# SRv6 网络编程

作者：王健　　指导老师：陆秋文

## 1 技术概述

分段路由（SR）协议由思科公司于 2013 年提出，它是一种源路由协议，支持在路径的起始点向报文中插入转发操作指令来指导报文在网络中的转发。其核心思想是将报文转发路径切割成不同的分段，并在路径的起始点往报文中插入分段信息指导报文转发。这样的路径分段被称为“Segment”，并通过 SID 标识。

目前 Segment Routing 支持 MPLS 和 IPv6 两种数据平面，基于 MPLS 数据平面的 Segment Routing 被称为 SR-MPLS，其 SID 为 MPLS 标签；基于 IPv6 数据平面的 Segment Routing 被称为 SRv6，其 SID 为 IPv6 地址。

下面通过一个比较经典的例子介绍 SR 的原理。假设机场行李托运的路线为德国慕尼黑-> 美国亚特兰大->日本东京，因为每个机场都有自己唯一的代号，所以只需要在起始机场将行李上依次贴上路线所经过机场的代号，那么当行李到达机场时，工作人员就知道这件行李该往哪座城市运送，最终送达目的地。

Segment Routing 也是一样的原理，在源节点为报文添加路径信息，指导路径的转发过程，如图 1 所示。

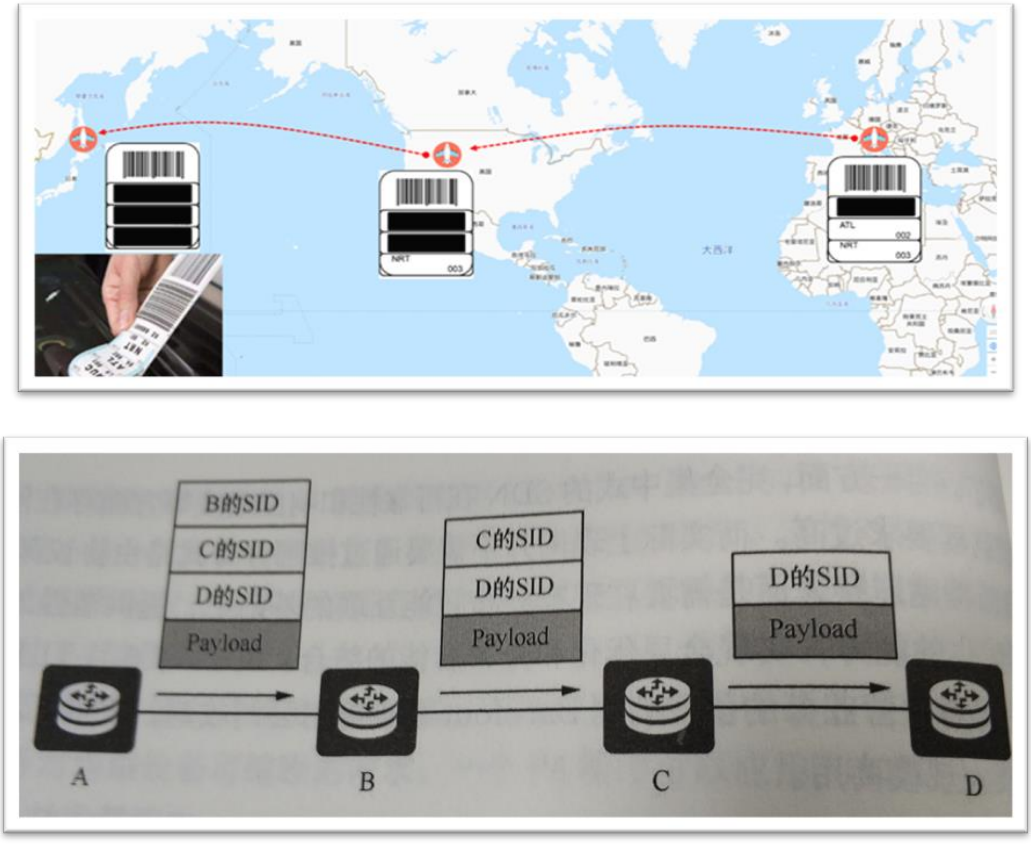


图 1　行李托运与 Segment Routing

因此，Segment Routing 有三个非常明显的特点：源路由，无状态和集中控制。在源节点添加路由信息，中间节点无需知道报文从哪里来，最终到哪里去，SID 和路径由控制器集中计算和下发。

## 2 基本原理

为了实现 SRv6，根据 IPv6 原有的路由扩展报文头定义了一种新类型的扩展报文头，称作 SRH。该扩展报文头通过携带 Segment List 等信息显式地指定一条 SRv6 路径。SRH 报文头格式如图 2 所示。

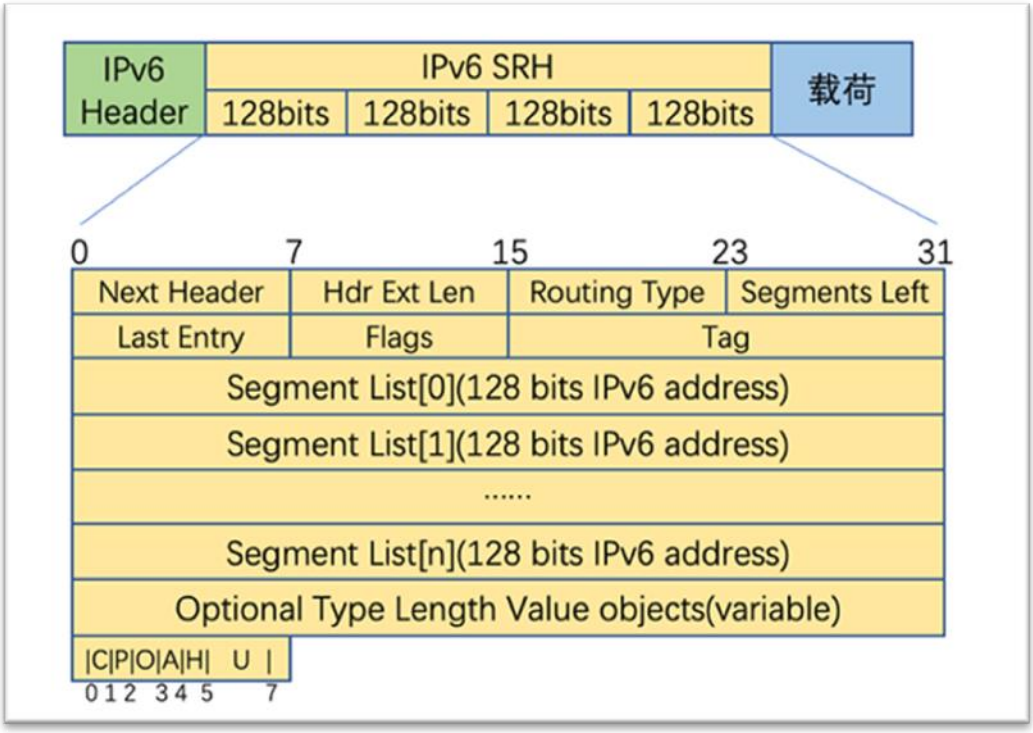


**图 2 SRH 格式**

各个字段的解释如下：

| 字段名 | 解释 |
|---|---|
| Next Header | 标识该 SRH 所封装的下层头类型。 |
| Hdr Ext Len | 该 SRH 头的长度（以 8 字节为基本单位），不包括前 8 字节。 |
| Routing Type | 4 |
| Segments Left | 表示还剩下多少个 segment 待转发。头节点在发出报文时，Segment Left 设置为 n-1，n 为 SR policy 的 segment 的总数。 |
| Last Entry | 索引，是指 SRH 中的 segment list[]数组中最后一个数组元素对应的下标（由于 SRH 中是逆序存放，Last Entry 实际上是逻辑上第一个 SID 的下标），它的作用是给出 SRH 中实际包含的 segment 的总数，取值一般为 n-1，reduced-SRH 模式时为 n-2，假设 n 为 SR policy 的 segment list 中所包含的 segment 的总数。 |
| Flags | 定义了以下标志：<br>O：OAM flag，OAM 报文时设置。 |

| | |
|---|---|
| Tag | 用于标识报文属于同一类或同一组，比如具有相同的属性集合的报文。 |
| Segment List[ ] | Segment List[0]表示最后一个 segment，Segment List[n-1]表示第一个 segment。 |
| Optional TLVs | 目前仅定义了 HMAC TLV 和 PAD TLV，注意这些 TLVs 不用于路由，不用于指导转发。<br>PAD TLV 用于使得 SRH 整体为 8 字节的整数倍。<br>HMAC 全称为 keyed Hashed Message Authentication Code，该 TLV 可选，用于校验报文的源头是否允许在报文的 DA 中使用当前 segment，并确保报文在传输时没有被修改。 |

利用 Wireshark 进行抓包，观察到 SRv6 的数据包格式如图 3 所示。

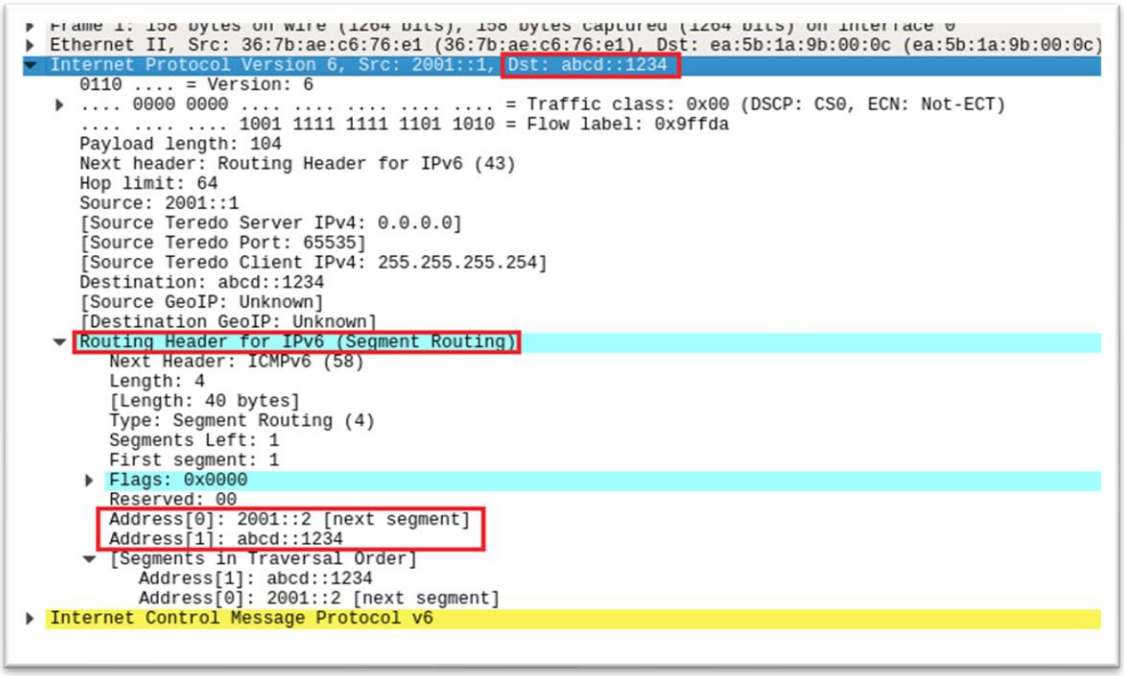


**图 3 SRv6 报文**

128 位 SRv6 SID 主要由三部分组成：标识节点位置的 Locator 字段（IPv6 前缀格式，可路由）、标识服务和功能的 Function 字段以及存储相关参数的 Args 字段，如图 4 所示。


**图 4 SRv6 SID 格式**

SRv6 相比较 SR-MPLS 具有更加强大的网络编程能力，SRH 具有三层编程空间，如图 5 所示。第一部分是 Segment 序列，它可以将多个 Segment 组合起来，形成 SRv6 路径，这跟 MPLS 标签栈比较类似；第二部分是对 SRv6 SID 的 128 比特的运用。众所周知，MPLS 标签封装主要是分成四个段，每个段都是固定长度

（包括 20 比特的标签）。而 SRv6 的每个 Segment 是 128 比特长，可以灵活分为多段，每段的长度也可以变化，由此具备灵活编程能力；第三部分是是紧接着 Segment 序列之后的可选 TLV（Type-Length-Value）。报文在网络中传送时，需要在转发面封装一些非规则的信息，它们可以通过 SRH 中 TLV 的灵活组合来完成。

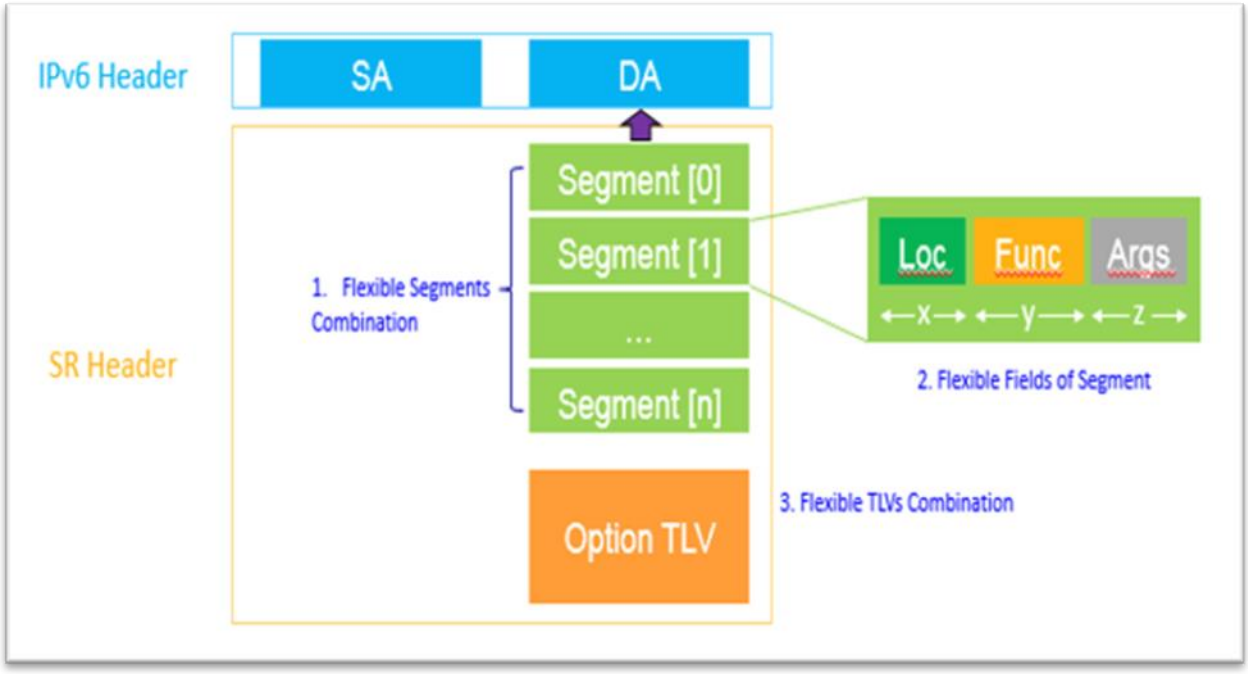


**图 5 SRH 的三层编程空间**

下面通过一个转发示例介绍 SRv6 数据包的转发过程。如图 6 所示的转发流程：

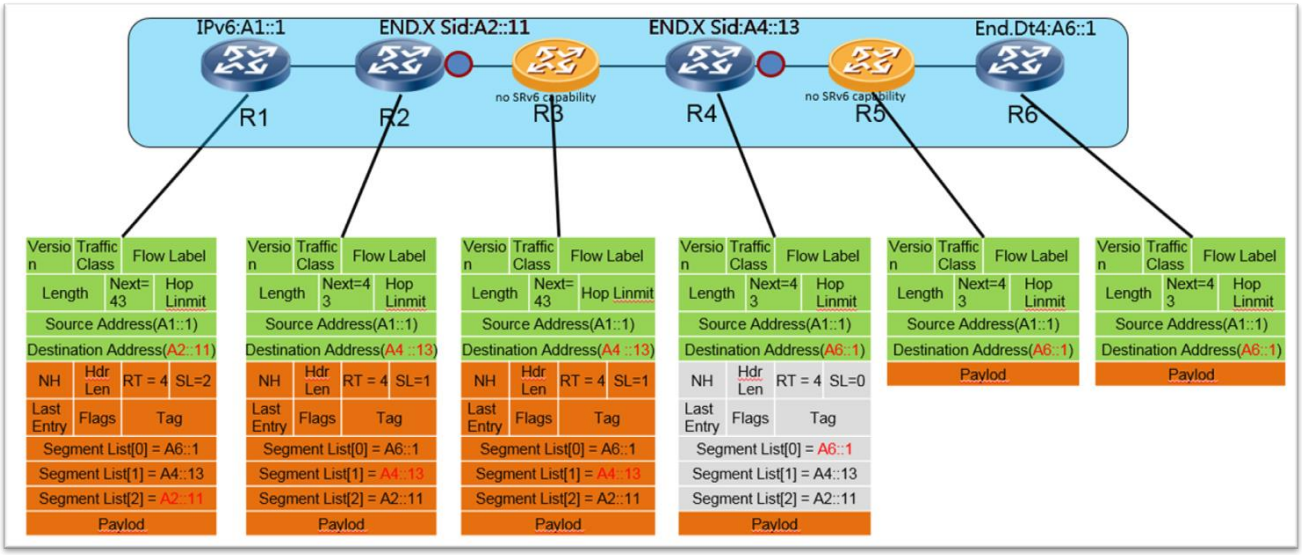


**图 6 SRv6 报文转发流程示例**

图 6 中蓝色节点为支持 SRv6 的路由节点，而黄色节点为仅支持 IPv6 转发的路由节点。

R1 将 SRv6 路径信息封装在 SRH 扩展头，指定 R2 和 R4 的 END.X SID，同时

初始化 SL = 2，并将 SL 指示的 SID A2::11 拷贝到外层 IPv6 头目的地址。R1 根据外层 IPv6 目的地址查路由表转发到 R2。

R2 收到报文以后，根据外层 IPv6 地址 A2::11 查找本地 Local SID 表，命中 END.X SID，执行 END.X SID 的指令动作：SL 减一，并将 SL 指示的 SID 拷贝到外层 IPv6 头目的地址，同时根据 END.X 关联的下一跳转发。

R3 根据 A4::13 查 IPv6 路由表进行转发，不处理 SRH 扩展头。具备普通的 IPv6 转发能力即可。

R4 收到报文以后，根据外层 IPv6 地址 A4::13 查找本地 Local SID 表，命中 END.X SID，执行 END.X SID 的指令动作：SL 减一，并将 SL 指示的 SID 拷贝到外层 IPv6 头目的地址，由于 SL = 0，弹出 SRH 扩展头，同时根据 END.X 关联的下一跳转发。

弹出 SRH 扩展头以后，报文就变成普通的 IPv6 头，由于 A6::1 是 1 个正常的 IPv6 地址，遵循普通的 IPv6 转发到 R6。

从上面的转发可以看出，对于支持 SRv6 转发的节点，可以通过 SID 指示经过特定的链路转发，对于不支持 SRv6 的节点，可以通过普通的 IPv6 路由转发穿越。这个特性使得 SRv6 可以很好地在 IPv6 网络中实现增量部署。

# 3 技术应用

## 3.1 SRv6 TE

TE 关注网络整体性能的优化，其主要目标是方便地提供高效、可靠的网络服务，优化资源的使用，优化网络流量的转发路径。

SRv6 通过 SRv6 Policy 实现了 TE。SRv6 Policy 利用 Segment Routing 的源路由机制，通过在头节点封装一个有序的指令列表来指导报文穿越网络。通过在 SRH 中封装一系列的 SRv6 Segment ID,可以显式地指导报文按照规划的路径转发，实现对转发路径端到端的细粒度控制，满足业务的高可靠、大带宽、低时延等 SLA 需求。

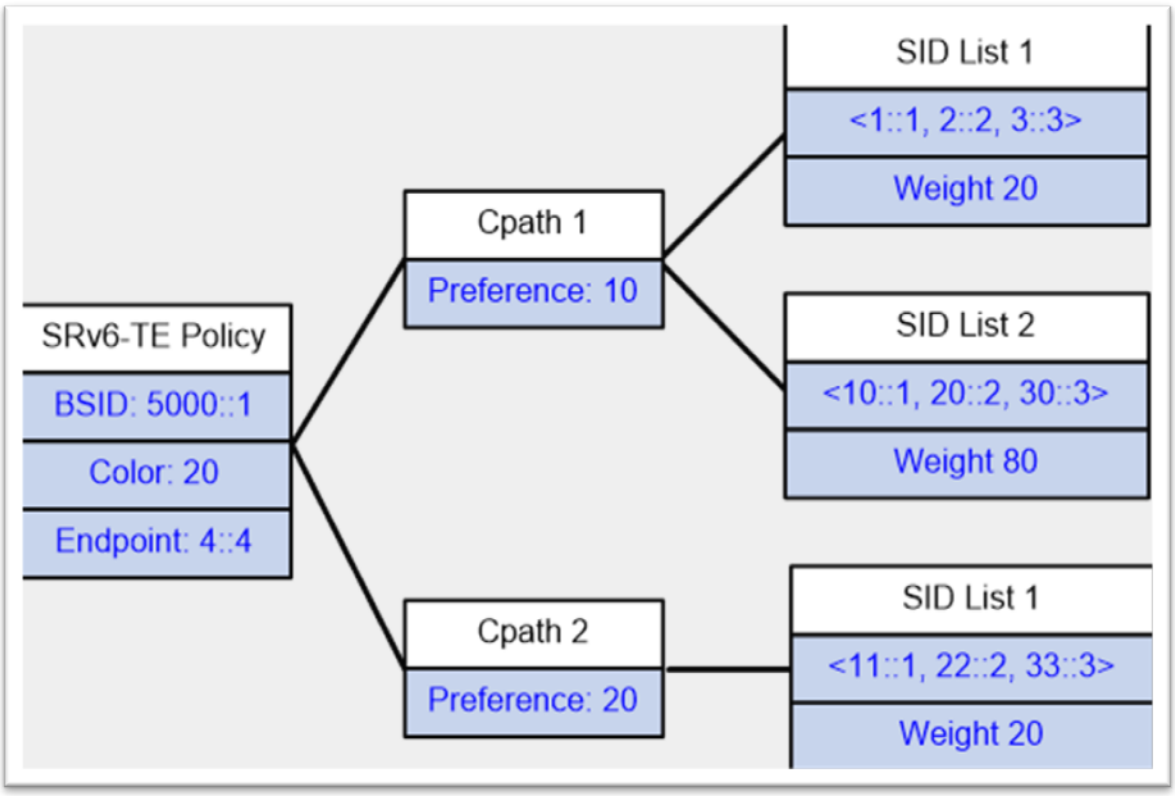


**图 7 SRv6 Policy 模型**

如图 7 所示，通过<BSID,Color,Endpoint>可以标识一个 SRv6-TE Policy，每个 SRv6 Policy 对应多个备选路径，每条路径设置有优先级（Preference），一般会选择优先级最高的路径作为主路径。 同时，每条路径可以关联多个 Segment List，通过 Segment List 附带的权重属性（Weight）来控制多个 SR 路径中的负载比例，从而实现 ECMP/UCMP。

## 3.2 SRv6 L3VPN

L3VPN over SRv6 控制平面工作流程如图 8 所示。

首先是基础网络联通配置，保证设备基础网络连通。

然后，VPN 配置与 Locator 路由发布。在 PE2 节点上配置 VPN 和 SRv6 SID，该节点配置 VPN1 接入 CE2 侧的 IPv4 业务，并且给每个 VPN 分配一个 SID。

之后 PE2 学习到来自 CE2 的私网路由，发布私网路由与 VPN SID。

最后，PE1 学习到私网路由，并生成 VPN 转发表。

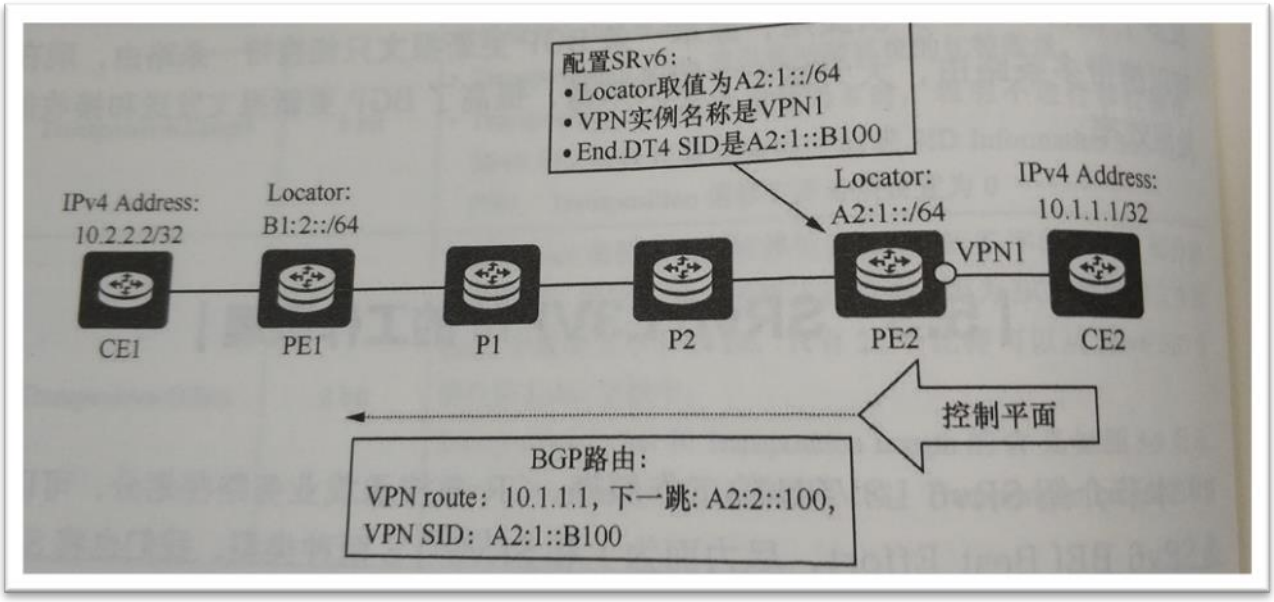


**图 8 L3VPN over SRv6 控制平面工作流程**

L3VPN 转发平面工作流程如图 9 所示。

当 PE1 收到从 CE1 发往 CE2 的报文时，PE1 会根据报文的入接口绑定的 VPN，查找相应的 VPN 实例转发表，将原始报文封装一层 IPv6 报文头往外转发。

网络中间的转发节点 P1 和 P2 可以是 SRv6 节点，也可以是不支持 SRv6 的普通 IPv6 节点。P1 和 P2 采用最普通的最长掩码匹配原则查找 IPv6 路由转发表，最终将报文沿最短路径转发至 PE2。

PE2 收到报文后查找本地 SID 表，弹出外层 IPv6 报文头的，然后使用内层报文的目的地址在 VPN 实例转发表中查找转发表项，并根据查找到的转发表项将报文转发到 CE2。

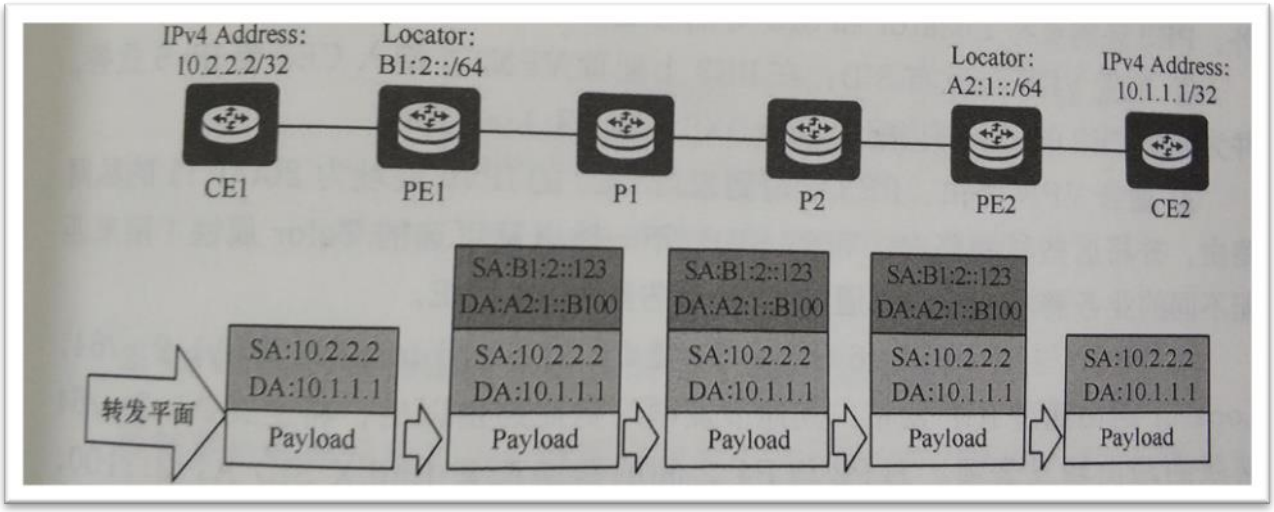


**图 9 L3VPN over SRv6 转发平面工作流程**

部署 SRv6 来支持 VPN 对现网的改动很小，只需将 PE 节点升级至支持 SRv6 即可。

## 3.3 基于 SRv6 的 SFC

Segment Routing 支持在入节点显示地编程数据报文的转发路径，这个能力天然可以支持 SFC。而且 Segment Routing 不需要在网络中间节点维护逐流的转发状态，这也让 Segment Routing 的业务部署比 NSH 简单很多。基于 Segment Routing 的 SFC 只需要在入节点下发 SFC 的策略即可，不需要对 SFC 中所有的网络节点进行配置，这也有效降低了部署基于 Segment Routing 的 SFC 难度，尤其是控制平面的部署难度。

基于 SRv6 的 SFC 部署方案如图 10 所示。

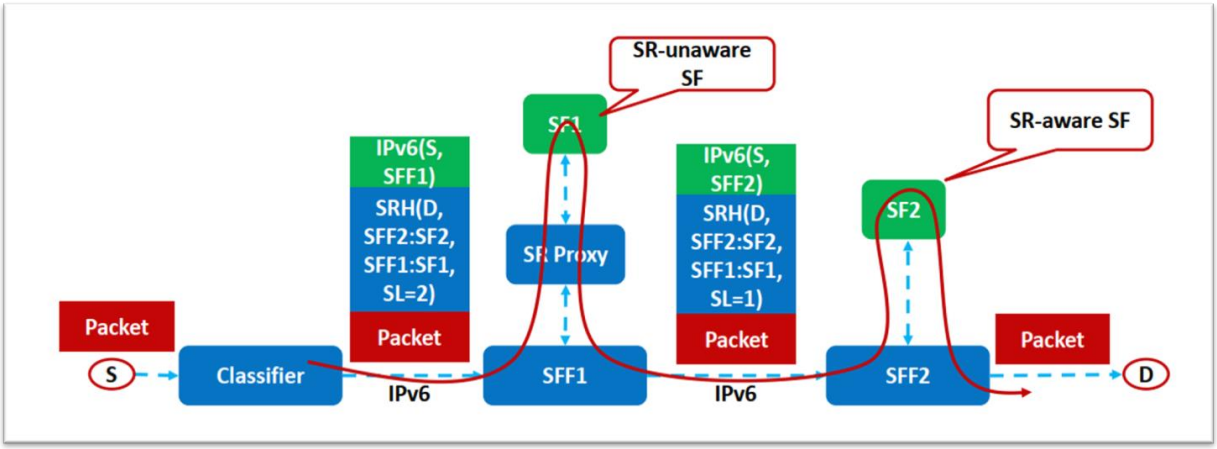


图 10 Stateless SRv6 SFC 方案架构

对于支持 SRv6 的 SF 节点（如图 10 中 SF2），可以直接连接到 SFF（Seivice Function Forwarder，业务功能转发器）。不支持 SRv6 的 SF 节点（如图 10 中 SF1），需要先在 SFF 和 SF 之间部署 SRv6 Proxy 来完成 SRv6 报文的处理。

# 4 SRv6 改进方案

虽然 SRv6 有很多优点，但它也存在如下问题：

从承载效率的角度分析，当前 SRv6 方案基于 SRH 扩展头实现，而 SID 长度为 128bits,对于一组 Segment List 就会增加 n * 128bits 的长度,而 SR-MPLS，每转一跳会弹出顶层标签，因此 SRv6 引入的协议开销远大于 SR-MPLS，造成了网络承载效率低；

从芯片能力的角度，SRH 扩展头的方式要求交换机芯片可以一次读取报头的深度更高，对硬件有特殊要求，而更换硬件需要增加新的投资成本。

所以业界也提出了一些压缩方案，其中 G-SRv6 相比较表现最佳。

## 4.1 G-SRv6

G-SRv6 不仅能够支持 SRv6 压缩,减少 SRv6 报文头开销,还可以与传统 SRv6 SID 在一个 SRH 中混合编程，从而支持 SRv6 向 SRv6 压缩方案的平滑演进。


**图 11 G-SRv6 方案**

如图 11 为 G-SRv6 压缩方案，其原理为：一个 SRv6 网络中很多 SID 都具有共同前缀（Common Prefix），一个 SID List 中 SID 的共同前缀部分都是重复冗余的。所以，SID list 中去掉 SRv6 SID 中的 Common Prefix 部分和其他冗余部分，保留 NodeID 和 FunctionID 作为压缩 SID，可以很好的减少报文头开销，如图 12 所示。

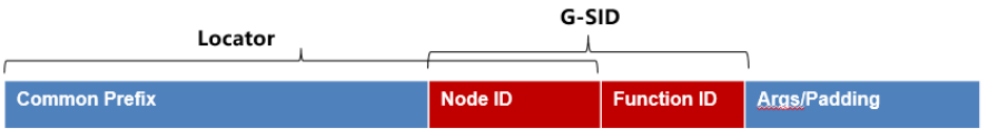


**图 12 支持压缩的 SRv6 SID 格式图**

G-SID 是可压缩 SRv6 SID 的一部分，和 CommonPrefix 以及 Argument/Padding 一起组合成完整的 SRv6 SID。G-SID 支持 32bit 或者 16bit 两种长度。实际从硬件处理效率、利旧和可扩展性的角度出发，32 bit 是一种更常用的压缩 SID 长度。

在 32bit 压缩中，G-SRv6 理论上可以将原有 SID 的封装效率提高到 75%左右，显著减少了报头开销，拓展了 SRv6 的应用部署范围。同时，G-SRv6 兼容 SRv6，支持压缩 SRv6 与原生 SRv6 混合编程，天然支持与原生 SRv6 互联互通，可支持从普通 SRv6 存量演进，平滑升级到 G-SRv6。

# 参考文献

【1】 李振斌等. 《SRv6 网络编程：开启 IP 网络新时代》.
【2】 draft-cl-spring-generalized-srv6-for-cmpr-01, IETF, 2020.
【3】 draft-ietf-spring-srv6-network-programming-19, IETF, 2020.
【4】 draft-ietf-idr-segment-routing-te-policy-09, IETF, 2020.
【5】 《中国移动 G-SRv6 技术白皮书》.
【6】 draft-cheng-spring-shorter-srv6-sid-requirement-02, IETF, 2020.
【7】 draft-filsfilscheng-spring-srv6-srh-comp-sl-enc-01, IETF, 2020.