值和先前Resv消息中的Label值相同，且所有的Path状态都应该在一半Recovery Time时间内完成刷新，如果有较多Path状态信息需要刷新时，Path消息需要间隔发送，避免给Restarter节点带来较大的CPU压力。

Helper节点不会主动发送和Restarter节点共享的Resv状态信息，只有收到相应的Path消息后，Helper节点才会发送Resv消息刷新Resv状态信息，状态处理过程才能正常工作。

RSVP-TE GR小结

我司目前实现了RSVP-TE Helper功能，没有实现RSVP-TE Restarter功能。我司Helper节点RSVP-TE GR功能实现是，当Helper节点连续发送几个Hello报文都没有收到对端节点的确认Hello报文时，Helper节点就进入GR流程。如果Helper节点在Restart时间内没有和对端节点重新建立协商，则Helper节点退出GR流程；其次Recovery定时器超时，Helper节点退出GR流程。

NSR

文 / 李 军

概述

随着网络的快速普及和应用的日益深入，各种增值业务得到了广泛部署，网络中断的发生也就可能影响大量业务，并造成重大损失。因此，作为业务承载主体的基础网络，其可靠性日益成为受关注的焦点，用户对于网络的可靠性也提出了越来越高的要求。因此，业务不间断转发能力成为了目前电信市场衡量设备HA（High Availability，高可靠性）技术的指标之一。

通常情况下，当设备协议重启或主备倒换期间，路由协议层面会出现邻居关系震荡。这种邻居关系的震荡将最终导致路由震荡，进而造成了业务转发中断，使网络可靠性大大降低。目前，解决以上问题的主要方法为业务不间断技术，随着业务不间断技术的应用，对于在设备协议重启或主备倒换期间，业务转发中断的问题已经得到了很好地解决。

实现业务不间断转发的技术主要包括GR （Graceful Restart，平滑重启）以及NSR（Nonstop Routing，不间断路由）。

GR （Graceful Restart，平滑重启）是一种在设备进行IP/MPLS转发协议重启或主备倒换时保证转发业务不中断的技术。它需要周边设备的配合来完成路由等信息的备份与恢复，如果在主备倒换期间拓扑发生变化，GR未退出有可能会造成路由黑洞，甚至路由环路。而且在运营商的实际网络应用中，通常PE（Provider Edge：服务提供商网络边缘设备）设备连接了很多CE（Customer Edge:用户网络边缘设备），而CE设备通常是不支持GR的。因此GR技术在实际网络应用中可能存在一定的使用限制。

NSR（Nonstop Routing，不间断路由）是一种在设备协议重启或主备倒换时保证转发业务不中断的技术。它通过将IP/MPLS的链路状态和路由等转发信息从主用主控板备份到备用主控板，使设备在发生协议重启或主备倒换时可以自行完成链路状态的恢复和路由的重新生成，从而在设备进行协议重启或

主备倒换时，无需周边设备配合即可完成上述信息的备份与恢复。

GR、NSR使用相同的基础架构来保持接口和转发信息，由于NSR不需要周边设备的配合即可完成路由信息的备份与恢复，实现业务的不间断转发，使其在网络应用中更加广泛。

技术简介

NSR原理

NSR的主要思想是在主用主控板故障的情况下，备用主控板能够无缝接管主用主控板的工作，整个过程对对端系统透明，对端系统完全感知不到这种变化，从而保证整个网络的拓扑稳定、路由稳定和业务转发稳定。NSR具体工作原理图如下所示：

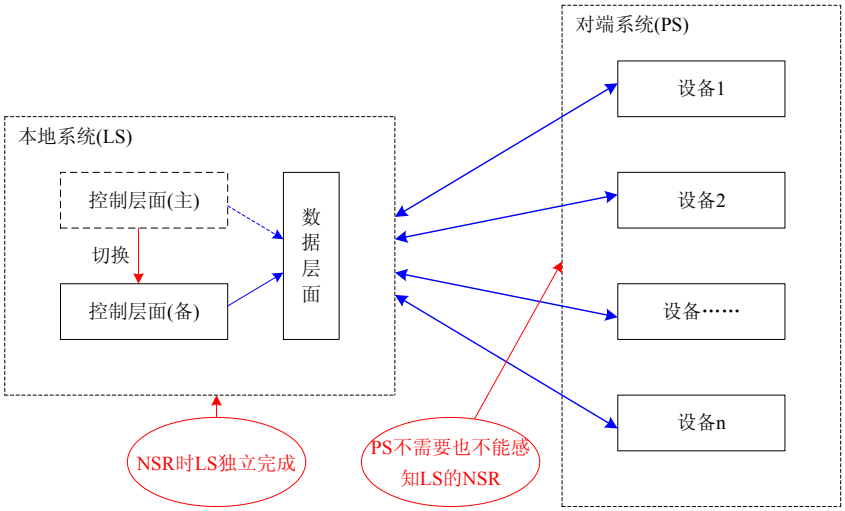


图1 NSR原理

当主用主控板由于某种原因出现故障时，由于备用主控板具有和主用主控板一样的协议状态和动态数据，备用主控板成为新的主用主控板后可以完全接管之前主控板的协议报文收发，继续和周边设备进行协议交互，而周边设备对本地主备板的交替过程并无感知，最重要的就是实现协议不中断，业务转发也不受影响。

NSR的实现，需要系统满足以下要求：

硬件要求：系统采用双主控冗余机制，一块作为主用主控板（AMB），处于工作状态，另一块作为备用主控板（SMB），处于备份状态。主用主控板重启，备用主控板成为新的主用主控板；分布式结构，数据转发和控制分离，使用专门的线卡（接口板）用于数据转发。

系统软件要求：主用主控板正常运行的过程中，会把配置信息、接口状态信息、协议状态信息、会话状态信息以及路由表项等备份到备用主控板，使得备用主控板具有和主用主控板相同的配置信息。当主用主控板因为硬件或软件失效出现故障时，备用主控板接管失效主控板的工作，重新启动控制平面和数据平面以及各业务处理板的工作。另外，由于系统具备控制与转发分离的分布式结构，主备倒换过程中接口板不会重启，接口板上的转发表被保留，可以继续业务转发。

目前业界主流的高性能设备在物理上都实现了将转发板（Data Board）与主控板（Control Board）分开，并且每块板都有自己的处理器与内存，在逻辑上实现了控制平面（Control Plane）和数据平面（Data Plane）相分离。主控板主要运行控制平面的软件，比如运行动态路由协议，维护路由过程中的必要数据库，并且计算出一个路由转发表。而转发板主要运行数据平面的软件，比如根据主控板计算出的

转发表，由转发板进行数据包的转发。对于分布式路由器来说，主控板用于学习和维护路由表，并把转发表FIB (Forwarding Information Base，转发信息库)下发到转发板，使用转发板进行数据包的转发。从逻辑功能上进行划分，主控板和转发板的功能是独立的，为实现NSR这一高可靠性技术奠定了基础。

NSR技术的应用，实现了在主控板切换的情况下，备用主控板能够无缝接管主用主控板的工作，保证路由不出现震荡，业务转发不中断。由于FIB在路由变化（很可能是网络拓扑发生了变化）的情况下，有可能是无效的，所以切换过程不应该花很长时间。切换的时间越短，风险越小。所以，保证备用主控板保持一份与主用主控板一致的配置，由于不需要事先学习，切换将变得很快。 这个切换的过程Cisco称作SSO(Stateful Switchover)，Juniper称作GRES（Graceful Routing Engine Switchover）。

相关技术实现

NSR特性要达到不间断转发的目的，需要对控制模块的运行数据进行备份，使备份系统的运行状态尽量接近主用系统，以保证主备主控板之间保持同步。

主备之间的同步机制，可以采用两种实现模型：一是数据备份模型，即备协议进程不运行，完全依赖主向备备份数据来保持主备协议状态的一致；二是事件备份模型，即备上同时运行协议，和主一样，依靠外部事件驱动协议运行。各个厂家的具体实现可能不太一样，但都是依据这两种模型来展开的。

目前Comware中具备NSR能力的路由协议主要有OSPF、IS-IS、BGP，下面将选取OSPF和BGP分别进行介绍。

对于我司V7平台BGP NSR特性的实现，首先要求BGP所依赖的TCP连接在主备倒换过程中不中断，即系统应首先支持TCP NSR属性，本端和对端建立一个TCP连接，对该连接进行备份，当发生主备倒换时，保证连接不中断，继续TCP通信，可以正确处理收发Sequence、收发buffer，使对端设备不会感知发生了变化；其次要求BGP连接不断。TCP NSR的实现如下图所示：

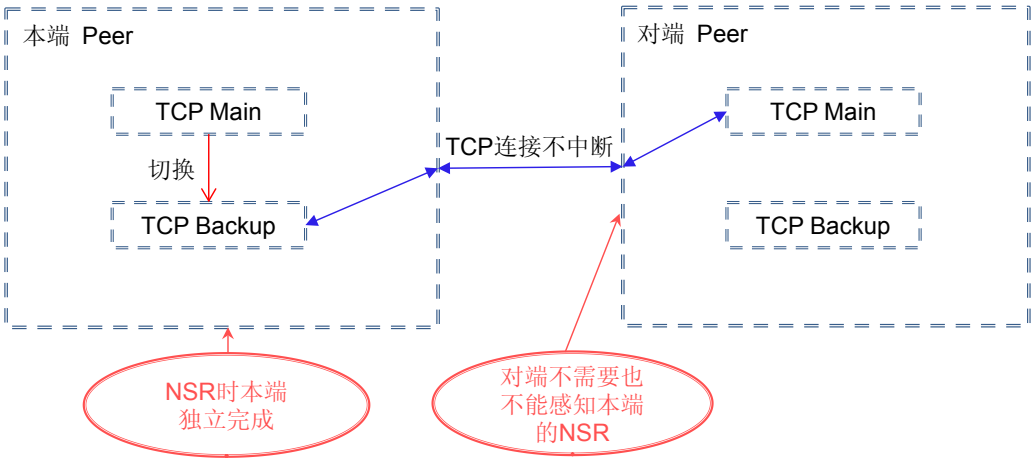


图2 TCP NSR原理

BGP连接不断，需要保持主备状态的一致，Primary和Backup上必须保持以下数据的一致：

1）BGP各项配置；

2）BGP邻居状态；

3）BGP协议表项（路由）。

1）的备份和NSR没有关系。2）的实现是NSR的基础，容易实现，BGP使能NSR后，将主进程上处于Established状态的BGP会话备份到备进程上。如果主进程上有邻居变为Established状态，要将该邻居状态备份到备进程上，如果有邻居状态从Established变为Idle也要将该邻居的状态备份到备进程上。

3）是NSR的难点，对于接收到的路由，为了减少备份信息数据量，将路由信息的备份分为三部分：属性备份，解析UPDATE报文生成属性信息后，首先将属性信息备份到备板，属性信息中包括属性ID；NEXTHOP备份，当NEXTHOP加入到NEXTHOP哈希后，向备板备份NEXTHOP，包括NEXTHOP ID；前缀信息，包含前缀的更新，修改和删除，当这些路由生成时，备份前缀，备份信息中包含属性ID和NEXTHOP ID。当发生主备倒换时，备进程升级为主进程再将以前收到的备份数据还原出来。

对于我司V7平台OSPF NSR特性，通过将主用主控板的运行数据实时同步到备板上，当主用板出现故障或是主备倒换时，备用板快速接管主用板的业务，使邻居不感知本设备发生的变化。要达到这样的目的，一般需要对控制模块的运行数据进行备份，使备份系统的运行状态尽量接近主用系统。备份的数据主要包括：

1）OSPF配置数据；

2）OSPF邻居状态数据、及对应的接口状态数据等运行状态数据；

3）OSPF路由信息，即LSA。

同BGP NSR一样，3)的备份是NSR的难点，在拓扑比较大时LSA的量有可能很大，且变化频繁。新的主用系统通过备份获取到Router ID、优先级、Hello Interval、DR/BDR、邻居列表、接口状态等信息。对于自身生成的LSA，为保证备用板和主用板一致，存在3种方式：

1）将主用板生成的LSA备份到备用板；

2）备份生成LSA所需要的条件，由备用板自己实时生成LSA；

3）备份生成LSA所需要的条件，由备用板倒换后做一次生成LSA处理。

目前我司V7平台采用方案1进行备份，V5平台采用方案2进行备份。

对于非自身生成的LSA，为保证备用板和主用板的一致，需要对这些LSA进行备份，而为了保证主备板处理的一致性，主用板需要在确认LSA备份成功以后，才能向对端回复ACK报文。LSA的泛洪由主用板完成，备用板不需要向外发送泛洪报文。但是备用板需要在倒换后，接管原主用板的泛洪功能。对于路由计算，需要确保路由管理 (RM) 进程路由老化前重新刷新路由，且保证前后路由的一致即可。

下图为OSPF NSR交互过程：

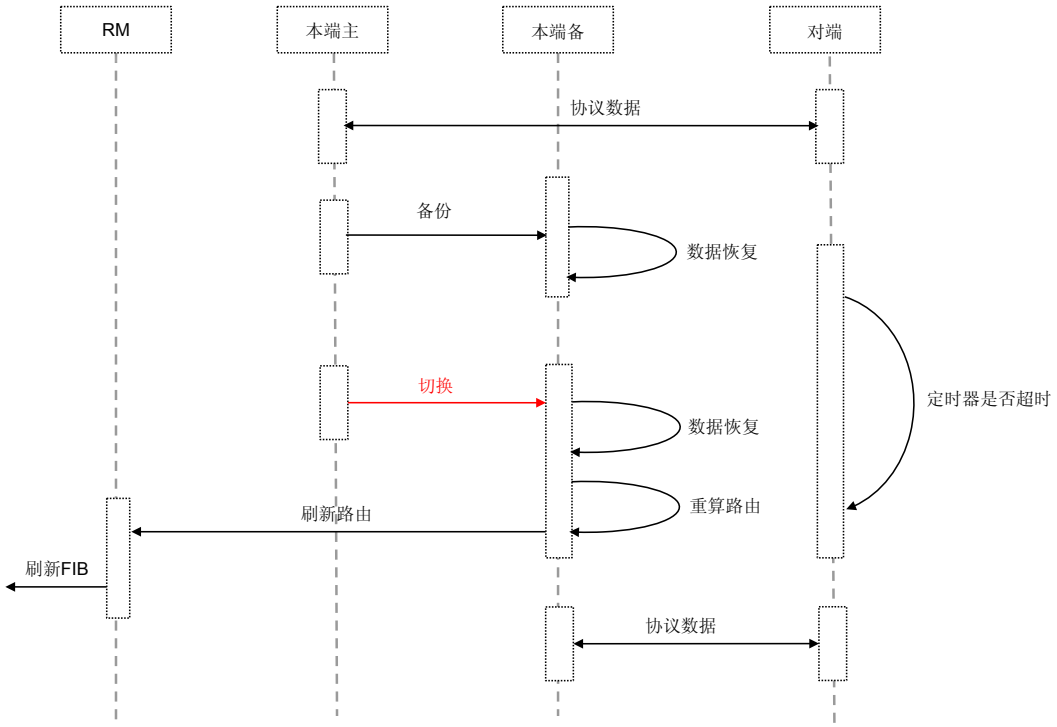


图3 OSPF NSR交互过程

在OSPF NSR整个交互过程中，NSR系统不需要对端系统提供支持，但是需要在对端系统定时器超时之前，重新启动和对端系统的交互(可以在数据完全恢复前、也可以在数据完全恢复后)，否则将造成网络中断。

典型应用

在运营商网络的边缘，即PE（Provider Edge）路由器是典型的NSR应用场所，特别是用户CE（Customer Edge）接入运营商网络的情况，当单点PE出现故障，或者出于维护目的（比如升级软件版本）导致PE路由器主备倒换发生，而通常要求用户CE设备具备GR Helper能力又不太现实，如果部署了NSR，则能够给用户的关键业务提供不间断转发的高可靠性保障。从设备可靠性上考虑，这时候NSR特性正好弥补GR特性的不足。具体应用如下图所示：

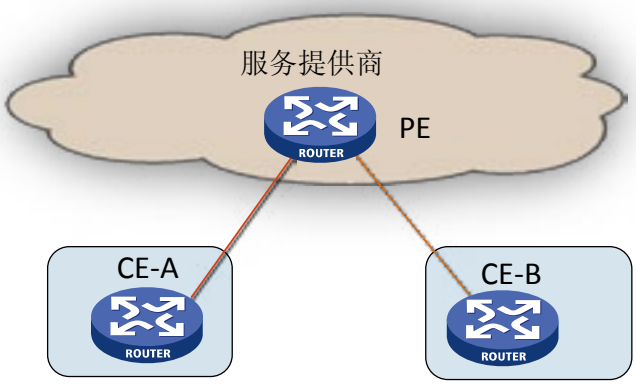
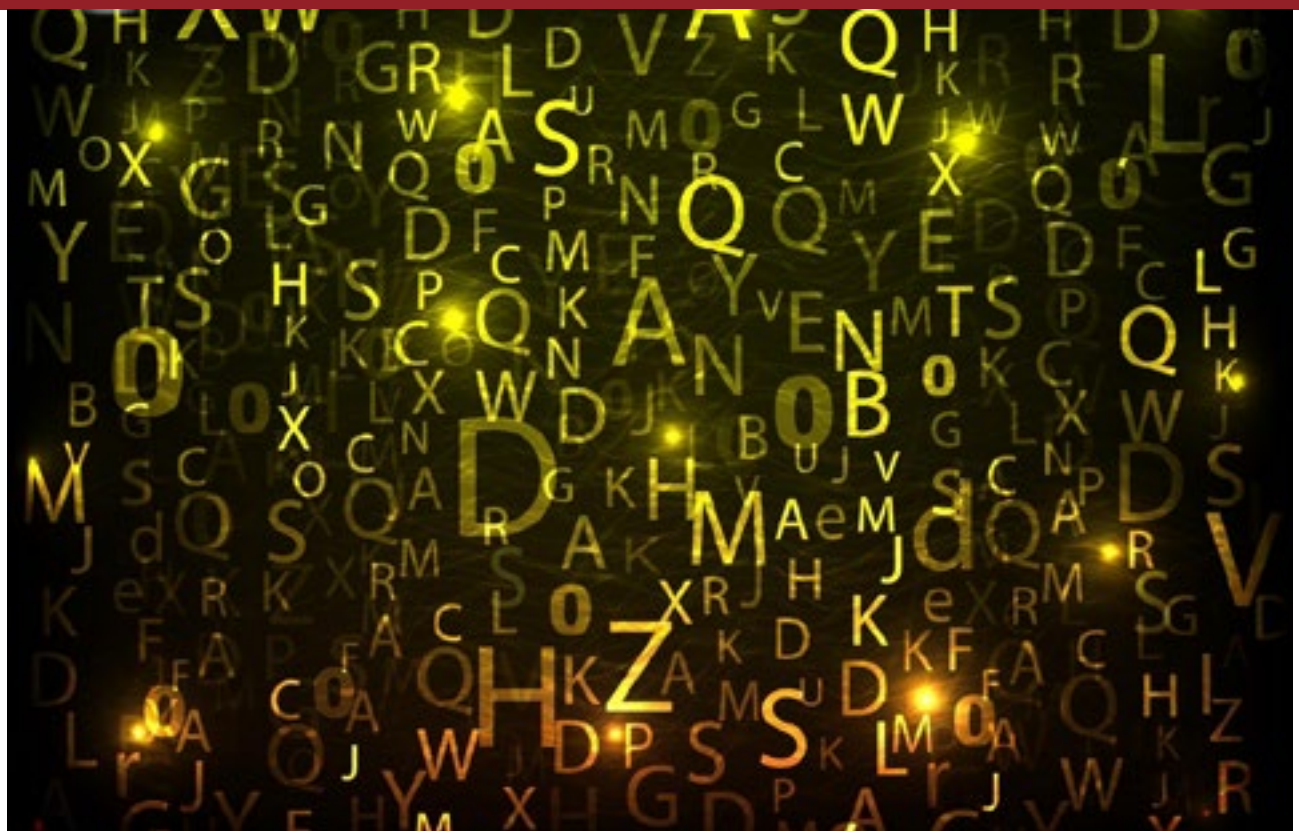


图4 NSR特性的典型应用场景

在图4中，在具备NSR能力的PE设备上进行了主备倒换，但CE-A、CE-B与PE之间的路由协议并没有中断，业务实现不间断转发，CE-A和CE-B并没有感知到PE设备发生了主备倒换。

结束语

在网络中，尤其是电信级多业务承载网络，可靠性成为越来越重要的指标。NSR最大的特点就是设备在主备倒换或是协议重启期间，能够提供不间断转发业务，提高网络的可靠性，同时不需要周边设备的辅助，这使得具备NSR能力的转发设备使用范围更加广泛。



CFD技术

文 / 李连华

CFD产生背景

传统的电信网络中，SDH有着专用的管理通道，可以快速发现网络连接中的问题，快速保护切换，同时提供了丰富的信息供管理员方便地定位故障所在。所以传统电信网络稳定性高，可管理性强，定位问题的手段丰富。

Ethernet网络由于其简单、低成本、高可扩展的优点，使得传统电信网络逐步向Ethernet发展，Ethernet取代SDH已经是大势所趋。但是Ethernet网络相比SDH，也有着很多不足，可管理性差就是其中一个主要缺陷。这包括很多：定位故障的手段不丰富、速度慢，从而导致网络恢复时间慢，维护成本高。这个问题不解决，Ethernet在电信网络里面的应用就会受到很大的局限，为了解决Ethernet网络中的可管理性问题，IEEE提出了802.1ag标准。

我司依据IEEE 802.1ag协议标准，推出CFD（Connectivity Fault Detection，连通错误检测）功能，它是一种二层链路的OAM（Operations，Administration and Maintenance，操作、管理和维护）机制，主要用于在二层网络中检测链路连通性、确定故障位置。

CFD简介

基本概念

CFD有维护域、维护集和维护点的概念。具体含义如下：

◆ MD（Maintenance Domain，维护域）：CFD所覆盖的网络部分，它的界限是由配置在端口上

的一系列维护点所定义的。用维护域名MD来标识维护域。

◆ MA（Maintenance Association，维护集）：维护域中一些维护点的集合。用“MD Name + MA Name”来标识。维护集服务于一个VLAN，每个MD内可以配置多个MA。

◆ MP（Maintenance Point，维护点）：配置在端口上，属于某个MA。可分为MEP（Maintenance association End Point，维护端点）和MIP（Maintenance association Intermediate Point，维护中间点）两种：

→MEP：维护连接的终结点，可以基于VLAN收发CFD报文。每个维护端点用一个数字标识，称为MEP ID。维护端点配置在端口上，它确定了维护域的范围。维护端点所属的维护集和维护域确定了维护端点所发出的报文的VLAN属性和级别属性。根据维护端点在维护域中所处的位置，维护端点的方向有内向和外向之分。

→MIP：维护连接的中间点，可处理和回应CFD报文。它所属的维护集和维护域确定了接收报文的VLAN属性和级别。

◆ MEP列表：MEP列表是同一MA中允许配置的本地MEP和需要监控的远端MEP的集合，它限定了MA中MEP的选取范围，不同设备上同一MA中的所有MEP都应包含在此列表中，且MEP ID互不重复。

CFD工作过程

MD/MA的作用

MD是指连通错误检测所覆盖的一个网络或网络的一部分，对于整个CFD组网来讲，MD维护域是CFD所覆盖的网络，它的界限是由配置在端口上的一系列维护点所定义的。为了准确定位故障点，在MD中引入了级别的概念。MD共分为八级，用整数0～7来表示，数字越大级别越高，MD的范围也就越大。不同MD之间可以相邻或嵌套，但不能交叉，且嵌套时只能由高级别MD向低级别MD嵌套，即低级别MD必须包含在高级别MD内部。

在图1中，MD_A为一个级别高的大MD，包含了五台设备，MD_B为一个级别较低的MD，包含三台设备，嵌套于MD_A中； 若整个组网中发生故障，首先可以查看级别较低的MD_B上是否存在问题，如果MD_B没有出现故障，那么可以确认Device C是没有问题的，排查范围便缩小为四台设备，提高了故障定位的效率。

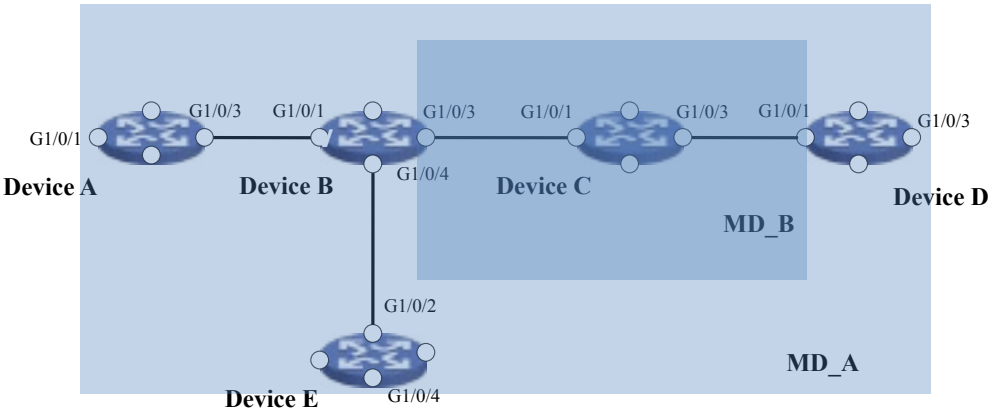


图 1 两个MD嵌套实例

MA就是维护点的集合，每个MA服务于一个VLAN，MA中的MP所发送的报文带有该VLAN的标签。一个MD中可以有多个MA服务于不同VLAN，类似于MSTP中不同的实例；而一个MA只能属于一个MD，该MA也继承了MD的level属性，只能属于一个特定的level。每个MA有一个MA name，这个name在它所属的MD里面是唯一的；每个MA有一个MA ID，它由两部分组成，分别是MD name和MA name。这样MA ID就是全局唯一的。

MEP工作方式

MEP确定了维护域的范围和边界，它是CFD中的核心工作部分，每个MEP都跟一个特定的MA相关联，配置在MA所在端口上。MEP负责发起所有的CFD报文，它是一个MA的终结点。MEP所属的维护域和维护集，确定了该MEP所发出报文的VLAN属性和级别。MEP收到高于自己级别的报文时，不会处理，按原有路径转发。只有收到小于等于自己级别的报文时，才会处理。

MEP分为两种，分别是内向MEP和外向MEP。若维护域中的报文是从配置端口接收的，则是外向的；反之，如果维护域中的报文是从其它端口接收的，则维护端点是内向的。

内向MEP的报文会往所有在该MEP所在VLAN的其他端口发送CCM报文，如图2中所示的内向MEP，当在SW A的G1/0/1端口配置了内向MEP后，SW A的其他端口会向该MEP所工作VLAN的其他端口发出CFD报文。

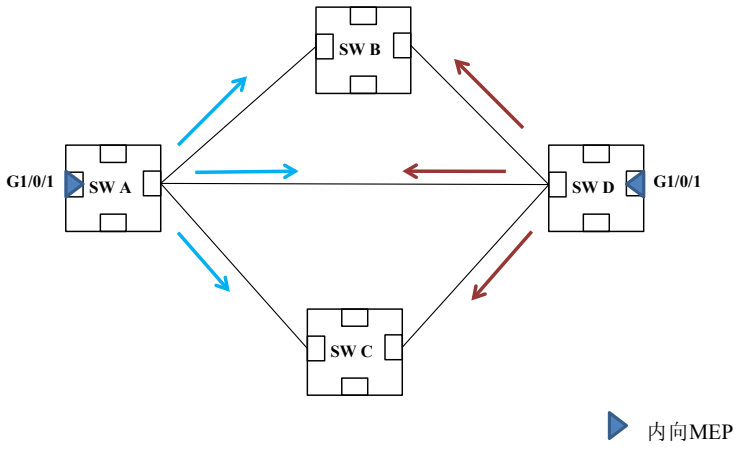


图 2 内向MEP

外向MEP则是一个相反的工作过程，只会往创建了外向MEP的端口发送CCM报文。图3中SW A的端口G1/0/1配置了外向MEP后，直接向端口所在链路发送CFD报文，而不会向其他端口发送。

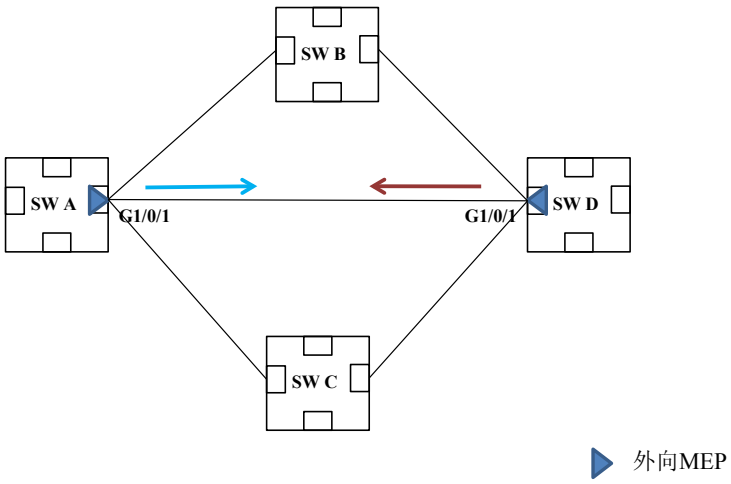


图 3 外向MEP

MIP的创建

MIP位于MA的内部，不能主动发出CFD协议报文，但可以处理和响应CFD协议报文。MIP可以配合MEP完成类似于ping和tracert的功能。当MIP收到不等于自己级别的报文时不会进行处理，而是将其按原有路径转发；只有当MIP收到等于自己级别的报文时才会进行处理。

MIP所属的MA确定了该MIP所能接收的报文所属的VLAN；MIP的级别由其创建规则和所属MD的级别共同确定。MIP的生成规则：Default和Explicit两种。

◆ Default：表示在低级别MA上没有MIP时，本级别创建MIP。

◆ Explicit：表示在低级别MA上没有MIP时，本级别是否创建MIP依赖于低级别MA上是否有MEP。

MIP由系统按照以下图4所示的流程在端口上创建：

◆ 如果端口上尚不存在MIP，就按照级别由低到高依次检查每个MD中的MA；

◆ 如果低级别MA存在MIP，则不在本级别创建MIP，否则进行MIP创建规则判断；

◆ 如果MIP创建规则为Default，在本级别创建MIP；如果创建规则为Explicit则进行低级别MA的MEP判断；

◆ 如果端口上低级别MA存在MEP，在本级别创建MIP，否则不在本级别创建MIP。

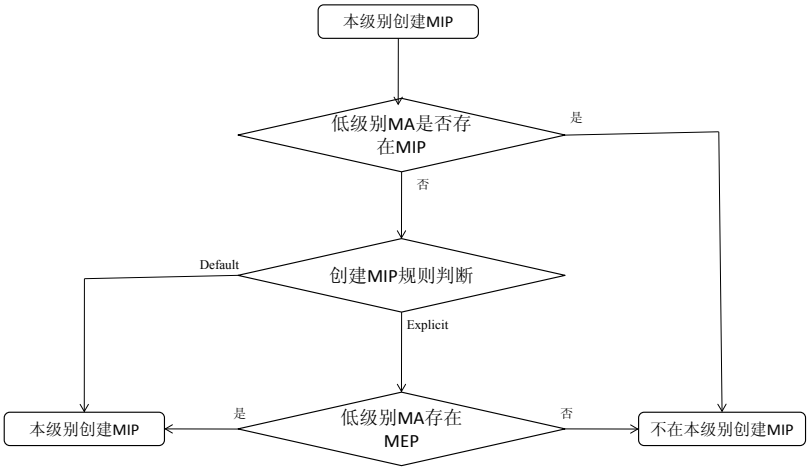


图 4 MIP创建流程

CFD链路检测功能

连续性检测功能

连续性检测是CFD的最基本功能，用来检测MEP间链路的连通性。通过在发送MEP端产生CCM（Continuity Check Message，连续性检测报文）报文，在接收MEP端处理CCM报文来实现。CCM报文以组播形式发出。

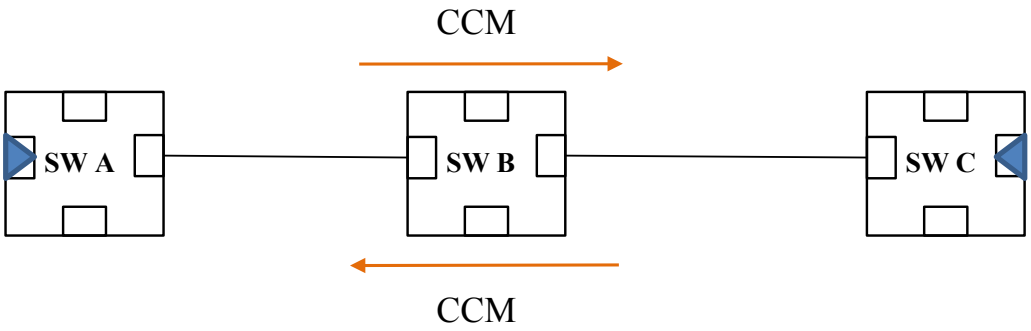


图 5 连续性检测

MEP端点周期性的发送CCM报文（CCM发送间隔如表1所示），每个MEP都可以接收到对方MEP的CCM报文，表示网络连接正常。若远端MEP端点在3.5个CCM报文发送周期内未收到对方MEP发来的CCM报文，则认为链路故障，输出日志告警。

在图5中SW A端配置了MEP节点后，发送CCM报文到整个MD网络，与SW C建立CFD通路；同理，SW C也会发送CCM报文至SW A。

表1 CCM发送间隔表

CCM报文中时间间隔值	CCM发送间隔	远端MEP超时时间
1	10/3ms	35/3ms
2	10ms	35ms
3	100ms	350ms
4	1s	3.5s
5	10s	35s
6	60s	210s
7	600s	2100s

CFD连接的建立并没有交换过程，整个网络依靠CCM报文来进行维持。当单向链路故障出现单通时（如图5中SW A到SW C的CCM报文无法到达，SW C到SW A的CCM报文可达），如何处理？802.1ag给出的解决办法是在CCM报文中加入RDI（Remote Defect Indication）标志位，当MEP接收不到自己对端MEP发送的CCM报文时，就在自己发送的CCM报文中将RDI置位。远端MEP收到了带RDI置位的CCM报文，就知道网络中出现了单通故障，从而进行告警。

环回功能

环回（Loopback，LB）功能用于验证与远端设备之间的连接状态。能类似于ping功能，一端的设备发送环回报文(Loopback Message，LBM)给远端节点，根据能否接收到对方反馈的环回应答报文(Loopback Reply，LBR)来检验连通性。LBM报文和LBR报文都是单播报文。LBM的目标是可以远端MEP，也可以是中间节点MIP。如果目标是MIP，需先通过链路跟踪功能获取到相应的MAC地址。

连续性检测(CC)适合检测单向的连通性失败，而环回适合检测双向的连通性失败。与连续性检测功能合用时通常作为故障确认的手段，而与链路跟踪功能合用时则可用于查找和确认故障点。

链路跟踪功能

链路跟踪（Linktrace，LT）功能用于确定源MEP到目标MEP的路径。类似于IP层的tracert，实现方式是：源MEP发送LTM（Linktrace Message，链路跟踪报文）报文给目标MEP，目标MEP及LTM报文所经过的MIP收到该报文后，会发送LTR（Linktrace Reply，链路跟踪应答）报文给源MEP，源MEP则根据收到的LTR报文来确定到目标MEP的路径。

LTM报文中带有TTL值，缺省值为64，报文每经过一个MP，TTL值减1；在到达目标前TTL减为0，则报文丢弃。LTM报文中还带有一个序列号，用来与回应的LTR报文进行比较，确定LTR报文回复的正确性。

LTR报文由接收LTM报文的MP回复给源MEP的报文。MIP接收LTM报文时要向前转发该报文，并且向源MEP回复LTR报文；MEP接收LTM时，仅向源MEP回应LTR报文。根据链路跟踪的特点，LTM设计为组播报文而LTR为单播报文。

图6形象的给出了链路跟踪的整个工作过程。SW A与SW C上配置有MEP节点，SW B为存在MIP节点的中间设备。当在SW A进行链路追踪时，SW A向目标MEP SW C发出LTM报文；SW B收到LTM报文后，在判断符合转发规则的情况下将报文的TTL减1并向SW C转发LTM报文，同时构造LTR报文回应

SW A；SW C收到LTM报文后，发现自己为目的节点，向SW A回应LTR报文，链路跟踪过程结束。

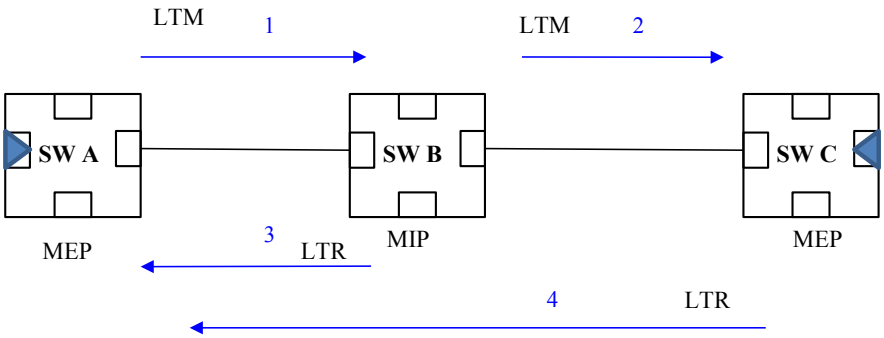


图 6 链路跟踪过程

CFD的链路跟踪功能可分为手动发送LTM报文和自动发送LTM报文两种方式。当管理员发现链路存在故障时可通过手动发送LTM的方式判断链路故障；或在设备上配置自动发送LTM报文，当链路出现故障后设备自动检测链路情况并上报给管理员。

单向丢包测试功能

单向丢包测试（Loss Measurement，LM）功能用来检测MEP间的单向丢包情况。源端发送LMM（Loss Measurement Message，丢包测量报文）报文给目标MEP，目标MEP收到该报文后，会发送LMR（Loss Measurement Reply，丢包测量应答）报文给源端，源端则根据两个连续的LMR报文来计算源端和目标维护端点间的丢包数，即源端从收到第二个LMR报文开始，根据本LMR报文和前一个LMR报文的统计计数来计算源端和目标MEP间的丢包数。LMM与LMR报文均为单播报文。

帧时延测试功能

帧时延测试（Delay Measurement，DM）功能用来检测MEP之间报文传输的时延情况，分为单向时延测试和双向时延测试两种：

◆ 单向时延测试功能的实现方式是：源端发送1DM（One-way Delay Measurement，单向时延测量）报文给目标MEP，该报文中携带有其发送时间。目标MEP收到该报文后记录其接收时间，并结合其发送时间来计算并记录链路传输的时延和抖动。需要注意的是单向时延测试结果需要到对端设备去查看，即本端设备发送1DM报文后无法得知时延测试结果，整个测试过程需要分别在两台设备上操作才能完成。

◆ 双向时延测试功能的实现方式是：源端发送DMM（Delay Measurement Message，时延测量报文）报文给目标MEP，该报文中携带有其发送时间。目标MEP收到该报文后记录其接收时间，然后再发送DMR（Delay Measurement Reply，时延测量应答）报文给源端，该报文中携带有DMM报文的发送和接收时间，以及DMR报文的发送时间。源端收到DMR报文后记录其接收时间，并据此计算出链路传输的时延和抖动（即时延变化值）。双向时延的计算需要扣除对端收到DMM后处理再发送DMR的这段时间的，它只表明链路传输时延。

帧时延测试中所用到的1DM、DMM和DMR报文均为单播报文。

比特错误测试功能

比特错误测试用来检测链路上比特错误发生的情况。工作方式是：源端MEP发送TST（Test，比特错误测试）报文给目标MEP，该报文中携带有伪随机序列或全0值。目标MEP收到该报文后，通过对报

文内容进行计算比较来确定错误比特的情况。

与单向丢包测试功能类似，比特错误测试功能也需要在进行检测的两端设备同时进行操作才能完成整个功能。

TST报文为单播报文。

告警抑制功能

连通性管理失败，CFD会启用告警功能。每个MEP都拥有独立的FNG（Fault Notification Generator）状态机，具备独立的故障通知功能

产生告警的条件可分为以下几种：

◆ MEP丢失对端MEP的CCM报文

◆ MEP接收到对端MEP的CCM报文中带有RDI标志

◆ MEP接收到错误CCM报文

◆ MEP接收到交叉CCM报文

由于CFD有域嵌套的概念，所以内层域已经发现链路故障并告警的情况下，外层域就没有必要再告警了，因为内层域已经将故障定位到了最小范围，此时就需要一个告警抑制功能使CFD故障链路对网络的冲击减小到最小。

我司设备在启用告警抑制功能后，如果外层域MEP收到内层域的AIS（Alarm Indication Signal，告警指示信号）报文后，会抑制本端的故障告警，并继续向更外层域发送AIS报文。而内层域MEP重新收到CCM报文，则停止发送AIS报文。AIS报文为组播报文。

CFD实现案例

CFD典型应用

下面通过一个典型组网应用来说明下CFD的使用。组网要求如下：

◆ 四台设备组成的CFD网络划分为MD_A和MD_B两个MD，级别分别为5和3，设备的所有端口都属于VLAN 100，且各MD中的MA均服务于VLAN 100。

◆ MD_A的边界端口为SW A的G1/0/1和SW D的G1/0/2，为内向MEP；MD_B的边界端口为SW B的G1/0/2和SW D的G1/0/1，为外向MEP。

◆ MD_A的MIP规划在SW B上，SW B上采用Default规则来创建MD_A的MIP。MD_B的MIP规划在SW C上，SW C上配置MD_B的MIP，且其创建规则为Default规则。

◆ MEP的分配情况为：SW A的G1/0/1使能内向MEP 1001，SW B的G1/0/2使能外向MEP 2011，SW D的G1/0/1使能外向MEP 4011，SW D的G1/0/2使能内向MEP 4001。

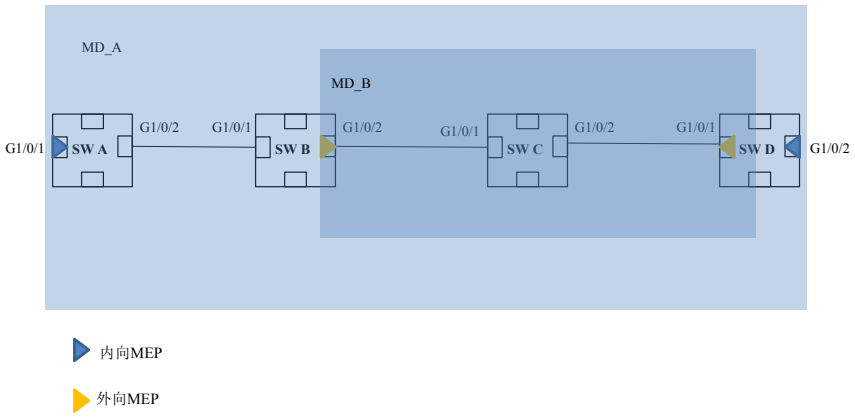


图 7 CFD应用组网

以SW A为例简单说下CFD的关键配置：

#首先需使能CFD功能

```
[SW A]cfd enable
```

#配置CFD实例

```
[SW A] cfd md MD_A level 5
[SW A] cfd ma MA_A md MD_A vlan 100
[SW A] cfd service-instance 1 md MD_A ma MA_A
```

#配置MEP并在端口使能

```
[SW A]cfd meplist 1001 4001 service-instance 1
[SW A]interface GigabitEthernet 1/0/1
[SW A-GigabitEthernet1/0/1]cfd mep 1001 service-instance 1 inbound
[SW A-GigabitEthernet1/0/1]cfd mep service-instance 1 mep 1001 enable
[SW A-GigabitEthernet1/0/1]cfd cc service-instance 1 mep 1001 enable
```

经过上面三步的CFD配置，MD域边缘交换机的基本配置已经完成，SW D的配置与A类似，中间MIP交换机在选择生成MIP规则时通过命令cfd mip-rule来配置。

在配置完成形成CFD链路后，可以对CFD的链路情况进行检查，在SW A上可以看到本端收到远端MEP 4001发出的CCM报文，形成CFD连接：

```
[SW A]display cfd remote-mep service-instance 1 mep 1001
MEP ID   MAC Address      State       Time                MAC Status
4001     0CDA-411A-5D13   OK          2000/05/05 21:47:52   DOWN
[SW A]
```

当SW B与SW C直接的链路出现故障后，由于SW A在CCM报文超时时间过后仍未收到对端MEP的CCM报文，此时远端CFD连接变为FAILED：

```
[SW A]display cfd remote-mep service-instance 1 mep 1001
MEP ID   MAC Address      State       Time                MAC Status
4001     0CDA-411A-5D13   FAILED      2000/05/06 01:14:32   DOWN
[SW A]
```

管理员发现链路故障进行定位时，通过CFD的链路跟踪功能便能发现从SW A到SW B（MAC 0CDA-411A-6513）的链路正常，但没有收到SW C的 LTR回应，便可以判定SW B与SW C间出现了问题：

```
[SW A]cfd linktrace service-instance 1 mep 1001 target-mep 4001
Linktrace to MEP 4001 with the sequence number 1001-7:
MAC Address            TTL     Last MAC         Relay Action
0CDA-411A-6513          63      000F-E000-0002    MPDB
[SW A]
```

单向/双向时延和单向丢包等功能的使用与链路跟踪功能类似，在链路正常时能够正确返回结果，当链路出现故障后通过具体的MEP ID定位到出现问题的链路。

CFD与Smart link联动

我司设备支持Smart link与CFD功能的联动。Smart link技术采用双上行方式将一台下游设备同时

连接到两台上游设备，最大限度地避免单点故障，提高网络可靠性的技术。但Smart link仅仅解决了本地设备的链路备份功能，当上游设备出现故障时依然会由于无法感知而导致流量转发的丢失。为此我司给出了与Monitor link、CFD两种联动方式来解决上游设备故障时Smart link的链路切换问题。本文只介绍Smart link与CFD联动的应用实例。

组网要求：

◆ 四台设备组成Smart link网络，并建立域MD；SW A为Smart link设备，SW A上VLAN 100、VLAN 101的流量通过SW B和SW C双上行到SW D。

◆ 通过Smart link与CFD联动，实现如下功能：正常情况下，VLAN 100的流量经SW A上Smart link组1的主端口G1/0/2向SW D转发，VLAN 101的流量经SW A上Smart link组2的主端口G1/0/3向SW D转发；当SW A与SW D之间的链路发生故障时，原本由各Smart link组的主端口转发的流量能够快速切换到从端口。

◆ CFD通过在一个MD内建立两个服务实例来分别对应两个Smart link组，MA_A与Smart link组1联动，MA_B与Smart link组2联动。MA_A的CFD报文运行于VLAN 100，MA_B的CFD报文运行于VLAN 101，Smart link两组的flush报文与CFD的VLAN严格对应。

◆ CFD MEP配置情况为：SW A端口G1/0/2配置实例1的外向MEP 1001，端口G1/0/3配置实例2的外向MEP 1002；SW D端口G1/0/3配置实例1的外向MEP 4001，端口G1/0/1配置实例2的外向MEP 4002。

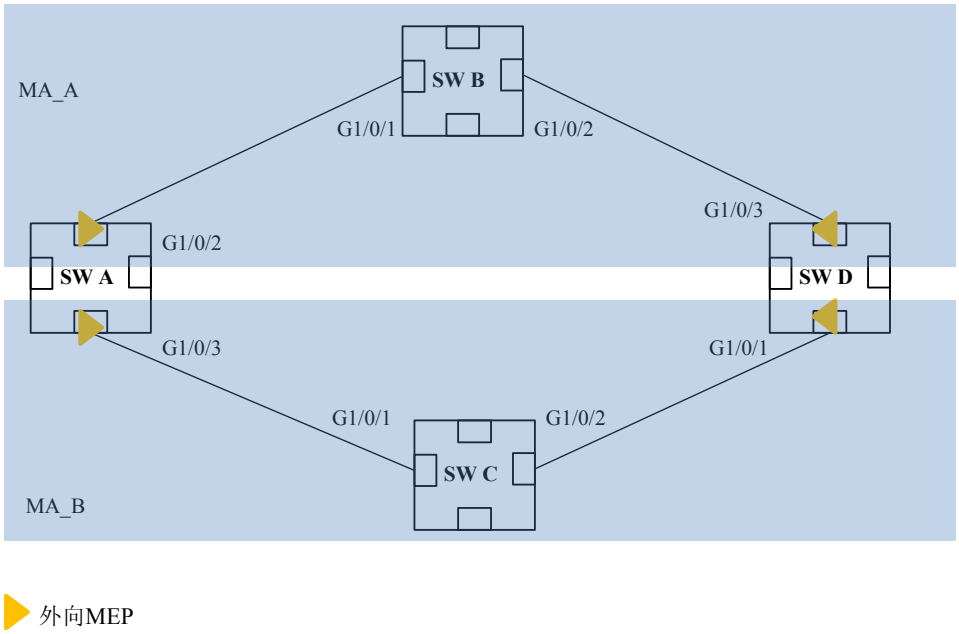


图 8 CFD与Smart link联动

设备的CFD配置需要建立MD和两个服务实例分别运行于VLAN 100和VLAN 101，并在SW A和SW D的四个端口配置上外向MEP；关于Smart link的配置：SW B、C、D需要在互连端口使能控制VLAN内的Smart link flush功能，而在SW A上则需要建立Smart link组，并配置主备端口等其他操作。下面以SW A为例介绍CFD的关键配置：

#首先需使能CFD功能

```
[SW A]cfd enable
```

#配置CFD实例：

```
[SW A]cfd md MD_A level 5
[SW A] cfd ma MA_A md MD_A vlan 100
[SW A] cfd ma MA_B md MD_A vlan 101
[SW A] cfd service-instance 1 md MD_A ma MA_A
[SW A]cfd service-instance 2 md MD_A ma MA_B
```

#配置MEP并在端口使能：

```
[SW A]cfd meplist 1001 4001 service-instance 1
[SW A]cfd meplist 1002 4002 service-instance 2
[SW A]interface GigabitEthernet 1/0/2
[SW A-GigabitEthernet1/0/2]cfd mep 1001 service-instance 1 outbound
[SW A-GigabitEthernet1/0/2]cfd mep service-instance 1 mep 1001 enable
[SW A-GigabitEthernet1/0/2]cfd cc service-instance 1 mep 1001 enable
[SW A-GigabitEthernet1/0/2]quit
[SW A]interface GigabitEthernet 1/0/3
[SW A-GigabitEthernet1/0/3]cfd mep 1002 service-instance 2 outbound
[SW A-GigabitEthernet1/0/3]cfd mep service-instance 2 mep 1002 enable
[SW A-GigabitEthernet1/0/3]cfd cc service-instance 2 mep 1002 enable
```

#设备的Smart link配置完成后，在MEP节点上配置CFD与Smart link的联动：

```
[SW A]interface GigabitEthernet 1/0/2
[SW A-GigabitEthernet1/0/2]port smart-link group 1 track cfd cc
[SW A]interface GigabitEthernet 1/0/3
[SW A-GigabitEthernet1/0/3]port smart-link group 2 track cfd cc
```

配置完成后，正常情况下Smart link与CFD均处于正常工作状态：

```
[SW A]display smart-link group all
 Smart link group 1 information:
 Device ID: 000f-e000-0002
 Preemption mode: ROLE
 Preemption delay: 1(s)
 Control VLAN: 100
 Protected VLAN: Reference Instance 1
 Member                  Role    State    Flush-count Last-flush-time
 ---------------------------------------------------------------------
 GigabitEthernet1/0/2      MASTER  ACTIVE   7         00:34:01 2000/05/07
 GigabitEthernet1/0/3      SLAVE   STANDBY  6         00:04:29 2000/05/07

 Smart link group 2 information:
 Device ID: 000f-e000-0002
 Preemption mode: ROLE
 Preemption delay: 1(s)
```

```
Control VLAN: 101
Protected VLAN: Reference Instance 2
Member                    Role    State    Flush-count Last-flush-time
-----------------------------------------------------------------
GigabitEthernet1/0/3      MASTER  ACTIVE   3          00:04:16 2000/05/07
GigabitEthernet1/0/2      SLAVE   STANDBY  1          21:15:45 2000/05/06

[SW A]display cfd remote-mep service-instance 1 mep 1001
MEP ID   MAC Address      State       Time              MAC Status
4001     0CDA-411A-5D13   OK          2000/05/07 00:34:00   UNKNOWN
[SW A] display cfd remote-mep service-instance 2 mep 1002
MEP ID   MAC Address      State       Time              MAC Status
4002     0CDA-411A-5D13   OK          2000/05/07 00:04:15   UNKNOWN
```

当上游设备链路发生故障时，如SW B与SW D间的链路Down掉后，由于CFD检查到链路故障，与Smart link发生联动， Smart link链路切换到备份端口：

```
[SW A]display smart-link group all
Smart link group 1 information:
Device ID: 000f-e000-0002
Preemption mode: ROLE
Preemption delay: 1(s)
Control VLAN: 100
Protected VLAN: Reference Instance 1
Member                    Role    State    Flush-count  Last-flush-time
-----------------------------------------------------------------
GigabitEthernet1/0/2      MASTER  DOWN     7          00:34:01 2000/05/07
GigabitEthernet1/0/3      SLAVE   ACTIVE   7          00:37:24 2000/05/07

Smart link group 2 information:
Device ID: 000f-e000-0002
Preemption mode: ROLE
Preemption delay: 1(s)
Control VLAN: 101
Protected VLAN: Reference Instance 2
Member                    Role    State    Flush-count Last-flush-time
-----------------------------------------------------------------
GigabitEthernet1/0/3      MASTER  ACTIVE   3          00:04:16 2000/05/07
GigabitEthernet1/0/2      SLAVE   STANDBY  1          21:15:45 2000/05/06
```

此时CFD实例1中显示远端MEP故障：

```
[SW A]display cfd remote-mep service-instance 1 mep 1001
MEP ID   MAC Address      State       Time              MAC Status
4001     0CDA-411A-5D13   FAILED      2000/05/07 00:37:24   UNKNOWN
```

关于CFD与Smart link的联动有几个注意的地方需要说明下：

1、CFD的CC检测VLAN与Smart link组控制VLAN必须一致，否则无法做到联动。

2、CFD实例与Smart link组的关系

CFD链路是基于端口来实现的，CFD用来检测一条链路的状态是否正常，故CFD与Smart link的联动实际是由CFD功能告知端口链路故障，由端口结合Smart link组策略来选择如何进行主备端口的选择。而Smart link不同的组是独立的，可以使用相同的端口来对应所有的Smart link组关系，所以不同的Smart link组完全可以通过相同的CFD实例来实现联动功能。

3、Smart link端口抢占功能是否对与CFD联动有影响

若Smart link组不配置端口抢占功能，则无法与双链路的CFD发生联动。因为在环形组网中，CFD的一个CC检测VLAN只能保证一条链路的联动检测；当主链路发生故障时会导致Smart link流量切换到备链路，若主链路恢复后不进行端口抢占，那么主链路所对应端口处于STANDBY状态；此时备链路的上游设备再发生故障后，由于CFD所对应VLAN只能匹配一个Smart link组，备链路会因为无法检测到CFD链路故障而仍处于Active状态，这就会导致Smart link的链路选择出现故障。所以端口抢占功能是必须要配置的。

4、Smart link只有一个组的情况如何与CFD联动

当Smart link只有一个组时情况类似，只是仅需在主链路配置CFD联动即可。

CFD的报文结构

CFD基本报文格式

1、CFD报文的MAC地址规范

CFD报文封装在以太网报文头里，报文类型为0x8902，以太网的报文结构在此不在赘述。CFD报文的源MAC为发送MP端的MAC地址，该MAC地址应该是一个单播地址，不能为组播地址。目的MAC规定如下：

◆ CCM、AIS报文的目的MAC为01-80-c2-00-00-3y，y根据MD的level确定，比如MD_A的level级别为3，则CCM的目的MAC为01-80-c2-00-00-33；

◆ LTM报文的目的MAC为01-80-c2-00-00-3y，y为MD的level+8，比如MD_A的level级别为3，则LTM的目的MAC为01-80-c2-00-00-3d；

◆ LBM、LBR、LTR、LMM、LMR、1DM、DMM、DMR、TST报文的目的MAC为目标设备的单播MAC地址，均为单播报文。

2、CFD报文的通用属性

CFD协议报文内容的基本结构如图7所示，各字段的定义如下：

MA Level（3bits）：第一字节的高3位，维护域的级别，取值0~7。

Version（5bits）：第一字节的低5位，协议版本号，都为0。接收时可以忽略。

OpCode（8bits）：表示不同的CFD报文类型，具体定义见表2。

Flags（8bits）：根据不同的OpCode分别定义。

First TLV Offset（8bits）：指从这个field往后，直到PDU中第一个TLV之间，有多少个字节。

Varies with value of OpCode（8bits）：是指具体的OpCode所对应的header

End TLV：Type为0的TLV，只有1个字节（0x00），标志CFD报文的结束

	Octet
MD Level	1(high-order 3 bits)
Version	1(low-order 5 bits)
OpCode	2
Flags	3
First TLV Offset	4
Varies with value of OpCode	5
End TLV(0)	First TLV Offset+5

图 9 CFD通用报文格式

OpCode在802.1ag中定义的含义字段如下：

表 2 OpCode字段定义

CFD PDU	OpCode
Reserved for 802.1	0
CCM	1
LBR	2
LBM	3
LTR	4
LTM	5
Reserved for 802.1	6~31
Defined by ITU–T Y.1731	32~63
Reserved for 802.1	64~255

ITU–T Y.1731中在802.1ag基础上新定义的OpCode：

表 3 ITU–T Y.1731 OpCode定义

CFD PDU	OpCode
AIS	33
LCK	35
TST	37
APS	39
MCC	41
LMR	42
LMM	43
1DM	45
DMM	46
DMR	47

3、CFD报文TLV的通用格式

CFD报文TLV格式如图10所示：

	0	1	2	3
1	Type	Length		(Value)

图 10 TLV格式

Type：指明类型。若为0，不存在Length和Value；若非0，至少存在Length。

Length：当Type不为0时存在。该区域长度为16bit，指明Value的字节长度。当Length为0时，Value不存在。

Value：该域可选。Type或Length为0时不存在。

在CFD的使用过程中， TLV的Type字段规定如表4所示：

表 4 TLY Type定义

TLV Type	Type Field
End TLV	0
Sender ID TLV	1
Port Status TLV	2
Data TLV	3
Interface Status TLV	4
Reply Ingress TLV	5
Reply Egress TLV	6
LTM Egress Identifier TLV	7
LTR Egress Identifier TLV	8
Reserved for 802.1	9~30
Organization Specific TLV	31
Defined by ITU-T Y.1731	32~63
Reserved for 802.1	64~255

CCM报文格式

CCM报文在CFD基本报文头中，OpCode为1，其标志位Flags包含RDI / Reserved / Lifetime三个部分：

◆ RDI：Flags的最高1bit，用于标志单通故障。

◆ Reserved：中间4个bits，置0保留。

◆ Lifetime：最低3bits，表明发送者的Lifttime设置，取值含义见表1。

CCM报文的结构如下：

	0	1	2	3
1	Common CFD Header			
	Transaction Identifier/Sequence Number			
5	MEP Identifier		MD Name Format	MD Name Length
9	MD Name…			
11+dnl	Short MA Name Format	Short MA Name Length	Short MA Name…	
13+dnl+snl	(Additional fields may be added, here, in future versions of the protocol)			
5+First TLV Offset	Optional CCM TLVs			
Last	End TLV(0)			

CCM Identifier：可发送该字段为0的CCM报文，或者每发送1个报文加1。

MEP ID：标志发送MEP，有效范围是1～8191，0 为保留值。

MD Name Format：指定MD名区域使用的格式。

MD Name Length：表明MD名区域的长度。

MD Name：根据MD名格式及MD名长度指定的MD名。MD名不能大于64字节。

Short MA Name Format：指定Short MA名的格式。

Short MA Name Length：指定Short MA区域的长度。

Short MA Name：根据Short MA名格式及Short MA名长度指定的Short MA名。

Optional CCM TLVs：每个CCM都需包括Port ID TLV和MAC Status TLV，Organization-Specific TLV可选。

End TLV：CFD报文的结束。

LBM/LBR报文格式

LBR报文的OpCode为2，LBM报文的OpCode为3，它们的Flags都为0。

报文的基本结构如下：

	0	1	2	3
1	Common CFD Header			
5	Transaction Identifier/Sequence Number			
9	(Additional fields may be added, here, in future versions of the protocol)			
9+First TLV Offset	Optional LBM/LBR TLVs			
Last	End TLV(0)			

各选项功能与CCM中的字段基本相同。

在Optional TLV字段，需要包含PortID TLV，Data TLV、Organization-Specific TLV字段可选。

LTM/LTR报文格式

LTR报文的OpCode为4、LTM报文的OpCode为5。

LTM报文中Flags第8bit：UseFDBonly位置1表示，接收LTM报文的维护中间点在硬件转发表中找不到目标MAC地址时，不对报文进行广播转发。第1~7bit：保留位，值为0。

LTR报文中第8bit：UseFDBonly位，与LTM报文中该位值相同。第7bit：FwdYes位，置1表示LTM报文被转发，0表示不转发。第6bit：TerminalMEP位，置1表示该MP是MEP。第1~5bit：保留位，值为0。

其通用报文格式如下，报文与LBM/LBR的报文相似，只是在Optional TLV字段有所区别。LTM在Optional TLV部分包含是Original Information TLV，LTR包含的是Reply Information TLV，Reply Ingress TLV和Reply Egress TLV

	0	1	2	3
1	Common CFD Header			
5	Transaction Identifier/Sequence Number			
9	(Additional fields may be added, here, in future versions of the protocol)			
9+First TLV Offset	Optional LTM/LTR TLVs			
Last	End TLV(0)			

1、Original Information TLV：

	0	1	2	3
1	Type	Length		TTL Fied
5	Original MAC Field			
9				
13	Target MAC Field			
17	Origin Flags Field			

TTL Field： LTM中的跳数，每经过一个MIP处理LTM时TTL值减1，LTM中的TTL Field的缺省值为64。

Original MAC Address Field：发起LTM的MEP的MAC地址，Original MAC是TLV内部的MAC，在报文发送过程中将保持不变。

Target MAC Address Field：LTM所要到达的最终设备的MAC地址。

Origin Flags Field：指定了某些选项，初始标记域的低序的7位被保留；最高位值置1，表示MAC地址只能在硬件转发表中被学习到，学到的MAC被用来向前传送LTM到下一跳的MP。

2、Reply Information TLV

	0	1	2	3
1	Type	Length		Return TTL Field
5	Forwarded Flags Field	Relay Action Field	Chassis ID Length Field	Chassis ID Subtype Field
9	Chassis ID Field			
9+Chassis ID Length Field	Port ID Length Field	Port ID Subtype Field	Port ID Field	

Return TTL Field：当MP接收到LTM并返回LTR的时候，减1；

Forwarded Flags Field：指示LTM是否将被转发。低七位被保留，最高位值为1时，表示LTM将被继续转发；

Relay Action Field：这是一个状态域，指示了回复LTR的状态：数据帧不能交付给MAC转发实体、下一跳的位置不能被识别、在转发表中找到了目标MAC地址、或者在当前活动的转发表中没有找到目标MAC地址，但是地址已经被记录到MEP的CCM数据库中，并且端口记录是在转发状态。

LTR TLV报文中Chassis ID域的参数字段在MEP处理LTM报文生成LTR时被填充。

3、Reply Ingress TLV

当LTM的入端口上存在与LTM报文MA相同的MP时，会用到此TLV，格式如下：

	0	1	2	3
1	Type	Length		Ingress Action Fied
5	Ingress MAC Field			
9				

Ingress Action Field：报告了接收MIP转发LTM数据桢的方式：

a. IngOK：成功转发到MAC转发实体；

b. IngBlocked：入端口被STP协议阻塞。

Ingress MAC Address Field：表示可以被用作LBM到该MP的目的地址。

4、Reply Egress TLV

当LTM被转发实体转发时会用到此TLV，格式如下：

	0	1	2	3
1	Type	Length		Egress Action Fied
5	Egress MAC Field			
9				

Egress Action Field：报告通过外出端口转发LTM数据帧的方式：

a. EgrNone：数据没有到达任何一个出端口；

b. EgrTTL：收到LTM时，TTL的值为1；

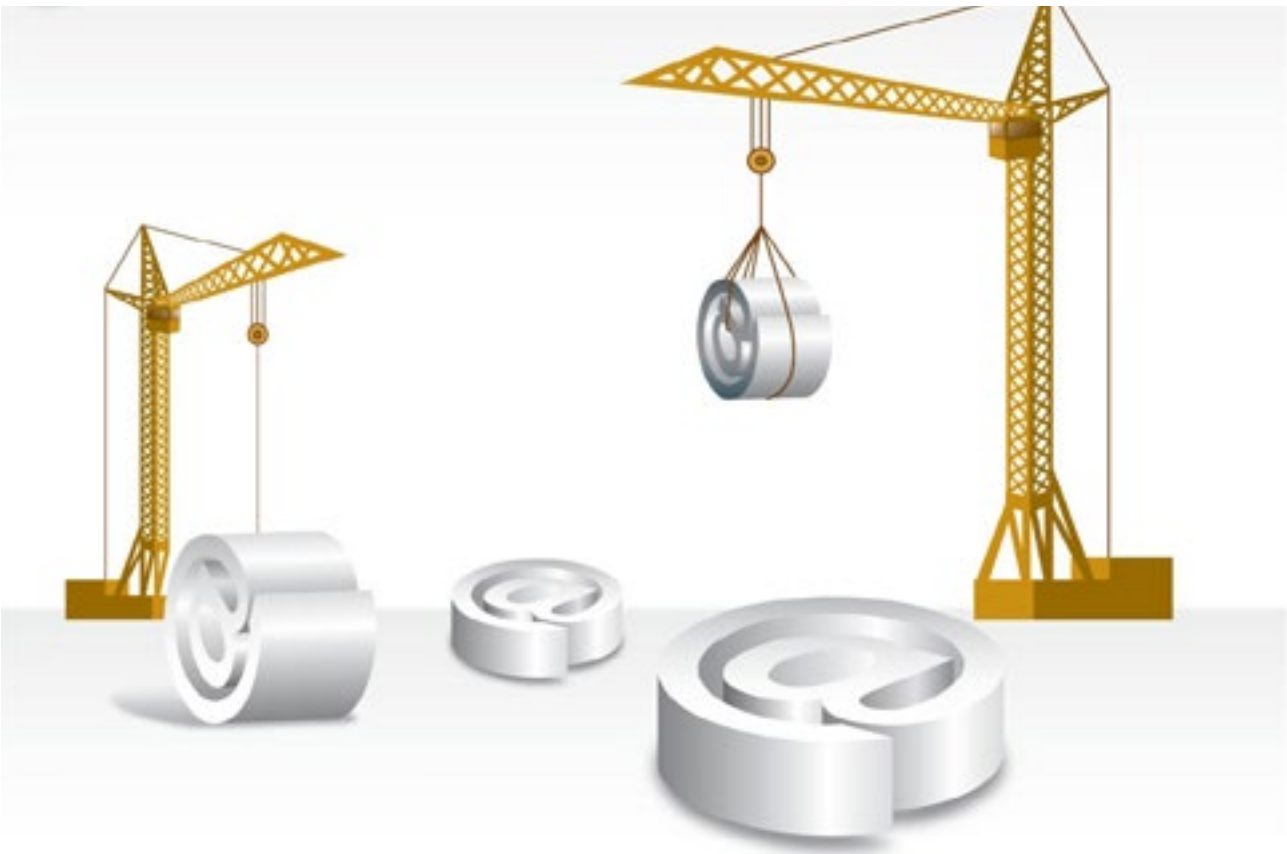
c. EgrDown：出端口的ifOperStatus的状态不是Up；

d. EgrBlocked：软件定义了LTM的下一跳，但是端口被STP阻塞；

e. EgrGVRP：软件定义了LTM的下一跳，但是被VLAN中的GVRP协议阻塞；

f. EgrOK：数据桢将由此端口转发。

Egress MAC Address Field：当需要应答发送TLV时才需要，否则不会出现。表示MAC地址可以被用作LBM到发送MP的目的地址。



BFD 原理与应用

文 / 曹佐清

BFD简介

为了保护设备关键应用，减小设备故障对业务的影响、提高网络的可用性，设备需要能够尽快检测到与相邻设备间的通信故障，以便能够及时采取措施，从而保证业务继续进行。

现有的故障检测方法主要包括以下几种：

◆ 硬件检测：例如POS链路中可以通过SDH（Synchronous Digital Hierarchy，同步数字体系）告警检测链路故障。硬件检测的优点是可以很快发现故障，但并不是所有介质都能提供硬件检测，如以太网链路不具备这样的检测机制。

◆ Hello机制：通常采用信令协议中的Hello报文机制。这种机制检测到故障所需时间为秒级。对于高速数据传输，例如吉比特速率级，超过1秒的检测时间将导致大量数据丢失；对于时延敏感的业务，例如语音业务，超过1秒的延迟也是不能接受的。并且这种机制依赖于信令协议，无法为其它协议提供快速故障检测。

◆ 其他检测机制：不同的协议有时会提供专用的检测机制，但在系统间互联互通时，这样的专用检测机制通常难以部署。

BFD（Bidirectional Forwarding Detection，双向转发检测）就是为了解决上述检测机制的不足而产生的，它是一套全网统一的检测机制，用于快速检测、监控网络中链路或者转发路径连通状况，保证邻居之间能够快速检测到通信故障，从而快速建立起备用通道恢复通信。BFD的检测速率能够以毫秒级

增量设定。当对等系统没有接收到预先设定数量的数据包时，它推断BFD保护的软件或硬件基础设施发生故障。

BFD工作原理

BFD提供了一个通用的、标准化的、介质无关、协议无关的快速故障检测机制，可以为各上层协议如路由协议、MPLS等统一地快速检测两台路由器间双向转发路径的故障。

BFD在两台设备上建立会话，用来监测网络设备间的双向转发路径，为上层协议服务。BFD本身并没有发现机制，而是靠被服务的上层协议通知其邻居信息建立会话，会话建立后如果在检测时间内没有收到对端的BFD控制报文则认为发生故障，通知被服务的上层协议进行相应的处理。

BFD工作流程

下面以OSPF与BFD联动为例，简要介绍BFD会话工作流程。

BFD会话建立过程

BFD会话建立流程如图1所示：

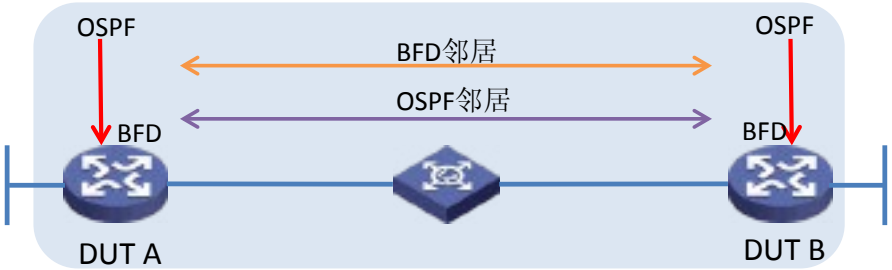


图1 BFD会话建立流程图

（1）OSPF通过自己的Hello机制发现邻居并建立连接；

（2）OSPF将邻居的参数及检测参数（包括目的地址和源地址等）通告给BFD；

（3）BFD根据收到的参数信息进行计算并建立起会话。

BFD故障发现处理流程

BFD故障发现处理流程如图2所示：

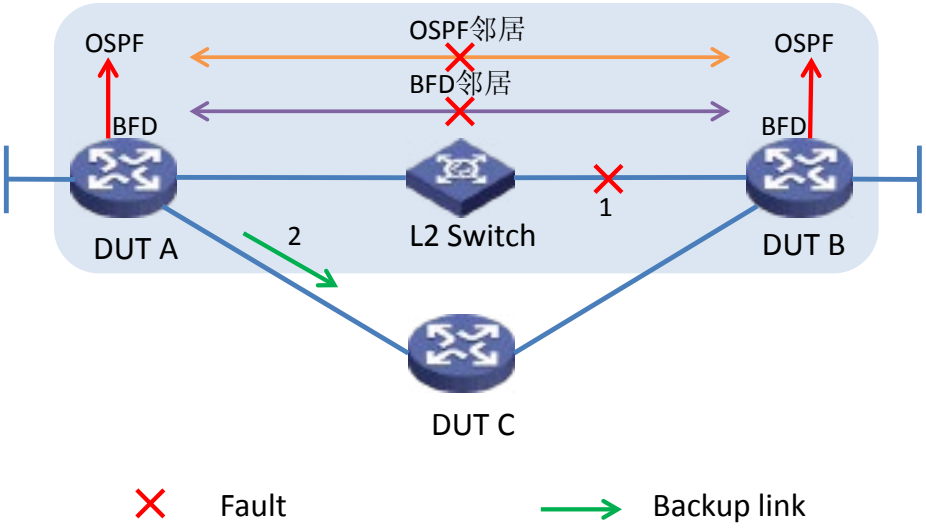


图2 BFD故障发现处理流程图

当网络出现故障时，DUT B与L2 Switch之间链路通信发生故障；

(1) DUT A与DUT B之间的BFD会话检测到故障，拆除BFD邻居会话；

(2) BFD通知本地上层运行的OSPF协议进程BFD邻居不可达；

(3) OSPF协议进行邻居Down过程的处理；

(4) DUT A选择备用路径，启用备份转发路径。

BFD检测方式

单跳检测：BFD单跳检测是指对两个直连系统进行IP连通性检测，这里所说的“单跳”是IP的一跳。

多跳检测：BFD可以检测两个系统间的任意路径，这些路径可能跨越很多跳，也可能在某些部分发生重叠。

双向检测：BFD通过在双向链路两端同时发送检测报文，检测两个方向上的链路状态，实现毫秒级的链路故障检测。

BFD会话工作方式

控制报文方式：链路两端会话通过控制报文交互监测链路状态，两端定时器分别协商，如图3所示，采用的是控制报文模式（Ctrl mode）

Echo报文方式：链路某一端通过发送Echo报文由另一端转发回来，实现对链路的双向监测。

```
<SR6608-X>display bfd session verbose

 Total session number: 1   Up session number: 1   Init mode: Active

 IPv4 session working under Ctrl mode:

        Local Discr: 5580            Remote Discr: 5519
          Source IP: 55.1.1.1       Destination IP: 55.1.1.2
     Session State: Up                   Interface: GigabitEthernet3/0/1
  Min Trans Inter: 400ms          Act Trans Inter: 400ms
   Min Recv Inter: 400ms        Act Detect Inter: 2000ms
  Running Up for: 00:02:15            Auth mode: None
    Connect Type: Direct              Board Num: 3
            Protocol: OSPF
           Diag Info: No Diagnostic
<SR6608-X>
```

图3 BFD会话信息

BFD运行模式

BFD会话建立前运行模式

BFD会话建立前有两种模式：主动模式和被动模式。

◆ 主动模式：在建立会话前不管是否收到对端发来的BFD控制报文，都会主动发送BFD控制报文，

如果某一端连续几个包都没有接收到，就认为会话为“Down”了。如图3所示，采用的是主模式（Init mode:Active）

◆ 被动模式：在建立会话前不会主动发送BFD控制报文，只有等收到BFD控制报文后才会向对端发送BFD控制报文。

在会话初始化过程中，通信双方至少要有一个运行在主动模式才能成功建立起会话。下面对两端都为主动模式的会话建立过程进行说明，一端主动模式一端被动模式的会话建立过程基本相同。

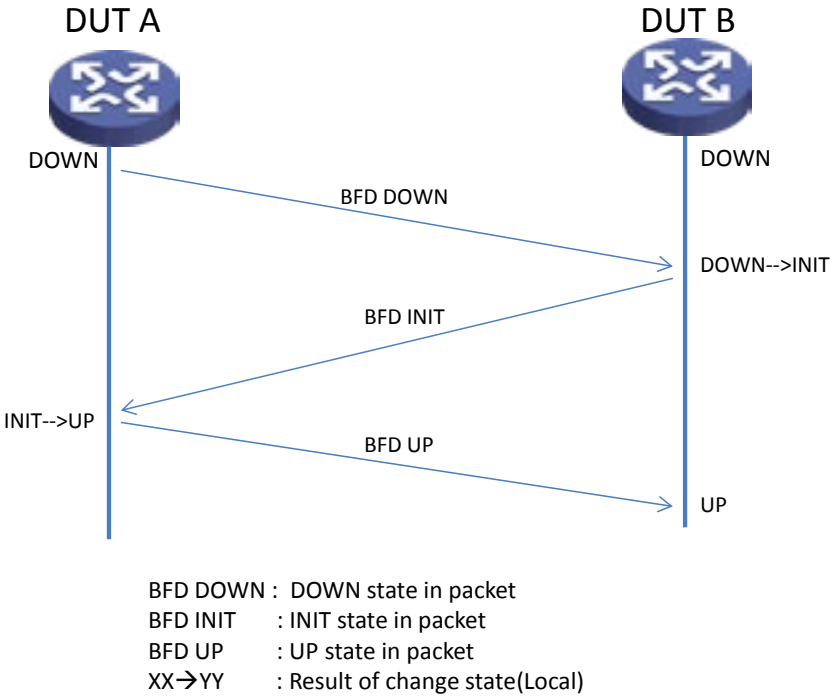


图4 BFD会话连接建立过程

如图4所示，BFD使用三次握手的机制来建立会话，发送方在发送BFD控制报文时会在Sta字段填入本地当前的会话状态，接收方根据收到的BFD控制报文的Sta字段以及本地当前会话状态来进行状态机的迁移，建立会话。

◆ DUT A和DUT B的BFD收到上层应用的通知后，发送状态为DOWN的BFD控制报文。DUT B的BFD状态变化同DUT A。

◆ DUT B收到对端状态为DOWN的BFD控制报文后，本地会话状态由DOWN迁移到INIT，随后发送的BFD控制报文中将Sta字段填为2表明会话状态为INIT。DUT A的BFD状态变化同DUT B。

◆ DUT A收到对端状态为INIT的BFD控制报文后，本地会话状态由INIT迁移到UP，随后发送的BFD控制报文中将Sta字段填为3表明会话状态为UP。DUT B的BFD状态变化同DUT A。

◆ BFD双方状态都为UP，会话将成功建立并开始检测链路状态。

BFD会话建立后运行模式

BFD会话建立后有两种模式：异步模式和查询模式。通信双方要求运行在相同的模式。

◆ 异步模式：以异步模式运行的设备周期性地发送BFD控制报文，如果在检测时间内对端没有收到BFD控制报文，则认为会话down。

◆ 查询模式：假定有一个独立的方法，确认自己和对端系统的连通性。这样，BFD会话建立后，会停止周期发送BFD控制报文，除非需要显式地验证连接性。

说明：目前Comware V5平台仅支持异步模式，而V7平台支持异步和查询两种模式。当BFD会话工

作于echo报文方式时，不受运行模式控制。在需要显式验证连接性的情况下，系统以协商的周期连续发送几个P比特位置1的BFD控制报文。如果在检测时间内没有收到返回的报文，就认为会话down；如果认为连通，则不再发送报文，等待下一次查询的触发。

BFD定时器协商

BFD定时器由Desired Min TX Interval、Required Min RX Interval、Required Min Echo RX Interval和Detect Mult进行计算，双方在邻居建立、邻居维护过程中均会进行计算。

BFD会话建立前BFD控制报文以1000ms的间隔周期发送以减小报文流量。在会话建立后则以协商的时间间隔发送BFD控制报文以实现快速检测。在BFD会话建立的同时，BFD控制报文发送时间间隔以及检测时间也会通过报文交互协商确定。在BFD会话有效期间，这些定时器可以随时协商修改而不影响会话状态。BFD会话中不同方向的定时器协商是分别独立进行的，双向定时器时间可以不同。

BFD控制报文发送时间间隔=MAX[本端Desired Min TX Interval，对端Required Min RX Interval]，即由比较慢的一方决定了发送频率。

BFD控制报文检测间隔=MAX[对端Desired Min TX Interval，本端Required Min RX Interval]*对端Detect Mult。

如果加大本端Desired Min TX Interval，那么本端实际发送BFD控制报文的时间间隔必须要收到对端F字段置位的报文后才能改变，这是为了确保在本端加大BFD控制报文发送时间间隔前对端已经加大了检测时间，否则可能导致对端检测定时器错误超时。

如果减小本端Required Min RX Interval，那么本端检测时间必须要等收到对端F字段置位的报文后才能改变，这是为了确保在本端减小检测时间前对端已经减小了BFD控制报文发送间隔时间，否则可能导致本端检测定时器错误超时。

然而如果减小Desired Min TX Interval，则本端BFD控制报文发送时间间隔将会立即减小；加大Required Min RX Interval，则本端检测时间将会立即加大。

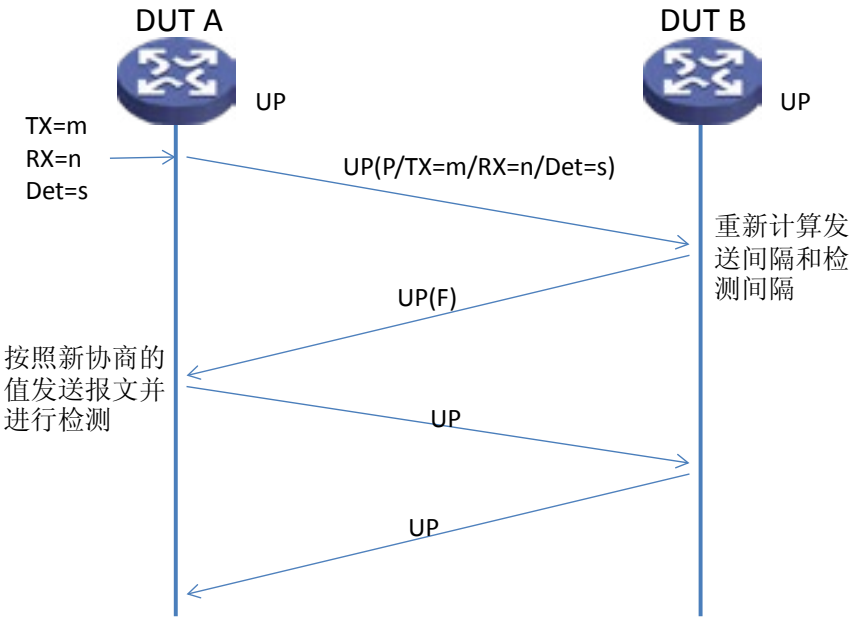


图5 BFD检测时间协商

下面介绍参数改变后定时器的协商过程，如图5所示，DUT A与DUT B双方已建立BFD会话，如果此时将DUT A的Desired Min TX Interval和Required Min RX Interval及Detect Mult分别修改为m、n、s，则参数改变后定时器的协商过程：

◆ DUT A向对端发送定时器参数（TX=m/RX=n/Det=s）修改后的BFD控制报文，并将控制报文中P字段置位；

◆ DUT B收到DUT A发送的P字段置位BFD控制报文，将根据报文中本端的定时器参数Desired Min TX Interval值与对端定时器参数Required Min RX Interval值为n之中的最大值决定了发送频率，重新计算发送间隔呵检查间隔。并将协商后的定时器参数的F字段置位的控制报文发送给DUT A。

◆ DUT A收到对端发来F字段置位的控制报文。根据对端协商后使用新的发送频率来改变本端定时器发送频率。

◆ 定时器协商结束两端按照协商后的时间间隔快速检测。

关于BFD定时器计算需注意如下事项：

◆ 区分配置的定时器和实际的定时器，实际定时器是计算出来的。

◆ 本地配置的发送定时器变化后，必须在收到对方F置位报文后才会改变实际发送间隔。

◆ 本地配置的接收定时器变化后，必须在收到对方F置位报文后才会重新计算并改变检测间隔。

◆ 收到对方接收定时器变化后，必须立刻重新计算并按照新计算的发送间隔发送（此时离上次发送报文时间大于/小于新发送间隔）。

◆ Echo功能使能后，协议规定设置本地控制报文接收间隔不能小于1000ms。

◆ 两端的检测时间分别计算，可以不一样。

BFD故障检测

BFD会话建立及定时器协商完成后，两端会以协商后的间隔发送BFD控制报文。每当收到BFD控制报文时，就会重置检测时间定时器，保持会话UP状态。如果在检测时间内没有收到BFD控制报文，BFD会话会迁移到DOWN状态，并通知该会话所服务的上层应用发生故障，由上层应用采取相应的措施。本端BFD会话DOWN后，发给对端的BFD控制报文中的Sta字段就填为1，通知对端会话DOWN，对端的BFD会话也迁移到DOWN状态。

BFD报文格式

BFD报文有两种类型分别是控制报文和Echo报文。其中协议没有规定Echo报文的具体格式。协议只规定了控制报文的格式，目前控制报文格式有两个版本(版本0和版本1)，当前BFD缺省采用版本1建立会话。

BFD 控制报文

BFD控制报文封装在UDP报文中传送，对于单跳检测其UDP目的端口号为3784，对于多跳检测其UDP目的端口号为4784。BFD echo报文与BFD控制报文格式类似（区别在于字段Desired Min TX Interval和Required Min RX Interval为空），其UDP目的端口号为3785。BFD控制报文包括强制部分和可选认证部分。

强制部分报文格式如图6所示。

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Vers |  Diag  |Sta|P|F|C|A|D|M|  Detect Mult  |    Length     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       My Discriminator                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Your Discriminator                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Desired Min TX Interval                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Required Min RX Interval                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Required Min Echo RX Interval                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

图6 BFD强制报文格式图

可选认证部分格式如图7所示。

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Auth Type   | Auth Len    |  Authentication Data...       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

图7 BFD可选认证报文格式图

BFD控制报文各字段含义如下表所示：

字段	含义
Vers	BFD协议版本号，目前版本号为1
Diag	诊断码，表明发送方最近一次会话Down的原因
Sta	报文类型（BFD会话当前状态），用0/1/2/3分别代表AdminDown/Down/Init/Up
P	Poll bit连接建立及参数变化时需要置位，表示需要Final bit置位的回复
F	Final bit，对Poll bit回应时置位
C	Control Plane Independent ，表示BFD快速检测不是由控制平面实现，而是由转发平面实现
A	Authentication Present bit，该字段置位表明报文包含认证部分，会话需要进行认证
D	Demand bit，该字段置位表明发送方希望以查询模式运行，不置位表明不希望以查询模式运行或不支持查询模式
M	Multipoint bit，为以后实现P2MP扩展预留，目前置0
Detect Mult	检测时间倍数
Length	BFD控制报文长度，单位为字节
My Discriminator	发送方产生的一个唯一非0值，用来标识不同的BFD会话
Your Discriminator	如果已经收到会话邻居发送的BFD控制报文则该值为收到报文中的My Discriminator，否则为0
Desired Min TX Interval	发送方支持的最小BFD控制报文发送时间间隔，单位为微秒。
Required Min RX Interval	发送方支持的最小BFD控制报文接收时间间隔，单位为微秒
Required Min Echo RX Interval	发送方支持的最小BFD Echo报文接收时间间隔，单位为微秒。为0表示不支持BFD Echo报文
Auth Type	认证类型
Auth Len	可选认证部分长度，包括Auth Type和Auth Len字段，单位为字节

关于M字段，关于M字段，在后面“多点BFD网络”小节中进行了简要介绍。

BFD Echo报文

BFD Echo报文是由本端发送、本端接收处理，BFD协议并没有对BFD Echo报文的格式进行定义，而是由实现者自己定义，唯一要求是发送方能够通过报文内容区分会话。Comware V5的实现中区分会话方法与BFD控制报文一致。

BFD Echo报文采用UDP封装，目的端口号为3785，目的IP地址为发送接口的地址，源IP地址由配置产生（配置的源IP地址要避免产生ICMP重定向）。

BFD多点网络检测

在BFD控制报文字段中预留了一个比特位M（MULTIPOINT），当它置为非0值时，用于点到多点（组播）网络扩展使用，IETF工作组已经制定了“BFD多点网络”草案，这篇Internet草案主要定义了一种通过使用BFD协议提供对多个发送者（“头节点”）和一组或多个接收者（“尾节点”)之间的多点（组播）故障检测机制。

多点BFD检测只是从发送者到接收者连接的多点路径上。虽然它可能会在两个方向使用单播路径，但多点 BFD是由发送者控制几乎所有的选项和时间参数。

在这篇草案中，“多点”的定义是一种机制，指由单个发送者发送数据包，一个或多个系统接收，具体应用包括IP组播网络、点到多点的MPLS网络。本草案将修改并在适当时候添加到BFD基本规范中。

点到多点网络中，让多个接收者快速检测多点连接网络故障是非常困难的。本草案确定由发送者来快速检测链路故障，通过发送者控制各种选项和时间参数，接收者也不会向发送者同步BFD数据包。若发送者认为与某个接收者之间有连接故障，则发送者直接发送单播BFD给这个接收者，若发送者认为与多个接收者之间有连接故障，则发送者发送多点BFD控制数据包给接收者，接收者收到后返回设置了F位的轮询（P）位单播BFD控制包，从而进行链路故障检测。

本草案对点到点BFD的状态参数进行了修改，新增了bfd.SessionType、bfd.SilentTail、bfd.ReportTailDown、bfd.UnicastRcvd状态参数和某些BFD选项，另外多点BFD对状态机也进行了修改，因为涉及到多对一映射，会话的建立与拆除不再采用3次握手方式，较明显变化是去掉了Init状态，保留了DONW和UP状态，详细描述请参看“draft-ietf-bfd-multipoint-01.txt”文档。

BFD在产品上的实现

我司低端路由器与交换机，多数为集中式设备，采用主控板CPU集中式处理BFD链路故障检测功能，BFD协议报文由主控板CPU产生、发送和处理。由于BFD需要在路由器之间建立会话，当路由器业务量较大或BFD会话数量较多时，主控CPU占用率高，BFD协议报文可能得不到调度，可能导致对端设备检测错误，网络无故振荡。因此，集中式BFD无法在全业务的情况下实现真正的30ms故障检测。

中高端路由器与交换机，大都是分布式框架结构，通过主控板及业务板的多核CPU分布式处理BFD链路故障检测，它们一般由主核CPU处理会话的建立以及状态机的变化，后续会话的检测由辅助CPU或者某个VCPU处理。交换机以S12500系列为例：S12500系列交换机由主控板的主CPU负责控制平面的工作，即BFD会话的建立与状态机的变化等，而由主控板上的OAM CPU负责报文等数据的收发，路由器以SR66系列为例：SR66系列路由器采用分布式框架结构，其主控板、业务板卡如FIP210、FIP600都是多核CPU，在FIP210、FIP600业务板上，BFD会话的建立以及状态机都在控制核（VCPU0）上处理，而会

话的检测及报文的收发由其中的数据核（分别由VCPU3、VCPU7）处理，。这样使得链路故障检测业务从主控CPU分离出来，既减轻主控CPU的负荷、保障主控CPU的安全性，同时提高链路故障检测业务的处理性能。

我司各路由交换设备是业界支持BFD关联协议最丰富的设备之一，关联的协议包括： OSPF与BFD联动、OSPFv3与BFD联动、IS-IS与BFD联动、IPv6 IS-IS与BFD联动、RIP与BFD联动、静态路由与BFD联动、BGP与BFD联动、IPv6 BGP与BFD联动、VRRP与BFD联动、MPLS与BFD联动、Track与BFD联动、IP快速重路由。

BFD组网应用

下面以VRRP与BFD联动组网应用进行描述。

VRRP协议主要是解决当Master出现故障时，Backup能够快速接替Master的转发工作，保证用户数据流在尽量短的时间内恢复，从而保持网络通信的连续性和可靠性。但是当Master出现故障时，VRRP依靠Backup设置的超时时间来判断是否应该抢占，切换速度最少在1秒以上，这对于吉比特速率级的高速数据传输，超过1秒的检测时间将导致大量数据丢失；对于语音等时延敏感的业务，超过1秒的延迟也是不能接受的。由此有必要将BFD应用于Backup对Master进行检测。

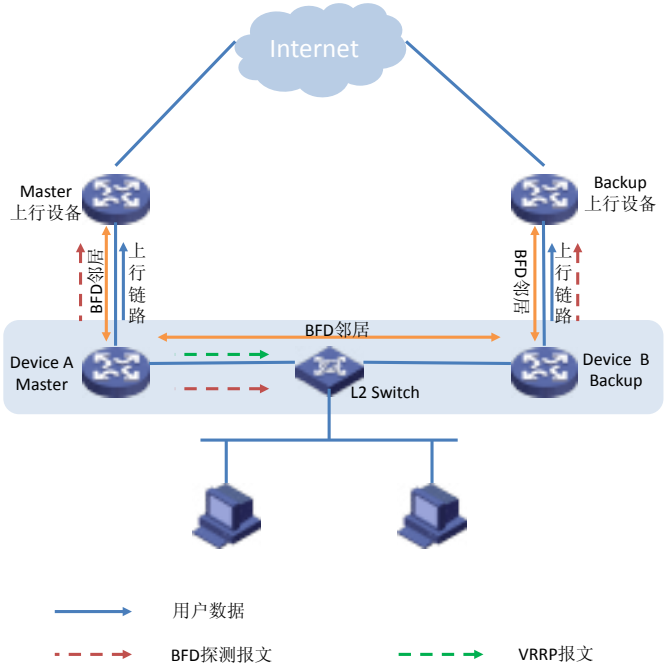


图8 VRRP与BFD联动组网图

1、BFD应用于Backup对Master的检测，可以实现对Master故障的快速检测，如图8所示，Device A与Device B通过L2 Switch互连，在设备上配置VRRP，在Device A和L2 Swicth之间的链路发生故障后，BFD能够快速检测并通告VRRP，触发VRRP主路由器降低优先级，引起主备切换，实现对Master故障的快速检测，缩短用户流量中断时间。

2、BFD应用于VRRP上行链路的快速检测，如图8所示，初始情况下，Device A作为Master设备，承担转发任务；Device B为Backup路由器，处于就绪监听状态。Device A使用BFD监视上行到达Internet的链路状态。如果Device A的上行链路发生故障，即使Master正常工作，用户报文实际上也是无法正常转发的。此时BFD能够快速检测并通告VRRP，触发Device A设备在毫秒级感知到网络变化，立即发送低优先级的VRRP报文给Device B。Device B将切换为新的Master设备，之后由新的Master设备进行数据转发，完成对VRRP上行链路的快速检测。

MPLS OAM

文 / 陈锦军

概述

MPLS OAM（Operation，Administration and Maintenance）技术为MPLS网络提供了一套缺陷检测及缺陷纠正机制，通过MPLS OAM及保护倒换机制可以完成CR-LSP转发平面的检测功能，并在缺陷发生后的50ms内完成保护倒换，从而将缺陷所产生的影响减小到最低。本篇文档介绍了MPLS OAM产生背景及工作原理，读者应具有基本的MPLS知识。本文档都是基于ITU-T定义的Y.1711而介绍的。

承载MPLS的各种服务层，比如SDH都具有完善的OAM机制，问题在于MPLS可以在多种不同的服务层上传送（甚至LSP可以跨越由不同服务层组成的网络），并且它的用户层也是多种多样，如IP、FR、ATM、Ethernet等等，为了在MPLS的用户平面能确定LSP的连通性，并且可以衡量网络的利用率以及度量网络的性能，以便能根据与用户签订的SLA协议提供业务，MPLS层需要提供一种完全不依赖于任何用户层或物理层的OAM机制。

MPLS OAM实际上为MPLS用户层单独提供了一套检测机制，独立于其他网络层并为用户提供LSP的状态信息，为网络管理以及维护人员提供丰富的LSP诊断接口，为网络性能测量以及用户计费提供依据；MPLS OAM在提供检测工具的同时，还具备完善的保护倒换机制，能够在MPLS层发生缺陷后50ms内完成用户数据的倒换动作，使用户数据的损失减小的最低。

为了更好的理解OAM原理，这里对保护倒换机制进行简单说明。保护倒换（Protection Switching）是为主隧道建立相应的备用隧道。主隧道和备用隧道构成一个保护组，在主隧道发生故障时，数据流能迅速的倒换到备隧道，从而提高网络的可靠性。当主隧道故障恢复时，还可以将数据流从备隧道回切到主隧道。目前业界实现有1:1和1+1两种保护倒换方式。

MPLS OAM实现原理

基本检测功能

MPLS OAM基本检测功能主要用来检测连通性，包括CV检测和FFD检测，两者的区别主要是FFD相比CV报文发送速率更快，检测周期更短。具体交互过程如图：

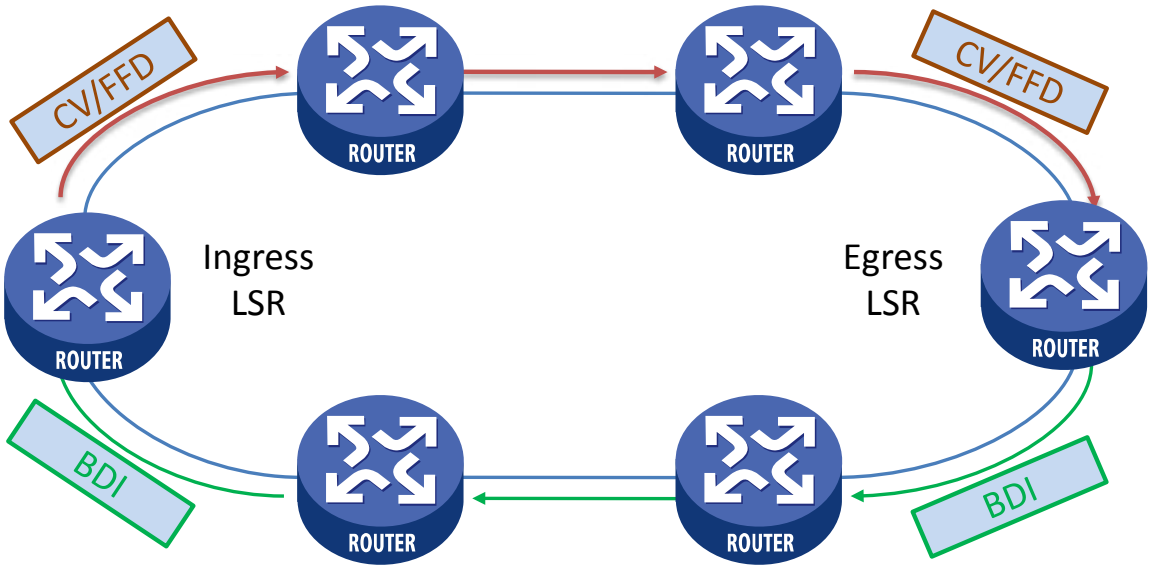


图1 MPLS OAM连通性检测示意图

1) 入节点发送CV/FFD报文，报文通过被检测的LSP到达出节点；

2) 出节点把接收到的报文类型、频率、TTSI等信息字段与本地记录的对应值相比较来判断报文的正误，并统计检测周期内收到的正确报文与错误报文的数量，从而对LSP的连通性随时进行监控；

3) 当出节点检测到LSP缺陷后，分析出缺陷类型，通过反向通道将携带缺陷信息的BDI报文发送给入节点，从而使入节点及时获知缺陷状态。

在配置OAM基本检测功能时，需要为被检测LSP绑定一个反向通道。反向通道是与被检测LSP具有相反的入节点和出节点的LSP，用于承载BDI报文，用来通告缺陷，有以下三种类型：

◆ 专用反向LSP。每条前向LSP都有自己的反向LSP，这种方法相对稳定，但可能造成资源浪费。

◆ 共享反向LSP。多条前向LSP共用一条反向LSP，所有LSP返回BDI报文均通过这一条反向LSP，这种方法可以减少资源浪费，但当多条前向LSP同时出现缺陷时，这条反向LSP上可能会出现拥堵。

◆ 非MPLS路径。这种方式容易受到攻击，不建议采用。

首包触发功能

当LSP入节点晚于出节点开启MPLS OAM功能，或出节点开启而入节点不开启MPLS OAM功能时，会造成出节点的LOCV（Loss of Connectivity Verification defect，连通性检测缺陷）告警。为了避免这种情况，可以通过首包触发功能来解决这个问题。

所谓首包触发功能，是指出节点在超时等待时间（OverTime）内接收到第一个CV/FFD报文，以此时作为连通性检测的起点；如果出节点在配置MPLS OAM功能的OverTime时间后，没有接收到CV/FFD报文，则产生BDI报文。注意只有在出节点上才能配置首包触发功能。

自适应

为了防止源宿配置CV/FFD类型或刷新频率不一致造成OAM检测功能失效，Egress节点以首次收到的CV/FFD报文作为后续报文校验的依据。Ingress停止OAM功能时，发送FDI报文(携带自动协议信息)通知Egress节点，Egress节点恢复到初始状态。

远程链路通告

RLSN（Remote Link Status Notification，远程链路状态通告）功能是指LSP出节点可以监控某一个接口的状态并发送BDI报文通知Ingress节点。监控的接口可以是出节点上的任意一个接口。

◆ 当接口down时，触发扩展BDI报文通知入节点，入节点不会通知保护倒换，但记录RLSN状态信息，以便用户查询。

◆ 当接口重新up时，停止发送扩展BDI报文，入节点不会通知保护倒换，但记录RLSN状态信息，以便用户查询。

OAM报文介绍

OAM报文类型

所有的OAM报文都是根据OAM报文的头部的功能类型码确定的，具体的功能类型码对应的报文类型如下：

表1 OAM报文功能类型定义

OAM function type codepoint (Hex)	First octet of OAM packet payload function type and purpose
00	Reserved
01	CV (Connectivity Verification)
02	FDI (Forward Defect Indicator)
03	BDI (Backward Defect Indicator)
04	Reserved for Performance packets
05	Reserved for LB-Req (Loopback Request)
06	Reserved for LB-Rsp (Loopback Response)
07	FFD (Fast Failure Detection)

OAM报文特征

目前定义了全局保留的标签值14作为”OAM Alert Label”，即收到的报文携带的标签如果是14，该报文就被认为是MPLS OAM报文。当源端构造OAM报文时，会在正常数据转发的标签前面插入一个OAM标签14来区别其他的普通MPLS用户数据报文。

Label字段：

在源端构造OAM报文时，会在正常数据转发的标签前面插入一个OAM标签，即标签14。

EXP字段：

OAM报文的标签栈的EXP字段，应该全部置为0，这样做的目的是为了尽可能的降低OAM的丢弃率，保证服务质量。

栈底标志：

OAM报文的标签栈必须置为1，保证标签14为最内层的标签。

TTL字段：

OAM报文的TTL值必须置为1，这样做的原因如下：

◆ OAM报文不能被转发出去，如果转发，会产生不可预见的错误；

◆ 减少环路的可能。

有效载荷

OAM报文的有效载荷是由OAM的功能类型、特定功能类型的报文数据和BIP16（报文校验和）组成。OAM报文的有效载荷必须最少达到44字节，以适应二层数据报文的最短报文限制（如Ethernet的60字节的长度）。当报文长度不足时，OAM报文使用填充0来补充报文，所有现在用来填充的位置都作为保留的字段，以备将来的扩展需要。

TTSI结构

每个CV/FFD报文都会携带一个唯一表示某一条LSP的标识，这个标识就是TTSI(Trail Termination Source Identifier LSP源宿连接标识)。

TTSI结构如下：

LSR ID	LSP ID
16 octet	4 octets

如上图所示，TTSI由16字节的LSR ID和4字节的LSP ID构成。注意，LSP ID的前两个字节目前都填充0，为将来可能的应用预留。

对于IPv4，LSR ID的前10字节填充为0x00，接着的2字节填充为0xFF，最后4字节为IPv4地址。

All 00Hex padding	All FFHex padding	IPv4 Address
10 octets	2 octets	4 octets

CV报文介绍

CV（Connectivity Verification）报文的检测速率较低，主要用在有效模型的检测(availability model)。CV报文是由LSP的Ingress LSR生成，频率为1个/S，并在LSP的Egress LSR终止的报文。CV报文会携带一个唯一的网络标识符（即TTSI）用来标识检测的LSP，Egress LSR会根据CV报文的情况产生各种缺陷类型。

CV报文有效载荷结构如下：

Function Type (01Hex)	Reserved (all 00Hex)	LSP Trail Termination Source Identifier	Padding (all 00Hex)	BIP16
1 octet	3 octets	20 octets	18 octets	2 octets

CV报文字段意义：

字段	含义
Function type	报文类型，0x01代表CV报文
Reserved	保留字段，取值为全0
LSP trail termination source identifier	TTSI，源宿连接标识，网络中一条LSP唯一的标识。由16字节的Ingress LSR ID与4字节的LSP ID组成。对于IPv4，Ingress LSR ID的前10字节填充为0x00，接着的2字节填充为0xFF，最后4字节为IPv4地址
Padding	填充字段，取值为全0
BIP16	报文校验和

FFD报文介绍

FFD（Fast Failure Detection）报文的检测速率较高，主要应用在对时延要求较高的场合，如保护倒换等。FFD报文同CV报文一样，也是由LSP的Ingress LSR生成，并在LSP的Egress LSR终止的报文。

FFD和CV报文是独立和互斥的，并且FFD的频率是可以调整的。FFD报文可支持10ms、20ms、50ms、100ms、200ms和500ms等多种发送频率。推荐的最佳频率值是50ms。FFD报文携带的信息与CV报文一致，不同的是FFD报文带有频率信息。

需要注意的是FFD只能用于检测最底层LSP的检测，即不能出现这种情况：底层LSP使用CV报文检测，而上层LSP使用FFD报文检测。这种情况下，如果底层的LSP发生缺陷，CV报文检测给上层的FFD检测发送FDI报文会不够快。导致上层先报出了缺陷，会误报给管理员上层LSP缺陷，但事实上是下层LSP缺陷。因为CV报文要上送FDI报文需要的时间是3s+1s，而FFD检测却只要150ms就可以检测出缺陷。

有效载荷结构如下：

Function Type (07Hex)	Reserved (all 00Hex)	LSP Trail Termination Source Identifier	Frequency	Padding (all 00Hex)	BIP16
1 octet	3 octets	20 octets	1 octet	17 octets	2 octets

FFD报文字段含义：

字段	含义
Function type	报文类型，0x07代表FFD报文
Reserved	保留字段，取值为全0
LSP trail termination source identifier	TTSI，源宿连接标识，网络中一条LSP唯一的标识。由16字节的Ingress LSR ID与4字节的LSP ID组成。对于IPv4，Ingress LSR ID的前10字节填充为0x00，接着的2字节填充为0xFF，最后4字节为IPv4地址
Frequency	FFD报文发送频率，可以包含的值有下列： 00 reserved 01 10msec 02 20msec 03 50msec (默认值) 04 100msec 05 200msec 06 500msec. 07–255 reserved

Padding	填充字段，取值为全0
BIP16	报文校验和

如果收到频率字段为保留值的FFD报文，则Egress节点认为Ingress节点不是一个有效的缺陷状态，因此会产生dLOCV缺陷。

FDI报文介绍

FDI报文即Forward Defect Indication，前向缺陷指示。FDI报文是为了对检测到的缺陷作出响应而产生的，缺省1个/S。主要目的是在下层LSP检测到缺陷之后抑制上层LSP产生的错误告警。FDI报文产生有以下两种途径：

◆ 第一次检测到dServer/dUnknown缺陷

◆ LSP的宿端发生任何MPLS的缺陷

当入节点检测到缺陷之后，就会向下游节点发送FDI报文。FDI报文在以下几种情况下是需要的：

◆ 抑制高层不必要的警告信息

◆ 指示高层缺陷发生在低层

◆ 允许高层产生适当的BDI报文

有效载荷结构如下：

Function Type (02Hex)	Reserved (00Hex)	Defect Type	TTSI (optional, if not used set to all 00Hex)	Defect Location	Padding (all 00Hex)	BIP16
1 octet	1 octet	2 octets	20 octets	4 octets	14 octets	2 octets

FDI报文字段含义：

字段	含义
Function type	报文类型，0x02代表FDI报文
Reserved	保留字段，取值为全0
Defect Type	LSP缺陷类型
LSP trail termination source identifier	TTSI，源宿连接标识，网络中一条LSP唯一的标识。由16字节的Ingress LSR ID与4字节的LSP ID组成。对于IPv4，Ingress LSR ID的前10字节填充为0x00，接着的2字节填充为0xFF，最后4字节为IPv4地址.
Defect location	缺陷定位信息
Padding	填充字段，取值为全0
BIP16	报文校验和

BDI报文介绍

BDI报文即Backward Defect Indication，反向缺陷指示。OAM BDI报文用于LSP的Egress节点在发现LSP缺陷后，将缺陷信息通过反向通道告知LSP的Ingress节点，发送速率与CV/FFD报文一致。

有效载荷结构如下：

Function Type (03Hex)	Reserved (00Hex)	Defect Type	TTSI (optional，if not used set to all 00Hex)	Defect Location	Padding (all 00Hex)	BIP16
1 octet	1 octet	2 octets	20 octets	4 octets	14 octets	2 octets

BDI报文字段含义：

字段	含义
Function type	报文类型，0x03代表BDI报文
Reserved	保留字段，取值为全0
Defect Type	LSP缺陷类型
LSP trail termination source identifier	TTSI，源宿连接标识，网络中一条LSP唯一的标识。由16字节的Ingress LSR ID与4字节的LSP ID组成。对于IPv4，Ingress LSR ID的前10字节填充为0x00，接着的2字节填充为0xFF，最后4字节为IPv4地址.
Defect location	缺陷定位信息
Padding	填充字段，取值为全0
BIP16	报文校验和

BDI报文的TTSI字段是可选的。一般在反向路径为共享LSP或者非MPLS路径时需要填写TTSI字段，如果反向路径是独占的LSP，则一般可以使用Label索引。如果反向隧道是双向隧道，则可以把TTSI字段填为全0。如果使用TTSI字段，则这里填写被检测的正向LSP的TTSI值。

错误报文处理

每个OAM报文都使用BIP16检测错误的报文，BIP16位于OAM报文有效载荷的最后两个字节。参与BIP16运算的包括OAM报文有效载荷的所有字段，包括OAM的功能类型、特定功能类型的报文数据和BIP16（报文校验和）。

BIP16的运算方法如下：G(x) = x16 + 1。

所有的报文都必须经过BIP16验证之后，才能作进一步的处理。对于接收到的BIP16不正确的报文，必须被丢弃。

当CV/FFD检测的时候，如果BIP16值错误，报文不能被正确上送处理。此时宿端就会上报dLOCV缺陷，这样的处理是没有问题的，但是建议对于这种上报的dLOCV缺陷，还需要告诉网管是由于报文校验和错误导致。

当一个设备不支持MPLS OAM，收到OAM报文能否能够正确处理？因为OAM报文的特殊标签为14，大多数设备对这个保留的标签不作处理，所以应该不会引入问题。

缺陷检测处理

缺陷类型定义

目前主要定义的缺陷类型，即DT有以下几种：

表2 FDI/BDI OAM 报文的缺陷代码

Defect Type	DT code (Hex)	含义
dServer	01 01	任何MPLS层的Server层的缺陷。这个缺陷不是有MPLS OAM状态机自己生成的，而是由MPLS OAM层的Server层通告给MPLS OAM的。
dPeerME	01 02	任何非MPLS子网生成的对等维护实例缺陷。只有当缺陷不是MPLS子网生成的时候，才产生这个缺陷。这个缺陷不是有MPLS OAM状态机自己生成的，而是由网络层通告给MPLS OAM的。
dLOCV	02 01	由于没有收到期望TTSI的CV或者FFD报文造成的缺陷。注意： 如果dLOCV是由于Server层导致的（如收到一个从Server层发来的FDI报文），则DT应该为dServer。 如果dLOCV是由于对等体维护导致的（如收到一个dPeerME的FDI报文），则DT应该为dPeerME。 dLOCV缺陷只是在MPLS层的连通性失效时产生。
dTTSI_Mismatch	02 02	dTTSI_Mismatch缺陷是由于收到携带未预料的和非期望的TTSI的CV或者FFD报文导致的。主要是用来检测错误的连接配置。注意：如果dLOCV和dTTSI_Mismatch同时发生，dTTSI_Mismatch有更高的优先级。
dTTSI_Mismerge	02 03	dTTSI_Mismerge缺陷是由于既收到携带期望TTSI的CV或者FFD报文，又收到携带不期望TTSI的CV或者FFD报文导致的。需要注意不要和dExcess缺陷混淆，因为尽管收到的CV或者FFD速率增加，但是可能不完全是由收到TTSI正确的CV/FFD报文引起的。
dExcess	02 04	dExcess缺陷是由于收到TTSI正确的CV/FFD报文的速率高于正常的配置值导致。这可能是本身的错误融合，源端LSR的错误，拒绝服务攻击（DoS）等原因引起的。
dUnknown	02 FF	dUnkonen缺陷是由于检测到MPLS层的未知缺陷导致的。这个缺陷主要是指MPLS节点内部检测到的并影响数据平面的缺陷。该缺陷不是由MPLS OAM检测的，而是由外部通告给MPLS OAM的。
None	00 00	保留
None	FF FF	保留

进入/退出缺陷的标准

dLOCV缺陷进入标准

当满足以下某个条件时，OAM进入dLOCV缺陷，并且LSP宿端进入近端缺陷状态：

◆ 对于采用CV检测的LSP，如果在任意3倍的检测周期内没有收到TTSI正确的报文的CV报文，则进入dLOCV缺陷。

◆ 对于采用FFD检测的LSP，如果在任意3倍的检测周期x内没有收到TTSI正确的报文的FFD报文，则进入dLOCV缺陷。这里的x可以是宿端手动配置的，也可以是FFD报文中携带的频率值，两者取任一都可以，这里的x必须小于1s。

对于检测到这类缺陷，该如何处理：

如果收到MPLS层的Server层发来的FDI消息，则在正向发送携带DT为dServer的FDI报文并且在反向发送BDI报文，DL也会正确填充，此时在MPLS层不会产生告警；

否则，如果收到MPLS层下层LSP发来的FDI消息，则把收到的FDI报文的DT/DL全部拷贝最新的FDI，然后使用新的FDI向该层LSP的Client层发送并且反向发送BDI报文，此时这条LSP不会产生告警；

否则，如果没有收到任何FDI报文，并且没有收到TTSI错误的CV或者FFD报文，则设置DT为0x 02 01，并发送携带此DT的FDI给宿端以及携带此DT的BDI给源端，DL也会正确填充，此时本地会产生告警信息。

dTTSI_Mismatch缺陷进入标准

当满足以下某个条件时，就会进入dTTST_Mismatch缺陷，并且LSP宿端进入近端缺陷状态：

◆ 如果某条LSP采用CV检测，在三倍检测周期内收到任意TTSI不正确的CV或者FFD报文，并且在相同周期内没有收到TTSI正确的CV报文。

◆ 如果某条LSP采用FFD检测，在三倍检测周期内收到任意TTSI不正确的CV或者FFD报文，并且在相同周期内没有收到TTSI正确的FFD报文。

一定要注意的是，dTTSI_Mismatch缺陷优先级比dLOCV缺陷高。

dTTSI_Mismerge缺陷进入标准

当满足以下某个条件时，就会进入dTTST_Mismerge缺陷，并且LSP宿端进入近端缺陷状态：

◆ 如果某条LSP采用CV检测，在三倍检测周期内收到任意TTSI不正确的CV或者FFD报文，并且在相同周期内收到TTSI正确的CV报文。

◆ 如果某条LSP采用FFD检测，在三倍检测周期内收到任意TTSI不正确的CV或者FFD报文，并且在相同周期内收到TTSI正确的FFD报文。

注意不要混淆dTTSI_Mismerge和dTTSI_Mismatch缺陷，前者是由同时收到TTSI正确和错误的报文引起的，而后者只能在收到的报文全部都是TTSI正确的报文。

dExcess缺陷进入标准

当满足以下某个条件时，就会进入dExcess缺陷，并且LSP宿端进入近端缺陷状态：

◆ 如果某条LSP采用CV检测，在三倍检测周期内收到TTSI正确的CV报文数量>=5。

◆ 如果某条LSP采用FFD检测，在三倍检测周期内收到TTSI正确的FFD报文数量>=5。

缺陷退出标准

当满足以下某个条件时，就会退出上述近端缺陷状态：

◆ 对于CV检测：在三倍检测周期内，收到TTSI正确的CV报文数量>=2且<=4，并且没有收到TTSI错误的CV或者FFD报文。

◆ 对于FFD检测：在三倍检测周期内，收到TTSI正确的FFD报文数量>=2且<=4，并且没有收到TTSI错误的CV或者FFD报文。

当满足以下某个条件时，就会进入和退出远端缺陷：

◆ 第一次收到BDI报文就进入远端缺陷；

◆ 在3s内未收到BDI报文则退出远端缺陷。

MPLS OAM状态机

LSP的缺陷状态一定程度上可以反映网络的使用情况，但是由于缺陷报告的只是在3倍检测周期内很短时间内的状态（CV报文检测为3s，FFD报文检测的周期更短）。如果利用这个短期的状态作为LSP转发层面的稳定性考量，显然是不太合理的，所以提出Available and unavailable 状态。

OAM状态机中定义了以下几个定时器：

T1定时器定义的是在近端缺陷状态到进入Unavailable状态或者退出近端缺陷状态的时间。当LSP进入近端缺陷状态之后，T1定时器启动，当LSP退出近端缺陷或者进入Unavailable状态之后，T1定时器停

止。T1定时器的最大值是10s。

T2定时器定义的是在近端进入Unavailable状态的时间减去10s。当近端进入Unavailable状态，T2定时器启动，当近端退出Unavailable状态10s之后停止。

T3定时器定义的是在远端缺陷状态到进入Unavailable状态或者退出远端缺陷状态的时间。当LSP进入远端缺陷状态之后，T3定时器启动，当LSP退出远端缺陷或者进入Unavailable状态之后，T3定时器停止。T3定时器的最大值是13s。

T4定时器定义的是在远端进入Unavailable状态的时间减去13s。当远端进入Unavailable状态，T4定时器启动，当近端退出Unavailable状态10s之后停止。

Short-Breaks

Short-Breaks指的是进入前面所述的缺陷但是没有进入unavailable状态，并且能在available状态中退出缺陷状态的情况。

Available/unavailable状态机定义

如果LSP的T1定时器超时，这条LSP就会进入unavailable状态，Unavailable状态从进入缺陷状态时算起(+10s)。

如果要重新进入available状态，则需要在第一次退出缺陷状态之后连续的10倍检测周期内之内，满足以下条件：

◆ 收到9~11个TTSI正确的CV/FFD报文。

◆ 没有收到TTSI错误的报文CV/FFD报文。

则从Unavailable状态进入Available状态，Available状态从这10s开始时算起。

MPLS OAM规范和应用

利用OAM的快速检测机制来检测链路或节点失效，达到数据层面的快速切换，使得切换时间最短可控制在50ms内。MPLS TE保护倒换机制正是利用OAM该特性来完成对主LSP进行检测，一旦发现故障，就触发流量从主LSP切换到备用LSP，从而实现业务流量的快速恢复转发，相关详细实现请参考MPLS TE文档。

目前ITU和IETF均对MPLS OAM有相关的规范。在ITU-T的规范中，MPLS OAM主要用于LSP检测和保护。主要规范有下面3个：

◆ ITU-T 1710中定义了用户的主要需求

◆ ITU-T 1711定义了OAM的检测机制和原理

◆ ITU-T 1720定义了保护倒换的原理

IETF在RFC5884中定义了关于BFD检测LSP技术，通过在LSP的Ingress节点和Egress节点之间建立BFD会话，利用BFD快速检测LSP的连通性。在Ingress节点为BFD控制报文压入FEC对应的标签，沿着LSP转发BFD控制报文，并根据收到的Egress节点的BFD控制报文来判断LSP的状态。当BFD检测到LSP故障后，就可以触发LSP进行流量切换。BFD和MPLS OAM都可以实现对指定LSP的检测并结合保护倒换技术完成主备LSP的切换。

VRRP/VRRPE 协议简介

文 / 李朝辉

概述

VRRP全称Virtual Router Redundancy Protocol（虚拟路由器冗余协议）。简单来说，VRRP是一种容错协议，它保证当主机的下一跳路由器坏掉时，可以及时的由另一台路由器来代替，从而保持通讯的连续性和可靠性。为了使VRRP工作，首先要创建一个虚拟IP地址和MAC地址，这样在这个网络中就加入了一个虚拟路由器。而这个网络上的主机与虚拟路由器通信，无需了解这个网络上物理路由器的任何信息。一个虚拟路由器由一个主路由器（Master）和若干个备份路由器（Backup）组成，主路由器实现真正的转发功能。当主路由器出现故障时，备份路由器成为新的主路由器，接替它的工作，如图：

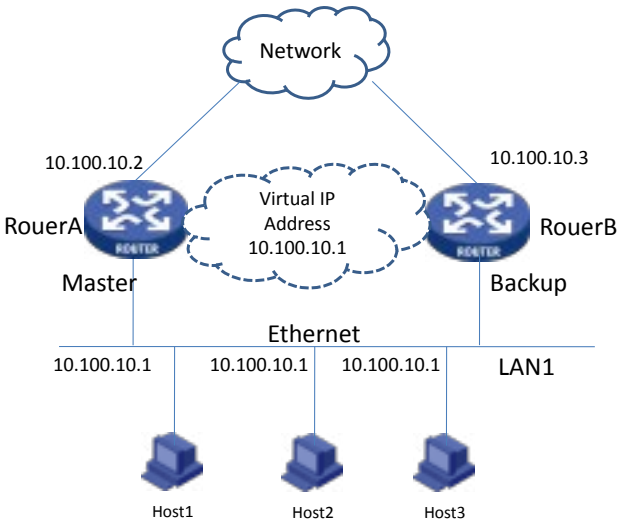


图1 虚拟路由器示意图

VRRP中只定义了一种报文——VRRP报文，这是一种组播报文，由主路由器定时发出来通告它的存在，使用这些报文可以检测虚拟路由器各种参数，还可以用于主路由器的选举。

VRRP中定义了三种状态模型：初始状态（Initialize）、活动状态（Master）、备份状态（Backup）。其中，只有活动状态可以为到虚拟IP地址的转发请求服务。

VRRP协议是在Cisco的私有协议HSRP协议的基础上制定出来的，对应的RFC是2338。 VRRP简化了HSRP提出的机制，尽量减少了由提供冗余功能给网络带来的额外负载，例如HSRP规定虚拟路由器有六种状态，而在VRRP中只有三种；HSRP中有两种状态可以发送报文，且报文类型有三种，而VRRP中只有Master状态的路由器可以发送报文，而且报文也只有一种。HSRP报文是封装在UDP报文上的，而VRRP报文是封装在IP报文上的，支持各种上层协议，同时VRRP还支持将真实接口IP地址设置为虚拟IP地址。

当前VRRP包括VRRPv2（RFC 3768）和VRRPv3（RFC 5798）两个版本。

VRRP的原理

如图1，VRRP（图中是RouterA和RouterB）组织成一个虚拟的路由器，这个虚拟的路由器拥有自己的IP地址10.100.10.1(这个IP地址可以和某个路由器的接口地址相同)。当然，物理路由器RouterA、RouterB也有自己的IP地址（RouterA的IP地址为10.100.10.2， RouterB的IP地址为10.100.10.3）。局域网内的主机仅仅知道这个虚拟路由器的IP地址10.100.10.1，而并不知道具体的路由器RouterA的IP地址10.100.10.2以及路由器RouterB的IP地址10.100.10.3，他们将自己的缺省路由设置为该虚拟路由器的IP地址10.100.10.1。于是，网络内的主机就通过这个虚拟的路由器来与其他网络进行通信。而对于这个虚拟路由器则需要进行如下工作：

1、根据优先级的大小挑选主路由器，优先级最大的为主路由器，若优先级相同，则比较接口的主IP地址，主IP地址大的就成为主路由器，由它提供实际的路由服务。

2、其它路由器作为备份路由器，随时监测主路由器的状态。当主路由器正常工作时，它会每隔一段时间（Advertisement_Interval）发送一个VRRP组播报文，以通知组内的备份路由器，主路由器处于正常工作状态。当组内的备份路由器长时间（Mater_Down_Interval）没有接收到来自主路由器的报文，则将自己转为主路由器。当组内有多台备份路由器时，将有可能产生多个主路由器。这时每一个主路由器就会比较VRRP报文中的优先级(priority)和自己本地的优先级，如果本地的优先级小于VRRP中的优先级，则将自己的状态转为备份路由器，否则保持自己的状态不变。通过这样一个过程，就会将优先级最大的路由器选成新的主路由器，完成VRRP的备份功能。

从上述分析可以看到，对于网络中的主机来说，它并没有做任何额外的工作，但是它对外的通信再也不会因为一台路由器坏掉而受到影响了。

VRRP的特点

作为提供可靠性的容错协议，VRRP具有如下特点：

1、IP地址的备份

这是VRRP最首要和重要的功能，该功能将网络路由黑洞的持续时间最小化，并能够实现负载分担功能。

如图2，路由器A作为备份组1的Master，同时又兼职备份组2的备份路由器，而路由器B正相反，作

为备份组2的Master，并兼职备份组1的备份路由器。一部分主机使用备份组1作网关，另一部分主机使用备份组2作为网关。这样，以达到分担数据流，而又相互备份的目的。

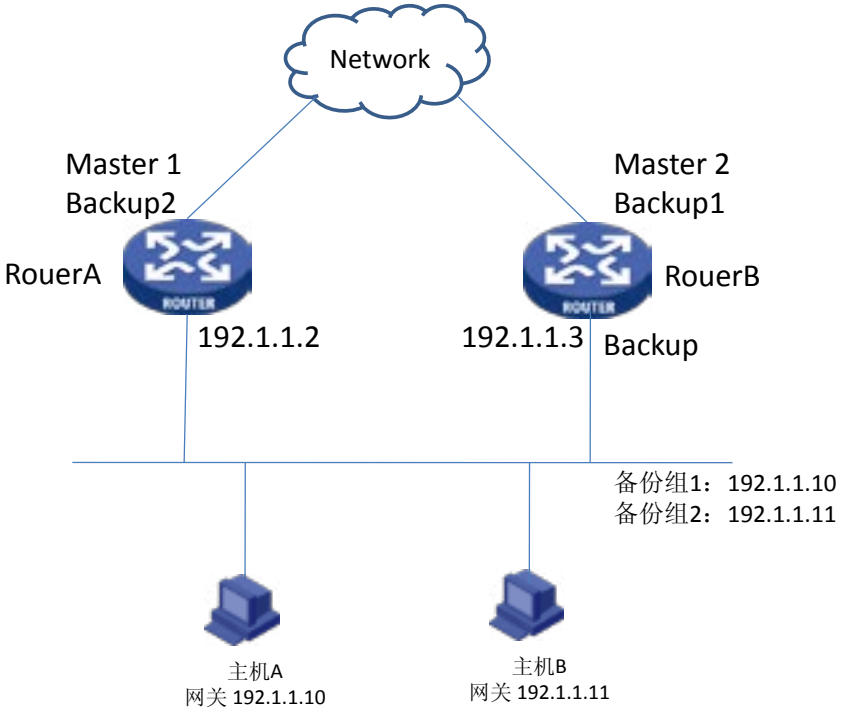


图2 VRRP负载分担

2、首选路径确定

正如前面讲的，VRRP用简单易行的方法从冗余路由器中选取主路由器，这种方法就是设定优先级和抢占方式。根据优先级来确定参与备份组的每台路由器的地位，用户可以通过设定优先级和抢占方式来指定某一路由器成为Master或Backup，达到控制虚拟路由器的行为的目的。

3、使不必要的中断最小化

当主路由器选好后，除了主路由器定时发送的VRRP组播报文，Master和备份路由器之间没有多余的通信。任何优先级低或相等的备份路由器不能发起状态转换，这样主路由器可以持续稳定地工作。

4、可扩展的安全性

对于安全程度不同的网络环境，可以在报头上设定不同的认证方式和认证字。任何没有通过认证的报文将做丢弃处理。

VRRP相关概念

为了更好的理解VRRP，首先介绍下面一些基本概念：

虚拟路由器号（VRID）——范围1～255，由用户配置。有相同VRID的一组路由器构成一个虚拟路由器。

IP地址（IP Address）——一个虚拟路由器可以拥有一个或多个IP地址，由用户配置，如果虚拟IP地址接口与实际地址相同，则该路由器被称为IP地址拥有者(IP address owner)，在同一个VRRP备份组中，只允许配置一个IP地址拥有者。

虚拟MAC地址（Virtual MAC Address）—— 一个虚拟路由器拥有一个虚拟MAC地址，根据RFC2338的规定，虚拟MAC地址的格式为： 00-00-5E-00-01-{VRID}。当虚拟路由器回应ARP请求

时，回应的是虚拟MAC地址，而不是接口的真实MAC地址。

优先级（Priority）—— VRRP中根据优先级来确定参与备份组的每台路由器的地位，备份组中优先级最高的路由器将成为主路由器，当优先级相同时，将会比较接口的主IP地址。优先级的取值范围为0到255 （数值越大表明优先级越高），缺省值是100，但是可配置的范围是1到254。0为系统保留为特殊用途来使用。255则保留给IP地址拥有者。

抢占模式（Preempt_Mode）——一备份组中的路由器有两种工作模式：抢占模式和非抢占模式。

一旦备份组中的某台路由器成为主路由器，只要它没有出现故障，其它路由器即使随后被配置更高的优先级，也不会成为主路由器，除非被设置为抢占方式。路由器如果设置为抢占方式，它一旦发现自己的优先级比当前的活动路由器的优先级高，就会成为主路由器，相应地，原来的主路由器将会变成备份路由器。在设置抢占的同时，还可以设置延迟时间。这样可以使得备份路由器延迟一段时间而不是立即成为主路由器。其目的是：在不够稳定或拥塞的网络环境里，备份路由器可能会因为链路的问题而没有收到主路由器的VRRP报文。如果未设置抢占时间，则备份路由器会立即成为主路由器。如果这时又收到了主路由器的VRRP报文，就会引发频繁的状态转换，造成网络不稳定，延迟时间能够很好的解决这个问题。

注：对于IP地址拥有者，无论如何配置，都将工作在抢占模式。

认证类型（Authentication Type）—— VRRP定义了三种认证方式：无认证（no authentication）、简单字符认证（simple clear text passwords）和 MD5认证 （MD5）。 在一个安全的网络中，可以将认证方式设置为不认证，路由器对要发送的VRRP报文不进行任何认证处理，而收到VRRP报文的路由器也不进行任何认证就认为是一个真实的，合法的VRRP报文，这种情况下，不需要设置认证字。在一个有可能受到安全威胁的网络中，可以将认证方式设置为SIMPLE，则发送VRRP报文的路由器就会将认证字填入到VRRP报文中，而收到的VRRP报文的路由器会将收到的VRRP报文中的认证字和本地配置的认证字进行比较，相同则认为是真实的，合法的VRRP报文，否则认为是一个非法的报文，将会丢弃。 这种情况下，应当设置长度为不超过8位的认证字。 在一个非常不安全的网络中，可以将认证方式设置为MD5，则路由器就会利用Authentication Header提供的认证方式和MD5算法来对VRRP报文进行认证。 这种情况下，应当设置长度为不超过8位的认证字。 对于没有通过认证的报文将做丢弃处理。

VRRP定时器

VRRP定时器分为三种：偏移时间、VRRP通告报文间隔时间定时器、VRRP抢占延迟定时器。

1.偏移时间：

偏移时间（Skew_Time）是用来避免当Master路由器出现故障时，备份组中会有多个Backup路由器在同一时刻同时转变为Master路由器，导致备份组中存在多台Master路由器。 偏移时间的值不可配置，系统根据VRRP不同的版本计算出来一个Skew Time.

VRRPv2：Skew_Time=（256－路由器在备份组中的优先级）/256

VRRPv3：Skew_Time=（（256－路由器在备份组中的优先级）×VRRP通告报文的发送时间间隔）/256

2. VRRP通告报文时间间隔定时器

VRRP备份组中的Master路由器会定时发送VRRP通告报文，通知备份组内的路由器自己工作正常。

用户可以通过设置VRRP定时器来调整Master路由器发送VRRP通告报文的时间间隔。如果Backup路由器在等待了3个间隔时间＋Skew_Time后，依然没有收到VRRP通告报文，则认为自己是Master路由器，并对外发送VRRP通告报文，重新进行Master路由器的选举。

3. VRRP抢占延迟定时器

为了避免备份组内的成员频繁进行主备状态转换，让Backup路由器有足够的时间搜集必要的信息（如路由信息），Backup路由器接收到优先级低于本地优先级的通告报文后，不会立即抢占成为Master，而是等待一定时间——抢占延迟时间＋Skew_Time后，才会对外发送VRRP通告报文取代原来的Master路由器。

VRRP报文结构

VRRP协议只有一种报文，即主路由器定时向其它成员发送的组播报文。当Master正常工作时，它会每隔一段时间(缺省为1秒)发送一个VRRP组播报文，以通知组内的备份路由器，主路由器处于正常工作状态。(注意：只有Master才发送VRRP报文)

VRRP广播报文封装在IP报文中，通过组播方式进行发送。源地址为发送报文的主接口地址(不是虚拟地址或辅助地址)，目的地址是224.0.0.18代表所有VRRP路由器，报文的TTL是255，协议号是112。VRRP广播报文的结构如下图：

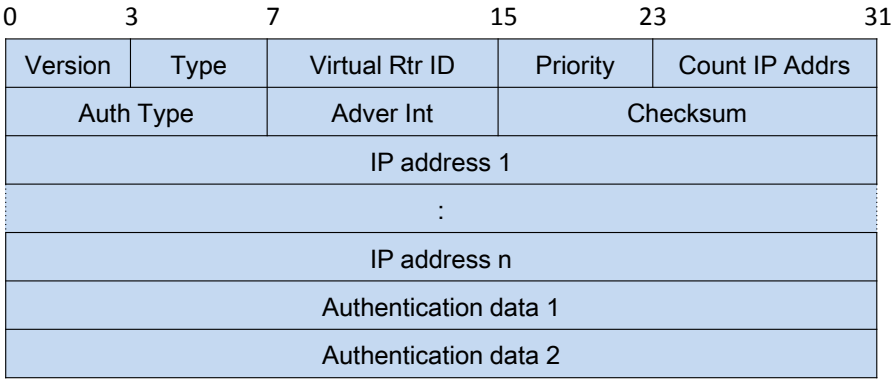


图3 VRRPv2报文结构

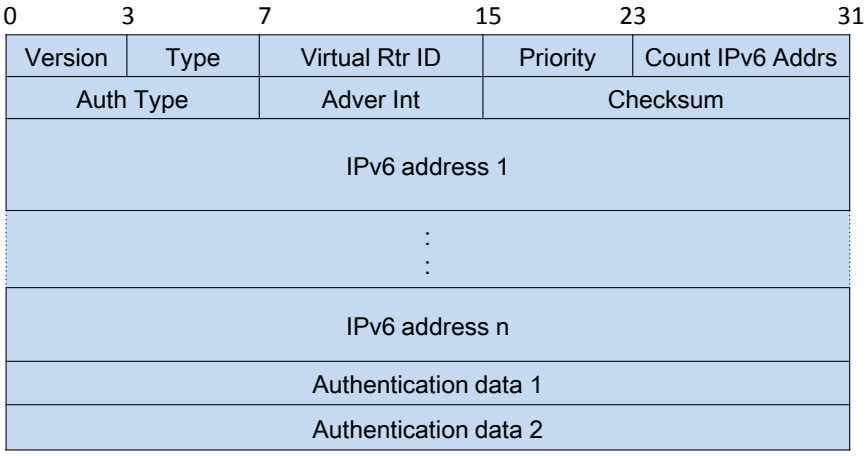


图4 VRRPv3报文结构

各字段的含义：

Version：协议版本号，VRRPv2对应的版本号为2；VRRPv3对应的版本号为3。

Type：VRRP报文的类型。VRRPv2和VRRPv3报文只有一种类型，即VRRP通告报文（Advertisement），该字段取值为1。

Virtual Rtr ID（VRID）：虚拟路由器号（即备份组号），取值范围1～255。

Priority：优先级（0～255），其中0表示路由器停止参与VRRP，用来使备份路由器尽快成为主路由器，而不必等到计时器超时；255则保留给IP地址拥有者，默认值是100，数值越大表明优先级越高

Count IP Addrs/Count IPv6 Addrs：：配置的备份组虚拟地址个数(1个备份组可对应多个虚拟地址)。

Authentication Type:验证类型，协议中指定了3种类型：

0 – No Authentication

1 – Simple Text Password

2 – IP Authentication Header

其中VRRPv3不支持认证。

Advertisement Interval ：发送通告报文的时间间隔。VRRPv2中单位为秒，缺省为1秒；VRRPv3中单位为厘秒，缺省为100厘秒。

Checksum： 16位校验和，用于检测VRRP报文中的数据破坏情况。

IP Address/IPv6 Address：配备份组虚拟IP地址表项。所包含的地址数定义在Count IP Addrs/Count IPv6 Addrs字段。

Authentication Data：：验证字，目前只有明文认证才用到该部分，对于其它认证方式，一律填0。

VRRP的状态机

组成虚拟路由器的路由器会有三种状态机，分别是Initialize、Master和Backup。

1、Initialize

系统启动后进入此状态，当收到接口startup的消息，将转入Backup或Master状态（优先级为255时）。在此状态时，不会对VRRP报文做任何处理。

2、Master

当路由器处于Master状态时，它将会做下列工作：

定期发送VRRP广播报文。响应对虚拟IP地址的ARP请求，并且响应的是虚拟MAC地址，而不是接口的真实MAC地址。转发目的MAC地址为虚拟MAC地址的IP报文。如果它是这个虚拟IP地址的拥有者（IP Address Owner），则接收目的IP地址为这个虚拟IP地址的IP报文。否则，丢弃这个IP报文。在Master状态中只有接收到比自己的优先级大的VRRP报文时，才会转为Backup，当接收到接口的Shutdown事件时，转为Initialize。

3、Backup

当路由器处于BACKUP状态时，它将会做下列工作：

接收Master发送的VRRP广播报文，从中了解Master。对虚拟IP地址的ARP请求，不做响应。丢弃目的MAC地址为虚拟MAC地址的IP报文。丢弃目的IP地址为虚拟IP地址的IP报文。

只有当Backup接收到MASTER_DOWN这个定时器到时的事件时，才会转为Master，而当接收到比自己的优先级小的VRRP报文时，它只是做丢弃这个报文的处理，从而就不对定时器做重置处理，这样

定时器就会在若干次这样的处理之后到时，于是就转为Master。当接收到接口的Shutdown事件时，转为Initialize。

三种状态机的转换如下图：

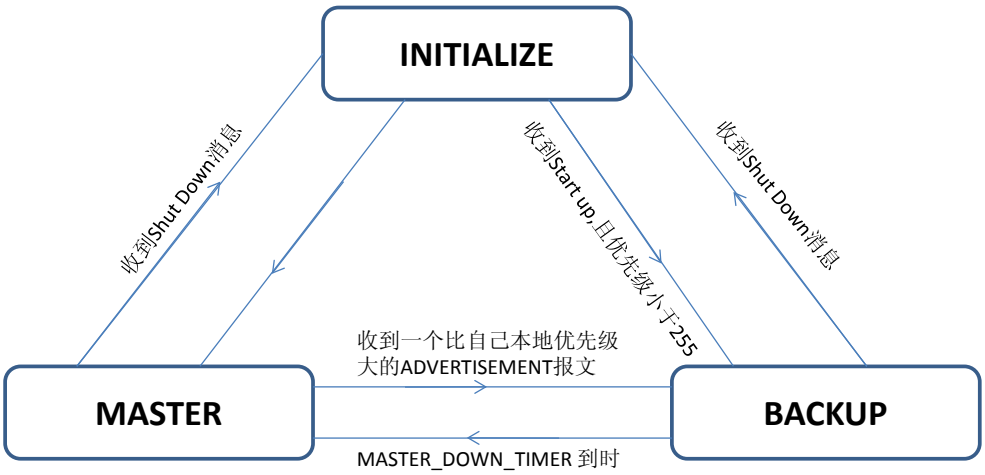


图5 VRRP状态机的转换

VRRP负载均衡功能（VRRPE）

在第3节VRRP的特点中我们讲了通过创建多个备份组可以使得一个路由器在一个备份组中作为Master路由器，在其他的备份组中作为Backup路由器，从而实现负载分担功能。虽然创建多个备份组可以实现多个路由器之间的负载分担，但是局域网内的主机需要设置不同的网关，增加了配置的复杂性。为解决这个问题，提出了VRRP负载均衡模式（也称为VRRPE功能）。

在VRRP负载均衡模式下VRRP不仅提供了虚拟网关冗余备份功能，同时增加了负载均衡功能。其实现原理为：将一个虚拟IP地址与多个虚拟MAC地址对应，VRRP备份组中的每个路由器都对应一个虚拟MAC地址；使用不同的虚拟MAC地址应答主机的ARP（IPv4网络中）/ND（IPv6网络中）请求，从而使得不同主机的流量发送到不同的路由器，备份组中的每个路由器都能转发流量。在VRRP负载均衡模式中，只需创建一个备份组，就可以实现备份组中多个路由器之间的负载分担，避免了VRRP备份组中Backup路由器始终处于空闲状态、网络资源利用率不高的问题。

虚拟MAC地址的分配

VRRP负载均衡模式中，Master路由器负责为备份组中的路由器分配不同的虚拟MAC地址，并为来自不同主机的ARP/ND请求，应答不同的虚拟MAC地址，从而实现流量在多个路由器之间分担。备份组中的Backup路由器不会应答主机的ARP/ND请求。

以IPv4网络为例，VRRP负载均衡模式的具体工作过程为：

(1) Master为备份组中的路由器（包括Master自身）分配虚拟MAC地址。如下图所示，虚拟IP地址为10.1.1.1/24的备份组中，Router A作为Master，Router B作为Backup。Router A为自己分配的虚拟MAC地址为000f-e2ff-0011，为Router B分配的虚拟MAC地址为000f-e2ff-0012。

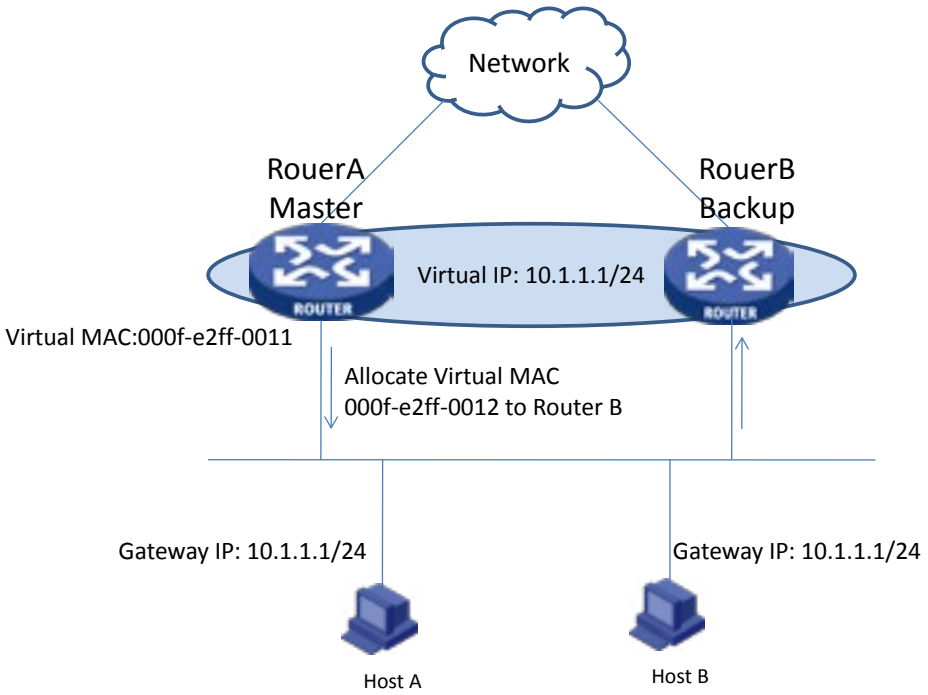


图6 Master分配虚拟MAC地址

(2) Master接收到主机发送的目标IP地址为虚拟IP地址的ARP请求后，根据负载均衡算法（依据主机的MAC地址进行hash计算，确定出相应的负责转发的路由器），使用不同的虚拟MAC地址应答主机的ARP请求。如下图所示，Host A发送ARP请求获取网关10.1.1.1对应的MAC地址时，Master（即Router A）使用Router A的虚拟MAC地址应答该请求；Host B发送ARP请求获取网关10.1.1.1对应的MAC地址时，Master使用Router B的虚拟MAC地址应答该请求。

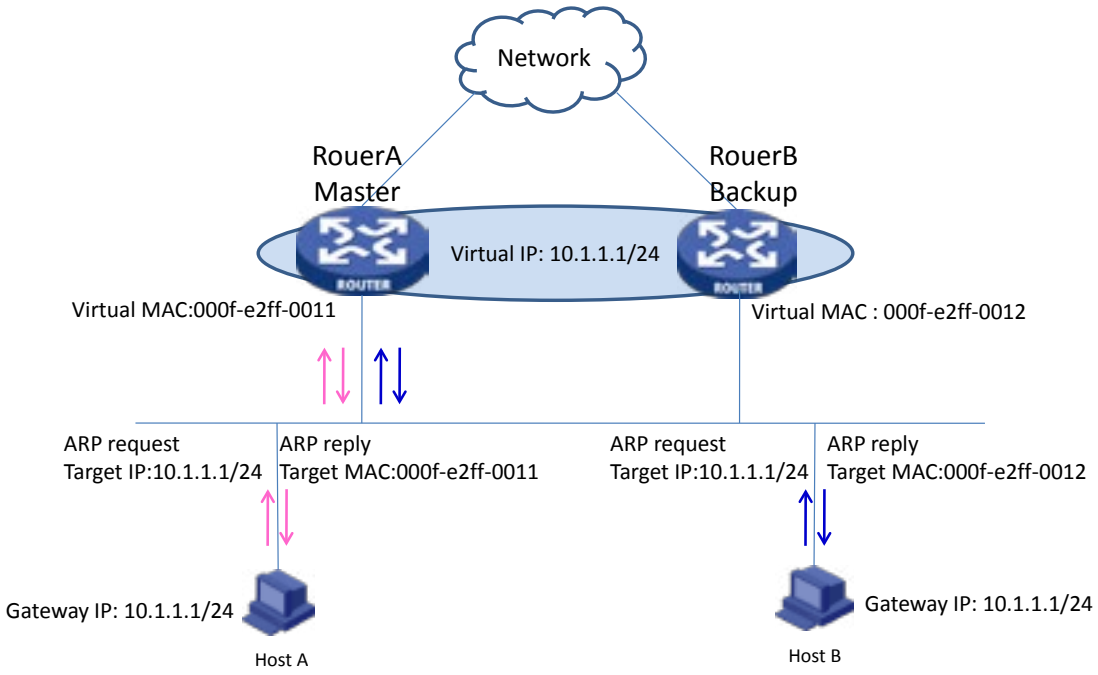


图7 Master应答ARP请求

(3) 通过使用不同的虚拟MAC地址应答主机的ARP请求，可以实现不同主机的流量发送给不同的路由器。如下图所示，Host A认为网关的MAC地址为Router A的虚拟MAC地址，从而保证Host A的流量通过Router A转发；Host B认为网关的MAC地址为Router B的虚拟MAC地址，从而保证Host B的流量通过Router B转发。

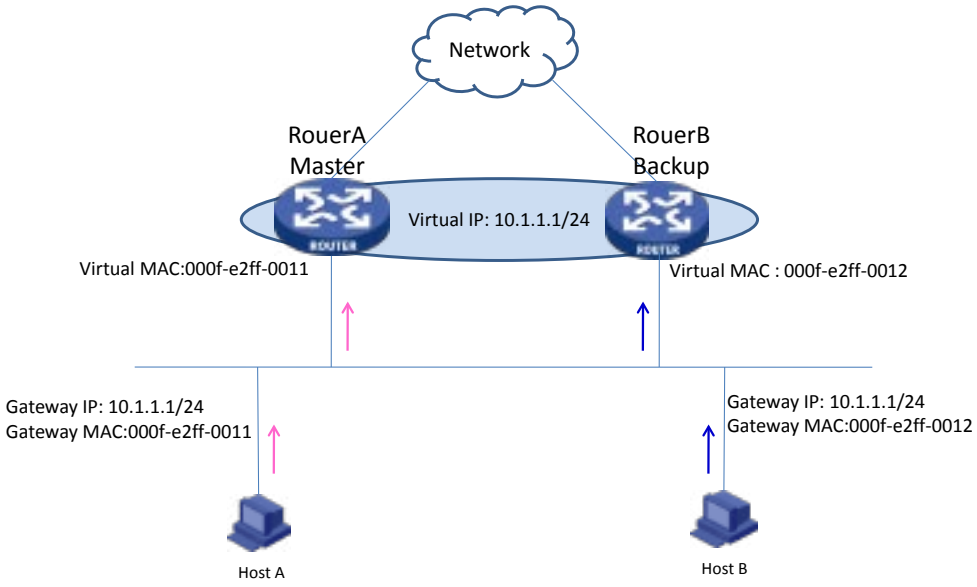


图8 主机通过不同路由器转发流量

虚拟转发器

◆ AVF：虚拟转发器(Active Virtual Forwarder)，作为AVF负责转发目的MAC地址为虚拟MAC地址的流量。

◆ LVF：备用虚拟转发器(Listening Virtual Forwarder)，LVF监视AVF的状态，当AVF出现故障时，LVF将选举出优先级最高的虚拟转发器作为AVF。

◆ VF Owner：虚拟转发器拥有者(Virtual Forwarder Owner)，通过Request-Reply方式获取虚MAC的AVF设备称为VF owner，AVF不一定就是VF owner，AVF也有可能是VF owner出现故障时，临时替代进行转发工作的LVF。

◆ VF Owner ID：虚拟转发器拥有者标志(Virtual Forwarder Owner Identifier)；当前用接口的MAC地址标志；

虚拟转发器的创建

通过Master路由器给备份组中各路由器分配虚拟MAC地址，和响应ARP请求，给不同主机分配置不同的网关MAC，实现了不同主机将流量发送给备份组中不同的路由器。为了使备份组中的路由器能够转发主机发送的流量，需要在路由器上创建虚拟转发器。每个虚拟转发器都对应备份组的一个虚拟MAC地址，负责转发目的MAC地址为该虚拟MAC地址的流量。

虚拟转发器的创建过程为：

(1) 备份组中的路由器获取到Master为其分配的虚拟MAC地址后，创建该MAC地址对应的虚拟转发器，该路由器称为此虚拟MAC地址对应虚拟转发器的VF Owner。

(2) 该路由器将虚拟转发器的信息通告给备份组内其他的路由器。

(3) 备份组内的路由器接收到虚拟转发器信息后，在本地创建该虚拟MAC地址对应的虚拟转发器。

由此可见，备份组中的路由器上不仅需要创建Master为其分配的虚拟MAC地址对应的虚拟转发器，还需要创建其他路由器通告的虚拟MAC地址对应的虚拟转发器，对其它的路由器进行备份，当其它的路由器down后系统根据相应的权重和优先级进行转发。

虚拟转发器的权重和优先级

虚拟转发器的权重标识了路由器的转发能力。权重值越高，路由器的转发能力越强。当权重低于一定的值（失效下限），路由器无法再为主机转发流量。

虚拟转发器的优先级用来决定虚拟转发器的状态：不同路由器上同一个虚拟MAC地址对应的虚拟转发器中，优先级最高的虚拟转发器处于Active状态，称为AVF，负责转发流量；其他虚拟转发器处于Listening状态，称为LVF，监听AVF的状态。虚拟转发器的优先级取值范围为0～255，其中，255保留给VF Owner使用。如果VF Owner的权重高于或等于失效下限，则VF Owner的优先级为最高值255。

设备根据虚拟转发器的权重计算虚拟转发器的优先级，如果权重高于或等于失效下限，且设备为VF Owner，则虚拟转发器的优先级为最高值255；如果权重高于或等于失效下限，且设备不是VF Owner，则虚拟转发器的优先级为权重/（本地AVF的数目＋1）；如果权重低于失效下限，则虚拟转发器的优先级为0。

虚拟转发器备份

备份组中不同路由器上同一个虚拟MAC地址对应的虚拟转发器之间形成备份关系。

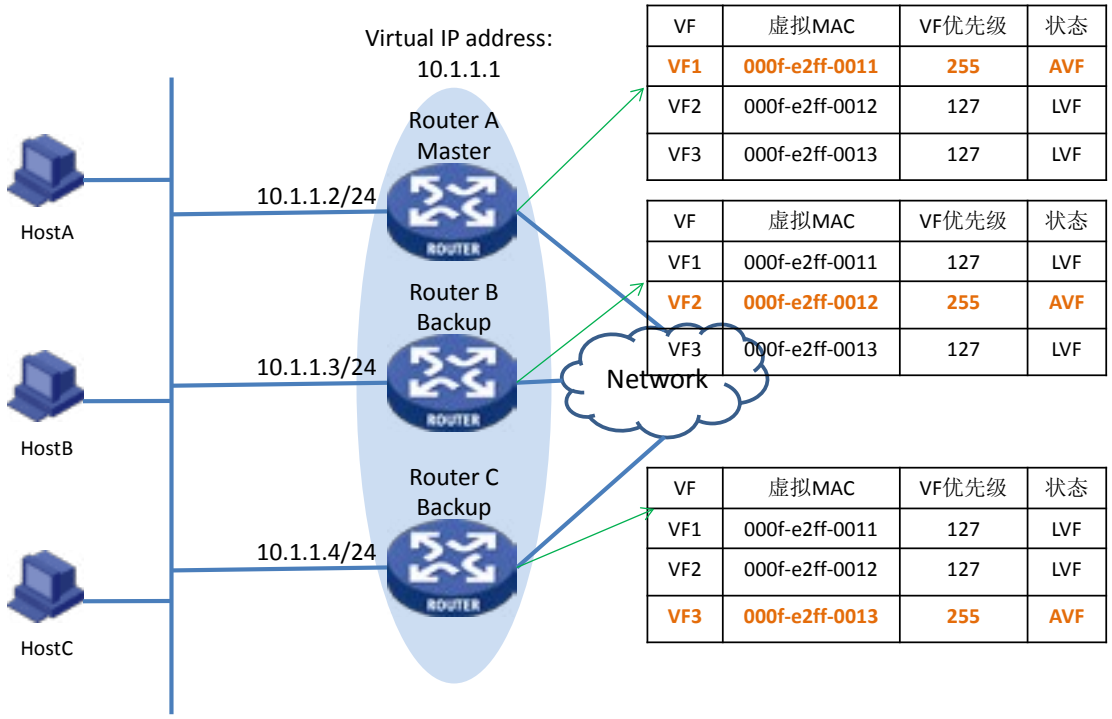


图9 虚拟转发器

上图举例说明了备份组中每个路由器上的虚拟转发器信息及其备份关系。Master路由器Router A为自己、Router B、Router C分配的虚拟MAC地址分别为000f-e2ff-0011、000f-e2ff-0012和000f-e2ff-0013。虚拟MAC地址对应的虚拟转发器分别为VF 1、VF 2和VF 3。在Router A、Router B和Router C上都创建了这三个虚拟转发器，并形成备份关系。例如，Router A、Router B和Router C上的VF 1互相备份：

1. Router A为VF 1的VF Owner，Router A上VF 1的虚拟转发器优先级为最高值255。因此，Router A上的VF 1作为AVF，负责转发目的MAC地址为虚拟MAC地址000f-e2ff-0011的流量。

2. Router B和Router C上VF 1的虚拟转发器优先级为：权重255/（本地AVF数目1＋1）＝127，低于Router A上VF 1的优先级。因此，Router B和Router C上的VF 1作为LVF，监视Router A上VF 1的状态。

3. 当Router A上的VF 1出现故障时，将从Router B和Router C上的VF 1中选举出虚拟转发器优先级最高的LVF作为AVF，负责转发目的MAC地址为虚拟MAC地址000f-e2ff-0011的流量。

实际设备上虚拟转发器的的情况，可以作为VF1的转发着同时还监扣兼顾VF2和VF3的运行情况：

```
[S58V2_IRF]dis vrrp inter vlan 200
IPv4 Virtual Router Information:
 Running Mode      : Load Balance
 Total number of virtual routers on interface Vlan-interface200 : 1
 Interface         VRID  State     Running Pri  Address     Active
 ------------------------------------------------------------
 Vlan200           1     Master    100          10.1.1.1    Local
 -----             VF1   Active    255          000f-e2ff-0011   Local
 -----             VF2   Listening 127          000f-e2ff-0012   10.1.1.3
 -----             VF3   Listening 127          000f-e2ff-0013   10.1.1.4
```

虚拟转发器的定时器

虚拟转发器的AVF出现故障后，接替其工作的新的AVF将为该VF创建Redirect Timer和Timeout Timer两个定时器。

Redirect Timer：VF重定向定时器。该定时器超时前，Master路由器还会采用该VF对应的虚拟MAC地址应答主机的ARP/ND请求；该定时器超时后，Master路由器不再采用该VF对应的虚拟MAC地址应答主机的ARP/ND请求。如果VF Owner在Redirect Timer超时前恢复，则VF Owner可以迅速参与流量的负载分担。

Timeout Timer：VF生存定时器，即AVF接替VF Owner工作的期限。该定时器超时前，备份组中的路由器上都保留该VF，AVF负责转发目的MAC地址为该VF对应虚拟MAC地址的报文；该定时器超时后，备份组中的路由器上都删除该VF，不再转发目的MAC地址为该VF对应虚拟MAC地址的报文。

虚拟转发器监视功能

AVF负责转发目的MAC地址为虚拟转发器MAC地址的流量，当AVF连接的上行链路出现故障时，如果不能及时通知LVF接替其工作，局域网中以此虚拟转发器MAC地址为网关MAC地址的主机将无法访问外部网络。

虚拟转发器的监视功能可以解决上述问题。利用NQA（Network Quality Analyzer，网络质量分

析）、BFD（Bidirectional Forwarding Detection，双向转发检测）等监测AVF连接的上行链路的状态，并通过Track功能在虚拟转发器和NQA/BFD之间建立联动。当上行链路出现故障，Track项的状态变为Negative，虚拟转发器的权重将降低指定的数额，以便虚拟转发器优先级更高的路由器抢占成为AVF，接替其转发流量。

虚拟转发器监视功能还可以用来在LVF上通过Track监视AVF的状态，当AVF出现故障时，工作在虚拟转发器快速切换模式的LVF能够迅速成为AVF，以保证通信不会中断。

VRRP负载均衡模式的报文

VRRP标准协议模式中只定义了一种报文——VRRP通告报文，且只有Master路由器周期性发送该报文，Backup路由器不会发送VRRP通告报文。

为了实现负载均衡，VRRP负载均衡模式中定义了四种报文：

Advertisement：不仅用于通告本路由器上备份组的状态，还用于通告本路由器上处于Active状态的虚拟转发器信息。Master和Backup路由器均周期性发送该报文。

Request：处于Backup状态的路由器如果不是VF Owner，则发送Request报文，请求Master路由器为其分配虚拟MAC地址。

Reply：Master路由器接收到Request报文后，将通过Reply报文为Backup路由器分配虚拟MAC地址。收到Reply报文后，Backup路由器会创建虚拟MAC地址对应的虚拟转发器，该路由器称为此虚拟转发器的拥有者。

Release：VF Owner的失效时间达到一定值后，接替其工作的路由器将发送Release报文，通知备份组中的路由器删除VF Owner对应的虚拟转发器。

这几个报文的格式与第6节中的报文格式基本一样，只是在其基础上增加了选项字段，用来携带实现负载均衡所需要的信息。

VRRP与Track联动

VRRP监视功能通过NQA（Network Quality Analyzer，网络质量分析）、BFD（Bidirectional Forwarding Detection，双向转发检测）等监测Master路由器和上行链路的状态，并通过Track功能在VRRP设备状态和NQA/BFD之间建立关联：

监视上行链路，根据上行链路的状态，改变路由器的优先级。当Master路由器的上行链路出现故障，局域网内的主机无法通过网关访问外部网络时，被监视Track项的状态变为Negative，Master路由器的优先级降低指定的数值。使得当前的Master路由器不是组内优先级最高的路由器，而其它路由器成为Master路由器，保证局域网内主机与外部网络的通信不会中断。

在Backup路由器上监视Master路由器的状态。当Master路由器出现故障时，监视Master路由器状态的Backup路由器能够迅速成为Master路由器，以保证通信不会中断。

被监视Track项的状态由Negative变为Positive或Notready后，对应的路由器优先级会自动恢复。

一些常见问题的解决方法

ICMP重定向问题

ICMP重定向会导致将路由器的真实接口IP地址通过重定向报文通知给网络内的主机，VRRP要避免这种情况发生，遵循以下的原则：

1、如果重定向报文中的目的地址的下一跳是一个备份组的MASTER，就将这个备份组的虚拟IP地址填充到ICMP重定向报文中的下一跳。

2、如果重定向报文中的目的地址地下一跳是一个备份组的BACKUP，就不发送ICMP重定向报文中。

3、如果重定向报文中的目的地址地下一跳是一个没有运行VRRP的路由器，就将这个路由器的真实接口IP地址填充到ICMP重定向报文中的下一跳。

Host ARP Requests

当主机发出对虚拟路由器IP地址的请求时，Master路由器必须以虚拟MAC地址回应，这样，无论哪个路由器是Master，客户机始终能够使用同一个MAC地址。

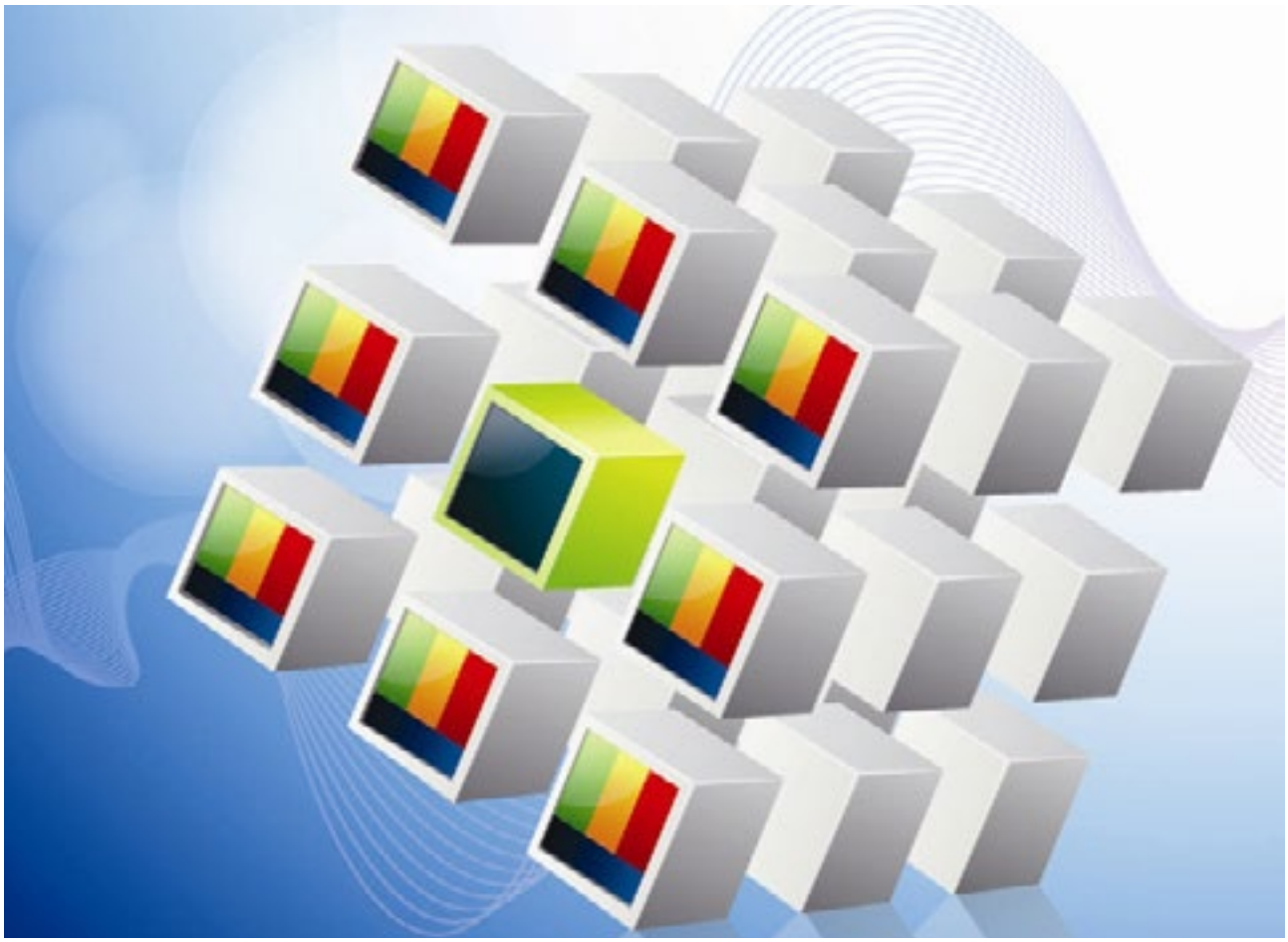
当一个VRRP路由器启动或重新启动时，它不能用自己的MAC地址来回应针对它的IP地址的ARP请求，而只能用虚拟MAC地址来回应。

虚拟IP地址应不应该能够ping通？

标准并没有规定虚拟IP地址应不应该ping通。目前在我司和C公司的设备上都提供了切换ping通虚拟IP地址的开关命令。RFC2338没有对这一行为作出规定，如果能够ping通虚拟IP地址，可以比较方便的监控虚拟路由器的工作情况，但是这样也带来了可能遭到ICMP攻击的隐患，同时也不能分辨当前虚拟路由器中是否存在IP地址拥有者。而如果虚拟IP地址不能ping通，会给监控虚拟路由器的工作情况带来一定的麻烦。两种实现方法应该说是各有利弊。

VRRP广播报文的TTL为何是255？

VRRP广播报文是用来在备份组中路由器之间维护Master/Backup关系的，只在本网段进行传播，不能被路由器转发。根据这些条件，VRRP广播报文的TTL设成1就可以了。但是，RFC2338规定VRRP广播报文的TTL必须是255，TTL不等于255的报文将被丢弃。这种设计主要是从安全方面考虑的。VRRP可以通过明文或MD5认证来保证安全，但是无论哪种认证方式都会消耗一些资源。在本网段的安全可以得到保证的情况下，攻击主要来自其它的网段，将TTL设定为255而不必使用认证就可以轻松应对这样的攻击。因为来自其它网段的报文要经过路由器转发，TTL必然小于255。



IRF MAD应用模型及技术分析

文 / 叶玉其

IRF概述

IRF（Intelligent Resilient Framework，智能弹性架构）是H3C自主研发的软件虚拟化技术。它的核心思想是将多台设备连接在一起，进行必要的配置后，虚拟化成一台设备。使用这种虚拟化技术可以集合多台设备的硬件资源和软件处理能力，实现多台设备的协同工作、统一管理和不间断维护。

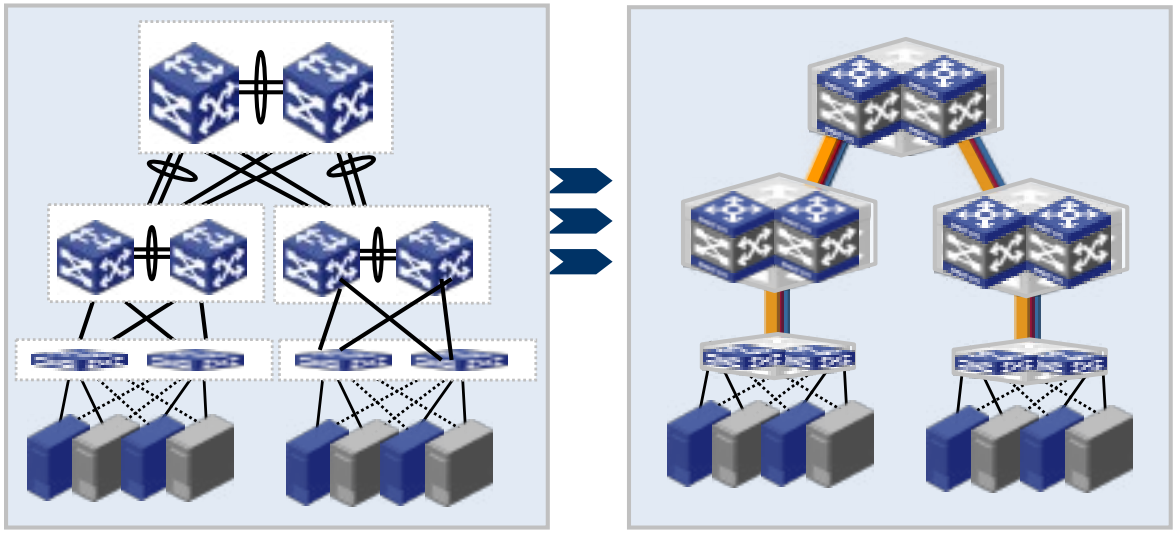


图1 IRF组网应用示意图

IRF的优点

IRF主要具有以下优点：

◆ 简化管理。IRF形成之后，用户通过任意成员设备的任意端口都可以登录IRF系统，对IRF内所有成员设备进行统一管理。

◆ 1:N备份。IRF由多台成员设备组成，其中，Master设备负责IRF的运行、管理和维护，Slave设备在作为备份的同时也可以处理业务。一旦Master设备故障，系统会迅速自动选举新的Master，以保证业务不中断，从而实现了设备的1:N备份。

◆ 跨成员设备的链路聚合。IRF和上、下层设备之间的物理链路支持聚合功能，并且不同成员设备上的物理链路可以聚合成一个逻辑链路，多条物理链路之间可以互为备份也可以进行负载分担，当某个成员设备离开IRF，其它成员设备上的链路仍能收发报文，从而，提高了聚合链路的可靠性。

◆ 强大的网络扩展能力。通过增加成员设备，可以轻松自如的扩展IRF的端口数、带宽。因为各成员设备都有CPU，能够独立处理协议报文、进行报文转发，所以IRF还能够轻松自如的扩展处理能力。

多IRF冲突检测（MAD功能）

机制介绍

IRF链路故障会导致一个IRF变成多个新的IRF。这些IRF拥有相同的IP地址等三层配置，会引起地址冲突，导致故障在网络中扩大。为了提高系统的可用性，当IRF分裂时我们就需要一种机制，能够检测出网络中同时存在多个IRF，并进行相应的处理尽量降低IRF分裂对业务的影响。MAD（Multi-Active Detection，多Active检测）就是这样一种检测和处理机制。它主要提供以下功能：

(1)分裂检测

通过ARP（Address Resolution Protocol）、ND（Neighbor Discovery Protocol）、LACP（Link Aggregation Control Protocol，链路聚合控制协议）或者BFD（Bidirectional Forwarding Detection，双向转发检测）来检测网络中是否存在多个IRF。

(2)冲突处理

IRF分裂后，通过分裂检测机制IRF会检测到网络中存在其它处于Active状态（表示IRF处于正常工作状态）的IRF。

◆ 对于BFD MAD/ ARP MAD/ND MAD检测，冲突处理会直接让Master成员编号小的IRF处于Active状态，继续正常工作；其它IRF迁移到Recovery状态（即禁用状态）。

◆ 对于LACP MAD检测，冲突处理会先比较两个IRF中成员设备的数量，数量多的IRF处于Active状态，继续工作；数量少的迁移到Recovery状态（即禁用状态）；如果成员数量相等，则Master成员编号小的IRF处于Active状态，继续正常工作；其它IRF迁移到Recovery状态（即禁用状态）。

IRF迁移到Recovery状态后会关闭该IRF中所有成员设备上除保留端口以外的其它所有物理端口（通常为业务接口），以保证该IRF不能再转发业务报文。缺省情况下，只有IRF链路物理端口是保留端口，用户也可以通过mad exclude interface命令行将其它端口设置为保留端口。

(3)MAD故障恢复

IRF链路故障导致IRF分裂，从而引起多Active冲突。因此修复故障的IRF链路，让冲突的IRF重新合并为一个IRF，就能恢复MAD故障。如果在MAD故障恢复前，处于Recovery状态的IRF也出现了故

障，则需要将故障IRF和故障链路都修复后，才能让冲突的IRF重新合并为一个IRF，恢复MAD故障；如果在MAD故障恢复前，处于Active状态的IRF出现其他故障，则可以通过命令行先启用Recovery状态的IRF，让它接替原IRF工作，以便保证业务尽量少受影响，再恢复MAD故障。

原理介绍

IRF支持的MAD检测方式有：LACP MAD检测、BFD MAD检测、ARP MAD检测和ND MAD检测。下面针对这四种MAD检测方式进行逐一阐述：

ARP MAD检测

(一)ARP MAD检测原理

ARP MAD检测是通过扩展ARP协议报文内容实现的，即将ARP协议报文中“Target MAC Address”字段扩展为IRF的DomainID（域编号）、对端ActiveID（即对端Master的成员编号）、自身ActiveID（自身Master的成员编号）和PktType（检测包类型）（图2）。其中检测包类型（PktType）包括如下三种类型：

a) 0x00，Hello报文，载荷为自身ActiveID

b) 0x01，Alive检测报文，载荷为自身ActiveID和对端ActiveID

c) 0x02，Alive确认报文，载荷为自身ActiveID和对端ActiveID

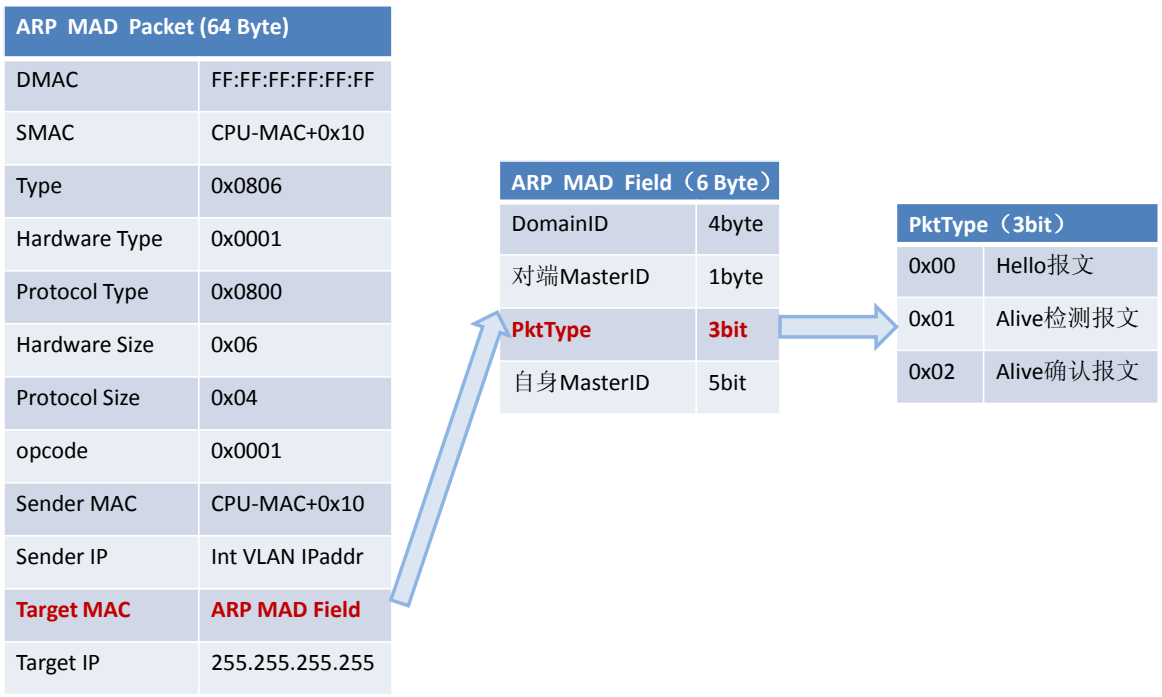


图2 ARP MAD PACKET

使能ARP MAD检测后，成员设备可以通过ARP协议报文和其它成员设备交互DomainID和ActiveID信息。

◆ 当成员设备收到ARP协议报文后，先比较DomainID。如果DomainID相同，再比较ActiveID；如果DomainID不同，则认为报文来自不同IRF，不再进行MAD处理。

◆ 如果ActiveID相同，则表示IRF正常运行，没有发生多Active冲突；如果ActiveID值不同，则表

示IRF分裂，进行ARP协议报文快速交互确认，检测到多Active冲突。

◆ 检测到多Active冲突后，会直接让Master成员编号小的IRF处于Active状态，继续正常工作；其它IRF上报MAD冲突事件给IRF模块，IRF模块将该IRF迁移到MAD Recovery状态。

其中IRF分裂后，进行ARP协议报文快速交互确认状态机和流程如下（图3）：

a) 初始情况下，IRF使能ARP MAD功能后，各成员设备ARP MAD状态机迁移到ARP_MAD_STS_WAIT_Hello，IRF内所有设备从ARP MAD VLAN对应的接口发送ARP MAD Hello报文（由于ARP MAD需要使能STP功能，相应端口会被STP Blocking，各成员端口发送的ARP MAD Hello报文无法相互接收到）。

b) 当IRF分裂后，各IRF都重新选择了Master，使用新Master的MAC地址封装STP报文，各IRF之间专门用于ARP MAD检测的链路STP状态迁移到Forwarding状态，ARP MAD模块可以收到对端IRF发送的MAD Hello报文，ARP MAD状态机迁移到ARP_MAD_STS_WAIT_ALIVEACK，并发送ARP MAD Alive检测报文。

c) 当收到ARP MAD Alive检测报文时，如果当前状态为ARP_MAD_STS_WAIT_Hello，则迁移到ARP_MAD_STS_WAIT_ALIVEACK，并发送ARP MAD Alive检测报文；否则发送ARP MAD Alive确认报文。

d) 如果收到ARP MAD Alive确认报文，且当前为ARP_MAD_STS_WAIT_ALIVEACK状态，则开始ARP MAD竞选。

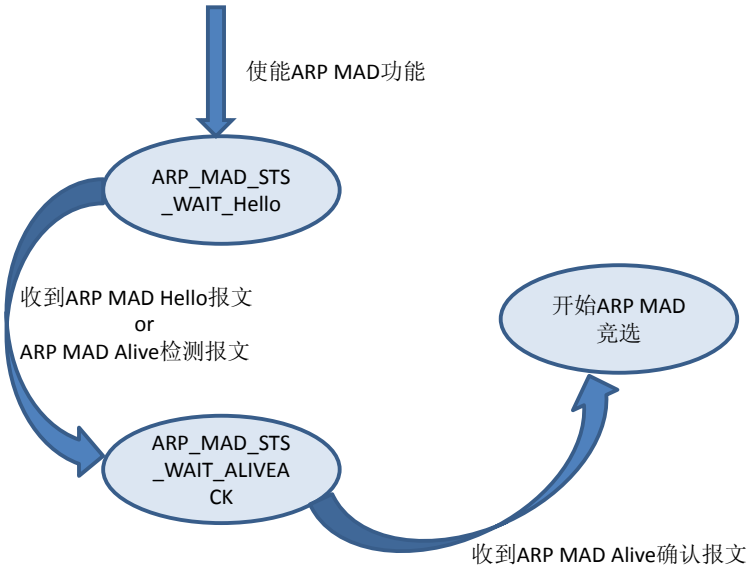


图3 ARP MAD检测状态机

(二)ARP MAD检测组网要求

ARP MAD检测方式可以使用中间设备来进行连接（图4），也可以不使用中间设备（图5）。不管是否使用中间设备来连接，都需要在IRF设备和中间设备上配置生成树功能，以防止形成环路。同时还需要在使能ARP MAD检测的IRF设备上配置IRF桥MAC不保留，这样才能触发IRF分裂后，STP状态快速切换，ARP MAD检测快速生效。

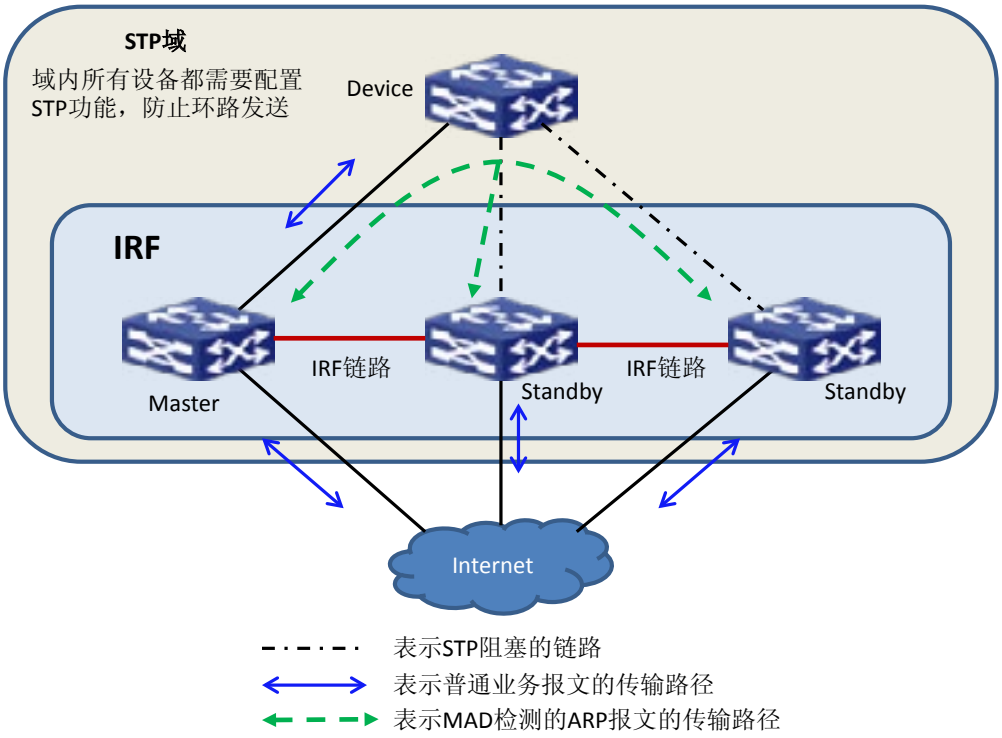


图4 ARP MAD检测组网示意图一

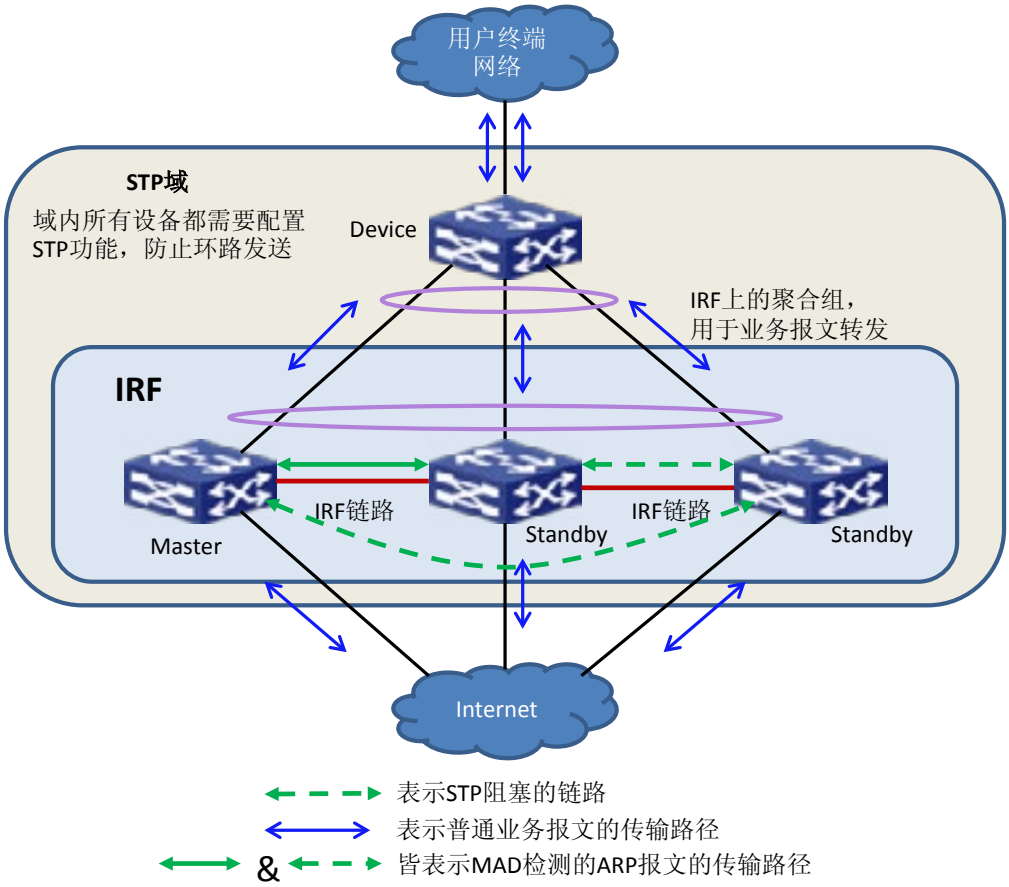


图5 ARP MAD检测组网示意图二

ND MAD检测

ND MAD检测是通过扩展ND协议报文内容实现的，即使用ND的NS协议报文携带扩展TLV选项数据来交互IRF的DomainID和ActiveID。ND MAD检测原理及组网要求皆和ARP MAD检测类似，请参考ARP MAD检测部分。

BFD MAD检测

(一)BFD MAD检测原理

BFD MAD检测是通过BFD协议来实现的。要使BFD MAD检测功能正常运行，除在三层接口下使能BFD MAD检测功能外，还需要在该接口上配置MAD IP地址。MAD IP地址与普通IP地址不同的地方在于：MAD IP地址与成员设备是绑定的，IRF中的每个成员设备上都需要配置，且所有成员设备的MAD IP必须属于同一网段。

◆ 当IRF正常运行时，只有Master上配置的MAD IP地址生效，Slave设备上配置的MAD IP地址不生效，BFD会话处于down状态；（使用display bfd session命令查看BFD会话的状态。如果Session State显示为Up，则表示激活状态；如果显示为Down，则表示处于down状态）

◆ 当IRF分裂形成多个IRF时，不同IRF中Master上配置的MAD IP地址均会生效，BFD会话被激活，此时会检测到多Active冲突。

◆ 检测到多Active冲突后，会直接让Master成员编号小的IRF处于Active状态，继续正常工作；其它IRF上报MAD冲突事件给IRF模块，IRF模块将该IRF迁移到MAD Recovery状态。

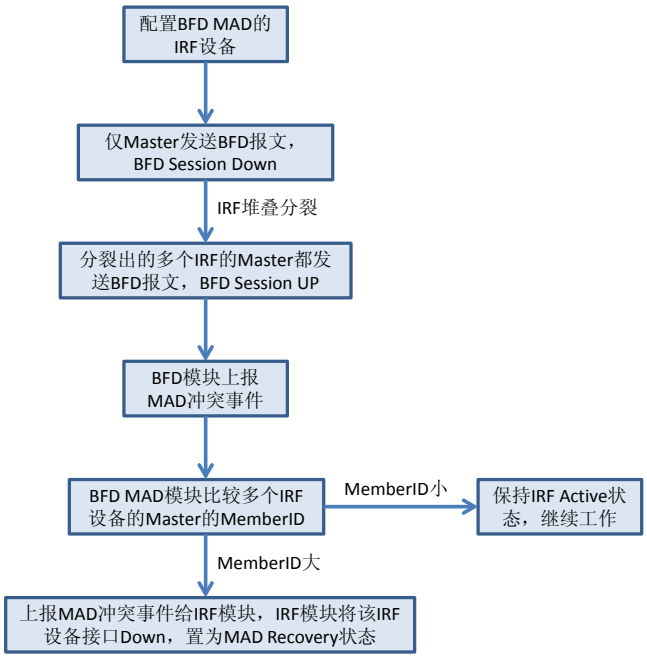


图6 BFD MAD交互流程

(二)BFD MAD检测组网要求

BFD MAD检测方式需要使用中间设备（图7），每个成员设备都需要连接到中间设备，这些BFD链路专用于MAD检测。这些链路连接的接口必须属于同一VLAN，在该VLAN接口视图下给不同成员设备配置同一网段下的不同IP地址。

在用于BFD MAD检测的接口下必须使用mad ip address命令配置MAD IP地址，而不要配置其它IP地址（包括使用ip address命令配置的普通IP地址、VRRP虚拟IP地址等），以免影响MAD检测功能。

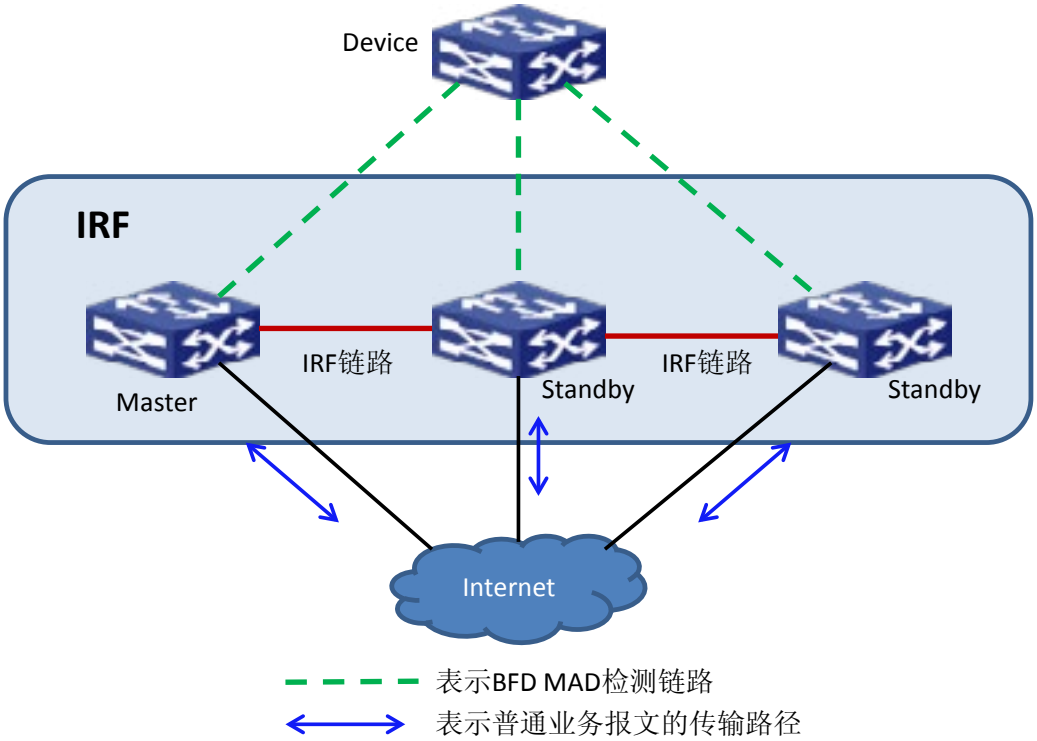


图7 BFD MAD检测组网示意图

LACP MAD检测

(一)LACP MAD检测原理

LACP MAD检测是通过扩展LACP协议报文内容实现的，即在LACP协议报文的扩展字段内定义一个新的TLV（Type/Length/Value，类型/长度/值）数据域——用于交互IRF的DomainID（域编号）、ActiveMemNum（当前IRF的成员数目）和ActiveID（等于Master的成员编号）。使能LACP MAD检测后，成员设备通过LACP协议报文和其它成员设备交互DomainID、ActiveMemNum和ActiveID信息。

使能LACP MAD检测后，聚合成员端口周期性地发送带有扩展TLV字段的LACP报文（缺省发送周期为30s，该LACP报文和动态LACP报文分开发送），对端设备（中间设备）收到带扩展TLV的LACP报文后，会从聚合组内除接收端口外的所有其他成员端口各转发一份。成员设备通过LACP协议报文和其它成员设备交互DomainID、ActiveMemNum和ActiveID信息。

◆ 当成员设备收到LACP MAD报文后，先比较DomainID。如果DomainID相同，再比较ActiveID；如果DomainID不同，则认为报文来自不同IRF，不再进行MAD处理，作为中间设备，仍然需要从聚合组内除接收端口外的所有其他成员端口各转发一份。

◆ 如果ActiveID相同，则表示IRF正常运行，没有发生多Active冲突；如果ActiveID值不同，快速进行LACP报文交互、确认冲突，确认后表示IRF分裂，检测到多Active冲突。

◆ 如果ActiveMemNum不同，ActiveMemNum大的为优，处于IRF Active状态继续工作，ActiveMemNum小的迁移到Recovery状态（即禁用状态）。

◆ 如果ActiveMemNum相同，继续比较ActiveID，ActiveID小的为优，处于IRF Active状态继续工作，ActiveID大的上报MAD冲突事件给IRF模块，IRF模块将该IRF迁移到MAD Recovery状态（即禁用状态）。

详细流程见下图8，其中当IRF分裂时，快速LACP MAD报文交换、确认冲突过程和ARP MAD类似。

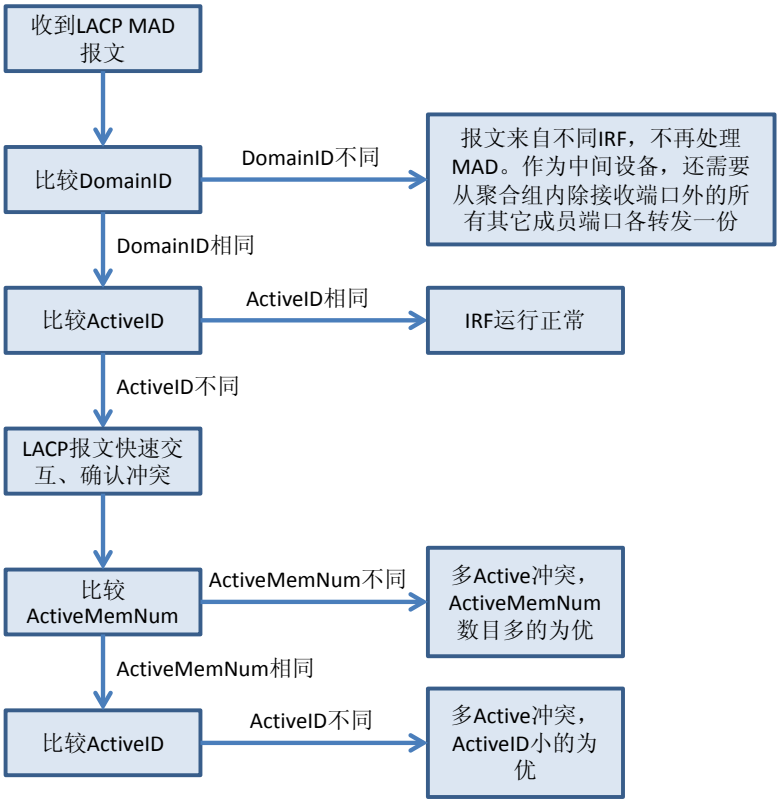


图8 LACP MAD交互流程

(二)LACP MAD检测组网要求

LACP MAD检测方式组网中需要使用H3C设备作为中间设备。通常采用如图9所示的组网，成员设备之间通过Device交互LACP扩展报文。

在LACP MAD检测组网中，如果中间设备本身也是一个IRF系统，则必须通过配置确保其IRF域编号与被检测的IRF系统不同，否则可能造成检测异常，甚至导致业务中断。

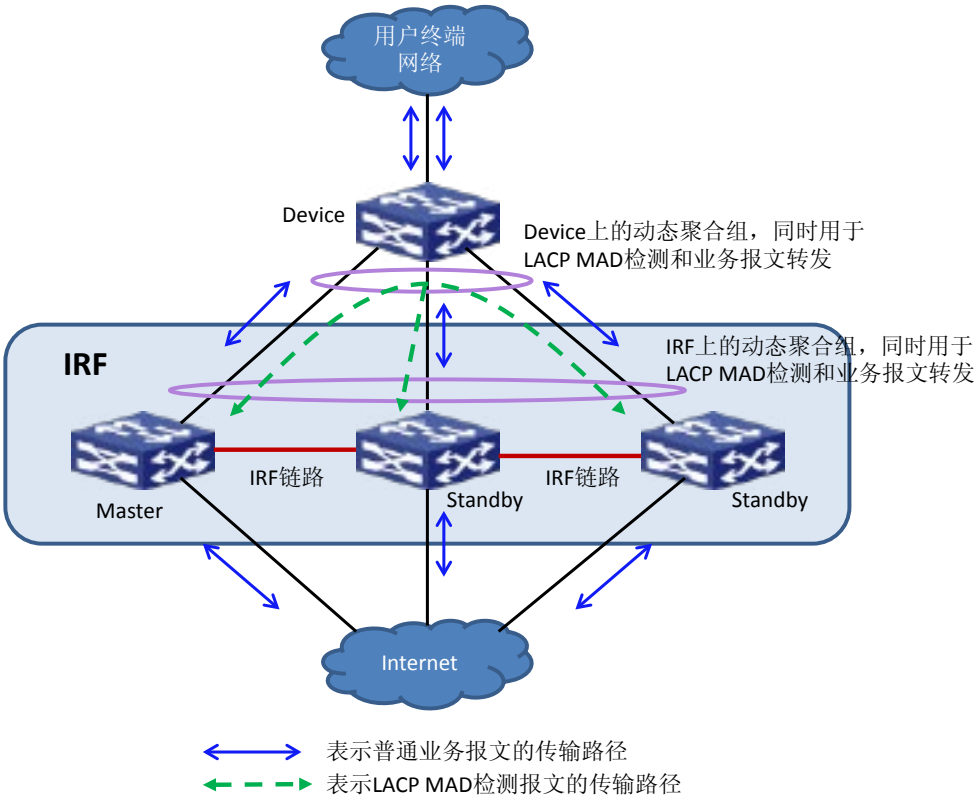


图9 LACP MAD检测组网示意图

IRF MAD总结

IRF支持的MAD检测方式有：LACP MAD检测、BFD MAD检测、ARP MAD检测和ND MAD检测。四种MAD检测机制各有特点，用户可以根据现有组网情况进行选择。由于LACP MAD和BFD MAD、ARP MAD、ND MAD冲突处理的原则不同，请不要同时配置。BFD MAD、ARP MAD、ND MAD这三种方式独立工作，彼此之间互不干扰，可以同时配置。

表1 MAD检测机制的比较

MAD检测方式	优势	限制
LACP MAD	检测速度快，利用现有聚合组网即可实现，无需占用额外接口，利用聚合链路同时传输普通业务报文和MAD检测报文（扩展LACP报文）	组网中需要使用H3C设备作为中间设备，每个成员设备都需要连接到中间设备
BFD MAD	检测速度较快，组网形式灵活，对其它设备没有要求	当堆叠设备大于两台时，组网中需要使用中间设备，每个成员设备都需要连接到中间设备，这些BFD链路专用于MAD检测
ARP MAD	非聚合的IPv4组网环境，和MSTP配合使用，无需占用额外端口。在使用中间设备的组网中对中间设备没有要求	检测速度慢于前两种。
ND MAD	非聚合的IPv6组网环境，和MSTP配合使用，无需占用额外端口。在使用中间设备的组网中对中间设备没有要求	检测速度慢于前两种

IP FRR 实现概述

文 / 丁炎

前言

当链路或节点发生故障时，在路由再次收敛前网路流量会发生中断，以现在使用最广泛的OSPF/ISIS协议来说，会经历如下五个过程：探测到故障、产生更新信息、泛洪到整个网络、重新计算路由表、下刷到FIB表。

为了缩短路由再次收敛时间，人们想了很多办法。比如利用BFD代替hello报文进行故障探测；通过定时器的退避算法以及fast flood缩短路由信息泛洪时间；利用ISPF、PRC缩短路由计算时间等等，但重新收敛时间缩短到几百毫秒已经到极限。随着语音/视频等实时性网络业务的兴起，网络运营商提供希望这个时间能小于50毫秒， IP FRR(IP Fast ReRoute)正是为了满足这种苛刻的需求而出现的。

IP FRR基本原理

IP FRR的基本原理就是对当前的路由的下一跳提前进行备份，把主用下一跳和备份下一跳下发到路由和FIB表，当检测到链路故障时，马上切换到备份下一跳进行数据转发，同时进行路由重新收敛的各个步骤，在重新收敛后再切回新的主用下一跳。这样中断时间就能缩短到故障探测和转发表替换的时间，从而大大缩短了流量中断时间。

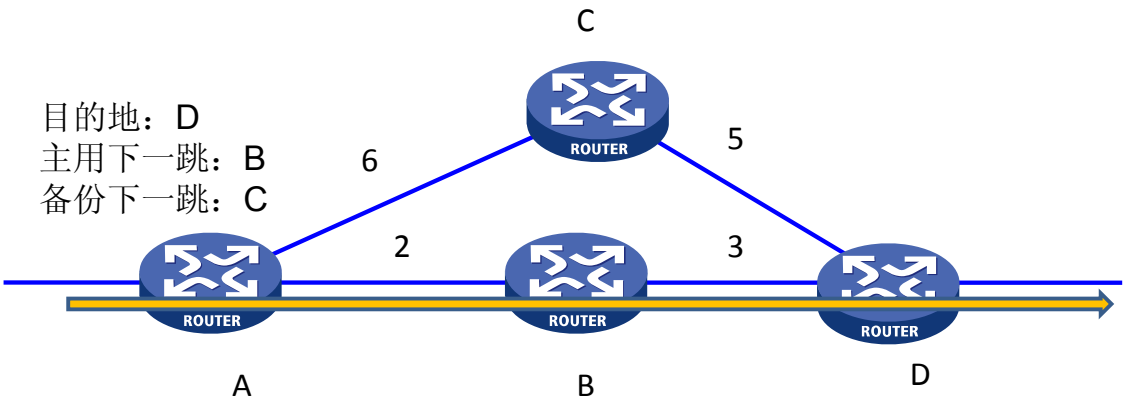


图1 IP FRR示意图

如图1所示，从路由器A到目的地D有两条路径：A->B->D和A->C->D。A->B->D的COST更小，所以B作为A到D的主用下一跳下发到路由表和FIB表中，指导数据转发。而C作为备份下一跳也下刷到路由表和FIB表。同时A到B的链路中使能BFD探测，以确保能快速感知到链路故障。

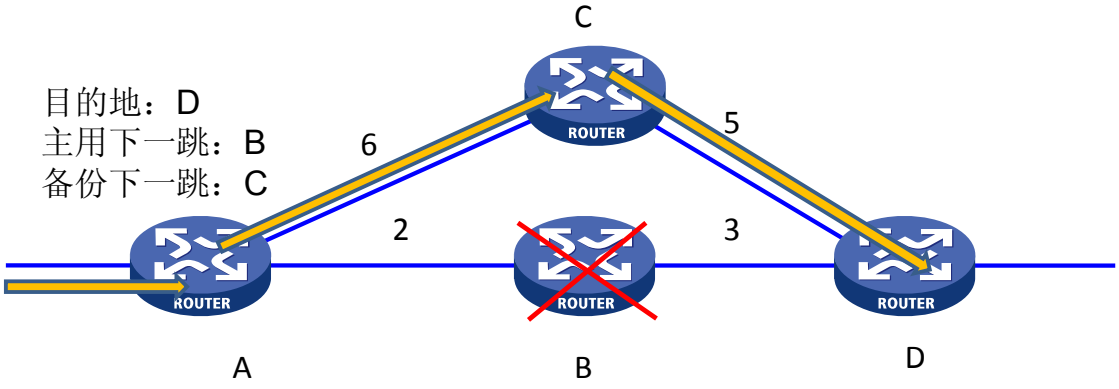


图 2 IP FRR故障示意图

如图2所示，当A通过物理层或BFD感知出到主用下一跳B出现故障后，马上切换到备份下一跳C进行转发，直到路由再次收敛，被新的最优路由替换掉。

在我司设备上，IP FRR提前备份的下一跳可以手工配置指定，也可以利用SPF和LFA算法，自动计算出备份下一跳。

手工配置是路由控制层面通过路由策略为目的IP地址的路由信息上指定一个备份出接口和下一跳地址，全凭人为指定，设备不会检查备份下一跳是否有效，相当于为目前的最优路由配置备份静态路由。当检测到故障后，临时切换到这条备份静态路由上进行数据转发。利用静态配置方式，优点是它可以为各种IGP路由协议进行IP FRR保护，但缺点就是十分麻烦，配置量大，并容易出错。在大型网络中，通过这种静态配置方式进行全面保护，在各台路由器上逐条进行配置，不太现实，这就需要有一种自动备份下一跳的技术来解决。

早期出现的几种主要IP FRR实现草案有：Loop Free Alternate、Not-Via Address、Interface-specific forwarding、Multi-topology protection等几种。几种方案各有优缺点，其中LFA实现最为简单，不用对路由协议进行扩展，它在06年和07年列为草案，并在08年的时候由ITEF推为正式标准---RFC5286。我司和其它几个设备厂商目前都是通过LFA来实现的IP FRR。

基于LFA实现IP FRR的基本原理

我们知道在大型网络上运行的IGP协议多为OSPF/ISIS这种链路状态协议，各台路由器维护同一个LSDB，最优路由就是通过LSDB这张地图，以自己为根，用SPF算法计算出来的。那么在运行链路状态协议的网络中，各台路由器是可以用LSDB把所有去往同一目的地的各个完整路径都计算出来，这就能够知道下一跳后的再下一跳是去往哪里。我们利用链路状态协议的这个特点，就可以让设备自动计算并优选出备份下一跳对下一跳进行保护。

基于Loop Free Alternate实现IP FRR就是通过SPF和LFA算法，自动计算并备份下一跳，最大的困难在于如何保证备份路由在故障发生后不会造成的临时环路，再有就是定制一个合理的优选规则，找出最优的备份下一跳。

确保无环路备份保护

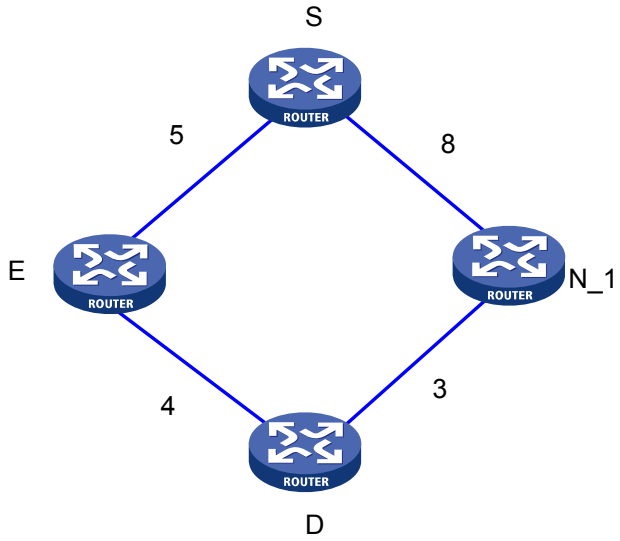


图3 无环路备份示意图

如图3所示，从始发设备S到D的最优下一跳为E，如果将N_1作为S到E的备份下一跳，当S和E间出现故障后，把去往D的数据包交给N_1，可以正常转发到D，所以N_1具备S到E的备份下一跳资格。

但如果当N_1到D的直连链路COST值大于17，在S和E之间发生故障且N_1上的路由重新收敛前，N_1到D的下一跳是S而不是D（N_1-S-E-D的链路Cost为17），如果S把去往D的数据报文转发给N_1，，N_1会把数据报文转发回给S，这样就形成了临时环路，因此当N_1和D之间直连链路的的COST值大于N_1->S->E->D时，N_1不具备S到E的备份下一跳资格。

在RFC5286中给出了满足无环路备份的条件，当且仅当满足下面不等式的时候，邻居路由器N_1能提供无环路备份保护：

Distance_opt(N_1，D) < Distance_opt(N_1，S) + Distance_opt(S，D)

注：Distance_opt[] 为计算两节点间开销或距离

对备份下一跳进行优选

当有若干个下一跳都满足Distance_opt(N_1，D) < Distance_opt(N_1，S) + Distance_opt(S，D)的

时候，就需要进行优选，主体思想是尽量选出保护能力更强且不会形成微环的备份下一跳。

优选的顺序如下：

节点保护>链路保护>满足下游路径优先>备份路径COST值小的优先>备份下一跳的出接口和主用下一跳出接口不同的优先>备份下一跳的IP的小的优先。

节点保护：

如图3所示，通过N_1的备份路径能绕过路由器E，当路由器E整机出现故障，备份下一跳N_1能够正常把数据流转发给D，这样N就具备节点保护能力。

当满足不等式 Distance_opt(N_1，D) < Distance_opt(N_1，E) + Distance_opt(E，D)的时候，备份下一跳就能对E提供节点保护。

链路保护：

当备份路径能够绕过故障链路，如图4所示，当节点S和E间的直连链路出现故障时，S把流量切换到N_1，能够正常转发到D，这种情况下，N_1具备链路保护能力。

链路保护要满足不等式Distance_opt(N_1，D) < Distance_opt(N_1，S) + Distance_opt(S，D)。

当满足节点保护时，一般而言也会同时满足链路保护。当仅满足链路保护时，下游节点发生故障时，可能会出现微环现象。如图4所示， 当S和N_1都启用了IP FRR，S和N_1会互把对方当做自己到D主路径的备份下一跳，这时候当下游节点E出现故障时，S和N_1就会把到D的报文相互发送，出现微环。所以规定节点保护的优先级大于链路保护。

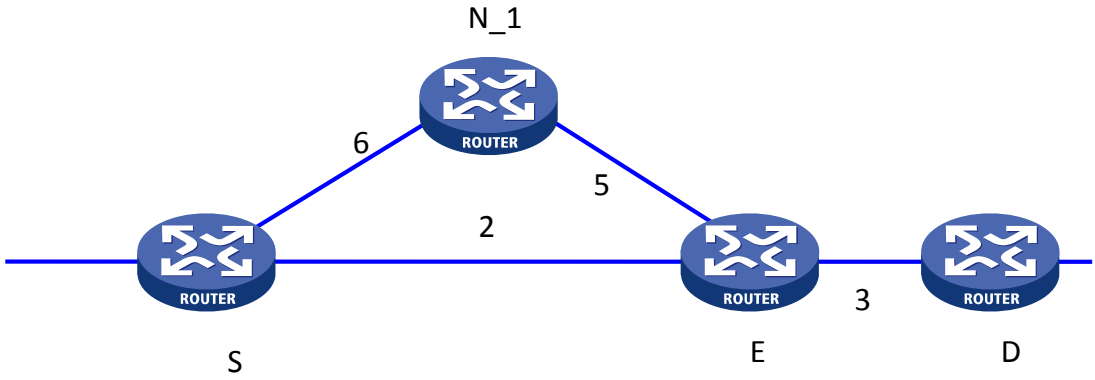


图4 链路保护示意图

下游路径标准：RFC中规定，当满足无环路不等式条件的子集：Distance_opt(N_1，D) < Distance_opt(S，D)时，称为满足下游路径标准。当节点保护和链路保护的情况相同时，优先选择符合下游路径标准的备份下一跳。满足下游路径标准的备份下一跳能够适应更复杂的故障情况。

当上述三个条件都相等时，优选各备份路径中COST值最小的。如果整个路径COST相等，选取出接口和主用出接口不同的备份下一跳。当所有优选条件都相同，通过选择备份下一跳IP地址小的来确定最优备份下一跳。

总结

基于LFA实现IP FRR相对简单，不需要对OSPF/ISIS协议进行扩展，利用现有的SPF算法将所有计算、备份都在本地完成，目前各个主流设备厂商都是基于此来实现IP FRR。但据统计分析基于LFA实现IP FRR保护率在70%左右，且路由计算量较大。而Not-via Address、Multi-topology protection等技术虽然理论上能解决所有故障情况，使得故障保护率达到100%，但实现复杂，且较多细节问题还待讨论。随着业界对各个方案的研究，IP FRR技术还会继续发展和完善。

MPLS TE FRR 技术分析

文 / 苏艳梅

MPLS TE FRR技术概述

FRR是快速重路由（Fast ReRoute）的简称，TE FRR是MPLS TE网络中实现局部修复的一种技术。在使能MPLS TE FRR功能之后，当被保护的LSP经过的某条链路或者某个节点失效时，流量会被PLR切换到保护隧道上，流量转发不中断或者仅有少量丢包。RFC 4090详细介绍了MPLS TE FRR技术，本文主要是结合我司设备实现介绍Facility方式的MPLS TE FRR实现原理。在详细介绍MPLS TE FRR实现原理之前，先介绍几个MPLS TE FRR技术中涉及的术语。

主LSP：被保护的LSP。

Bypass LSP：旁路LSP，保护主LSP的LSP。

PLR（Point of Local Repair，本地修复节点）：Bypass LSP的头节点，必须在主LSP的路径上，并且不能是主LSP的尾节点。

MP（Merge Point，汇聚点）：Bypass LSP的尾节点，必须在主LSP的路径上，并且不能是主LSP的头节点。

MPLS TE FRR实现原理

MPLS TE FRR的实现方式

MPLS TE FRR包括两种方式，一种是One-to-One方式，另一种是Facility方式。

One-to-One方式的FRR是指PLR节点为每一条主LSP都创建一条旁路LSP，当有多条主LSP通过PLR节点时，会创建多条旁路LSP分别保护每条主LSP。

Facility方式的FRR是指PLR节点为多条主LSP创建一条旁路LSP的方式。

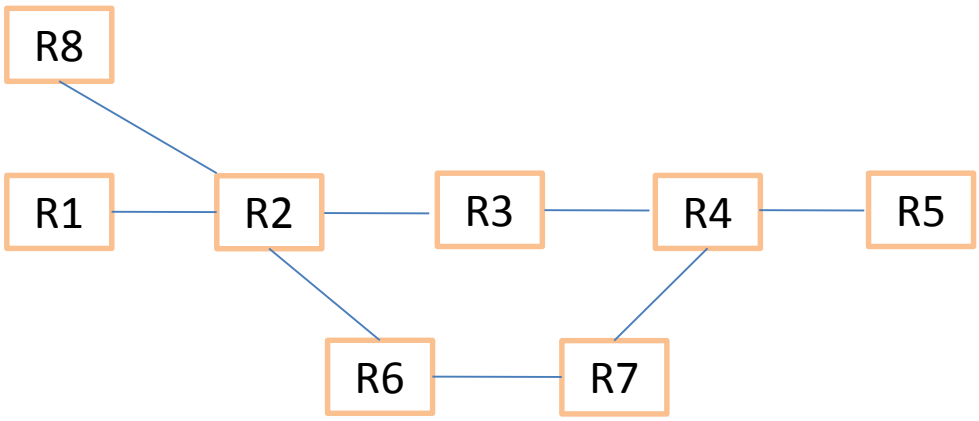


图1 Facility方式FRR说明图

Facility方式MPLS TE FRR举例说明如下：

假设有三条LSP通过节点R2，分别是

Protected LSP 1：[R1->R2->R3->R4->R5]

Protected LSP 2：[R8->R2->R3->R4]

Protected LSP 3：[R2->R3->R4->R5]

在节点R2（也就是PLR节点）上建立旁路LSP[R2->R6->R7->R4]隧道，用来保护主LSP1、LSP2、LSP3三条隧道，当节点R3或者链路[R2->R3]出现故障时，通过主LSP转发的流量会切换到旁路LSP隧道内转发。

One-to-One方式的MPLS TE FRR我司没有实现，我司实现的是Facility方式的MPLS TE FRR。

Facility方式的MPLS TE FRR有两种实现方式，一种是手工方式，就是在PLR节点上主LSP对应的出接口上配置保护隧道；另一种是自动方式，如果主LSP使能了FRR能力，且在PLR节点上配置TE Auto FRR能力，PLR节点会自动建立保护隧道。本文主要是结合我司设备实现讲解手工方式的MPLS TE FRR实现原理。

Facility 方式RSVP TE FRR实现原理

Facility 方式的RSVP TE FRR用到的标志位

RSVP-TE协议是RSVP协议的扩展，用于TE隧道的建立和维护。RSVP-TE协议通过不同类型消息对象来实现不同的功能。而实现TE FRR功能用到了SESSION_ATTRIBUTE对象、RRO对象中的标志位。头节点使能FRR之后，SESSION_ATTRIBUTE对象中的Local Protection desired、label recording desired、SE Style desired标志位都要置1，RSVP消息SESSION_ATTRIBUTE对象具体格式如下：

```
⊟ SESSION ATTRIBUTE: SetupPrio 7, HoldPrio 7, Local Protection, Label Recording, SE Style,  [Tunnel1]
    Length: 16
    Object class: SESSION ATTRIBUTE object (207)
    C-type: 7 - IPv4 LSP (No Resource Affinities)
    Setup priority: 7
    Hold priority: 7
  ⊟ Flags: 0x07
      .... ...1 = Local protection desired
      .... ..1. = Label recording desired
      .... .1.. = SE style desired
      .... 0... = Bandwidth protection not desired
      ...0 .... = Node protection not desired
    Name length: 7
    Name: Tunnel1
```

图2： Rsvp消息SESSION_ATTRIBUTE对象

当PLR节点FRR生效绑定成功之后，Path、Resv消息中的RRO对象对应的IPv4子对象以及Label子对象中的标志位也会相应置位，我司实现只用到Flag标志位的低两位，IPv4子对象中报文格式如下：

```
⊟ IPv4 Subobject - 100.1.1.1, Local Protection Available
    Type: 1 (IPv4)
    Length: 8
    IPv4 hop: 100.1.1.1
    Prefix length: 32
  ⊟ Flags: 0x01
      .... ...1 = Local Protection Available
      .... ..0. = Local Protection Not In Use
      .... .0.. = Bandwidth Protection Not Available
      .... 0... = Node Protection Not Available
      ...0 .... = Address Doesn't Specify a Node-id Address
```

图3：IPv4子对象报文格式

```
⊟ Label Subobject - 1035, , Local Protection Available
    Type: 3 (Label)
    Length: 8
  ⊟ Flags: 0x01
      .... ...1 = Local Protection Available
      .... ..0. = Local Protection Not In Use
      .... .0.. = Backup Tunnel Does Not Have Bandwidth
      .... 0... = Backup Tunnel Goes To Next-Hop
    C-Type: 1
    Label: 1035
```

图4 Label子对象

Facility 方式TE FRR协议流程

Facility方式的TE FRR实现主要是由头节点、PLR节点、MP节点完成的。FRR能力实现可以分为两个阶段，第一阶段是FRR切换之前，第二阶段是FRR切换之后。

FRR切换之前，头节点、PLR节点、MP节点各需要完成那些工作?

FRR能力是由LSP头节点决定，在使能FRR能力时，必需使能Label record能力，同时资源预留样式必须是SE类型。头节点在使能FRR之后，其后续发送Path消息，SESSION_ATTRIBUTE对象中的Local Protection desired、label recording desired、SE Style desired标志位都要置位。

主LSP经过的所有LSR节点（尾节点除外）都可能成为PLR节点，根据网络规划，最后由用户配置决定PLR节点的位置。PLR节点通过SESSION_ATTRIBUTE对象中的Flag标志位来确定LSP是否需要本地保护，如果是需要本地保护的LSP，PLR节点需要查找符合条件的Bypass LSP，如果能够找合适的Bypass LSP，向上游节点发送Resv息时以及向下游节点发送Path消息时，其RRO对象中的Label子对象、IPv4子对象中的Local protection available标志位置1；同时生成主LSP和Bypass LSP的保护关系，并下刷LSP表项以及MPLS转发表项。这时候，PLR才算正式具备了FRR能力。

MP节点是主LSP、Bypass LSP的汇聚点，必是Bypass LSP的尾节点，也可以是主LSP的尾节点。在FRR切换之前，MP节点和普通节点功能相同，不需要做特殊处理。

FRR切换之后，头节点、PLR节点、MP节点各需要完成那些工作?

在FRR切换之后，头节点在收到PLR节点发送的Path Error消息之后，重新进行CSPF计算，建立新隧道LSP，在新的LSP建立之前，老的LSP不会删除，当新的LSP建立之后，头节点发送Path Tear消息拆除老的LSP。如果新的LSP和旧的LSP部分路径相同，则在这些路径上需要重复为新旧CRLSP预留带宽，造成带宽资源的浪费。为此提出了make-before-break机制，它是一种在尽可能不丢失数据，也不占用额外带宽的前提下改变MPLS TE隧道的机制。RSVP-TE协议在建立LSP隧道时，有两种资源预留方

式。一种是为每个发送者单独预留资源，同一会话中的不同发送者不能共享资源，该资源预留方式称为Fixed-Filter。另外一种方式是为同一个会话中的不同发送者预留同一个资源，不同发送者之间可以共享资源，该资源预留方式称为Shared-Explicit。make-before-break机制就是通过Shared-Explicit资源预留方式实现的。

在FRR切换之后，PLR要做协议层、转发层两方面的工作。PLR检测链路或者节点故障后进行FRR切换，首先会通知MPLS转发层面，把数据流量切换到Bypass隧道上，协议报文也要通过Bypass隧道转发到MP节点。接着通知协议层向上游节点发送Path Error消息，通知头节点重新建立新隧道LSP，同时更新LSP对应的PSB和RSB信息，Path消息中的Local Protection In Use标志位置位。当新的主LSP建立之后，PLR节点才把数据流量以及协议报文重新切换到主LSP隧道上，并且清除相应的标志位。

FRR切换之后，MP节点收到Path消息的入接口和Hop地址发生了变化，直接更新本地PSB中的Hop地址和入接口，Hop地址为PLR发送Path消息的出接口地址。

举例说明

举例说明：

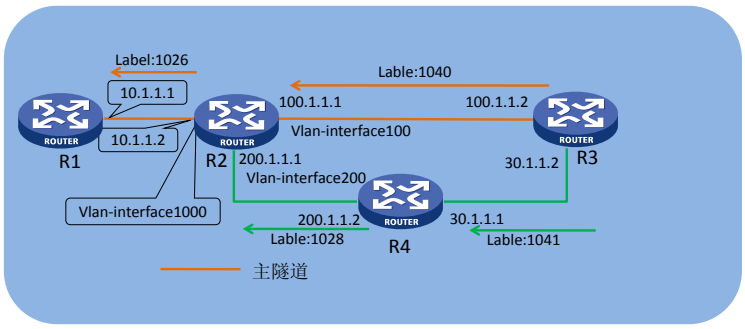


图5 TE FRR 举例组网图

如上组网图中建立两条隧道：

主隧道路径为R1->R2->R3

保护隧道路径为R2->R4->R3，其中R2为PLR节点，R3为MP节点

R1_RouterID：1.1.1.1　　R2_RouterID：1.1.1.2

R4_RouterID：1.1.1.4　　R3_RouterID：1.1.1.3

PLR节点是TE FRR功能的关键节点，大部分工作是由PLR节点完成的。

◆ 在主备隧道绑定之后，FRR切换之前，PLR节点发送的协议报文

PLR节点向上游节点R1发送Resv消息，其RRO对象中的Label子对象以及IPv4子对象中的Local Protection Available标志位置1。具体信息如下：

```
⊟ RECORD ROUTE: IPv4 10.1.1.2, Label 1026, IPv4 1.1.1.2, ...
    Length: 60
    Object class: RECORD ROUTE object (21)
    C-type: 1
  ⊞ IPv4 Subobject - 10.1.1.2
  ⊞ Label Subobject - 1026, , Local Protection Available
  ⊞ IPv4 Subobject - 1.1.1.2
  ⊞ IPv4 Subobject - 100.1.1.1, Local Protection Available
  ⊞ IPv4 Subobject - 100.1.1.2
  ⊞ Label Subobject - 1040, , Local Protection Available
  ⊞ IPv4 Subobject - 1.1.1.3
```

图6 FRR切换之前，Resv消息中的RRO对象

PLR节点向下游节点R3发送Path消息，其RRO对象中的IPv4子对象以及Label子对象中的Local

Protection Available标志位置1。具体信息如下：

```
⊟ RECORD ROUTE: IPv4 100.1.1.1, IPv4 1.1.1.2, IPv4 10.1.1.2, ...
    Length: 44
    object class: RECORD ROUTE object (21)
    C-type: 1
  ⊞ IPv4 Subobject - 100.1.1.1, Local Protection Available
  ⊞ IPv4 Subobject - 1.1.1.2
  ⊞ IPv4 Subobject - 10.1.1.2
  ⊞ Label Subobject - 1026, , Local Protection Available
  ⊞ IPv4 Subobject - 10.1.1.1
```

图7 FRR切换之前，Path消息中的RRO对象

在PLR节点上，Bypass LSP和普通LSP一样，不需要特殊处理。

◆ FRR切换之后，PLR节点的FRR处理过程

当R2和R3之间发生链路路障时，PLR节点就会响应FRR切换事件，把数据流量，协议报文切换到旁路LSP上，同时向PLR节点向上游节点发送Path Error消息，通知头节点重新建立新隧道LSP。PLR节点会继续向下游节点发送Path消息，Path消息是通过Bypass LSP 标签转发到MP节点，其RRO对象中IPv4子对象中的Local Protection In Use标志位置1。PLR节点向下游节点发送的Path消息报文如下：

```
⊞ SENDER TEMPLATE: IPv4-LSP, Tunnel Source: 200.1.1.1, LSP ID: 5.
⊞ SENDER TSPEC: Intserv, Token Bucket, 125000 bytes/sec.
⊞ LABEL REQUEST: Basic: L3PID: IP (0x0800)
⊞ ADSPEC
⊞ EXPLICIT ROUTE: IPv4 1.1.1.3
⊟ RECORD ROUTE: IPv4 200.1.1.1, IPv4 1.1.1.2, IPv4 10.1.1.2, ...
    Length: 44
    object class: RECORD ROUTE object (21)
    C-type: 1
  ⊞ IPv4 Subobject - 200.1.1.1, Local Protection In Use
  ⊞ IPv4 Subobject - 1.1.1.2
  ⊞ IPv4 Subobject - 10.1.1.2
  ⊞ Label Subobject - 1026    , Local Protection Available
  ⊞ IPv4 Subobject - 10.1.1.1
```

图8 FRR切换之后，PLR节点向下游节点发送的Path消息

在PLR节点上，对比FRR切换前后LSP信息，Bypass In Use标志位由Not Used变为In Use。

```
PLR节点， FRR切换之前LSP(1.1.1.3)的详细信息：
 No.            : 1
 IngressLsrID      : 1.1.1.2
 LocalLspID        : 6
 Tunnel-Interface    : Tunnel2
 Fec            : 1.1.1.3/32
 Nexthop          : 200.1.1.2
 In-Label         : NULL
 Out-Label        : 1027
 In-Interface      : ----------
 Out-Interface      : Vlan-interface200
 LspIndex        : 7175
 Tunnel ID        : 0xc0021
 LsrType         : Ingress
 Bypass In Use      : Not Exists
 BypassTunnel      : Tunnel Index[---]

 No.            : 2
 IngressLsrID      : 1.1.1.1
 LocalLspID        : 7
 Tunnel-Interface   : Tunnel1
 Fec            : 1.1.1.3/32
 Nexthop          : 100.1.1.2
 In-Label         : 1026
 Out-Label        : 1040
 In-Interface      : Vlan-interface1000
 Out-Interface      : Vlan-interface100
 LspIndex        : 7179
 Tunnel ID        : 0xc0025
 LsrType         : Transit
 Bypass In Use      : Not Used   FRR备份关系，当前状态时Not Used
 BypassTunnel      : Tunnel Index[Tunnel2], InnerLabel[1045]
```

```
PLR节点，FRR切换之后，LSP(1.1.1.3)的详细信息：
 No.            : 1
 IngressLsrID      : 1.1.1.2
 LocalLspID        : 6
 Tunnel-Interface    : Tunnel2
 Fec            : 1.1.1.3/32
 Nexthop          : 200.1.1.2
 In-Label         : NULL
 Out-Label        : 1027
 In-Interface      : ----------
 Out-Interface      : Vlan-interface200
 LspIndex        : 7175
 Tunnel ID        : 0xc0021
 LsrType         : Ingress
 Bypass In Use      : Not Exists
 BypassTunnel      : Tunnel Index[---]

 No.            : 2
 IngressLsrID      : 1.1.1.1
 LocalLspID        : 7
 Tunnel-Interface   : Tunnel1
 Fec            : 1.1.1.3/32
 Nexthop          : 100.1.1.2
 In-Label         : 1026
 Out-Label        : 1040
 In-Interface      : Vlan-interface1000
 Out-Interface      : Vlan-interface100
 LspIndex        : 7179
 Tunnel ID        : 0xc0025
 LsrType         : Transit
 Bypass In Use      : In Use   旁路LSP正在使用
 BypassTunnel      : Tunnel Index[Tunnel2], InnerLabel[1045]
```

图9 PLR节点上，FRR切换前后LSP详细信息对比图

在PLR节点上，对比FRR切换前后PSB信息块，Next Hop地址、出接口发生了变化，Next Hop地址变为实际发送物理接口地址，出接口变为Bypass隧道，同时增加了PLR In used信息。

```
PLR节点上，FRR切换之前PSB信息：
dis mpls rsvp-te psb-content  1.1.1.1 7 1 1.1.1.3
The PSB Content:
 Tunnel Dest: 1.1.1.3              Session Tunnel ID: 1
 Tunnel ExtID: 1.1.1.1
 Ingress LSR ID: 1.1.1.1           Local LSP ID: 7
 Previous Hop : 10.1.1.1           Next Hop : 100.1.1.2
 Incoming / Outgoing Interface: Vlan-interface1000  / Vlan-
interface100
 InLabel : 1026            OutLabel : 1040
 Reverse InLabel : NULL       Reverse OutLabel : NULL
 Send Message ID : 36              Recv Message ID : 0
 Session Attribute-
   SetupPrio: 7       HoldPrio: 7
   SessionFlag: Local Protect desired. Label Record. SE Style desired.
 ERO Information-
     L-Type        ERO-IPAddr      ERO-PrefixLen
   ERHOP_STRICT    100.1.1.2          32
 RRO Information-
 RRO-Flag is  1 Local repair available.
   RRO-CType: IPV4     RRO-IPAddress: 10.1.1.1          RRO-
IPPrefixLen: 32
 SenderTspec Information-
   Token bucket rate: 125000.00
   Token bucket size: 1000.00
   Peak data rate: 125000.00
   Minimum policed unit: 0
   Maximum packet size: 1500
 Path Message arrive on Vlan-interface1000 from PHOP 10.1.1.1
 Path Message sent to NHOP 100.1.1.2  on Vlan-interface100
 Resource Reservation OK
 Graceful Restart State: Normal
 Bandwidth Constraint Mode: PreStandard DS-TE RDM
```

```
PLR节点上，FRR切换之后PSB信息：
dis mpls rsvp-te psb-content  1.1.1.1 7 1 1.1.1.3
The PSB Content:
 Tunnel Dest: 1.1.1.3              Session Tunnel ID: 1
 Tunnel ExtID: 1.1.1.1
 Ingress LSR ID: 1.1.1.1           Local LSP ID: 7
 Previous Hop : 10.1.1.1           Next Hop : 30.1.1.2
 Incoming / Outgoing Interface: Vlan-interface1000  / Tunnel2
 InLabel : 1026            OutLabel : 1040
 Reverse InLabel : NULL       Reverse OutLabel : NULL
 Send Message ID : 36              Recv Message ID : 0
 Session Attribute-
   SetupPrio: 7       HoldPrio: 7
   SessionFlag: Local Protect desired. Label Record. SE Style desired.
 ERO Information-
     L-Type        ERO-IPAddr      ERO-PrefixLen
   ERHOP_STRICT    100.1.1.2          32
 RRO Information-
 RRO-Flag is  2 Local repair enable.
   RRO-CType: IPV4     RRO-IPAddress: 10.1.1.1          RRO-
IPPrefixLen: 32
 SenderTspec Information-
   Token bucket rate: 125000.00
   Token bucket size: 1000.00
   Peak data rate: 125000.00
   Minimum policed unit: 0
   Maximum packet size: 1500
 Path Message arrive on Vlan-interface1000 from PHOP 10.1.1.1
 Path Message sent to NHOP 30.1.1.2  on Tunnel2
 Resource Reservation OK
 PLR in used
 Graceful Restart State: Normal
 Bandwidth Constraint Mode: PreStandard DS-TE RDM
```

图10 PLR节点上，FRR切换前后PSB信息对比图

在PLR节点上，对比FRR切换前后RSB信息块，只有RRO-IPAddress地址发生了变化。

```
PLR节点上，FRR切换之前RSB信息：
dis mpls rsvp-te rsb-content  1.1.1.1 12 1 1.1.1.3 100.1.1.2
The RSB Content:
 Tunnel Dest: 1.1.1.3              Session Tunnel ID: 1
 Tunnel ExtID: 1.1.1.1
 Next Hop: 100.1.1.2               Resevation Style: SE Style
 Reservation Incoming Interface: Vlan-interface100
 Reservation Interface: Vlan-interface100
 Message ID : 0
 Filter Spec Information-
   The filter number: 1
   Ingress LSR ID: 1.1.1.1        Local LSP ID: 12      OutLabel: 1040
 Graceful Restart State: Normal
 RRO Information-
 RRO-Flag is
   RRO-CType: IPV4   RRO-IPAddress: 100.1.1.2          RRO-
IPPrefixLen: 32
   RRO-CType: Label  RRO-Label: 1040
   RRO-CType: IPV4   RRO-IPAddress: 1.1.1.3            RRO-IPPrefixLen:
32
 FlowSpec Information-
   Token bucket rate: 125000.00
   Token bucket size: 1000.00
   Peak data rate: 125000.00
   Minimum policed unit: 0
   Maximum packet size: 1500
   Bandwidth guarantees: 0.00
   Delay guarantees: 0
   Qos Service is Controlled
 Resv Message arrive on Vlan-interface100 from NHOP 100.1.1.2
 Graceful Restart State: Normal
```

```
PLR节点上，FRR切换之后RSB信息：
dis mpls rsvp-te rsb-content  1.1.1.1 12 1 1.1.1.3 100.1.1.2
The RSB Content:
 Tunnel Dest: 1.1.1.3              Session Tunnel ID: 1
 Tunnel ExtID: 1.1.1.1
 Next Hop: 100.1.1.2               Resevation Style: SE Style
 Reservation Incoming Interface: Vlan-interface200
 Reservation Interface: Vlan-interface100
 Message ID : 0
 Filter Spec Information-
   The filter number: 1
   Ingress LSR ID: 1.1.1.1        Local LSP ID: 12      OutLabel: 1040
 Graceful Restart State: Normal
 RRO Information-
 RRO-Flag is
   RRO-CType: IPV4   RRO-IPAddress: 30.1.1.2          RRO-IPPrefixLen:
32
   RRO-CType: Label  RRO-Label: 1040
   RRO-CType: IPV4   RRO-IPAddress: 1.1.1.3            RRO-IPPrefixLen:
32
 FlowSpec Information-
   Token bucket rate: 125000.00
   Token bucket size: 1000.00
   Peak data rate: 125000.00
   Minimum policed unit: 0
   Maximum packet size: 1500
   Bandwidth guarantees: 0.00
   Delay guarantees: 0
   Qos Service is Controlled
 Resv Message arrive on Vlan-interface200 from NHOP 100.1.1.2
 Graceful Restart State: Normal
```

图11 PLR节点上，FRR切换前后RSB信息对比图

在MP节点上，对比FRR切换前后PSB信息块，其Previous Hop、入接口发生了变化。

```
MP节点上，FRR切换之前PSB信息：
dis mpls rsvp-te psb-content  1.1.1.1 12 1 1.1.1.3
The PSB Content:
 Tunnel Dest: 1.1.1.3              Session Tunnel ID: 1
 Tunnel ExtID: 1.1.1.1
 Ingress LSR ID: 1.1.1.1           Local LSP ID: 7
 Previous Hop : 100.1.1.1          Next Hop :  -----
 Incoming / Outgoing Interface: Vlan-interface100  /  -----
 InLabel : 1040            OutLabel : NULL
 Reverse InLabel : NULL       Reverse OutLabel : NULL
 Send Message ID : 0               Recv Message ID : 0
 Session Attribute-
   SetupPrio: 7       HoldPrio: 7
   SessionFlag: Local Protect desired. Label Record. SE Style desired.
 ERO Information-
    -----
 RRO Information-
 RRO-Flag is  0
   RRO-CType: IPV4    RRO-IPAddress: 100.1.1.1         RRO-
IPPrefixLen: 32
   RRO-CType: IPV4    RRO-IPAddress: 1.1.1.2          RRO-
IPPrefixLen: 32
   RRO-CType: IPV4    RRO-IPAddress: 10.1.1.2         RRO-
IPPrefixLen: 32
   RRO-CType: Label    RRO-Label: 1026
   RRO-CType: IPV4    RRO-IPAddress: 10.1.1.1         RRO-
IPPrefixLen: 32
 SenderTspec Information-
   Token bucket rate: 125000.00
   Token bucket size: 1000.00
   Peak data rate: 125000.00
   Minimum policed unit: 0
   Maximum packet size: 1500
 Path Message arrive on Vlan-interface100 from PHOP 100.1.1.1
 Resource Reservation OK
 Up stream node was Protected
 Graceful Restart State: ---
 Bandwidth Constraint Mode: PreStandard DS-TE RDM
```

```
MP节点上，FRR切换之后PSB信息：
dis mpls rsvp-te psb-content  1.1.1.1 12 1 1.1.1.3
The PSB Content:
 Tunnel Dest: 1.1.1.3              Session Tunnel ID: 1
 Tunnel ExtID: 1.1.1.1
 Ingress LSR ID: 1.1.1.1           Local LSP ID: 7
 Previous Hop : 200.1.1.1          Next Hop :  -----
 Incoming / Outgoing Interface: Vlan-interface300  /  -----
 InLabel : 1040            OutLabel : NULL
 Reverse InLabel : NULL       Reverse OutLabel : NULL
 Send Message ID : 0               Recv Message ID : 0
 Session Attribute-
   SetupPrio: 7       HoldPrio: 7
   SessionFlag: Local Protect desired. Label Record. SE Style desired.
 ERO Information-
    -----
 RRO Information-
 RRO-Flag is  0
   RRO-CType: IPV4    RRO-IPAddress: 200.1.1.1         RRO-
IPPrefixLen: 32
   RRO-CType: IPV4    RRO-IPAddress: 1.1.1.2          RRO-
IPPrefixLen: 32
   RRO-CType: IPV4    RRO-IPAddress: 10.1.1.2         RRO-
IPPrefixLen: 32
   RRO-CType: Label    RRO-Label: 1026
   RRO-CType: IPV4    RRO-IPAddress: 10.1.1.1         RRO-
IPPrefixLen: 32
 SenderTspec Information-
   Token bucket rate: 125000.00
   Token bucket size: 1000.00
   Peak data rate: 125000.00
   Minimum policed unit: 0
   Maximum packet size: 1500
 Path Message arrive on Vlan-interface300 from PHOP 200.1.1.1
 Resource Reservation OK
 MP in used
 Graceful Restart State: ---
 Bandwidth Constraint Mode: PreStandard DS-TE RDM
```

图12 MP节点上，FRR切换前后PSB信息对比图

头节点收到Path Error消息之后，重新进行CSPF计算，建立新隧道LSP，在新的LSP建立之前，老的LSP不会删除。这段期间，头节点会发送分别对应新老LSP的两条Path消息，同样下游节点会逐跳向上游节点发送Resv消息，由于新、老LSP其实是一个隧道，就用一条Resv消息回应两条Path消息，具体内容如下：

```
⊟ Resource Reservation Protocol (RSVP): RESV Message. SESSION: IPv4-LSP,
  ⊞ RSVP Header. RESV Message.
  ⊞ SESSION: IPv4-LSP, Destination 1.1.1.3, Tunnel ID 1, Ext ID 1010101.
  ⊞ HOP: IPv4, 10.1.1.2
  ⊞ TIME VALUES: 30000 ms
  ⊞ STYLE: Shared-Explicit (18)
  ⊞ FLOWSPEC: Controlled Load: Token Bucket, 125000 bytes/sec.
  ⊞ FILTERSPEC: IPv4-LSP, Tunnel Source: 1.1.1.1, LSP ID: 5.
  ⊞ LABEL: 1026
  ⊞ RECORD ROUTE: IPv4 10.1.1.2, Label 1028, IPv4 1.1.1.2, ...
  ⊞ FILTERSPEC: IPv4-LSP, Tunnel Source: 1.1.1.1, LSP ID: 6.
  ⊞ LABEL: 1029
  ⊞ RECORD ROUTE: IPv4 10.1.1.2, Label 1029, IPv4 1.1.1.2, ...
```

图13 FRR切换之后，PLR节点发送的Resv消息

LSP ID等于5对应的是老LSP，LSP ID等于6对应的是新LSP。头节点当新的LSP建立之后，数据报文就通过新的LSP隧道进行转发，同时发送Path Tear消息删除老LSP。PLR节点从新LSP隧道收到的数据报文从新LSP隧道转发，从老的LSP隧道收到的数据报文从备份隧道转发。PLR节点收到Path Tear消息之后，先通过备份LSP向下游发送Path Tear消息，然后去绑定老LSP和备份LSP之间的绑定关系，同时拆除老的LSP，后续不再通过备份LSP隧道转发数据，TE FRR回切过程完成。

2.3 链路保护和节点保护

TE FRR根据保护对象不同又可以分为链路保护和节点保护。

如下图所示，当PLR节点和MP节点之间有直接链路连接，当这条链路失效之后，PLR节点把流量切换到备份LSP，这种组网方式称为链路保护。

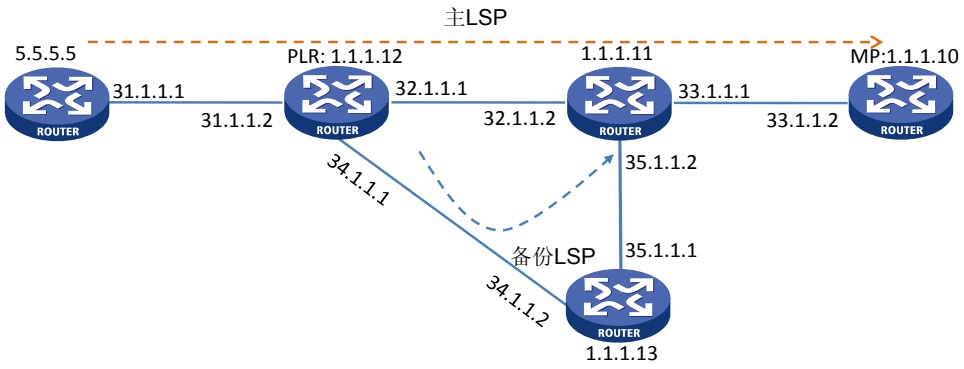


图14 链路保护

如下图所示，当PLR节点和MP节点之间有一台设备连接，当该设备失效时，PLR节点把流量切换到备份LSP，这种组网方式称为节点保护。

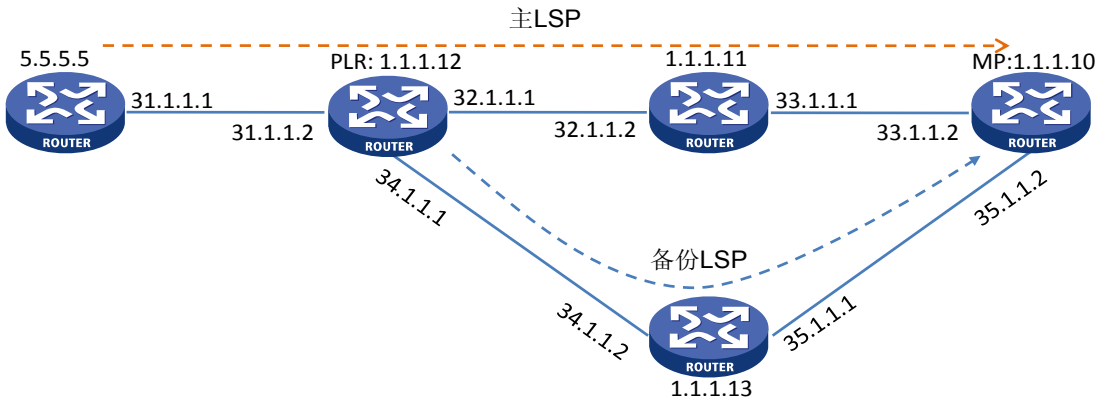


图15 节点保护

我司设备实现TE FRR技术，其节点保护和链路保护处理流程一样。在头节点使能FRR能力之后，下游节点向上游节点发送的RESV消息中的RRO对象中包含所有下游节点对应的Label子对象，其Local Protection Available标志位置1。下面是图15中PLR节点收到的下游节点1.1.1.11发送的RRO对象，从该RRO对象可知，PLR节点的直连节点是1.1.1.11，出标签是69249，非直连的下一跳节点是1.1.1.10，出标签是2287。当在PLR节点上主LSP经过的出接口上配置备份隧道时，PLR节点根据备份隧道的目的地址来确定MP节点，再根据MP节点找到FRR切换后的内层标签。如果MP节点是1.1.1.11，即直连下一跳节点，FRR切换需使用的内层标签是其分配的标签69249，这种情况属于链路保护；如果MP节点是1.1.1.10，是非直连下一跳节点，FRR切换需使用的内层标签是其分配的标签2287，这种情况属于节点保护。最后，在PLR节点上，主备LSP形成绑定关系，下刷控制平面和数据平面，为主LSP链路故障或者节点故障进行TE FRR切换做好了准备工作。

```
⊟ RECORD ROUTE: IPv4 32.1.1.2, Label 69249, IPv4 1.1.1.11, ...
    Length: 60
    Object class: RECORD ROUTE object (21)
    C-type: 1
  ⊞ IPv4 Subobject - 32.1.1.2
  ⊞ Label Subobject - 69249, , Local Protection Available
  ⊞ IPv4 Subobject - 1.1.1.11
  ⊞ IPv4 Subobject - 33.1.1.1
  ⊞ IPv4 Subobject - 33.1.1.2
  ⊞ Label Subobject - 2287, , Local Protection Available
  ⊞ IPv4 Subobject - 1.1.1.10
```

图16 RRO对象

不管节点保护还是链路保护，在PLR节点上的配置是一样的，就是在PLR节点主LSP出接口上手工配置保护隧道，主隧道和保护隧道能否形成主备关系，还需要判断保护隧道可用带宽，如果保护隧道的可用带宽大于等于主隧道带宽，主隧道和保护隧道形成主备关系，否则主隧道和保护隧道不能形成主备关系。

TE FRR切换之后，新LSP建立之前数据转发过程

TE FRR不管是链路保护还是节点保护都需要结合BFD检测机制或者RSVP Hello机制对主LSP进行故障检测，来达到TE FRR的快速切换。PLR节点检测到主LSP发生故障之后，会直接把流量切换到备份隧道转发，尽可能的减少数据丢包。

以 TE FRR节点保护方式举例说明TE FRR切换之后，新LSP建立之前的数据转发过程。

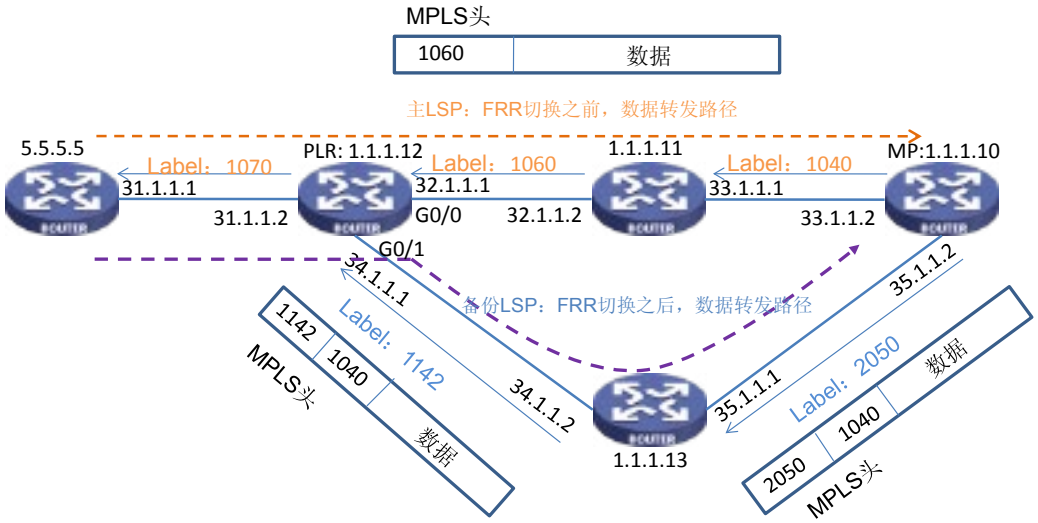


图17 TE FRR切换之后，新LSP建立之前的数据转发过程

FRR切换之前，数据报文通过主LSP进行转发，转发路径是5.5.5.5–>1.1.1.12–>1.1.1.11–>1.1.1.10。在PLR节点上，出方向的数据报文只需封装一层MPLS标签，标签值就是下游节点1.1.1.11向上游PLR节点分配的标签1060。当BFD或者RSVP Hello机制检测到节点故障后，PLR节点进入TE FRR节点保护流程，把主LSP上的MPLS数据报文以及协议报文在BypassSP隧道内转发。如图17所示，主备LSP会合节点即MP节点是PLR节点的非直连下一跳节点是节点保护方式，此时，PLR节点转发报文有两层标签，其内层标签是MP节点1.1.1.10向上游节点1.1.1.11分配的标签1040，外层标签是备份LSP对应的出标签1142。

从头节点5.5.5.5上指定源地址7.1.1.1 Ping MP节点上的loopback地址58.4.0.1，FRR切换之后，PLR

节点在出接口G0/1转发的报文截图如下：

```
⊞ Frame 8: 114 bytes on wire (912 bits), 114 bytes captured (912 bits)
⊞ Ethernet II, Src: Hangzhou_d2:8f:7f (c4:ca:d9:d2:8f:7f), Dst: 0c:da:41:b1:f9:4a (0c:da:41:b1:f9:4a)
⊞ 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 4095
⊞ MultiProtocol Label Switching Header, Label: 1142, Exp: 0, S: 0, TTL: 254
⊞ MultiProtocol Label Switching Header, Label: 1040, Exp: 0, S: 1, TTL: 254
⊞ Internet Protocol Version 4, Src: 7.1.1.1 (7.1.1.1), Dst: 58.4.0.1 (58.4.0.1)
⊞ Internet Control Message Protocol
```

图18 FRR切换之后，PLR节点转发的报文截图

MPLS TE FRR小结

本文详细讲解了Facility方式的TE FRR实现细节，再重新总结一下PLR节点在FRR切换前后的主要变化：

主备LSP绑定之后，FRR切换前：

在PLR节点上，主备LSP已经形成了绑定关系，并在控制平面，数据平面下刷主备LSP绑定关系以及相关数据信息。此时，PLR节点发送数据报文通过主LSP发送，封装一层MPLS标签。

当链路故障或者节点故障导致FRR切换后：

在PLR节点上，主LSP控制平面信息保持不变，只是其Path消息走Bypass LSP进行转发，同时Path消息中Hop地址变为实际发送的物理接口地址，且RRO本地出接口HOP的Flag中Local Protection in use位置1。此时，PLR节点通过备份隧道发送数据报文，封装双层MPLS标签，其内层标签是主LSP对应的出标签，外层标签是Bypass LSP对应的出标签。

VPN FRR 实现概述

文 / 蒋益群

前言

在网络高速发展的今天，三网合一的需求日益迫切，运营商对网络故障时的业务收敛速度非常重视，在任何一个节点发生故障时，相邻节点业务倒换小于50ms，端到端业务收敛小于1s已经逐步成为承载网的门槛级指标。

为了达到相邻节点业务倒换小于50ms、端到端业务收敛小于1s的要求，MPLS TE FRR技术、IGP路由快速收敛技术都应运而生。但上述方法均有其局限性，IGP FRR快速收敛速度与公网路由规模有关，网络中部署MPLS TE FRR比较复杂，而且上述方法都无法与私网直接关联，无法解决CE双归属网络中PE设备节点故障时端到端业务收敛慢的问题。VPN FRR致力于解决CE双归这种最普遍的网络模型的端到端业务收敛问题，将PE节点故障情况下的端到端业务的收敛时间控制在毫秒级别。

VPN FRR技术实现

VPN FRR技术主要应用于CE双归属网络模型，在远端PE上部署，并可以使用路由匹配策略挑选需要保护的远端CE路由，以解决主用PE故障时的业务端到端快速收敛问题。

CE双归属组网分析

以L3VPN为例，典型的CE双归PE的组网图如图1所示。客户端接入设备CE-A同时接入两台PE设备PE-A和PE-B，PE-A和PE-B为客户端CE同时提供多条出口线路互为备份，这种组网就叫CE双归属。对于这种冗余链路组网可实现的技术很多：VRRP，浮动静态路由，策略路由，动态路由协议等。

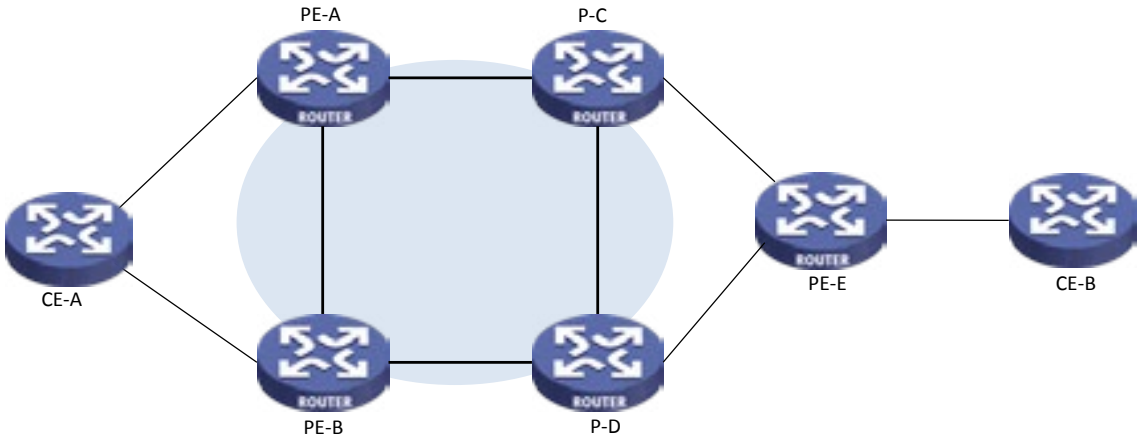


图1 CE双归属网络模型

假设图1组网中CE–B访问CE–A的路径为：CE–B——PE–E——P–C——PE–A——CE–A。当PE–A节点故障之后，CE–B访问CE–A的路径收敛为：CE–B——PE–E——P–D——PE–B——CE–A。

按照标准的MPLS L3 VPN技术，PE–A和PE–B都会向PE–E发布指向CE–A的路由，并分配私网标签；

在传统技术中，PE–E根据策略优选一个MBGP邻居发送的VPNV4路由，在这个例子中，优选的是PE–A发布的路由，并且只把PE–A发布的路由信息（包括转发前缀、内层标签、选中的外层LSP隧道）填写在转发引擎使用的转发项中，指导转发。

当PE–A节点故障时，PE–E感知到PE–A的故障（BGP邻居DOWN或者外层LSP隧道不可用），重新优选PE–B发布的路由，并重新下发转发表项，完成业务的端到端收敛，在PE–E重新下发PE–B发布的路由对应的转发项之前，由于转发引擎的转发项指向的外层LSP隧道的终点是PE–A，而PE–A节点故障，这段时间之内，CE–B是无法访问CE–A的，端到端业务中断。在传统技术中，端到端业务收敛的时间包括：1）PE–E感知到PE–A故障；2）PE–E重新优选PE–B发布的VPN V4路由；3）PE–E将新的转发项下刷到转发引擎中。很明显，步骤2和步骤3的速度与VPNV4路由的规模相关。当L3VPN中承载了大量的路由时，按照传统的收敛技术，当远端PE出现故障时，所有这些VPN路由都需要重新迭代到新的隧道上，端到端业务故障收敛的时间与VPN路由的数量相关，VPN路由数量越大，收敛时间越长。

VPN FRR基本原理

VPN FRR利用基于VPN的私网路由快速切换技术，通过预先在远端PE中设置指向主用PE和备用PE的主备用转发项，并结合PE故障快速探测，旨在解决CE双归PE的MPLS VPN网络中，PE节点故障导致的端到端业务收敛时间长（大于1s）的问题，同时解决PE节点故障恢复时间与其承载的私网路由的数量相关的问题。

VPN FRR外层隧道可以选择LDP LSP，也可以是RSVP TE，甚至可以是GRE等传统IP VPN隧道，我们利用检测技术感知外层隧道的状态，之后修改VPN路由迭代的外层公网隧道在转发引擎中的状态，从而进行快速的内层标签的倒换和备份路径的切换。因此在VPN FRR技术中，私网路由收敛的时间取决于两点：1、远端PE故障到本端检测到故障所用的时间；2、检测到故障后修改转发引擎中对应公网隧道状态的时间，实现此技术后VPN路由收敛速度与VPN私网路由的数量无关。

以图1中的典型CE双归属网络模型为例进行说明。PE-E设备可配置匹配策略选择符合条件的VPNV4路由，对于这些私网路由，除了包括优选的PE-A发布的路由信息（包括转发前缀、内层标签、选中的外层LSP隧道），同时添加次优的PE-B发布的路由信息（包括转发前缀、内层标签、选中的外层LSP隧道）作为备份。

当PE-A节点故障时，PE-E通过BFD、MPLS OAM等技术感知到PE-E与PE-A之间的外层隧道不可用，在典型组网中，端到端故障感知时间小于50ms。

当PE-E感知到MPLS VPN依赖的外层LSP隧道不可用之后，将LSP隧道状态表中的对应标志设置为不可用并下刷到转发引擎中，转发引擎命中一个转发项之后，检查该转发项对应的LSP隧道的状态，如果为不可用，则使用本转发项中携带的次优路由的转发信息进行转发，这样，报文就会打上PE-B分配的内层标签，沿着PE-E与PE-B之间的外层LSP隧道转发到PE-B，再转发给CE-A，从而恢复CE-B到CE-A方向的业务，实现PE-A节点故障情况下的端到端业务的快速收敛。

当VPN FRR与LDP FRR/MPLS TE FRR等技术组合使用时，遵循的原则是VPN FRR是比外层隧道切换级别要高的倒换技术，其故障检测时间需要配置得长于LDP FRR/MPLS TE FRR等外层隧道的故障检测＋隧道倒换时间，以保证在外层隧道能够进行倒换的情况下，不触发VPN FRR这种高级别的倒换技术，这正是网络中通用的低级别倒换优先原则的一个具体实例。

应用场景分析

VPN FRR的路径备份方式分为两种：VPNv4路由备份VPNv4路由；VPNv4路由备份IPv4路由。

VPNv4路由备份VPNv4路由

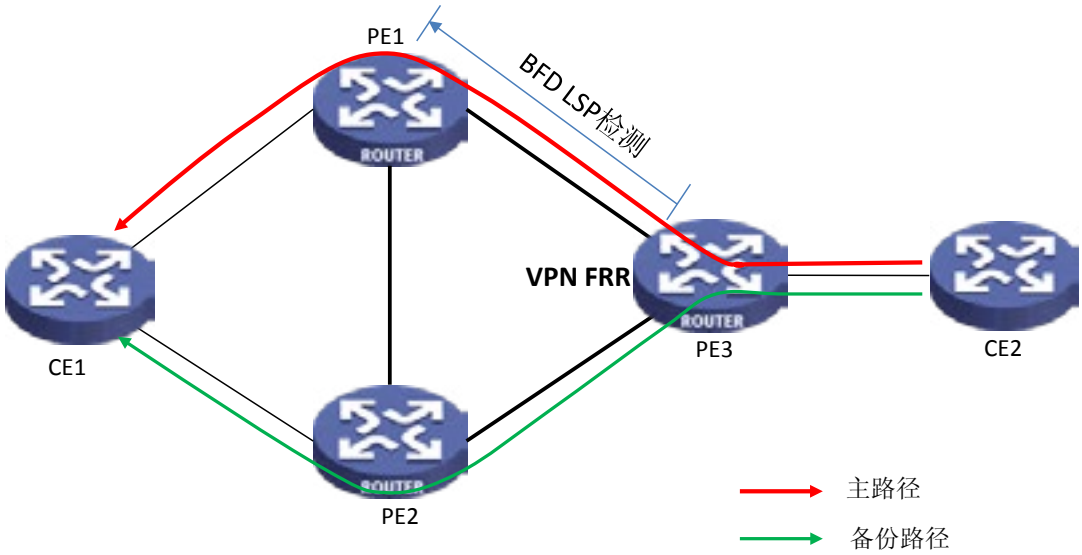


图2 VPN FRR场景1

参见图2，在入节点PE 3上指定VPN 1的FRR备份下一跳为PE 2，则PE 3接收到PE 1和PE2发布的到达CE 1的VPNv4路由后，PE 3会记录这两条VPNv4路由，并将PE 1发布的VPNv4路由当作主路径，PE 2发布的VPNv4路由当作备份路径。

在PE 3上配置BFD检测LSP功能，通过BFD检测PE 3到PE 1之间公网LSP的状态。当公网LSP正常工作时，CE 1和CE 2通过主路径CE2—PE 3—PE 1—CE 1通信。当PE3检测到该公网LSP出现故障时，PE 3将通过备份路径CE 2—PE 3—PE 2—CE 1转发CE 2访问CE 1的流量，此时私网标签和公网标签都会改变。

在这种备份方式中，PE3负责主路径检测和流量切换。

VPNv4路由备份IPv4路由

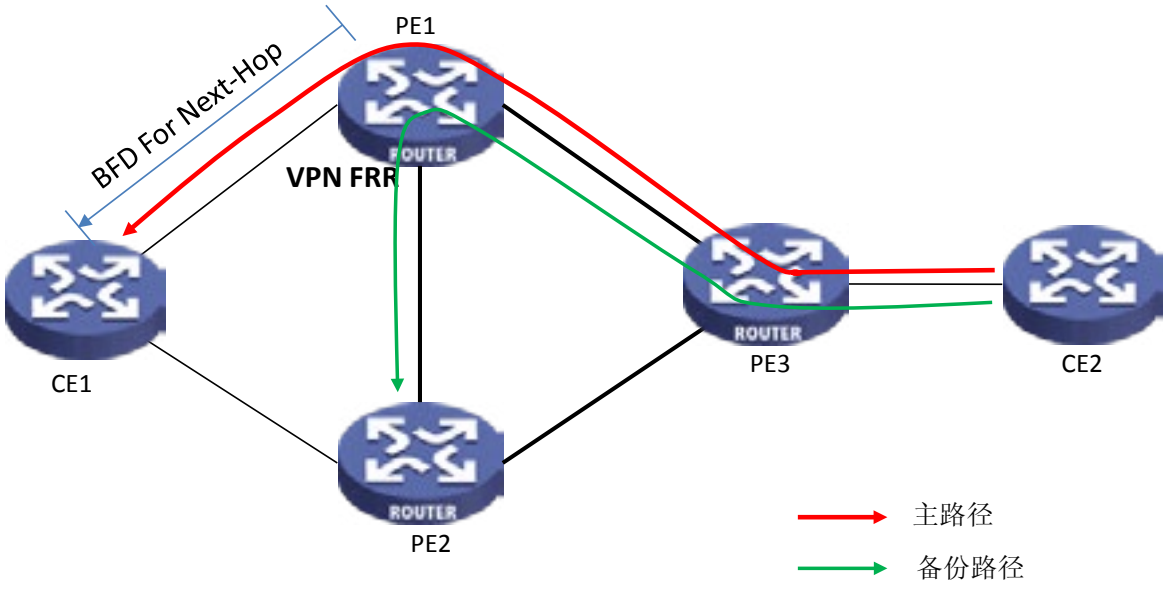


图3 VPN FRR场景2

参见图3，在PE1上配置VPN FRR。在出节点PE1上指定VPN FRR备份下一跳为PE2，则PE1接收到CE1发布的IPv4路由和PE2发布的到达CE1的VPNv4路由后，PE1会记录这两条路由，并将CE1发布的IPv4路由当作主路径，PE2发布的到达CE1的VPNv4路由当作备份路径。同时PE1将创建Echo报文方式的BFD会话，检测PE1—CE1这条路径的状态。

当主路径正常工作时，CE1和CE2通过主路径CE2—PE3—PE1—CE1通信。当PE1检测到路径PE1—CE1出现故障时，快速切换到路径PE1—PE2—CE1，CE2将通过备份路径CE2—PE3—PE1—PE2—CE1访问CE1。从而，避免路由收敛（切换到路径CE2—PE3—PE2—CE1）前VPN流量转发中断。由PE2通告的VPNv4路由备份从CE1直接过来的普通IPv4路由，私网侧链路故障时从普通IPv4路由向VPNv4路由切换完成FRR保护。

在这种备份方式中，PE1负责主路径检测和流量切换。

结束语

与MPLS TE FRR技术相比较，TE FRR着重解决公网隧道途经的链路和节点本地修复问题，而VPN FRR技术解决了在CE双归属网络中隧道尾结点发生故障或尾节点与CE之间链路发生故障时的快速收敛问题，故障恢复时间与私网路由的规模无关，并且简单、可靠，部署方便，而且除了PE之间的故障快速检测机制之外，不依赖于周边设备的配合。

VPN FRR关注的是内层标签，或者说内层隧道的快速切换，采用类似的技术，它同样在VLL/VPLS VPN中适用，并有效的缩短尾节点PE故障引起的业务中断时间。

MPLS TP

文 / 贾欣武

MPLS-TP产生背景

传统TDM传输网络面临的挑战

当前，在移动运营商网络中，类似3G数据回传的专线业务需求量很大。在之前2G网络中移动运营商基站回传的数据业务主要通过MSTP（multiservice transport platform ，多业务传输平台）来承载，但在3G回传应用中MSTP面临许多问题。3G网络中数据回传主要承载Internet应用，Internet业务具有突发性，某些热点地区的基站到基站控制器之间在某些时段可能需要很大的带宽才能满足其传输需求，尤其是智能终端的普及，使移动大流量应用更为普遍，这需要传输网络具有弹性更加智能。传统MSTP的优势在于承载TDM恒定带宽业务，其带宽是刚性恒定的，无法应付突发应用，在开通大量专线的情况下，会出现有的专线利用率不足有的专线可能带宽不够的情况。如果每条专线再加上保护链路的话该问题更加突出，因为其保护链路一直处于空闲备用状态。另外传统传输设备中继带宽不能统计复用，没有收敛比。由于以上原因，MSTP在承载3G数据回传业务时的每bit成本比较高。

近几年业界提出的PTN技术就是在上述需求背景下提出的，PTN在承载Internet应用时在成本和新业务扩展上拥有巨大优势。目前的MSTP网络本质上是一种电路交换网络，而PTN是包交换网络，其具有弹性管道，有统计复用，带宽规划可按收敛比、带宽利用率高等特点。 PTN的目的就是设计一种能够为分组报文（如IP报文）提供专线传输的廉价弹性网络，IP报文能够在端到端的专线中透明传输还能享受到类似MSTP的APS（automatic protection switching ，自动保护倒换）保护，且有丰富的OAM手段，在开通大量此类专线的情况下成本也可控，这就是PTN的初衷。

三种主要的PTN技术

PBB（ Provider Backbone Bridge ）与PBT（ PBB-TE）：将用户的以太网数据帧再封装一个运营商的以太网帧头，这时以太帧具有两个MAC地址。PBB的主要优点是：具有清晰的运营商和用户间的界限，可以屏蔽用户侧 PTN网络架构信息，实现二层信息的完全隔离，解决网络安全性问题。 PBB技术是在MAC-IN-MAC技术上发展起来的，尽管有一些厂商在坚持，而且在国内的个别运营商有部署，但总体呈没落趋势，未来不被看好。

MPLS-TP（MPLS Transport Profile ）及T-MPLS（Transport MPLS）：T-MPLS是ITU-T定义的一种PTN技术标准， T-MPLS的目标是成为一种通用的分组传送网， 是一种面向连接的网络技术，转发方式使用MPLS的一个功能子集。 2008年2月，IETF的MPLS Interoperability Design Team与ITU-T的T-MPLS特别工作组(T-MPLS ad hoc group)形成JWT（联合工作组）一起开发T-MPLS/MPLS-TP标准。JWT将T-MPLS更名为MPLS-TP，MPLS-TP在OAM方面与T-MPLS相比有较大的改动，在转发方式及原理上与T-MPLS基本相同。

CE（Carrier Ethernet）：分组传送演进的另一个方向，如VPLS、QinQ等，也属于利用IP数据网络模拟传输网的技术。VPLS技术属于MPLS L2VPN技术的一个子集，也具有一些类专线特征，也能满足PTN的部分要求。但其没有对提供商网络的隧道本身做明确的规范，在承载专线业务时没有考虑保护和OAM等内容。在PTN中，运营商网络的中的Tunnel才是技术重点，因为PTN中用Tunnel来仿真MSTP中的专线。

当前PTN技术的主流是MPLS-TP，相对于其他几种技术，MPLS-TP更符合PTN的要求，MPLS-TP的技术标准也是各大厂商激烈争夺的阵地。

MPLS-TP的技术特点

MPLS-TP目前主要用来在Internet 专线业务中替代传输网络，所以从设计之初就要求保留传输网络的一些优点，要求必须具备面向连接、双向隧道、高可靠性（保护倒换）、丰富的OAM等特性，因为在传输网络中这些都是基本功能。在具备了上述优势的同时又克服了TDM传输网络的部分缺点，MPLS-TP技术继承了MPLS TE的显式路由与带宽保证等优点，可以灵活地在网络中预留期望的带宽，与TDM传输网络固定接口带宽相比，带宽利用率提高，而且扩展灵活。

MPLS-TP与传统的MPLS技术相比主要有如下区别：

◆ 可承载多业务，业务层面采用PWE3技术，可以为TDM、ATM、Ethernet提供专线

◆ 可以预留带宽：在建立隧道时就可以预留带宽资源，为不同的业务提供不同的带宽时延保证

◆ 显式路由：建立的专线可以显式绕开拥塞节点，战略或战术实施流量工程

◆ 双向LSP：MPLS LSP都是单向的，而SDH传输网络通常使用的都是双向连接。因此MPLS-TP网络将两条方向相反的单向LSP组合成一条双向LSP

◆ MPLS-TP OAM：采用Y.1711 OAM（T-MPLS）或BFD For LSP（MPLS-TP）做CC/CV信令，还有LM/DM、LB、LT、VCCV等工具

◆ 保护倒换（PS）：采用保护倒换机制来保护专线，工作隧道检测到错误后就会自动切换到保护隧道。

◆ 双向倒换（双向高可用）：采用T-MPLS 定义的APS协议或者MPLS-TP定义的PSC协议来实现双向切换（这使得MPLS-TP专线更像一条SDH/SONET电路）

◆ 不使用倒数第二跳弹出PHP：采用PHP的话在隧道的尾节点无法区分哪个隧道过来的业务，传统的MPLS网络如VPLS中不需要知道从哪个隧道来的业务，只需要知道从哪里发出即可，而MPLS-TP网络中在尾节点上需要知道报文来自哪个隧道。

◆ 不使用ECMP，采用ECMP的话OAM报文可能经过不同的路径，使得OAM处理非常麻烦。

◆ 不使用标签合并操作：在利用LDP分发标签时标签合并是最常见的操作，可以在一定程度上节省标签和表项，但在MPLS-TP中不能采用标签合并动作，原因与前述两个原因类似。

从以上可以看出MPLS-TP与T-MPLS原理是完全相同的，仅仅是在OAM及双向切换信令的具体定义上有区别，T-MPLS由ITU-T主导，MPLS-TP是ITU-T与IETF成立的联合工作组开发的标准。T-MPLS中CC/CV采用Y.1711，双向切换采用的协议为APS协议，标签值为14；MPLS-TP中PSC协议采用GAL标签13，双向切换采用PSC实现。APS PDU定义也与PSC PDU定义不同，所以T-MPLS无法与MPLS-TP互通。MPLS-TP中专线业务封装格式如下：

承载以太业务的封装格式：

二层报头	隧道标签	PW标签	Ethernet Payload

承载ATM业务的封装格式：

二层报头	隧道标签	PW标签	ATM控制字	ATM Payload

承载TDM业务的封装格式：

二层报头	隧道标签	PW标签	TDM控制字	TDM Payload

图1 MPLS-TP报文封装格式

双向隧道

双向隧道是MPLS-TP中的关键技术之一，MPLS-TP双向隧道由正、反两个单向的CR-LSP组成。MPLS-TP双向隧道的建立有Co-routed方式、Associated信令方式和Associated非信令方式这三种方式。

Co-routed方式

在GMPLS（通用多协议标记交换）体系中提出了利用RSVP协议建立双向隧道的方法（RFC 3473）,在MPLS-TP中的双向Co-routed隧道也借用了此方法。其基本原理为：对RSVP-TE协议进行扩展，通过RSVP-TE信令协议建立MPLS TE双向隧道。通过Path消息将上游LSR分配的标签通告给下游LSR，在Path消息传递的过程中建立一个方向的CR-LSP，再通过RESV消息将下游LSR分配的标签通告给上游LSR，在RESV消息传递的过程中建立另一个方向的CR-LSP。Co-routed方式建立的MPLS-TP双向隧道的正、反两个方向CR-LSP使用的是相同的路径。

以下图为例说明Co-routed方式双向隧道建立过程：

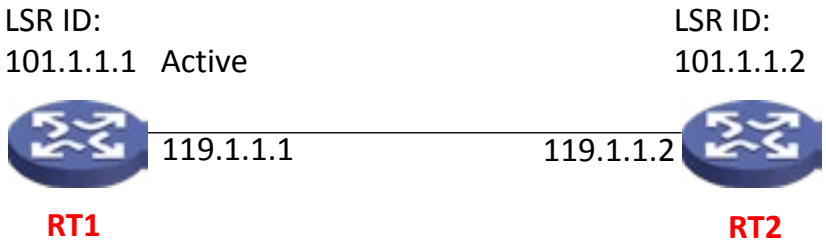


图2 Co-routed方式

1、在本例中RT1是Active端，RT1启动CSPF计算，计算时和传统MPLS TE计算不同的地方是：链路两端都满足约束条件才能参与计算

2、如果计算成功说明从RT1到RT2双向都能够建立满足约束条件的LSP

3、由RT1发送Path消息建立隧道，同时分配自己的入标签“3083”（RT2的出标签）

```
⊞ Frame 23 (238 bytes on wire, 238 bytes captured)
⊞ Ethernet II, Src: c4:ca:d9:3e:2f:00 (c4:ca:d9:3e:2f:00), Dst: 3c:e5:a6:45
⊞ Internet Protocol, Src: 101.1.1.1 (101.1.1.1), Dst: 101.1.1.2 (101.1.1.2)
⊟ Resource Reservation Protocol (RSVP): PATH Message. SESSION: IPv4-LSP, Des
  ⊞ RSVP Header. PATH Message.
  ⊞ SESSION: IPv4-LSP, Destination 101.1.1.2, Tunnel ID 1, Ext ID 65010101.
  ⊞ HOP: IPv4, 119.1.1.1
  ⊞ TIME VALUES: 30000 ms
  ⊞ SENDER TEMPLATE: IPv4-LSP, Tunnel Source: 101.1.1.1, LSP ID: 408.
  ⊞ SENDER TSPEC: IntServ: Token Bucket, 0 bytes/sec.
  ⊞ LABEL REQUEST: Basic: L3PID: IP (0x0800)
  ⊞ ADSPEC
  ⊞ EXPLICIT ROUTE: IPv4 119.1.1.2
  ⊞ RECORD ROUTE: IPv4 119.1.1.1
  ⊞ SESSION ATTRIBUTE: SetupPrio 7, HoldPrio 7, Label Recording,  [Tunnel1]
  ⊞ UPSTREAM LABEL: 3083
```

图3 Path消息

4、Passive端（RT2）收到后记录前一步Active端（RT1）为本端分配的出标签，并向上游发送RESV消息分配本端的入标签“4054”（上游的出标签）

```
⊞ Frame 24 (174 bytes on wire, 174 bytes captured)
⊞ Ethernet II, Src: 3c:e5:a6:45:25:00 (3c:e5:a6:45:25:00), Dst: c4:ca:d9:3
⊞ Internet Protocol, Src: 119.1.1.2 (119.1.1.2), Dst: 119.1.1.1 (119.1.1.1
⊟ Resource Reservation Protocol (RSVP): RESV Message. SESSION: IPv4-LSP, D
  ⊞ RSVP Header. RESV Message.
  ⊞ SESSION: IPv4-LSP, Destination 101.1.1.2, Tunnel ID 1, Ext ID 65010101.
  ⊞ HOP: IPv4, 119.1.1.2
  ⊞ TIME VALUES: 30000 ms
  ⊞ STYLE: Fixed Filter (10)
  ⊞ FLOWSPEC: Controlled Load: Token Bucket, 0 bytes/sec.
  ⊞ FILTERSPEC: IPv4-LSP, Tunnel Source: 101.1.1.1, LSP ID: 408.
  ⊞ LABEL: 4054
  ⊞ RECORD ROUTE: IPv4 119.1.1.2, Label 4054, IPv4 101.1.1.2
```

图4 RESV消息

双向隧道建立后在设备上显示信息如下：

```
[R1]display mpls lsp pro rsvp
-------------------------------------------------------------------------------
              LSP Information: RSVP LSP
-------------------------------------------------------------------------------
FEC              In/Out Label  In/Out IF       Vrf Name
*101.1.1.1/32      3083/NULL     RAGG101/-
101.1.1.2/32       NULL/4054     -/RAGG101
```

图5 双向隧道LSP

Associated非信令方式

通过配置手工将两条方向相反的单向CR-LSP绑定，从而形成MPLS TE双向隧道。绑定在一起的两条单向CR-LSP可以通过不同的方式建立。例如一个方向上的CR-LSP使用静态方式建立，而另一个方向上的CR-LSP使用RSVP-TE信令建立。绑定在一起的两条单向CR-LSP使用的路径可以相同也可以不同。此方式的特点：

◆ 可以完全不依靠RSVP信令来建立双向的LSP

◆ 两条方向相反的LSP关联起来的过程也不需要信令参与

◆ 双向的CR-LSR关联采用tunnel name + tunnel ID方式进行。如果反向隧道用信令建立就利用软状态信息绑定；如果反向隧道用静态CR-LSP建立就利用配置信息进行绑定。

Associated信令方式

利用信令将两条方向相反的单向CR-LSP绑定起来，此信令采用G-Ach（Generic Associated Channel）封装（G-ACh是隧道带内协议报文的封装方式，后面章节介绍），携带源端的LSP信息和目的端的LSP信息沿着正向LSP传送到目的端。如果目的端能够接收绑定消息并满足绑定条件则回应成功消息，双向绑定成功。目前ComwareV5不支持这种信令双向绑定方式，支持上一种Associated非信令方式。

MPLS-TP中的OAM

为了满足电信级网络管理和维护的需求，PTN网络中也要求有类似MSTP中丰富的SDH OAM功能，针对我司ComwareV5中的MPLS-TP OAM实现介绍如下：

主动故障管理：支持BFD Over G-Ach和Y.1711 OAM这两种方法，这两种方法都能够实现规范所要求的对隧道的CC（Continuity Check 连续性检查）和CV（Connectivity Verification，连通性检测）

按需故障管理：支持LB（Loopback，LSP环回ping）和LT（LinkTrace，LSP链路跟踪）

按需性能监测：支持LM（Loss Measurement，丢包测量）和DM（Delay Measurement，时延测量）

自动双向倒换信令：支持基于GAL和A-Ach的PSC协议

BFD 方式故障监测

在IETF草案draft-ietf-mpls-tp-cc-cv-rdi中建议MPLS-TP OAM中的CC/CV检测功能由BFD来实现。并定义了封装方式： MPLS-TP采用GAL（G-ACH标签，Label值为13）和G-ACH（通用关联信道）来表示Section和LSP的OAM报文。报文头如下：

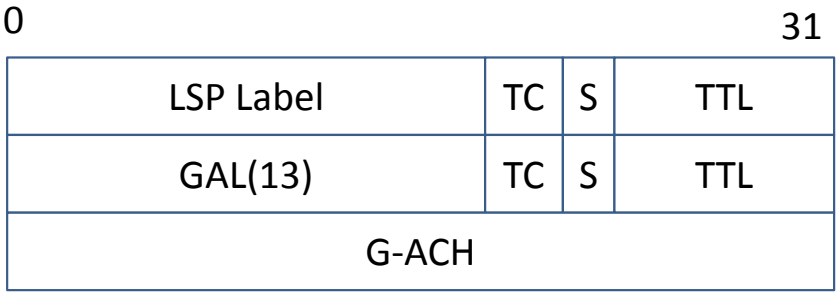


图6 GAL+G-Ach封装

其中GAL标签“13”在RFC5586中定义，采用G-ACH封装的BFD报文头格式如下：

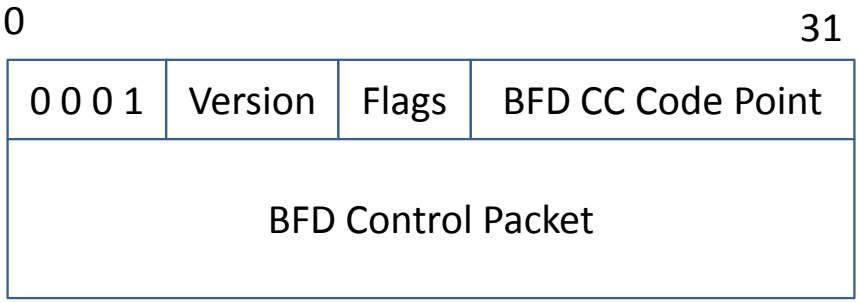


图7 采用G-Ach封装的BFD

采用G-Ach封装的报文以“0001”开头，版本和flags字段均为“0”，在我司Comware实现中，Channel Type如果为“0x0013”代表协商MPLS-TP BFD鉴别值的LSP ping报文（只有动态协商鉴别值时才用到LSP Ping）。

在G-Ach头之后是BFD报文（带有Router alert的IP报文）。BFD建立会话过程与通常BFD For LSP类似，遵循RFC5884。具体过程简述如下：

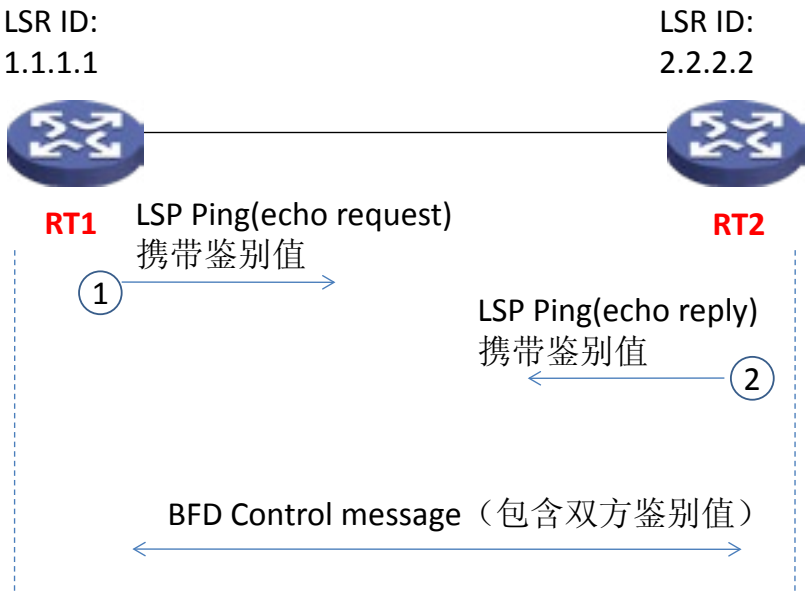


图8 BFD会话建立

首先RT1发送G-Ach封装的LSP ping报文（如下图），此报文是动态BFD在建立会话前的第一个协商鉴别值的报文，抓包解析如下：

```
⊞ Frame 71: 130 bytes on wire (1040 bits), 130 bytes captured (1040 bits) on interface 0
⊞ Ethernet II, Src: Hangzhou_ea:d5:24 (3c:e5:a6:ea:d5:24), Dst: Hangzhou_ad:89:f6 (3c:e5:a6:ad:89:f6)
⊞ 802.1Q Virtual LAN, PRI: 7, CFI: 0, ID: 1
⊟ MultiProtocol Label Switching Header, Label: 2069, Exp: 0, S: 0, TTL: 255
    0000 0000 1000 0001 0101 .... .... .... = MPLS Label: 2069
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...0 .... .... = MPLS Bottom of Label Stack: 0
    .... .... .... .... .... .... 1111 1111 = MPLS TTL: 255
⊟ MultiProtocol Label Switching Header, Label: 13 (Generic Associated Channel Label (GAL)), Exp: 0, S: 1, TTL: 255
    0000 0000 0000 0000 1101 .... .... .... = MPLS Label: Generic Associated Channel Label (GAL) (13)
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...1 .... .... = MPLS Bottom of Label Stack: 1
    .... .... .... .... .... .... 1111 1111 = MPLS TTL: 255
⊟ Generic Associated Channel Header
    Version: 0
    Channel Type: Unknown (0x0013)
⊟ Data (100 bytes)
    Data: 4600006c000240000111000001010101 7f000001940400 00...
    [Length: 100]

0000  3c e5 a6 ad 89 f6 3c e5  a6 ea d5 24 81 00 e0 01   <.....<. ...$....
0010  88 47 00 81 50 ff 00 00  d1 ff 10 00 00 13 46 00   .G..P... ......F.
0020  00 6c 00 02 40 00 01 11  00 00 01 01 01 01 7f 00   .l..@... ........
0030  00 01 94 04 00 00 05 ee  0d af 00 4c 00 00 00 01   ........ ...L....
0040  00 00 01 02 00 00 00 00  00 01 00 00 00 01 d6 99   ........ ........
0050  77 18 1e b8 00 00 00 00  00 00 00 00 00 00 00 01   w....... ........
0060  00 18 00 03 00 14 02 02  02 02 00 00 00 01 01 01   ........ ........
0070  01 01 01 01 01 01 00 00  00 3f 00 0f 00 04 00 00   ........ .?......
0080  02 43                                              .C
```

图9 协商鉴别值的LSP Ping

栈顶是MPLS-TP隧道标签2069，栈底标签是GAL标签13，随后是G-ACh头，之后是LSP ping (echo request)报文，由于是BFD动态会话（没有静态配置鉴别值），所以用LSP ping报文来协商BFD的鉴别值，此方式遵循RFC5884。

其中LSP Ping echo request的报文格式为：目的IP为127.X.X.X/8，目的端口为3503，遵循RFC4379。

对端收到后用LSP Ping（echo reply）回应，双方BFD session建立，后续将用BFD Control message来进行CC探测，回应的LSP Ping（echo reply）报文解析如下：

```
⊞ 802.1Q Virtual LAN, PRI: 7, CFI: 0, ID: 1
⊞ Internet Protocol Version 4, Src: 11.1.1.2 (11.1.1.2), Dst: 1.1.1.1 (1.1.1.1)
⊞ User Datagram Protocol, Src Port: lsp-ping (3503), Dst Port: vpvd (1518)
⊟ Multiprotocol Label Switching Echo
    Version: 1
  ⊞ Global Flags: 0x0000
    Message Type: MPLS Echo Reply (2)
    Reply Mode: Reply via an IPv4/IPv6 UDP packet (2)
    Return Code: Replying router is an egress for the FEC at stack depth RSC (3)
    Return Subcode: 1
    Sender's Handle: 0x00000001
    Sequence Number: 1
    Timestamp Sent: Jan 10, 2013 17:21:28.119995000 UTC
    Timestamp Received: Jul  7, 2007 10:53:19.418991000 UTC
  ⊟ Target FEC Stack
      Type: Target FEC Stack (1)
      Length: 24
    ⊟ FEC Element 1: RSVP IPv4 Session Query
        Type: RSVP IPv4 Session Query (3)
        Length: 20
        IPv4 Tunnel endpoint address: 2.2.2.2 (2.2.2.2)
        Must Be Zero: 0
        Tunnel ID: 1
        Extended Tunnel ID: 0x01010101 (1.1.1.1)
        IPv4 Tunnel sender address: 1.1.1.1 (1.1.1.1)
        Must Be Zero: 0
        LSP ID: 63
  ⊟ Unknown TLV type (0x000F)
      Type: Unknown TLV type (15)
      Length: 4
      Value: 00000243
```

图10 LSP Ping（echo reply）

回应的报文中最后的新增TLV字段中包含鉴别值“0x00000243”。

BFD检测报文发送的频率与产品实现相关，周期性检测失效后会触发保护倒换，关于保护倒换的细节在后面章节介绍。

Y.1711方式故障监测

在ITU-T的标准文档Y.1711中定义了一种检测MPLS 隧道的方法，目前我司ComwareV5中除支持BFD方式外也支持Y.1711方式作为MPLS-TP隧道的CC信令。与MPLS-TP BFD不同的是，采用Y.1711方式时栈底标签为“14”，其机制为头端发送CV或者FFD，尾端通过接收头端发送的报文来判断是否发生故障。如果尾端检测到故障则向头端发送BDI报文通知错误，头端收到故障报告后会作为保护倒换等操作的输入条件。

头端发送的检测报文如下：

```
⊞ Frame 2: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface C
⊞ Ethernet II, Src: Hangzhou_ea:d5:24 (3c:e5:a6:ea:d5:24), Dst: Hangzhou_ad:89:f6 (
⊞ 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 1
⊞ MultiProtocol Label Switching Header, Label: 2050, Exp: 0, S: 0, TTL: 255
⊟ MultiProtocol Label Switching Header, Label: 14 (OAM Alert), Exp: 0, S: 1, TTL: 1
    0000 0000 0000 0000 1110 .... .... .... = MPLS Label: OAM Alert (14)
    .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
    .... .... .... .... .... ...1 .... .... = MPLS Bottom of Label Stack: 1
    .... .... .... .... .... .... 0000 0001 = MPLS TTL: 1
⊟ Y.1711 OAM
    Function Type: FDD (Fast Failure Detection) (0x07)
    Error: these bytes are padding and must be 0x00
    LSR ID: 1.1.1.1
    LSP ID: 96119
    Frequency: 500 ms (0x06)
    Error: these bytes are padding and must be 0x00
    BIP16: 0x89fe
```

图11 Y.1711方式OAM

其中栈底标签采用14，代表Y.1711方式的OAM带内检测信令

Function Type为：FFD

LSR ID：头端的LSR ID

LSP ID：头端的LSP ID

Frequency：FFD报文发送的频率

BIP16：校验和

关于Y.1711方式MPLS OAM的报文内容和状态机在本期季刊有专文介绍，请参考，此处不再重复。

保护倒换

在传统的传输网络中利用APS（Automatic Protection Switching，自动保护倒换）技术来提高信道的高可用性，PTN为了实现高可用性，也有对应的自动保护倒换技术。在ITU-T的G.8131规范中定义了T-MPLS框架下的APS方法，在MPLS-TP的相关规范中，自动保护倒换的基本原理还与G.8131类似，但APS信令有了变化，MPLS-TP中的APS信令称为PSC（Protection Switching Control，保护倒换控制协议）。

在G.8131之前，ITU-T的Y.1720定义的PS（保护倒换）与G.8131的APS主要区别在于：PS仅仅能够支持单向保护倒换，并没有考虑双向隧道同时倒换这样的需求；而APS考虑了双向保护倒换这种情况，并设计了详细的处理流程和信令。在PTN中主要部署双向隧道，有双向保护倒换的需求，所以需要APS技术。

从广义上来说，MPLS TE FRR也属于PS的一种技术，TE FRR的优势是能够实现1:N保护，但TE FRR关注链路和节点的局部修复，不关注整个LSP路径，从原理上来说不属于路径保护；PS和APS关注

整个LSP（传输隧道）路径的通断，对于PTN来说，路径保护更有意义。

保护倒换方式

保护倒换有以下两种方式：

1:1保护倒换：在Tunnel的头节点和尾节点之间提供主备两条Tunnel。正常情况下，数据在主Tunnel上传输；当头节点通过检测机制（如MPLS OAM/BFD）发现主Tunnel发生缺陷或者收到对端PSC倒换通知，需要进行保护倒换时，将数据切换到备Tunnel上继续传输。1:1保护倒换支持单向隧道和双向隧道。

1+1保护倒换：在Tunnel的头节点和尾节点之间提供主备两条Tunnel。正常情况下，数据在主Tunnel、备Tunnel上都传输，尾节点选择接收主Tunnel或者备Tunnel的数据；通过检测机制（如MPLS OAM）发现其中一条Tunnel发生缺陷时，在头节点进行保护倒换，这时尾节点只能接收另一条隧道的流量。1+1保护倒换只支持双向隧道。

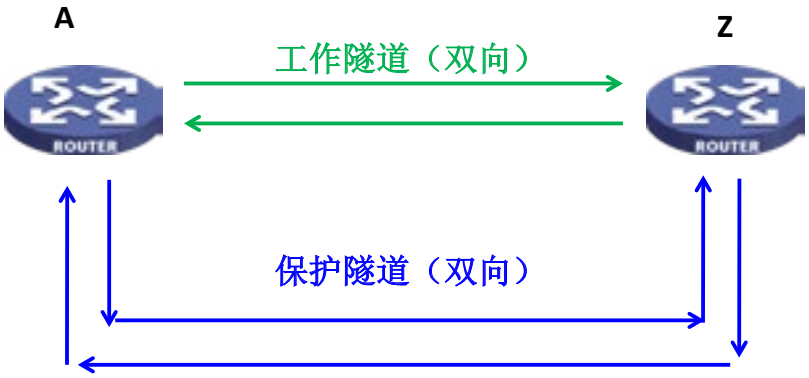


图12 保护方式

在1：1保护倒换时，工作隧道上承载业务，保护隧道上不承载业务，在发送端只发送一份业务，在接收端不需要进行选收判断；1：1保护的优点是节省资源。

在1＋1保护倒换时，工作隧道和保护隧道上都承载业务，在发送端会发送两份业务，在接收端需要进行选收判断，转发一份丢弃一份；1+1保护的优点是发生故障后切换快，丢包少。

保护倒换触发方式

触发保护倒换的方式有如下几种：

(1) 外部倒换是指通过手工配置的命令触发的保护倒换。外部倒换的优先级由高到低为：

清除倒换（Clear）：清除所有的外部倒换命令。

锁定倒换（Lockout of Protection）：数据流锁定在主LSP上传输。

强制倒换（Forced Switch）：强制数据流在备份LSP上传输。

手工倒换（Manual Switch）：手动将数据流从主LSP倒换到备LSP上传输。

(2)信令倒换（Signal Fail）是指通过协议信令触发的保护倒换。可以触发保护倒换的信令包括BFD检测MPLS-TE隧道功能、MPLS OAM、PSC信令。

外部倒换命令及信令倒换优先级从高到低依次为：

清除倒换

锁定倒换

备份LSP的信令倒换

强制倒换

主LSP的信令倒换

信令清除倒换

手工倒换

只有在手工输入的外部倒换命令比当前启用的倒换命令优先级高的情况下，外部倒换才会生效。设备上已经配置了外部倒换命令时，若要将其修改为低优先级的外部倒换命令，则需要先配置清除倒换（Clear）命令，再配置低优先级的外部倒换命令。

路径切换方式

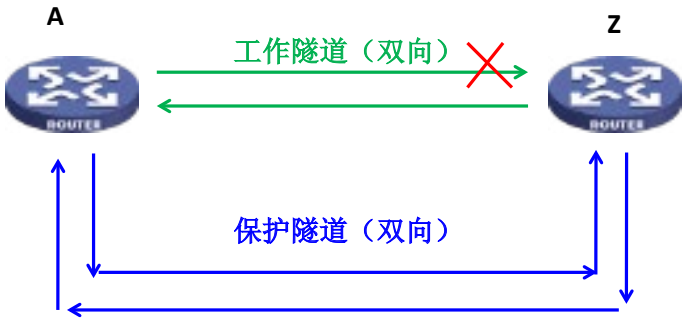


图13 切换方式

保护组内主备Tunnel的路径切换方式分为以下几种：

单向路径切换：进行保护倒换时，只在头节点进行倒换操作，不会通知尾节点进行相应的倒换操作。在上图中，如果在A端检测到A到Z的工作隧道发生故障，A发向Z的业务会通过保护隧道转发，但Z发向A的业务有可能还在工作隧道上转发。

双向路径切换：在头节点或者尾节点进行保护倒换时，通过PSC协议报文通知另一端节点进行保护倒换操作。只有双向隧道的保护组才支持双向路径切换。在上图中，如果在A端检测到A到Z的工作隧道发生故障，A发向Z的业务会通过保护隧道转发，同时A会发送PSC报文通知Z也进行切换，PSC协议规定如果本地检测到倒换事件需要向对端连续快速发送3个PSC报文，用于通知对方切换，这三个报文的间隔建议不超过3.3ms，后续报文应该以5s为间隔。如果是因为远端消息引发的状态变化则不需要快速发送报文。

在T-MPLS中定义的双向切换协议为APS协议；在MPLS-TP中定义的双向切换协议为PSC。

APS协议

T-MPLS中定义的APS协议报文格式与Y.1711定义的相同，也采用14标签（route alert标签），在Y.17TOM中定义APS的function type为0x0027，在G.8131中定义了APS PDU的格式：

<table>
<tr><td colspan="2">0</td><td colspan="2">31</td></tr>
<tr><td colspan="2">1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td rowspan="2">Request/ state</td><td colspan="2">Protection type</td><td rowspan="2">Requested Signal</td><td rowspan="2">Bridged Signal</td><td rowspan="2">Reserved</td></tr>
<tr><td>A</td><td>B D R</td></tr>
</table>

图14 APS PDU格式

其中各个字段定义如下：

Field		Value	Description	Priority
Request/State		1111	(LO)锁定倒换	highest
		1110	(SF-P)保护路径信令失败	↑
		1101	(FS)强制倒换	|
		1011	(SF)工作路径信令失败	|
		1001	为未来保留	|
		0111	(MS)手动切换	|
		0101	(WTR) 等待回切	|
		0100	为未来保留	|
		0010	为未来保留	|
		0001	(DNR)不可回切	↓
		0000	(NR)无请求	lowest
		Others	为未来保留	
Protection Type	A	0	非APS通道	
		1	APS通道	
	B	0	1+1 永久桥	
		1	1:1 选择器桥	
	D	0	单向切换	
		1	双向切换	
	R	0	不回切	
		1	回切	
Requested Signal		0	空信令	
		1	普通流量信令	
		2-255	保留	
Bridged Signal		0	空信令	
		1	普通流量信令	
		2-255	保留	

图15 APS各个字段定义

目前，T-MPLS标准不再推进，尽管有部分厂商支持T-MPLS定义的APS协议，但IETF主导的MPLS-TP是未来发展的主流，我司V5目前不支持APS协议，V5中支持的PSC协议（draft-ietf-mpls-tp-linear-protection）是基于MPLS-TP框架的协议。

PSC协议

PSC协议最新的标准文档为RFC6378：MPLS Transport Profile (MPLS-TP) Linear Protection，其中规定PSC报文采用RFC5586定义的G-Ach（Generic Associated Channel）封装，完整报文格式为：

二层报头	隧道标签	GAL（13标签）	G-Ach	PSC 协议

图16 PSC完整报文格式

其中G-Ach封装的PSC如下：

0 31

0 0 0 1	Version	Reserved	PSC-CT
Ver	Request	PT	R	Reserved1	FPath	Path
TLV Length	Reserved2
Optional TLVs

图17 PSC PDU格式

“0001”代表通用关联信道头，需要被CPU处理

PSC-CT（channel type）：0x0024

Version（G-ACh）：为0

Channel Type：我司实现为0x0012

Ach TLV Header和Optional TLVs为空

Version（PSC）：为0

其他字段定义如下：

Field	Value	Description
Request	14	（LO）锁定倒换
	12	(FS)强制倒换
	10	（SF）信令故障
	7	(SD)信令降级
	5	（MS）手动切换
	4	(WTR)等待回切
	1	(DNR)不可回切
	0	(NR)无请求
Protection Type	3	双向切换采用永久桥接
	2	双向切换采用选择器桥接
	1	单向切换采用永久桥接
	0	为未来保留
Revertive	0	不回切模式
	1	回切模式
Fault path (FPath)	0	保护路径异常
	1	工作路径异常
	2-255	为未来保留
Data path (Path)	0	保护路径不承载用户流量 (1:n) 或保护路径承载的用户流量是冗余的 (1+1)
	1	保护路径承载用户流量
	2-255	为未来保留

图18 PSC各个字段定义

处理流程（以本地强制倒换为例）如下：

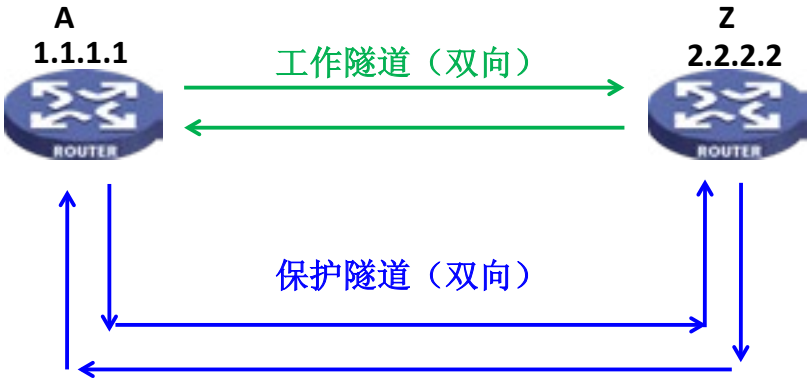


图19 保护倒换

◆ 在倒换之前A、Z设备上的保护状态

```
[A]disp mpls te protection tunnel all verbose
-----------------------------------------------------------------
Verbose information about the protection-group 1
-----------------------------------------------------------------
Work-tunnel id                : 3
Protect-tunnel id              : 4
Protection mode                : 1+1
Switching mode                 : bidirectional
Switch result                  : work-tunnel
Work-tunnel defect state       : No-defect
Protect-tunnel defect state    : No-defect
Holdoff                        : 0ms
WTR                            : 0s
Mode                           : revertive
Holdoff timer                  : not start
WTR timer                      : not start
The current external command    : --
-----------------------------------------------------------------
Total PG Num: 1

[Z]display mpls te protection tunnel all verbose
-----------------------------------------------------------------
Verbose information about the protection-group 1
-----------------------------------------------------------------
Work-tunnel id                : 5
Protect-tunnel id              : 6
Protection mode                : 1+1
Switching mode                 : bidirectional
Switch result                  : work-tunnel
Work-tunnel defect state       : No-defect
Protect-tunnel defect state    : No-defect
Holdoff                        : 0ms
WTR                            : 0s
Mode                           : revertive
Holdoff timer                  : not start
WTR timer                      : not start
The current external command     : --
-----------------------------------------------------------------
Total PG Num: 1
```

图20 倒换之前的保护状态

在A和Z上切换模式都是“bidirectional”表示双向切换，在本端发生路径保护倒换时会发送PSC报文通知对端进行保护倒换

◆ 在A设备上执行强制倒换：

A设备将业务切换到保护隧道，并且快速发送三个PSC报文通知Z端进行切换，PSC报文同时说明了切换原因等信息：

```
⊞ Frame 78: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
⊞ Ethernet II, Src: Hangzhou_ea:d5:24 (3c:e5:a6:ea:d5:24), Dst: Hangzhou_ad:89:f6 (3c:e5:a6:ad:89:f6)
⊞ 802.1Q Virtual LAN, PRI: 7, CFI: 0, ID: 2
⊞ MultiProtocol Label Switching Header, Label: 2091, Exp: 7, S: 0, TTL: 255
⊞ MultiProtocol Label Switching Header, Label: 13 (Generic Associated Channel Label (GAL)), Exp: 7, S: 1, TTL: 255
⊞ Generic Associated Channel Header
⊞ Data (34 bytes)

0000  3c e5 a6 ad 89 f6 3c e5  a6 ea d5 24 81 00 e0 02   <.....<. ...$....
0010  88 47 00 82 be ff 00 00  df ff 10 00 00 12 34 00   .G...... ......4.
0020  01 01 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

图21 发送给对端的PSC报文

Z端收到PSC通知后即使本地没有发生信令故障也会立即切换，状态如下：

```
[Z]display mpls te protection tunnel all verbose
-----------------------------------------------------------------
Verbose information about the protection-group 1
-----------------------------------------------------------------
Work-tunnel id                  : 5
Protect-tunnel id               : 6
Protection mode                 : 1+1
Switching mode                  : bidirectional
Switch result                   : protect-tunnel
Work-tunnel defect state        : No-defect
Protect-tunnel defect state     : No-defect
Holdoff                         : 0ms
WTR                             : 0s
Mode                            : revertive
Holdoff timer                   : not start
WTR timer                       : not start
The current external command     : remote force-switch
-----------------------------------------------------------------
```

图22 倒换之后的状态

状态显示当前的切换结果为“protect-tunnel”，本端切换的原因是“remote force-switch”。

PSC协议在发送中仅在保护路径发送，不在工作路径发送，原因是PSC的作用就是通知对方将业务切换到保护路径，切换的前提就是保护路径是通的，如果保护路径不通则切换的前提条件不存在，PSC协议也没有意义了，所以PSC只通过保护路径发送。

MPLS-TP主要应用场景

目前MPLS-TP主要应用在移动回传网络中，移动回传网络指基站到RNC/BNC（基站控制器）之间的专线网络。典型的MPLS-TP应用如下：

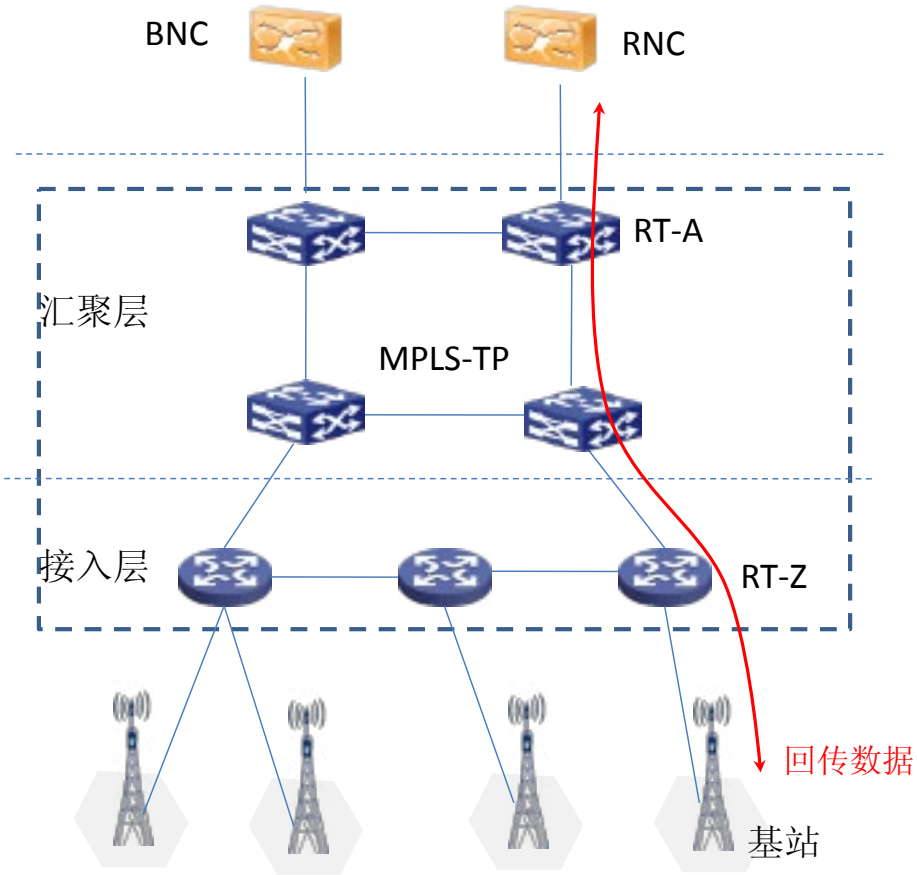


图23 移动回传网络

在上图中，PTN网络采用MPLS-TP技术，为基站与RNC/BNC之间的数据回传业务提供专线。RT-A工作在汇聚层，RT-Z工作在接入层，RT-A和RT-Z充当MPLS-TP网络中的PE设备，基站与RNC都为CE设备，基站和RNC之间的TDM业务、IP业务等均可以通过MPLS-TP网络提供的专线进行传送。

ISSU

文 / 蒋茂勇

前　言

ISSU（In-Service Software Upgrade，不中断业务升级）是一种可靠性高的升级设备软件的方式。它可以保证在系统软件升级过程中数据转发不中断。业界主要数据通信厂商思科、华为都推出支持ISSU特性的产品。我司在Comware V5平台上已经开始支持ISSU特性，在Comware V7平台上该特性得到全面加强。

ISSU的基本原理

Comware V7平台引入软件包概念，将系统软件中比较稳定的基础进程和相对比较活跃的业务进程分离，便于软件的管理和维护。软件包按功能划分成Boot软件包、System软件包、Feature软件包和补丁包。这些包是软件特性的基本单元，每个包提供特定的功能集：

◆ Boot软件包：包含操作系统内核，提供进程管理、内存管理、文件系统、应急shell等功能。

◆ System软件包：软件基础包，包含设备运行所必须的支撑模块和基本业务模块，比如设备管理、接口管理、配置管理和路由模块等。

◆ Feature软件包：用于业务定制的软件包，能够提供更丰富的业务。是否支持Feature包以及支持哪些Feature包与设备的型号有关。

◆ 补丁包（Patch包）：用于在不重启设备的情况下快速修复系统Bug。

◆ 多个软件包可以组合打包成IPE文件（Image Package Envelope，复合软件包套件），系统可以把它解压成单个的软件包，批量安装。通过包管理工具，可以在线安装、卸载补丁包或feature包，而不影响系统整体运行。通过分析升级前后版本间的差异部分，对差异程序实施不同的升级处理，为不同的

修改范围提供不同的ISSU方式。Comware V7系统支持如下几种ISSU方式：

◆ 热补丁升级：是一种直接而简单的ISSU方式。但是热补丁方式的升级只适合于内核简单错误的修改及已经运行的用户态进程的小范围修改，使用中有局限性。

◆ 增量升级：是一种进程级的ISSU方式。对于支持主备进程的模块，通过进程级备份及HA机制实现ISSU。对于单进程的模块，通过使用DBM对进程的配置与运行数据的进行存储与恢复来实现ISSU。

◆ 软重启：把系统运行瞬间的数据（运行数据、配置数据、硬件数据）通过DBM或内存文件存储到系统的备份内存。然后通过软重启CPU重新加载软件包，并从备份内存中恢复数据。CPU软重启期间，所在单板的硬件继续提供转发能力。

◆ 重启：通过主备倒换机制完成主控板的软件升级，通过热补丁、增量、软重启等机制完成业务板的软件升级。如果是主控出业务口的产品，可以结合聚合重定向功能，在单板重启前把已知单播流量重定向到聚合组的备份端口，在一定范围内提供ISSU能力。

Boot包只支持软重启和重启方式的ISSU，system包和feature包支持增量、软重启、重启方式的ISSU。

ISSU的实现机制

升级策略选择

ISSU升级前先要进行版本兼容性检查，只有兼容的版本之间才能进行ISSU升级。不兼容的版本，需要通过不兼容的方式升级版本。

在打包过程中，除boot包之外，在每个软件包中放入一个版本兼容描述文件。版本描述文件中包含该软件包的版本号、与之兼容软件包的版本号列表、它所依赖的软件包的版本号列表。其中对boot包的版本号采用平台版本号，system和feature包的版本号采用产品版本号。

System包的版本描述文件格式：

```
[current]
V600R001B01D009

[compatible]
V600R001B01D009
[boot]
7.1.039
```

Feature包的版本描述文件格式：

```
[current]
V600R001B01D009

[compatible]
V700R001B39D001
V600R001B01D009

[system]
V700R001B39D001
V600R001B01D009
```

版本的兼容性检查，分为纵向依赖性检查和横向兼容性检查两个方面。先进行依赖性检查，检查通过后，再进行兼容性检查。

System包的依赖包是boot包，feature包的依赖包是system包，纵向依赖性检查：

◆ boot升级，查看升级包的版本号是否是在当前系统system包的依赖包列表中，如果不是说明该boot包和当前系统的system包不配套，不能进行升级。

◆ system包升级，查看当前系统的boot包版本号是否是在升级包的依赖包列表中，如果不是，说明该system包和当前系统的boot包不配套，不能进行升级。

◆ system包升级，查看升级包的版本号是否是在当前系统的每个feature包的依赖包列表中，如果不是，说明该system包和当前系统上至少一个feature包是不配套的，不能进行升级。

◆ feature包升级，查看当前系统的system包版本号是否是在升级包的依赖列表中，如果不是，说明该feature包和当前系统的system包不配套，不能升级。

◆ boot包和system包同时升级，查看boot升级包的版本号是否是在system升级包的依赖列表中，如果不是，说明这两个被升级包是不配套的，不能升级。

◆ system包和feature包同时升级，查看system升级包的版本号是否是在每个feature升级包依赖列表中，如果不是，说明被升级的system包至少和一个被升级的feature包是不配套的，不能升级。

横向检查是待升级的包与当前运行包之间的兼容性检查，这个兼容性主要是模块或特性的板间通信的接口或消息格式的一致性：

◆ boot包升级，boot包不涉及板间通信，不需要兼容性检查。

◆ system包升级，检查当前系统的system包版本号是否是在升级包的兼容列表中，如果不是，检查升级包的版本号是否是在当前系统的system包的兼容列表里，如果也不是，则是版本不兼容，可以通过不兼容方式的流程进行升级，否则走兼容的ISSU方式升级。

◆ feature包升级，如果系统已经存在该特性的feature包，检查当前系统的feature包版本号是否是在升级包的兼容列表中，如果不是，检查升级包的版本号是否是在当前系统的feature包的兼容列表里，如果也不是，则是版本不兼容，可以通过不兼容方式的流程进行升级，否则走兼容的ISSU方式升级；如果该feature包还没在系统安装，按从无到有方式增量安装该特性包。

软件包以特性集为单位，采用三级版本号的方式统一管理升级策略。每个特性集的版本号用Major.Minor.Maintenance的形式编号，每级为一个数字，其中：

◆ Major：主版本号，这个版本号变化表示该特性需要重启方式升级；

◆ Minor：软重启版本号，这个版本号表示该特性需要软重启方式升级；

◆ Maintenance：维护版本号，这个版本号表示特性需要增量方式升级；

对于一个特性集，可能会经历多次升级，两个升级包的三级版本号也就存在不止一处变化情况。比如，STP特性先后有三个升级包A、B、C。A包和B之间，主版本号发生了变化，其它版本号相同，B包和C包之间，维护版本号发生了变化，其它版本号相同，那么A包和C包之间主版本号和维护版本号都不相同。对于这种情况，遵循重启 > 软重启 > 增量升级的原则来选择本次软件升级应采用的升级策略，因此A包和C包之间使用重启方式升级。

我们以MSR 36产品为例，对boot包、system包、feature包进行解包，看下这些软件包含的特性集列表及特性集的版本号描述。先解压boot包，boot解压后的文件结构如下：

```
C:\>tree  unpack_msr36-cmw710-boot-a0005.bin /f
卷 OS 的文件夹 PATH 列表
卷序列号为 0063002D E099:55E6
```

```
C:\UNPACK_MSR36-CMW710-BOOT-A0005.BIN
 │  8048 BASIC BootWare!
 │  package
 │
 ├──cen.elf_info
 │     cen.elf
 │
 └──script
    └──00003600
          boot.plist
          feature.lst
```

打开feature.lst，内容是boot.plist，所以boot包的特性集列表其实是在boot.plist中描述：

```
[app version]
disk.ko = 1.1.0
libc.so = 1.2.0
busybox = 1.0.4
os_kernel = 3.2.0

[lib version]

[plugin version]
```

特性集里包括应用程序进程或内核线程、链接库、插件三个子集，子集里的每个特性都用三级编号标识其版本号。特性集的子集可能为空，比如boot包的特性集里的链接库和插件子集都是空的。

system包的解压目录如下：

```
C:\UNPACK_MSR36-CMW710-SYSTEM-A0005.BIN
├─cen.cpio.gz_info
│  ├─etc
│  │  ├─comsh
│  │  ├─h3c
│  │  ├─hp
│  │  ├─license
│  │  ├─patch
│  │  ├─pkg
│  │  │  └─version
│  │  ├─raddb
│  │  ├─rbac
│  │  ├─scm
│  │  ├─snmp
│  │  ├─tcl8.5
│  │  ├─v5cfgconvert
│  │  └─xmlui
│  ├─lib
│  │  ├─cellular
│  │  ├─cli
│  │  ├─modules
│  │  ├─security
│  │  ├─snmp
│  │  ├─web
│  │  └─xmlui
│  ├─sbin
│  ├─var
│  │  ├─run
│  │  └─tmp
│  └─www
└─script
   └─00003600
```

在script/00003600目录下有存放system包的所有特性集列表，记录了system包里支持的所有特性集：

```
C:\unpack_msr36-cmw710-system-a0005.bin\script\00003600
2012/12/26  08:33    <DIR>          .
2012/12/26  08:33    <DIR>          ..
2012/12/26  16:06               747 aaa.plist
2012/12/26  16:06               126 acl.plist
2012/12/26  16:06               100 apr.plist
2012/12/26  16:06                71 aspf.plist
2012/12/26  16:06               100 bfd.plist
2012/12/26  16:06               275 bgp.plist
2012/12/26  16:06                69 ccf.plist
……................................................................................................
```

其中的bgp特性的版本描述文件bgp.plist：

```
[app version]
bgpd  =  1.0.14

[lib version]
libbgpip4u.so  =  1.0.1
libbgpipv4.so  =  1.0.1
libbgpvpnv4.so  =  1.0.2
libbgpcommon.so  =  1.0.5
libbgpip6u.so  =  1.0.1
libbgpipv6.so  =  1.0.1
libbgpvpnv6.so  =  1.0.3

[plugin version]
bgp.cli  =  1.0.11
bgp4.snmp  =  1.0.1
```

Voice包是MSR的一个feature包，解包后特性集列表如下：

```
iva.plist
em.plist
dpl.plist
msc.plist
sip.plist
cmc.plist
r2.plist
voice.plist
lgs.plist
```

其中的msc特性集的版本信息如下：

```
[app version]
mscd  =  1.0.3

[lib version]
libmsc.so  =  1.0.0

[plugin version]
msc.cli  =  1.0.2
```

软件包兼容升级时，包管理模块会遍历升级前后两个软件包内的特性集列表，以及两者间特性集三级版本号的差异，遵循重启 > 软重启 > 增量升级的原则来选择本次软件升级应采用的升级策略。比如升级system.bin时，通过对比升级前后包的特性集列表，发现有如下变化：

Route特性集的routed进程从1.0.6 变化至1.0.7（增量升级）

STP性集的stpd进程从1.0.19 变化至1.1.19（软重启升级）

基于软重启高于增量升级的原则，本次升级应采用软重启策略实施升级。

通过display version comp-matrix命令可以显示软件包版本信息以及在当前系统上的ISSU升级方式：

```
<H3C>display version comp-matrix file system cfa0:/S12500-CMW710-SYSTEM-
A732001sp2.bin
System image: cfa0:/S12500-CMW710-SYSTEM-A732001sp2.bin
  Version:
  S12500-CMW710-A732001
  Version compatibility list:
  S12500-CMW710-A732001
  Version dependency boot list:
  7.1.039
  Chassis   Slot          Upgrade Way
  2         0             Service Upgrade
  2         0.1           File Upgrade

 Influenced service according to following table on chassis 2 slot 0:
cfa0:/S12500-CMW710-SYSTEM-A732001sp2.bin
    routed
```

以上是Route特性集变化引起的增量升级，在显示版本升级方式的同时，也显示了本次升级影响的服务进程。

除了升级决策以外，升级前还要进行一些状态的检查、版本拷贝、版本校验等基础工作，在这过程如果出现错误，则终止本次升级，并给用户相应的提示信息。

增量升级过程

如果升级的范围只涉及用户态进程，并且升级版本所用的系统数据与当前版本完全兼容，可以使用

增量方式升级，也就是进程级别的升级。增量升级的操作过程在管理MDC控制，进程的升级过程在管理和用户MDC内同时进行。

增量升级可以分基于HA机制的增量升级和基于DBM的增量升级两种方式。基于HA机制的升级过程为：先升级备份进程到新版本（对业务运行无影响），通过HA把主进程的系统数据同步到升级后的备进程，再通过HA进行进程倒换，使新版本进程成为主进程，然后把备进程也升级成新版本，通过HA把运行数据同步新的备进程，从而完成版本更新。

HA机制的ISSU过程中，进程倒换使用的是进程异常并重启的倒换方式。ISSU模块在执行主备倒换前，调用HA模块的接口通知HA被升级进程下次倒换新Master所在板号。主备倒换时，先将被升级模块所在系统主控板的进程（不一定是该模块的主进程）重启，HA模块发现该模块存在进程重新行为后，进一步判断该进程是否需要主备倒换，根据ISSU模块注册的标记，决定进行主备倒换，把当前的主进程降级为备进程，根据新Master的板号，把已升级成新版本的备进程升级为主进程。

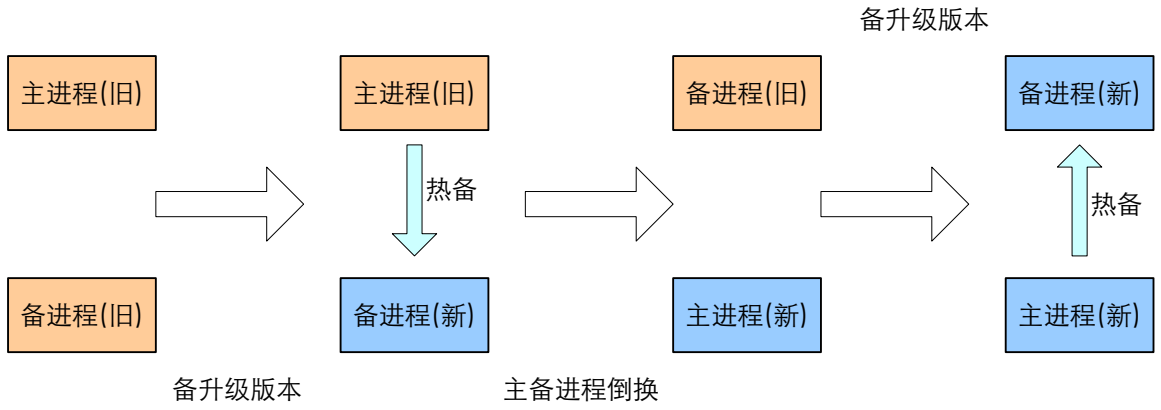


图1 HA机制的增量升级

对于不存在备份进程的服务可以使用基于DBM的增量升级方式。进程在升级前先把系统数据存入DBM，关闭当前被升级进程，将进程软件包升级成新版本，启动进程后从DBM中读出系统数据，恢复进程的系统运行数据和配置数据。由于不存在备份进程，升级过程相对比较简单。

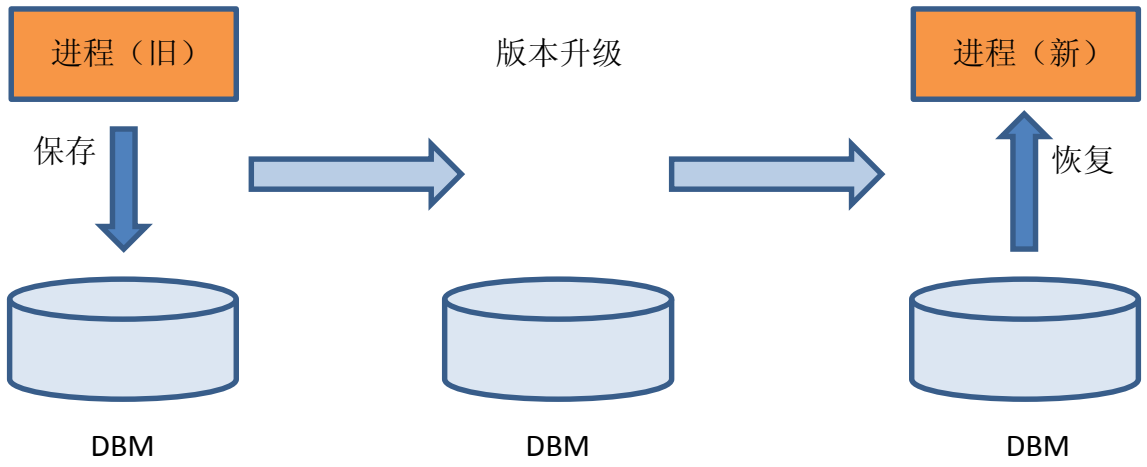


图2 DBM机制的增量升级

软重启升级过程

软重启的升级，主要是将系统运行瞬间的数据（运行数据、配置数据、硬件数据）全部保存在系统的Pramfs文件系统中，再重启CPU（重启期间硬件继续提供转发能力），启动新软件从上次保存的数据状态继续运行。软重启前的过程：

(1) 主控板上启动PTA（Packet Transmission Agent，发包代理）进程。接口软重启过程中，所有进程都会重启，在这期间运行在接口的模块无法和对端保持报文交互，可能导致对端协议的连接状态中断。为此在进行软重启前先在主控板启动发包代理进程，为模块进行发包代理。PTA进程优先在备用主控板启动。

(2) 主控板上发布软重启事件（所有主控板上都发布该事件），业务模块根据本模块特点对该事件做出响应，设备管理将接口板和主控之间的超时握手时间加长，避免接口板在软重启期间，因为握手超时而重启。

(3) 接口板上进行报文重定向，接口板下驱动通知把协议报文重定向到备用主控板处理，如果系统是单主控，则重定向到主用主控板。

(4) 接口板本板上报事件，通知本板的协议模块关闭对应的超时计时器。协议报文已经重定向到主板处理，而模块进程保存数据过程可能比较长，可能对导致部分模块进程认为连接超时，而导致协议状态变化错误，所以要关闭超时计时器

(5) 接口板本板上停止用户态进程，用户MDC内的用户态进程响应事件，把系统数据保存到DBM中，然后退出，DBM进程在有其他进程退出之后，把DBM数据保存在特定的内存文件。用户MDC的进程退出以后，管理MDC开始DBM数据的保存工作。

(6) 接口板通知内核模块保持数据，各个MDC都通知一遍，先用户MDC内核模块保存数据，然后管理MDC保存。内核数据直接保存在文件中。

(7) 接口板解压软重启需要的程序。

(8) 接口板通知主控板软重启准备完成，主控板等所有接口板均返回软重启准备完成后，对接口板下发软重启命令。

(9) 接口板收到软重启命令后发布软重启事件。

(10)接口板执行软重启动作。

软重启期间，驱动的VLAN配置、双工速率、QoS、端口转发状态等硬件数据保持不变，接口板软重启后：

(1) 内核初始化时各业务模块从高端内存文件中读运行数据恢复到内存中。模块和驱动约定采用两段式下发方式：

a) 第一阶段下发软重启前保存的数据，供驱动恢复软件表项，但驱动不需要真实下发硬件。

b) 第二阶段和主板数据进行平滑，此时下发动作和正常下发一致，驱动需要维护软件表项，也需要下发硬件。

(2) DBM先从根文件系统中读文件把业务数据load到内存中，各业务进程从DBM读数据进行恢复

(3) 驱动响应第三阶段结束事件取消报文上送主控板，改为本板自行处理，同时利用该事件清除系统的软重启标记。

软重启方式的升级，需要设备硬件支持，并且需要预留一定的系统内存。操作系统把这部分内存映射成Pramfs文件系统、做成块设备，内核启动后把系统的linux根目录挂接在这个块设备上，这样CPU软重启期间保存的系统数据不会丢失。

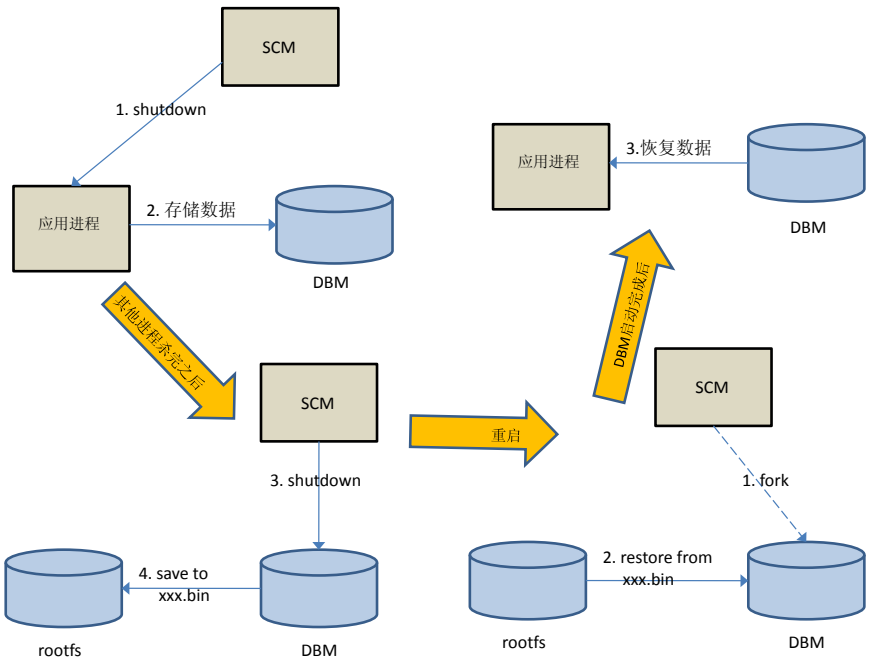


图3 软重启升级

重启升级过程

重启本单板加载新软件包完成升级。如果该单板有备板，可以利用主备板的切换来实施ISSU。重启升级方式在没有备板或者备板出业务口的情况下，软件包自身无法保证ISSU（比如接口板重启或单主控重启）。

重启方式的ISSU是单板级的ISSU。在主备环境下，先通过重启方式升级备板成新版本，然后主板把数据热备到新升级的备板，主板用旧的版本重启，触发主备倒换，之后对备用主控板逐一升级成新版本，并完成热备。如下图所示：

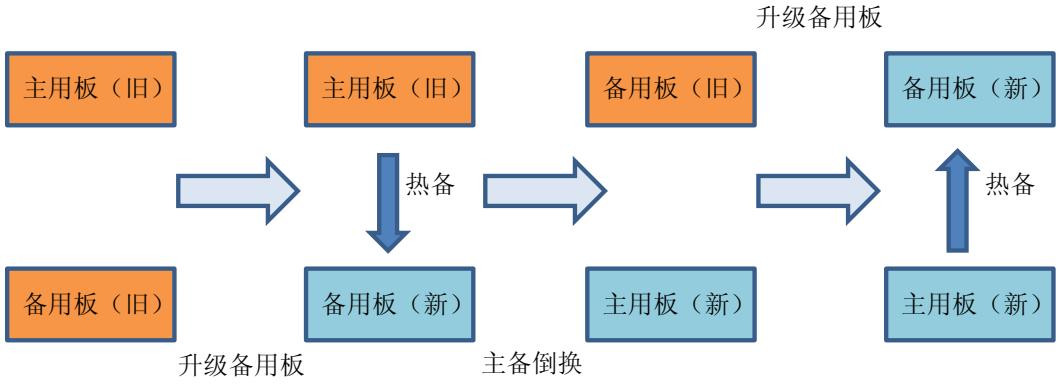


图4 重启升级

ISSU时序

具体应用的时候，一次ISSU升级过程，可能同时升级多个软件包，也可能同时升级不同类型的业务板卡、并且使用不同的升级方式。ISSU升级时遵守以下原则：

(1) 多主控的环境下，需要先升级备用主控板，再升级主用主控板。

(2) 升级方式遵循的时序原则为：增量升级–>逐次重启–>软重启–>重启。

(3) 除增量升级外，板类型遵循的升级时序原则为：接口板–>主控板（增量升级相反）。

(4) 本框的主控板负责本框单板的增量、软重启、重启升级。

(5) 该板决策为软重启时，但该板又不支持软重启，则更改该板升级方式为重启。

(6) 如果网板是重启升级时，需获取该板是否支持逐次重启，支持则更改升级方式为逐次重启。

(7) 如果本框没有备板，而主控板又决策为重启，则最终升级方式为整框重启。

(8) 如果非单框的IRF系统，IRF板的升级方式是重启，则最终方式为整框重启。

比如一个包对主控板、网板、接口板的升级方式分别为增量、逐次重启、软重启。那么升级的时候，先升级主控板，再升级网板，最后升级接口板。一个分布式的IRF系统包括多块主控板、网板、接口板，一次ISSU过程的时序是以上原则的总和。

逐次重启方式是一种特殊的重启方式的升级，只有网板才支持的这种升级方式。当网板需要重启升级时，为了避免流量中断，系统会自动升级完毕一块网板后，再升级下一块网板，这样能保证至少有一块网板在升级期间是正常工作的。因此，这种类型设备ISSU升级时，需要两块以上网板的作为硬件保证。

ISSU升级的时序是为实现ISSU功能做的系统设计，不同版本和产品的可能会做出调整，表现出差异性，但不会影响ISSU的功能和ISSU状态机的变化。

升级回滚

ISSU升级过程可能遇到异常情况，导致升级过程不能正常完成。用户也可能根据需要终止升级，回退回升级前版本。为此ISSU提供了两种回滚方式：一种是自动回滚，一种是手工回滚。如果ISSU过程中在某个特性的阶段停留的时间过长，导致计时器超时，ISSU会自动回滚回升级前版本，这就是自动回滚。用户也可以直接发命令，执行回滚操作，这就是手工回滚。

ISSU的升级过程

ISSU支持两种升级方式，一种是Install命令升级方式，另一种是ISSU命令升级方式。这两种方式均可以完成ISSU升级，其主要差异在于：

◆ Install命令仅适用与兼容版本之间的升级，Issu命令适用于兼容版本之间以及不兼容版本之间的升级。

◆ Install命令支持补丁的安装和卸载，ISSU命令不支持补丁操作。

◆ Install命令是对每块主控板实施升级操作，因此可以将每个升级动作对系统的影响控制在最小的范围内，属于高级应用方式，需要操作者对系统有较深入的了解，对升级每个动作所产生的影响有着准确的判断。

◆ ISSU命令属于批处理命令，在IRF环境中是以成员设备为升级对象，对操作者要求不高，适合大多数用户。

◆ Install命令支持Feature包的安装和卸载，ISSU命令支持对feature包的安装，但不支持卸载操作。

◆ Install命令可以删除设备上非激活状态的软件包，ISSU无此功能。

ISSU不支持多用户升级，必须等一个用户升级过程完成后，才能进行下一个升级过程。

Install命令方式的升级过程

1、升级前准备

升级前需先把软件包拷贝到设备存储介质的根目录，并保证备用主控板的存储介质上预留够的空间来安放软件包。正式升级前可以通过测试命令预知该软件包的升级方式。

2、激活软件包

Install命令方式和ISSU命令方式都能安装和升级软件包。对于Boot包和System包，在设备出厂时已经安装好，只存在升级的情况；Feature包和补丁包存在安装和升级两种情况。当Feature包或补丁包中的软件特性在系统中尚未运行时，加载或激活该软件包，是对该特性进行从无到有的安装过程；当Feature包或补丁包中特性已经运行的情况下，再次加载或激活包含该特性的一个新版本时，是对该特性的升级过程。

对于Install方式，第一次激活Feature包或补丁包，即安装新Feature包或补丁包时，必须先激活主用主控板；升级已有软件包时，若存在备用主控板，必须先激活备用主控板，再激活主用主控板。对于升级操作，从转发业务不中断角度考虑，要求用户必须从备用主控板开始升级。从系统的稳定性和可用性来说，即时用户从主用主控板开始升级过程，也不应该出现系统挂住或异常。

Install方式升级，用户操作的对象是主控板，对每块主控板依次实施升级操作。对于分布式设备，本框的接口板、网板、随本框的主用主控板一起升级。同种类型的升级，按先升级网板、再升级接口板、最后升级主控板的顺序进行；不同类型的升级，按增量、逐次重启、软重启、重启的顺序进行。非主控板的升级及时序由系统自动分析与控制，不需要用户干预。Install方式的安装与升级支持批量操作，可以一次同时激活多个软件包。

3、软件包确认

增量方式升级，软件包安装或升级后，只在设备本次运行过程中生效，用户只有通过命令确认后，才能使此次安装或升级的软件包在系统重启后仍处于激活状态；对于软重启和重启方式升级，用户在安装或升级软件包时，系统已经修改了下次启动软件列表，升级软件包会在系统重启后继续生效，不需要确认过程。需要注意的是，一个软件包，既有增量、又有重启或软重启方式的升级时，也不需要确认过程，升级效果永久生效。

4、软件包卸载

Boot包和System包是设备运行的基础软件，不支持卸载，只能卸载Feature包和补丁包。Feature包或补丁包被卸载后，可以通过install remove命令将卸载后的软件包从存储介质上彻底删除。卸载操作也只在设备本次运行过程中生效，用户只有通过命令确认后，才能使此次卸载永久生效。

5、软件包回滚

每次激活或者卸载软件包，系统中将运行新的软件特性集，系统将这些变化记录为回滚点。回滚点用自然数编号，依次递增。系统最多保存50个回滚点。当回滚点个数达到上限以后，最老的回滚点被删除，新的回滚点被保存。通过回滚功能，可将系统回滚到之前的某个历史状态。执行确认命令或者重启设备后，全部的回滚点都会被清除；当升级方式为软重启或重启时，旧的回滚点都被删除，只保留最近的一次回滚点，即只支持回滚到系统升级前的状态。

增量升级时，用户只有通过命令确认后，回滚操作的效果才有永久生效，否则只对设备本次运行过程中生效。

ISSU命令方式的升级过程

执行ISSU命令方式升级前同样要做软件包拷贝等基础准备工作。ISSU命令方式的升级是由状态机方式控制的升级过程。用户只需要下发命令依次进入每个升级状态，即可完成升级，不需要关心板间升级的先后顺序。ISSU命令方式升级主要有如下操作与状态。

(1) Issu load操作，对目标板或框执行升级操作，升级开始后，ISSU状态由Init状态进入loading状态，升级结束后进入Loaded状态，并开始配置备份与数据同步过程。对于非单主控的分布式设备，操作的目标必须是备板或备框。

(2) Issu run switchover操作，对于增量升级情况下，执行进程级倒换；对于软重启、重启升级，执行板级倒换。执行倒换命令前，会先判断HA的状态，如果备份或同步过程没有结束，会提示用户存在风险。倒换开始后，ISSU状态进入Switching状态，升级结束后进入Switchover状态，并开始备份与同步过程。

(3) Issu accept操作，取消自动回滚定时器，ISSU状态进入Accepted状态。该过程是可选过程。

(4) Issu commit操作，对未升级的单板或框执行升级操作，命令执行前，会先判断HA的状态，如果备份或同步过程没有结束，会提示用户存在风险。升级开始后，ISSU状态进入Commiting状态。全部未升级的单板升级结束后，ISSU状态重新回到初始状态。

(5) Issu rollback操作，在issu commit操作前，用户均可以执行该命令，终止升级并回退版本。

ISSU命令方式升级必须按照状态机的状态变化顺序依次进行，对非单主控的分布式设备，Load操作的目标必须是备板或备框，否则系统会给出相应的提示。ISSU命令方式升级适合大多数用户，是系统推荐的ISSU升级方式。在内部具体实现上，不同形态的产品有所区别。主要可以分这样几种产品形态：单主控设备、单台分布式设备、集中式IRF设备、分布式IRF设备。

1、单主控设备

单主控设备包括集中式设备、单主控的分布式设备、单台集中式IRF设备。对于业务模块来说，单主控设备没有备份进程，所以增量升级时，执行load过程完成后直接进入Accepted状态。此时用户可以通过issu rollback命令回滚到升级前版本，也可以通过issu commit命令结束本次升级，但不能自动回滚。

软重启、重启、或者不兼容的升级，执行load过程完成后，升级过程即结束，整个设备已经是完全更新到新版本。

2、单台分布式设备

典型的单台分布式设备存在双主控板、多块网板、块接口板。分布式设备存在单机设备和IRF两种情况，ISSU命令方式升级时，采取了不同的处理方式（Upgrade method）。对于两台及两台以上的分布式IRF系统，采用的是Chassis by chassis方式，以框为ISSU命令的操作单位，先升级一台standby设备，再升级包括Master设备在内的其它备。而对于单台分布式设备，采用的是Card by card方式，以主控板为ISSU命令的操作单位，先升级备用主控板，再升级主用主控板。

Card by card方式升级，用户执行issu load命令时需要输入备用主控板的板号，否则系统会提示错误，并终止当前操作。升级时，系统自动保证软件包和单板的升级时序。

3、集中式IRF设备

集中式IRF设备的ISSU命令方式升级，也是Card by card方式升级，先升级其中一个备板，然后再升级包括主板在内的其它板。不同的是，对于IRF环境，软重启或重启方式升级，执行主备倒换时，会选择已经升级成新版本的standby设备为新的Master设备，与普通的IRF选举规则不同。

4、分布式IRF设备

分布式IRF设备的ISSU命令方式升级，是Chassis by chassis方式升级，先升级其中一个备框，然后

再升级包括主框在内的其它框。Chassis by chassis方式升级在整体处理流程与状态机变化上与Card by card方式是一样的。不同的是,按框升级时load和commit操作都是对框操作，run switchover操作是对全局主控板操作。系统自动处理框内的升级时序，对于重启升级，如果被操作的框内只有一块主控板，则该框的升级方式退化成整框重启。

不兼容升级

升级的版本和原版本之间可能存在兼容性问题。比如，两个版本之间的板间通信接口不兼容了，此时需要进行不兼容方式升级。不兼容升级时，备用主控板（或IRF的standby设备）在升级成新版本启动后，首先进入RPR（Route Processor Redundancy，路由处理器冗余模式）。这种模式会block主控板间通信的物理口（即此时板间除了升级过程本身的通信报文和加载报文外，其他报文全部阻断）。对于分布式设备整个系统而言，此时是双Master系统。

因此对IRF系统进行不兼容升级时，会对loaded状态的设备进行MAD down处理。框式设备也会对loaded状态的主控板的网管口做MAD down处理，避免网络冲突。不兼容升级又叫对半升级，本身不能保证业务不中断。对于IRF系统，它能先升级系统中的一半的成员设备，然后再升级另外一半设备。这样升级过程中可以基本上维持一半的转发能力。差异性较大的版本之间可以选择这种升级方式。

ComwareV7 HA实现综述

文 / 吴玲

HA简介

HA概念

HA即High Availability，高可用性，从系统的角度来说，HA可作如下定义：系统各相关单元为达到业务的高可用性所进行的各种操作的总集。整个过程涉及的各单元包括硬件、业务模块及HA框架，涉及的各种操作包括批量备份、实时备份、主备切换等。

系统主：硬件或者STM（Stack Topology Management ：堆叠拓扑管理）确定的具有主身份的主用主控板。在Comvare V5上系统的主一定是业务的主，但是在V7上系统主不一定是业务进程的主。

主进程：实际提供控制、业务功能的实体。

备进程：不实际提供控制、业务功能的实体，处于备用状态，只能接收运行数据，不能发送运行数据。它随时准备代替主提供控制、业务功能的实体。

HA在Comware中的演进

HA在Comware中的演进主要有两个阶段，即从Comware V3的1:1备份演进到Comware V5 IRF的1: N备份。在这个过程中，变化的是备份板的数量，不变的是如下一些特点：

◆ 以板为控制单位，不面向业务级。即：当主用主控板不能正常工作时，备用主控板将迅速取代主

用主控板来处理业务，以保证设备的正常运行。

◆ 状态变更完全依赖驱动触发，属于被动受控。即：当主用主控板发生故障或者被拔出或通过命令行手工触发主备倒换。

V7上的HA将不受以上限制，有如下特点：

◆ 支持业务进程级的HA，即当业务主进程不能正常工作时，备进程将迅速取代主进程来处理业务，以保证业务的正常运行。

◆ 利用HA实现ISSU动态升级。ISSU的增量升级先升级备份进程到新版本，再通过HA进行进程倒换，控制权交给新版本进程，从而完成版本更新。

◆ 用户可以配置进程分布策略，执行分布优化命令，让主进程运行在期望的位置。

HA的业务结构和生命周期

业务的组织结构

在V7 HA环境下，各业务按照业务间约束关系，划分成单元（Unit）和组（Group），来实现HA的管理。在1：N的HA模型中，有两个概念：

◆ SU：Service Unit。系统中运行于一个Node（Node即节点，是指设备上的CPU，包括主CPU、辅助CPU、OAM CPU等）上的一个、或多个业务（如图1中SU1就包含了S1和S2业务）的集合。

◆ SG：Service Group。系统中运行在所有Node上的具有相同业务集的SU的集合。

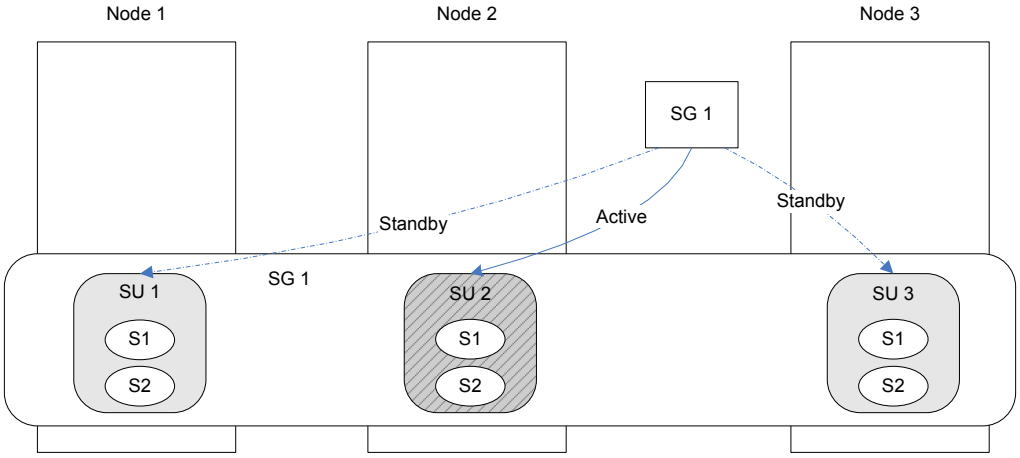


图1 HA的管理结构

大部分模块实现都是一个SU包括一个业务，也有少部分SU是包含多个业务的。SU中有多个业务时，若某个业务主进程发生异常，主SU中所有进程将被强制被动停止，转为备进程，被动停止不允许停止失败。停止完成后，HA选举一个新的主SU来继续工作。

HA的大部分控制是以SU为单位的。HA的批备、实备、平滑操作，以及状态、身份变化均以SU为单位执行。在一个SG的众多SU中，仅存在一个主SU（Active），其余都是备SU（Standby）。主SU上的进程为业务主进程，备SU上的为业务备进程。如上图1所示，SU2是主SU，SU2上的业务进程S1、S2是业务主进程；SU1和SU3是备SU，运行在SU1和SU3上的S1、S2是业务备进程。

业务间的关系

业务在被HA在管理之前，首先要按照一定的约束关系划分成若干个SG。划分SG后，还必须识别SU中多个业务之间的优先级。优先级的高低决定了备份和平滑操作的先后。只有优先级高的业务完成操作后，次优先级的业务才能开始相应的操作。SU多个业务之间的优先级通常根据依赖性来划分，被依赖的模块优先级高。各个SG间没有相关性，各个SG间可以并行进行批备、实备、平滑操作。

业务的生命周期

从业务生命周期来看，HA模块需要提供批备、实备、平滑管理功能，以及提供主动、被动倒换功能。

◆ 业务进程的启动

业务进程启动后首先向HA注册，由HA决策身份后（Active或Standby）方可继续运行。

◆ 业务进程的数据备份

业务主进程启动完毕后，会不断产生运行数据，需要将这些数据同步到其他备进程。HA提供点对多点的批备数据通道和实备数据通道，用于完成业务数据的备份。

◆ 业务进程的几种状态

批备发送：业务主进程发送批量备份数据。

实备发送：业务主进程初始化完毕，行使主控功能时，通常处于该状态。该进程检测到运行数据发生变化后，发送实时数据给备进程。

停止状态：业务进程需要由Active身份变成Standby时，先进入停止状态。如果停止失败，则转入平滑状态后返回Active状态；如果停止成功，则根据用户配置的亲和力值选出的备进程转入Smooth状态，同时处于停止状态的主进程转入Standby状态，然后处于Smooth状态的备进程经过平滑过程后进入Active状态。

降级状态：业务进程停止成功后进入降级状态，降级完成后业务进程变成Standby身份。

平滑状态：业务进程变成Active身份后，行使主控功能前，需要平滑自身业务数据。

实备接收：业务备进程初始化完毕，行使备份功能，通常处于该状态。

批备接收：业务备进程接收批量备份数据。

◆ 业务进程的倒换

业务进程的倒换是指HA控制业务进程在Active和Standby状态间进行切换。V7的倒换分两种：主动倒换、被动倒换。

◆ 业务进程的终结

在任意状态去注册HA，均可进入终结状态。

HA的三种通道

实际运行时，每个业务进程都有三种通道，分别为：控制通道、批量备份数据通道、实时备份数据通道。

控制通道和实时备份通道是在进程启动阶段，HA就为该业务进程创建（实备通道建立是指进程初始化过程中主备进程加入组播组，即：主进程时启动，以P身份先加入组播组，此时通道就建立了，如果备

进程没有启动主进程发送的数据没有接收者。当备进程启动后，以S身份加入组播组，此时主进程再发送的数据备进程就可以收到），业务进程退出时才撤销；批量备份数据通道是业务进程批量备份时，根据业务进程的优先级，在主进程和备进程中建立的，业务批备完后，撤销批量备份数据通道。

控制通道使用单播方式，用于连接HA主进程和业务主进程。批量备份数据通道和实时备份数据通道使用组播方式，用于连接业务主进程和业务备进程，分别用于承载业务进程的批备、实备数据。

HA的两种倒换方式

主动倒换

主动倒换：指通过命令行、或某种配置策略、或业务升级时，在保持主控节点正常运行的前提下，把备变为主、同时把主变为备的过程，又称平滑倒换，V5不支持该倒换方式。

V7上有部分不支持进程倒换的业务，当该业务进程在发生异常（如致命错误）或主动终止（如升级）时，主身份的进程永远跟随系统的主（Master）走，而不是由HA来动态决策。目前如：ACL、AAA等进程都不支持由HA动态决策，对于这些进程配置重优化策略时，会提示不支持。 对于V7上支持主动倒换方式的业务进程，如：STP、OSPF、 ISIS 、BGP等，配置进程优化策略后，再执行进程分布优化，就可以达到让主进程运行在期望的位置上，这样就能充分利用多个节点的CPU和内存资源，提高了业务的高可靠性，更有效地实现系统的负载分担。

用户可以通过命令行配置进程的亲和力来使业务运行在所选择的节点上，包括以下几类：

◆ 具体位置亲和力（affinity location-set）：表示业务对指定节点具体位置的偏好，节点位置指的是节点的绝对物理位置，即通过chassis/slot/cpu唯一确定的节点位置，其中，chassis参数只对分布式IRF设备有效，CPU参数只对存在多块CPU的单板或成员设备有效，只支持指定主控板上的CPU，不支持接口板的CPU配置。

◆ 指定节点类型的亲和力（affinity location-type）：表示业务对某个类型位置的偏好，系统有3种节点类型：

1) current：主进程当前运行的位置；

2) paired：所有备份进程当前运行的位置；

3) primary：系统的主用主控板（分布式设备），或全局主用主控板（分布式IRF设备），或Master（集中式IRF设备）。

◆ 不同业务间亲和力（affinity program）：表示本业务和其它某个业务运行于相同位置的偏好，即不同进程之间的依赖关系；

◆ 业务多实例间亲和力（affinity self）：表示同一业务的多个实例间运行于相同位置的偏好；

亲和力的偏好分为正向偏好和反向偏好，取值范围为[1，100000]。正向偏好表示互容性，反向偏好表示互斥性，值配置的越大表示要求越强烈。另外正向和反向都有缺省偏好，设备的缺省偏好值为200。配置亲和力前，需要执行placement program命令进入具体的业务配置视图。当亲和力配置完成后，不会立即改变进程的分布，需要执行placement reoptimize命令，HA才会对SU的位置分布进行优化，优化后的分布能更好地满足亲和力配置要求，使业务运行在最合适的节点上。

配置举例：

#配置OSPF进程对1号槽主控板的正向偏好值100

<Sysname>system-view

```
[Sysname]placement program ospf
[Sysname-program-ospf]affinity location-set slot 1 attract 100
#配置LSM进程对current节点类型的反向偏好值50000
[Sysname]placement program lsm
[Sysname-program-lsm] affinity location-type current repulse 5000
#进行进程分布优化
[Sysname]placement reoptimize
Predicted changes to the placement
Program                    Current location      New location
-----------------------------------------------------------------
lsm                        1/0/0                 1/1/0
routepolicy                1/0/0                 1/0/0
slsp                       1/0/0                 1/0/0
rib6                       1/0/0                 1/0/0
rib                        1/0/0                 1/0/0
staticroute6               1/0/0                 1/0/0
staticroute                1/0/0                 1/0/0
eviisis                    1/0/0                 1/0/0
stp                        1/0/0                 1/0/0
ospf                       1/0/0                 1/1/0
Continue? [y/n]:y
Re-optimization of the placement start. You will be notified on completion
Re-optimization of the placement complete. Use 'display placement' to view the new
placement
[Sysname]
#显示分布优化后进程的运行位置
<Sysname> dis placement location all
Program(s) placed at location: 1/0/0
  syslog
  l3vpn
  dns
  track
  lagg
  ipaddr
  trange
  rtm
  routepolicy
  ip6addr
  lauth
  slsp
```

```
  rib6
  rib
  acl
  aaa
  staticroute6
  tunnel
  staticroute
  qos
  copp
  ethbase
  ipcim
  ip6base
  ipbase
  bfd
  eviisis
  stp
  xmlcfg
Program(s) placed at location: 1/1/0
ospf
  lsm
<Sysname>
```

被动倒换

被动倒换：现有主控节点异常而触发状态机被动的进行倒换。如：插拔主进程所在的板或reboot主进程所在的板，都属于被动倒换方式。

HA的备份方式

HA备份与DBM备份对比

在V7上DBM（Database in Memory：内存数据库）也提供了主备备份的功能，DBM的备份分成批量备份和实时备份。当需要备份的数据发生改变时，通过DBM模块修改备份数据；当需要进行备份数据恢复时，直接从DBM模块获得备份数据。 DBM会根据系统运行的情况将数据保存在适当的场所，各模块无需关心DBM如何保存数据。

HA与DBM两种备份方式的特点参考如下：

备份方式	HA备份	DBM备份
主备状态管理	支持	不支持
空间占用	低	高。 有冗余数据

备份性能	高	低
倒换恢复性能	高 备进程直接接替	低 需从DBM恢复
适用场景	有主备进程； 接口板不适用	可以无需主备环境 接口板可以存储DBM数据 但不支持跨板备份
备份数据的统一控制	不可以	可以
实现复杂度	高	低
数据源限制	主进程可分布在主控板或备板上	必须在系统主控板
运行数据量	适用于运行数据量比较大的模块	适用于运行数据量比较小的模块

实备

主、备进程完成了批量备份后，主进程上的运行数据和配置数据发生变化，需要将实时备份数据同步到所有备进程。

实时备份过程：

1) 业务主进程运行时产生实时备份数据；

2) 业务主进程通过实时备份数据通道发送运行数据；

3) 判断当前是否正在批量备份，若正在批量备份，将运行数据插入到批量备份数据通道中，具体参见图3；

4) 所有就绪的业务备进程从该通道接收数据。

HA允许业务主进程在批备期间，发送实备数据。通过把实备数据插到批备通道中发送，保证批备、实备数据的时序，即：业务主进程交替发送批备/实备数据时，业务备进程按照发送的顺序收到批备/实备数据，如下图所示。

“圈”表示实备消息，“框”表示批备消息，中间的数字表示报文发送顺序

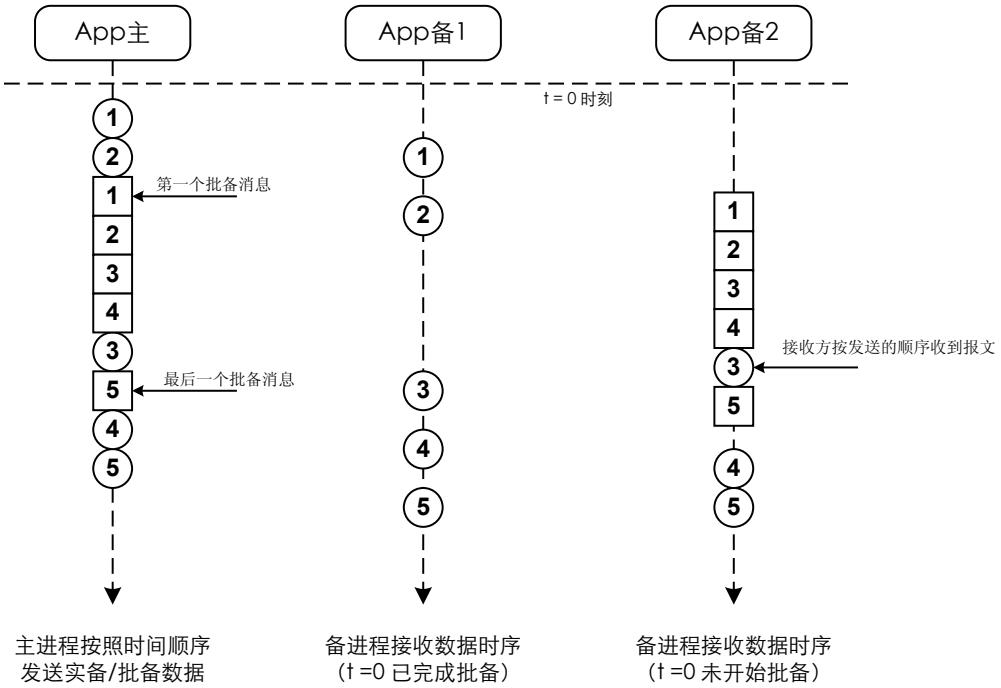


图2 业务备进程接收数据时序

批备

批量备份是业务主进程向多个业务备进程组播批量发送数据的过程，在此过程中，如果又有新的备份节点加入，则需要等待本次批备结束后再进行批备。

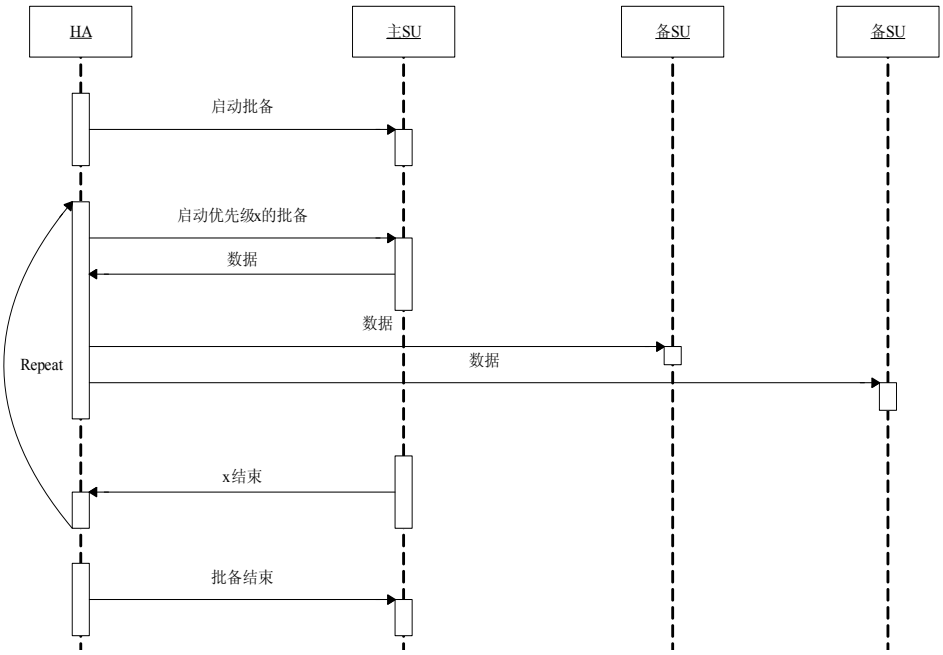


图3 批量备份

具体过程如下：

1) HA主进程从当前需要批备的业务里选出优先级最高的业务，加入到

到批备组播组中；

2) 各个业务加入完毕后，回复HA主进程的业务处理模块：“I’m ready”；

3) HA主进程通知业务主进程“start batch”；

4) 业务主进程使用组播通道发送批量数据给业务备进程；

5) 业务主进程向HA回复“send over”；业务备进程收到标识为结束的数据报文后，向HA回复“receive over”；

6) HA主进程通知业务进程撤销批量备份数据通道；

7) HA主进程按照优先级关系选取下一个业务进程，直到所有选中的成员均回复后，通知FSM（Finite State Machine）状态发生转换。

平滑

业务进程由备身份切换成主身份时，由HA通知业务进程开始平滑；具体平滑的过程由各业务模块自己完成，业务进程平滑完毕后，需要通知HA主进程平滑结果。业务进程可以在HA通知开始平滑时，同步业务时隙数据（运行数据可能在业务进程切换身份过程中发生了变化，这部分变化的数据称为时隙数据）。

平滑由于是各业务模块自己完成，所以不同的业务模块可能在平滑方式上存在不同，但是大致过程

如下：新的主控节点向所有受控节点收集数据，和自己已有的数据比较，进而消除两者之间的差异； 同时在平滑过程中，需要向新的备份节点备份平滑导致的数据变化，最终达到全局一致，平滑完毕。即：新主控板先从接口板拉时隙数据，再推给各备板，从而达到所有板的数据一致（适用于接口板上存在运行数据的模块）；对于接口板上没有运行数据的模块，如单播路由模块，由于V7上接口管理现在做成完全分布式，单播协议倒换后从本板（本板指的是协议主进程所在的板）的接口管理去拉接口数据即可。

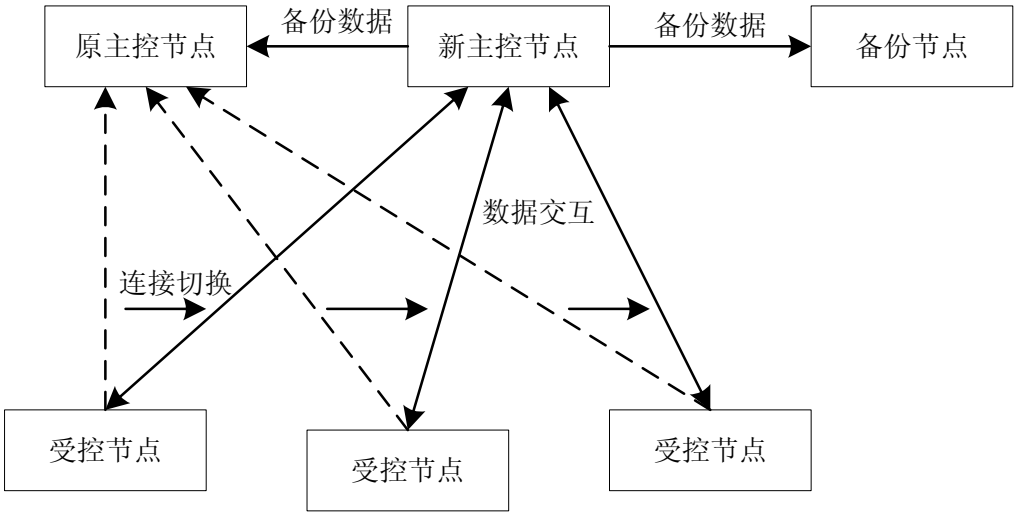


图4 平滑

网络之路

主办单位：H3C测试中心

策　　划：刘　宇 陈旭盛 杜祥宇

主　　编：陈　翀

本期编委：陈旭盛 吴　帅 王茂葳 涂勇军 李　冰 张　岩 乔　辉 苏艳梅 李　军 李连华
曹佐清 陈锦军 李朝辉 叶玉其 丁　炎 苏艳梅 蒋益群 贾欣武 蒋茂勇 吴　玲