

数据包处理的方法及设备

相关申请的交叉引用

本公开基于申请号为 202010334603.0、申请日为 2020 年 04 月 24 日的中国专利申请提出，并要求该中国专利申请的优先权，该中国专利申请的全部内容在此以引入方式并入本公开。

技术领域

本公开涉及通信技术领域，具体涉及一种数据包处理的方法及设备。

背景技术

运营商网络中对分段路由（Segment Routing，SR）标签层数要求较高。以第五移动通信技术（5th Generation，5G）承载网为例，随着 5G 核心网集中化部署，基站的流量需要穿过城域网以及互联网协议（Internet Protocol，IP）骨干网。典型场景下，在城域网中，接入环有 8-10 个节点，汇聚环有 4-8 个节点，核心环也有 4-8 个节点；在 IP 骨干网，流量还需穿过多个路由器节点。同时，由于网络切片、高可靠服务等级协议（Service-Level Agreement，SLA）、可管可控的要求，运营商网络需要能够指定显式路径，端到端 SR 隧道会有 10 跳甚至以上。因此，目前国内外多数部署多协议标签交换（Multi-Protocol Label Switching，MPLS）-SR 的运营商都要求支持 8 层以上段标识（Segment ID，SID）标签。

当前，SRv6 是基于互联网协议第 6 版（Internet Protocol Version 6，IPv6）扩展的 SR 解决方案，SRv6 方案基于分段路由头（Segment Routing Header，SRH）实现转发，其 SID 长度为 128 比特（bit）。按照 8 层 SID，为数据包带来 128 字节（Byte）的开销，对于平均长度 256Byte 的应用净荷，SRv6

带来的开销超过 1/3，带宽利用率则下降为 67%以下。而相同场景下，SR-MPLS 的开销只有 32Byte，带宽利用率仍有 89%。SRv6 和 SR-MPLS 在 SID 个数从 1-10 时承载效率的对比分析如图 1 所示（仅简单对比 SRH 和 SR-MPLS SID 的开销）。

5 开销的增大一方面造成了网络利用率的降低，另一方面为支持深层数据包深层负载均衡、带内遥测（In-Band Telemetry）、网络服务包头（Network Service Header，NSH）带来更大挑战。

另外，SRv6 部署必然会和 SR-MPLS 网络共存，由于网络利用率的不同可能会导致网络边界接口不平衡的问题，从而导致投资浪费。在 SR-MPLS
10 网络与 SRv6 网络域对接时，考虑 100G 链路，256byte 数据包，8 层 SID 的情况，由于链路利用率差异较大，SR-MPLS 域中的 1 个 100GE 链路在 SRv6 域中可能需要 2 条 100 千兆以太网（Gigabit Ethernet，GE）链路才能匹配。

在运营商应用中，SRv6 需要在网络芯片在数据包中插入超过 128Byte 长度的字段，相当于 32 层 MPLS-SR 标签深度，超出了已部署网络芯片的
15 能力，如果在芯片内部采用环回的解决方案，将大幅降低网络性能并引入更高的时延和抖动。在重新设计的网络芯片中，支持 SRv6 需要进一步扩大内部处理总线带宽，其是芯片成本和功耗的关键因素。

SRv6 在中间节点要求网络芯片读取完整 SRH，然后根据指针指示的位置提取需要处理的 Segment 并进行转发。对比 MPLS-SR 仅需读取最外层标
20 签，引入的复杂性进一步增加网络芯片的处理时延。

低功耗和低时延是运营商 5G 解决方案的关键因素，SRv6 复杂性对网络芯片带来的功耗、成本、时延的增加为其落地应用带来挑战。

根据以上分析，现有 SRv6 数据包开销较大，增加网络芯片的复杂性和难以平滑升级，导致 SRv6 难以快速部署到运营商网络中，需要在 SRv6 技
25 术基础上进一步进行演进。

发明内容

本公开实施例的一个目的在于提供一种数据包处理的方法及设备，解决现有 SRv6 数据包开销较大的问题。

第一方面，本公开实施例提供一种数据包处理的方法，包括：获取第一段标识（SID）的第一信息和第二信息，所述第一信息用于指示所述第一 SID 所在的容器在数据包的 SID 列表中的位置，所述第二信息用于指示所述第一 SID 在所述容器中的位置；根据所述第一信息和第二信息，得到所述第一 SID 在 SID 列表中的位置；将所述第一 SID 复制到数据包的目的地地址中，发送所述数据包。

10 在本公开的一些可选实施例中，所述获取第一 SID 的第一信息和第二信息，包括：获取第二 SID 的第三信息和第四信息，所述第三信息用于指示所述第二 SID 所在的容器在所述数据包的 SID 列表中的位置，所述第四信息用于指示所述第二 SID 在所述容器中的位置；根据所述第二 SID 的第三信息和第四信息，得到所述第一 SID 的第一信息和第二信息。

15 在本公开的一些可选实施例中，所述获取第二 SID 的第三信息和第四信息，包括：从所述数据包的段路由头（SRH）中获取所述第二 SID 的第三信息；从所述数据包的目的地地址中获取所述第二 SID 的索引，所述第二 SID 的索引包括：第四信息。

20 在本公开的一些可选实施例中，所述根据所述第二 SID 的第三信息和第四信息，得到所述第一 SID 的第一信息和第二信息，包括：根据所述第二 SID 的第四信息或根据所述第二 SID 的属性信息，确定所述第一 SID 与所述第二 SID 是否在同一个容器中；如果所述第一 SID 与所述第二 SID 在同一个容器中，则所述第一 SID 的第一信息与所述第二 SID 的第三信息相同，以及根据所述第二 SID 的第四信息，得到所述第一 SID 的第二信息；
25 或者，如果所述第一 SID 与所述第二 SID 不在同一个容器中，则根据所述

第二 SID 的第三信息，得到所述第一 SID 的第一信息，以及根据所述第一 SID 的 SID 类型，确定所述第一 SID 的第二信息。

在本公开的一些可选实施例中，所述第一 SID 的 SID 类型是由所述第二 SID 的属性信息指示的。

5 在本公开的一些可选实施例中，所述方法还包括：将所述第一 SID 的第一信息更新到所述数据包的 SRH 中；将所述第一 SID 的第二信息作为所述第一 SID 的索引更新到所述数据包的地址中。

在本公开的一些可选实施例中，所述将所述第一 SID 复制到数据包的地址中，发送所述数据包，包括：将所述第一 SID 复制到数据包的地址中，所述第一 SID 与所述地址中的公共前缀或地址块组成得到一个新的地址，并根据所述新的地址将所述数据包转发给下一个节点；或者，将所述第一 SID 和第二信息复制到数据包的地址中，所述第一 SID 和第二信息与所述地址中的公共前缀或地址块组成得到一个新的地址，并根据所述新的地址将所述数据包转发给下一个节点；或者，将所述第一 SID 和第二信息复制到数据包的地址中，将所述数据包转发给下一个节点。

在本公开的一些可选实施例中，所述获取第一 SID 的第一信息和第二信息，包括：获取第二 SID 的第三信息、第四信息和第五信息，所述第三信息用于指示所述第二 SID 所在的容器在所述数据包的 SID 列表中的位置，所述第四信息用于指示所述第二 SID 在所述容器中的位置，所述第五信息用于指示所述第二 SID 的 SID 类型和/或指示所述第二 SID 是否压缩；根据所述第二 SID 的第三信息、第四信息和第五信息，得到所述第一 SID 的第一信息和第二信息；其中，在所述 SID 列表中所述第二 SID 为所述第一 SID 的上一个 SID。

25 在本公开的一些可选实施例中，所述获取第二 SID 的第三信息、第四

信息和第五信息，包括：从所述数据包的 SRH 中获取所述第二 SID 的第三信息；从所述数据包的地址中获取所述第二 SID 的索引，所述第二 SID 的索引包括：第四信息和第五信息。

在本公开的一些可选实施例中，所述根据所述第二 SID 的第三信息、
5 第四信息和第五信息，得到所述第一 SID 的第一信息和第二信息，包括：
根据所述第二 SID 的 SID 类型和所述第一 SID 的 SID 类型，确定所述第一
SID 与所述第二 SID 是否在同一个容器中；如果所述第一 SID 与所述第二
SID 在同一个容器中，则所述第一 SID 的第一信息与所述第二 SID 的第三
信息相同，以及根据所述第二 SID 的第四信息，得到所述第一 SID 的第二
10 信息；或者，如果所述第一 SID 与所述第二 SID 不在同一个容器中，则根
据所述第二 SID 的第三信息，得到所述第一 SID 的第一信息，以及根据所
述第一 SID 的 SID 类型，确定所述第一 SID 的第二信息。

在本公开的一些可选实施例中，所述容器用于容纳一个标准 SID 或者
多个压缩 SID，所述压缩 SID 包括：节点标识和所述节点的功能（function）
15 的取值。

第二方面，本公开实施例还提供一种节点，包括：

获取模块，配置为获取所述第一 SID 的第一信息和第二信息，所述第一信
息用于指示所述第一 SID 所在的容器在数据包的 SID 列表中的位置，所述
第二信息用于指示所述第一 SID 在所述容器中的位置；

20 处理模块，配置为根据所述第一信息和第二信息，得到所述第一 SID
在 SID 列表中的位置；

发送模块，配置为将所述第一 SID 复制到数据包的地址中，发送
所述数据包。

第三方面，本公开实施例还提供一种节点，包括：收发机、处理器和
25 用于存储能够在处理器上运行的计算机程序的存储器；

所述收发机用于在所述处理器的控制下发送和接收数据;

所述处理器用于读取所述存储器中的计算机程序时, 执行本公开前述第一方面实施例的数据包处理方法的步骤。

第四方面, 本公开实施例还提供一种计算机可读存储介质, 所述计算机可读存储介质上存储有计算机程序, 所述计算机程序被处理器执行时实现包括如本公开实施例第一方面所述的数据包处理的方法的步骤。

在本公开实施例中, SID 列表中至少包括压缩 SID, 在更新数据包的地址时, 通过将压缩 SID 和当前地址中的公共前缀拼接形成新的 SID, 继续查表转发, 从而实现数据包的压缩, 有效的减少数据包的报头开销。此外, 还可以通过将标准 SID 和压缩 SID 混合编排在 SID 列表中, 从而支持压缩节点与普通节点混合编程, 实现存量演进, 平滑升级。

附图说明

通过阅读下文优选实施方式的详细描述, 各种其他的优点和益处对于本领域普通技术人员将变得清楚明了。附图仅用于示出优选实施方式的目的, 而并不认为是对本公开的限制。而且在整个附图中, 用相同的参考符号表示相同的部件。在附图中:

图 1 为净荷长度 256 字节 (B) 时不同 SID 个数 SR 承载效率对比分析示意图;

图 2 为标准的 SRv6 扩展头的封装格式示意图;

图 3 为本公开实施例的中的数据包处理的方法的流程示意图;

图 4 为支持网络可编程标准 SRv6 SID 格式示意图;

图 5 为支持网络可编程典型 SRv6 Locator 示意图;

图 6 为本公开实施例中的支持压缩的 SRv6 SID 格式示意图;

图 7 为本公开实施例中 32-bits 压缩 SID Container 格式示意图;

图 8 为本公开实施例中 128bit SRv6 SID 和 32bit X-SID 混合编程在

X-SRH 中的格式示意图;

图 9 为本公开实施例中 128 bit SRv6 SID 和 32 bit X-SID 混编在 X-SRH 中的编码示意图;

图 10 为本公开实施例中 XI 的示意图;

5 图 11 为本公开实施例中 X-SID 的更新示意图;

图 12 为本公开实施例中混编场景中数据包的转发示意图;

图 13 为本公开实施例中纯压缩场景中数据包的转发示意图;

图 14 为本公开实施例的节点的示意图之一;

图 15 为本公开实施例的节点的示意图之二。

10 具体实施方式

SR是一种源路由技术,基于软件定义网络(Software Defined Network, SDN)理念,构成面向路径连接的网络架构,支撑未来网络多层次的可编程需求,可以满足5G超大连接和切片的应用场景下的连接需求。SR-MPLS是基于当前主流MPLS转发面形成的SR解决方案。

15 SRv6是基于IPv6扩展的SR解决方案。IPv6技术成是新一代网络的主体技术,基于IPv6的SRv6长远考虑是未来网络的演进趋势,关于SRv6技术的机制研究是业界热点。

标准SRv6的128比特(bit)SID采用IPv6地址格式的SID,相比MPLS 标签(Label)格式的SID具备可路由属性,简化域间路径创建,实现在IPv6
20 网络中简化建立端到端路径的能力。同时,SRv6 SID支持可编程能力,能够满足灵活的网络和业务功能处理,结合集中式和分布式控制平面的协同支持,能灵活满足各种业务和网络功能的需求,适应网络和业务发展的需要。

SR通过头结点封装一系列的指令来引导数据包通过网络,在SRv6架构
25 中,一个指令是一个128比特的IPv6地址,参见图2,为标准SRv6扩展头的

封装格式。其中，对于IPv6头，包括：版本（Version）字段、流量类型（Traffic Class）字段、流标签（Flow Label）字段、有效净荷长度（Payload Length）字段、下一报头（Next Header）字段、跳数限制（Hop Limit）字段、源地址（Source Address）字段以及目的地址（Destination Address）字段。对于

5 SRH扩展头，包括：下一报头（Next Header）字段、Hdr Ext len字段（表示SRH头的长度）、路由类型（Routing Type）字段、段剩余（Segment left）字段（表示到达目的节点前应当访问的中间节点数）、最后一项（Last Entry）字段（表示分段列表中包含分段列表的最后一个元素的索引）、标记（Flags）字段（表示数据包的一些标识）、标签（Tag）字段（用于标识同组数据包）、

10 分段列表（Segment list）字段（包含128bit的IPv6地址）、可选TLV（Optional TLV）对象字段（可变的）和有效净荷（Payload）字段。

通过SRv6扩展头封装格式可知，SRv6扩展头封装长度为：40 Byte（IPv6头）+ 8 Byte（SRH固定头）+ $16 \times N$ Byte（Segment List）。所以，随着SRv6指定的SID数量增加，SRv6扩展头封装带来的额外的开销就会越大。为了解

15 决这个问题，需要实现SRv6扩展头缩短的功能。

标准SRv6虽然具备上述诸多优点，但是缺点同样明显。SRv6技术在网络中实际部署当前主要面临二个方面的挑战：一是SRv6数据包开销大、网络链路带宽利用率低，256byte包长8层SID的情况下带宽利用率只有60%左右；二是SRv6数据包处理对芯片要求高，现网设备难以支持深度的SRH头的复制和操作，并且128bit的SRH头处理会降低现有芯片的处理效率。

20 的复制和操作，并且128bit的SRH头处理会降低现有芯片的处理效率。

针对标准SRv6技术存在的问题，发明人发现可以通过对标准SRv6的SRH进行压缩优化，采用更简洁短小的X-SID，如典型的32bit SID标识，以优化SRv6的SID，将标准128bit的SID采用32bit的X-SID来表示并替代其在SRH头中的封装，从而将标准SRv6的SRH封装开销优化，在支持SRv6功能

25 的同时大大优化SRv6的性能，支持SRv6技术规模部署。

下面将结合本公开实施例中的附图，对本公开实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本公开一部分实施例，而不是全部的实施例。基于本公开中的实施例，本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例，都属于本公开实施例保护的范

5 护的范围。

本公开实施例的说明书和权利要求书中的术语“包括”以及它的任何变形，意图在于覆盖不排他的包含，例如，包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元，而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步

10 骤或单元。此外，说明书以及权利要求中使用“和/或”表示所连接对象的至少其中之一，例如A和/或B，表示包含单独A，单独B，以及A和B都存在三种情况。

在本公开实施例中，“示例性的”或者“例如”等词用于表示作例子、例证或说明。本公开实施例中被描述为“示例性的”或者“例如”的任何实施例或设计

15 方案不应被解释为比其它实施例或设计方案更优选或更具优势。确切而言，使用“示例性的”或者“例如”等词旨在以具体方式呈现相关概念。

本文所描述的技术不限于 5G 系统以及后续演进通信系统，以及不限于长期演进(Long Term Evolution, LTE)/LTE 的演进(LTE-Advanced, LTE-A)系统，并且也可用于各种无线通信系统，诸如码分多址(Code Division

20 Multiple Access, CDMA)、时分多址(Time Division Multiple Access, TDMA)、频分多址(Frequency Division Multiple Access, FDMA)、正交频分多址(Orthogonal Frequency Division Multiple Access, OFDMA)、单载波频分多址(Single-carrier Frequency-Division Multiple Access, SC-FDMA)和其他系统。

25 术语“系统”和“网络”常被可互换地使用。CDMA 系统可实现诸如

CDMA2000、通用地面无线电接入（Universal Terrestrial Radio Access，UTRA）等无线电技术。UTRA 包括宽带 CDMA（Wideband Code Division Multiple Access，WCDMA）和其他 CDMA 变体。TDMA 系统可实现诸如全球移动通信系统（Global System for Mobile Communication，GSM）之类的无线电技术。OFDMA 系统可实现诸如超移动宽带（Ultra Mobile Broadband，UMB）、演进型 UTRA（（Evolution-UTRA，E-UTRA））、IEEE 802.11（（Wi-Fi））、IEEE 802.16（（WiMAX））、IEEE 802.20、Flash-OFDM 等无线电技术。UTRA 和 E-UTRA 是通用移动通信系统（Universal Mobile Telecommunications System，UMTS）的部分。LTE 和更高级的 LTE（如 LTE-A）是使用 E-UTRA 的新 UMTS 版本。UTRA、E-UTRA、UMTS、LTE、LTE-A 以及 GSM 在来自名为“第三代伙伴项目”（3rd Generation Partnership Project，3GPP）的组织的文献中描述。CDMA2000 和 UMB 在来自名为“第三代伙伴项目 2”（3GPP2）的组织的文献中描述。本文所描述的技术既可用于以上提及的系统和无线电技术，也可用于其他系统和无线电技术。

参见图3，本公开实施例提供一种数据包处理的方法，该方法的执行主体可以为节点，例如SRv6节点，该方法包括：

步骤 301：获取第一SID的第一信息和第二信息，所述第一信息用于指示所述第一SID所在的容器在数据包的SID列表中的位置，所述第二信息用于指示所述第一SID在所述容器中的位置。

在一些可选实施例中，第一信息也可以是段剩余（Segment Left，SL）信息，标识剩余Segment的数量；第二信息可以是索引（Index）信息，索引（Index）信息也可称为SID Index信息，在本公开实施例中对其不做限定。

容器（或者称为SID Container）中可以包括一个或多个第一SID，比如容器为 128bit，该容器可以包括：一个标准SID（128bit的SID），或者多个

压缩SID，也就是说，上述第一SID可以是标准SID，或者也可以是压缩SID，该压缩SID可以是 32bits的压缩SID或者 16bits的压缩SID，本实施例中不限于此。

以 32bits的压缩SID为例，在一个 128bit的容器中可以包括：三个 32bits
5 的压缩SID和一个用于补齐 128bit的填充（padding），或者包括：两个 32bits
的压缩SID和两个用于补齐 128bit的padding，或者包括：一个 32bits的压缩
SID和三个用于补齐 128bit的padding，或者包括：四个 32bits的压缩SID。
可以理解的是，上述第一SID不包括用于补齐 128bit的填充。

以一个 128bit的容器中包含三个 32bits的压缩SID和一个用于补齐
10 128bit的padding为例，从左到右依次可以为：padding、32bits的压缩SID、
32bits的压缩SID、32bits的压缩SID和 32bits的压缩SID，各自的第二信息可
以分别为：0、1、2、3。

以一个 128bit的容器中包含：128bit 的标准SID为例，该容器中只有一个
标准SID，该标准SID的第二信息可以为空或者为预设值。

15 以SID列表（SID list）包括五个容器为例，该容器在SID列表中的位置
由下至上可以表示为：SL=0、SL=1、SL=2、SL=3 和SL=4，可以理解的是
在本公开实施例中“容器在SID列表中的位置”的具体表达形式不做具体
限定。

步骤 302：根据所述第一信息和第二信息，得到所述第一SID在SID列
20 表中的位置。

示例性的，如果第一SID为压缩SID，则可以根据该压缩SID的SL与Index
在SID列表中定位得到对应的一个压缩SID，则如果该第一SID为 128bit的
SID，该第二信息可以为空，则可以根据该第一SID的SL在SID列表中定位
得到对应的一个 128bit的SID。

步骤 303: 将所述第一SID复制（或更新）到数据包的目的地址中，发送所述数据包。

方式 1: 将第一SID复制到数据包的目的地址中，所述第一SID与所述目的地址中的公共前缀或地址块组成得到一个新的目的地址，并根据所述新的目的地址将所述数据包转发给下一个节点；

比如，第一SID可以为压缩SID，基于该压缩SID的SL与Index，可以在SID列表中定位得到对应的一个压缩SID，将该压缩SID复制到数据包的目的地址中，与公共前缀或地址块组成得到一个新的目的地址，根据新的目的地址将该数据包转发给下一个节点。

方式 2: 将所述第一SID和第二信息复制到数据包的目的地址中，所述第一SID和第二信息与所述目的地址中的公共前缀或地址块组成得到一个新的目的地址，并根据所述新的目的地址将所述数据包转发给下一个节点；

比如，第一SID可以为压缩SID，基于该压缩SID的SL与Index，可以在SID列表中定位得到对应的一个压缩SID，将该压缩SID和Index复制到数据包的目的地址中，与公共前缀或地址块组成得到一个新的目的地址，根据新的目的地址将该数据包转发给下一个节点。

方式 3: 将所述第一SID和第二信息复制到数据包的目的地址中，将所述数据包转发给下一个节点。比如，该第一SID为 128bitSID，基于该 128bitSID的SL与Index，可以在SID列表中定位得到对应的一个 128bitSID，将该 128bitSID复制到数据包的目的地址中，并将该数据包转发给下一个节点。

在本公开的一些可选实施例中，步骤 301 可以包括：

步骤 3011: 获取第二SID的第三信息和第四信息，所述第三信息用于指示所述第二SID所在的容器在所述数据包的SID列表中的位置，所述第四信息用于指示所述第二SID在所述容器中的位置。

在一些可选实施例中，第三信息也可以是段剩余（SL）信息，标识剩余Segment的数量；第四信息也可以是索引（Index）信息，在本公开实施例中对其不做限定。

步骤 3012：根据所述第二SID的第三信息和第四信息，得到所述第一
5 SID的第一信息和第二信息。

在一些可选实施例中，在所述SID列表中所述第二SID为所述第一SID的上一个SID，或者在该SID列表中第二SID排序在第一SID之前，第二SID和第一SID之间可以包括一个或多个Padding（比如用“0”表示），比如，SID列表中先后顺序为：第二SID、Padding、第一SID。

10 上述“所述第二SID为所述第一SID的上一个SID”可以理解为，第一SID为第二SID的下一个SID，也即，与第一SID匹配的节点是与第二SID匹配的节点的下一个节点，比如与第一SID匹配的节点为N3，与第二SID匹配的节点为N2。

在一些可选实施例中，在步骤 3011 之前，可以判断第二SID的下一个
15 SID是否为padding，如果不是，则执行步骤 3011，如果是，则默认第一SID是一个 128bit的SID，此时该第一SID的第一信息可以基于第二SID的第三信息计算得到，并将该第一SID的第二信息设置为空。

示例性的，可以基于第二SID的属性信息识别第二SID的下一个SID是否为padding。也就是说，在本地SID（Local SID）表中命中到节点本地发布的
20 的END.X SID（第二SID），没有配置COC Flavor（该COC Flavor用于指示下一个SID的SID类型），则第二SID的下一个SID是否为padding。

在本公开的一些可选实施例中，步骤 3011 可以包括：

步骤 30111：从所述数据包的SRH中获取所述第二SID的第三信息；

步骤 30112：从所述数据包的地址中获取所述第二SID的索引，所
25 述第二SID的索引包括：第四信息。也就是说，第二SID的索引中可以包括

一个指示比特或多个指示比特。在本公开实施例中对第二SID的索引的格式不做具体限定。

在本公开的一些可选实施例中，步骤 3012 可以包括：

步骤 30121：根据所述第二SID的第四信息或根据所述第二SID的属性
5 信息，确定所述第一SID与所述第二SID是否在同一个容器中。

一个示例中，如果第二SID的第四信息大于零，则表示第一SID与所述第二SID在同一个容器中；如果第二SID的第四信息等于零，则表示第一SID与第二SID不在同一个容器中。

另一个示例中，第二SID的属性信息没有指示下一个SID的SID类型，则
10 表示第一SID与第二SID不在同一个容器中。也就是说，在Local SID表中命中到节点本地发布的END.X SID（第二SID），没有配置COC Flavor（该COC Flavor用于指示下一个SID的SID类型），则默认指示下一个SID（第一SID）是 128bit的SID，也即，第二SID和第一SID之间可以包括：一个Padding。

步骤 30122：如果所述第一SID与第二SID在同一个容器中，则所
15 述第一SID的第一信息与第二SID的第三信息相同（即SL相同），以及根据第二SID的第四信息，得到第一SID的第二信息，比如第二SID的第四信息（Index）减一，得到第一SID的第二信息（Index）；或者，

如果所述第一SID与第二SID不在同一个容器中，则根据第二
SID的第三信息，得到第一SID的第一信息，比如第二SID的第三信息减
20 一（SL-1），得到第一SID的第一信息（SL），以及根据第一SID的SID类型，确定第一SID的第二信息。

示例性的，第一SID的SID类型为 32bit，则第一SID的第二信息为“3”。

在本公开的一些可选实施例中，所述第一SID的SID类型可以由所述
第二SID的属性信息（COC Flavor）指示的，可以理解的是，第二SID的属
25 性信息可以在当前节点的本地SID表中查表得到，如表所示。

表 1

SID Flavor 类型	功能描述
None	不增加 Flavor; SID 的默认 Flavor 指示第一 SID 是 128 bits SRv6 SID
COC32	标识下一个节点 (第一 SID) 是 32bit 的压缩 SID
COC16	标识下一个节点 (第一 SID) 是 16bit 的压缩 SID

在本公开的一些可选实施例中, 所述方法还包括: 将所述第一SID的第一信息更新到所述数据包的SRH中; 将所述第一SID的第二信息作为所述第一SID的索引更新到所述数据包的目的地址中。

5 在本公开的一些可选实施例中, 步骤 301 可以包括:

 步骤 3013: 获取第二SID的第三信息、第四信息和第五信息, 所述第三信息用于指示所述第二SID所在的容器在所述数据包的SID列表中的位置, 所述第四信息用于指示所述第二SID在所述容器中的位置, 所述第五信息用于指示所述第二SID的SID类型和/或指示所述第二SID是否压缩。

10 示例性的, SID类型可以包括 8bit、16bit、32bit、64bit或者 128bit等。

 在一些可选实施例中, 第五信息可以包括一个指示比特或多个指示比特。

 步骤 3014: 根据所述第二SID的第三信息、第四信息和第五信息, 得到所述第一SID的第一信息和第二信息。

15 其中, 在所述SID列表中所述第二SID为所述第一SID的上一个SID, 或者在该SID列表中第二SID排序在第一SID之前, 第二SID和第一SID之间可以包括一个或多个Padding (比如用“0”表示)。

 在本公开的一些可选实施例中, 步骤 3013 可以包括:

 步骤 30131: 从所述数据包的SRH中获取所述第二SID的第三信息;

20 步骤 30132: 从所述数据包的目的地址中获取所述第二SID的索引, 所述第二SID的索引包括: 第四信息和第五信息。

在一些可选实施例中，第二SID的索引中可以包括一个指示比特或多个指示比特。

在本公开的一些可选实施例中，步骤 3014 可以包括：

- 步骤 30141：根据所述第二SID的SID类型和所述第一SID的SID类型，
5 确定所述第一SID与所述第二SID是否在同一个容器中。

示例性的，如果第二SID的SID类型为 32bit，第一SID的SID类型为 32bit，则确定所述第一SID与所述第二SID在同一个容器中；如果第二SID的SID类型为 32bit，第一SID的SID类型为 128bit，则确定所述第一SID与所述第二SID不同一个容器中。

- 10 步骤 30142：如果所述第一SID与所述第二SID在同一个容器中，则所述第一SID的第一信息与所述第二SID的第三信息相同（即SL相同），以及根据所述第二SID的第四信息，得到所述第一SID的第二信息，比如第二SID的第四信息减一（Index-1），得到第一SID的第二信息；或者，

- 如果所述第一SID与所述第二SID不在同一个容器中，则根据所述第二
15 SID的第三信息，得到所述第一SID的第一信息，比如第二SID的第三信息减一，得到第一SID的第一信息，以及根据所述第一SID的SID类型，确定所述第一SID的第二信息。

示例性的，第一SID的SID类型为 32bit，则第一SID的第二信息为“3”，即第一SID的Index为 3。

- 20 在本公开实施例中，所述容器用于容纳一个标准SID或者多个压缩SID，所述压缩SID包括：节点标识和所述节点的功能（function）的取值。

在本公开实施例中SID列表中至少包括压缩SID。在更新数据包中的目的地址时，通过将压缩SID和当前目的地址中的公共前缀拼接形成新的SID，继续查表转发，从而实现数据包的压缩，有效的减少数据包的报头开销。

- 25 此外，在本公开实施例中，还可以通过将标准SID和压缩SID混合编排

在SID列表中,从而在SRv6路径中支持压缩节点与普通节点混合编程,实现存量演进,平滑升级。同时,在多域的场景中,因为地址规划受限,无法保证多个域的SID有公共前缀,通过混编则可以方便地实现建立跨多域的SRv6路径。

5 下面以IPv6的场景为例进行介绍,当然可以理解的是,本公开实施例也可以适用于其他版本的IP技术(比如,IPv5、IPv7、IPv9、IPv10等),而不仅限于IPv6技术,适用于其他版本的IP技术的方式与此类似,在此不再赘述。

 当SRv6部署中,需要通过地址规划划分一个地址块,用于SID的分配,该地址块可以称为SID空间(Space)。因此,在一个SRv6域中,从SID Space
10 分配的SID可以具有公共前缀(Common Prefix)。通常情况下,SRH中的SID均具有相同的前缀,而这部分内容是冗余的,去除这部分冗余的信息,可以有效的减少SRv6报头开销。

 因此,在本公开实施例中,将SID列表(SID List)中SID的Common Prefix
 移除,仅携带变化的压缩SID(或者称为短SID,或者称为X-SID)。在替换
15 目的地址时,通过将压缩SID和当前IPv6目的地址中的公共前缀拼接
 (Stitching)形成新的SID,继续查表转发,从而实现SRv6的压缩。

 此外,在本公开实施例中,还通过将传统SRv6 SID和压缩SID混合编排在SRv6(比如,该SRv6称为X-SRv6)中,从而在SRv6路径中支持压缩节点与普通SRv6节点混合编程,实现存量演进,平滑升级。同时,在多域的场景中,因为地址规划受限,无法保证多个域的SID有公共前缀,通过混编则可以方便地实现建立跨多域的SRv6路径。
20

(一) 关于标准128bit SRv6 SID格式的介绍:

 SRv6 Segment Identifier: 用于标识SRv6 Segment的ID,标准SRv6 SID
 是一个128bit的IPv6地址,如果考虑支持SRv6网络可编程,它通常由三部分
25 组成,如图4所示。

(1) 位置 (Locator) 是网络拓扑中分配给一个网络节点的标识, 用于路由和转发数据包到该节点, Locator标识位置信息。Locator对应的路由会被节点通过内部网关协议 (Interior Gateway Protocol, IGP) 发布到网络中, 用于帮助其他设备将数据包转发到发布该Locator的节点。在SRv6 SID中,

5 Locator长度可变, 用于适配不同规模的网络。

(2) 功能 (Function) 用来表达该指令要执行的转发动作, 相当于计算机指令的操作码。在SRv6网络编程中, 不同的转发行为由不同的Function来表达, 如转发数据包到指定链路, 或在指定表中查表转发等。

(3) 变量 (Arguments) (以下简称Args) 是一个可选字段。它是指令
10 在执行时对应的参数, 这些参数可能包含流, 服务或任何其他相关的信息。

进一步的, 参见图5, Locator又可以细分为B: N, 其中“B”标识SRv6 SID块 (Block), 一般由运营商分配给某个子网, 通常用前缀 (Prefix) 来表示; “N”表示该子网内区分节点的标识。因此在一个SRv6网络中SID都具有共同
的共同前缀B, 所以在一个SID List中SID的B部分都是重复冗余的。

15 (二) 关于压缩SRv6 SID格式的介绍:

根据标准SRv6 SID的格式, 可以看出在一个典型的支持SRv6网络可编程的应用中, 其SID格式是有规律的, 充分利用SID格式的规律可以实现标准SRv6 SID格式优化。

本公开实施例定义典型的32bit SRv6压缩SID格式, 该压缩SID可以是由
20 标准128bit SID中的Node ID和Function ID组成。

标准128bit SID格式可以称为完整SID, 其定义沿用标准SRv6 SID; 32bit SID为压缩SID (或者称为X-SID), 其是完整SID的变化部分。格式如图6所示。

类似的, 也可以定义16bits的压缩SID。下面以32bits的压缩SID为例描
25 述, 可以理解的是, 16bits的压缩SID的实现方式与32bits的压缩SID的实现

方式类似，在此不再赘述。

完整SID和压缩SID的转换关系可以根据SRv6 Locator的规律，可以采用如下方式：

完整SID = B (Block, 即通用Prefix) + N (节点ID) + F (Function) + A
5 (Args) (可选) + Padding (可选) ;

压缩SID = N (节点ID) + F (Function) 。

由此可见，压缩SID加上公共前缀/地址块 (Block)，可以完整的组成一个SID。

为支持压缩，需要规划支持压缩的SRv6 SID格式。比如Common Prefix
10 为96 bits，压缩SID为32bits；比如Common Prefix为64bits，紧跟压缩SID为32bits，剩余低32位为0。为标识特定的SRv6 SID为可压缩SID，需要扩展对应的控制面协议，将可压缩SID发布出去，发布SID可压缩的能力以及SID的结构信息。

(三) 关于容纳压缩SID和标准SID的容器 (Container) 格式的介绍：

15 当压缩SID和标准SID被编排在SRH SID List 中时，为准确定位压缩SID，需要满足压缩SID按照128bit对齐的方式来编排，即一行128bit需要放4个32bits压缩SID，或多个其他长度的缩短压缩SID。如果排不满，则需要补齐填充 (Padding)，对齐128bit。

为方便理解，在本公开实施例中定义SID Container 的概念，其是一个
20 128bit。一个SID Container可以包括：

(1) 一个SRv6标准SID；

(2) 多个压缩SID (X-SID)，比如，4个32bits的压缩SID或者8个16bits的压缩SID。

当多种SID混编时，需要保证128bit对齐。当压缩SID无法排满128bit时，
25 需要补齐Padding。以32bits的压缩SID为例，SID Container可能的格式如图7

所示。

由于SRv6 SRH中可以编码多种类型的SID Container，所以这个SRH可以称之为压缩SRH。是一种新型SRv6的实现，可以称之为压缩SRv6。下面内容介绍数据面压缩SRv6的方案细节。

5 （四）关于压缩SRv6数据面方案：压缩SRH的介绍：

压缩SRH与SRH[RFC8754]格式保持一致，没有对其格式和字段语义进行改动，但支持将128bit的SID和32bit的压缩SID混合编程在压缩SRH中。其格式如图8所示，包括：下一报头（Next Header）字段、Hdr Ext len字段、路由类型（Routing Type）字段、段剩余（Segment left）字段、最后一项（Last Entry）字段、标记（Flags）字段、标签（Tag）字段、X-SID容器（X-SID Container）字段（包含128bit value）和可选TLV（Optional TLV）对象字段（可变的）。

10

当128 bit的SID和32 bit X-SID混编在X-SRH中时，其编码示例如图9所示。

15 一段压缩SRv6路径可能由SRv6子路径和SRv6压缩子路径组成。SRv6子路径由SRv6 SID编码。SRv6压缩子路径由一个支持压缩的128bit SRv6 SID引导开始，并由随后跟随的多个压缩SID组成。在压缩SID List结束时，需以128bit为界结束，下一个可以为标准的128bit SRv6 SID或者下一段不同前缀的SRv6压缩子路径。

20 为标识SID list中SRv6压缩路径的起始和结束，也即128bits SID和32bits的SID之间的边界，需要再新增若干种类型的属性（Flavor），并发布对应Flavor的SID。COC32属性（COC32 Flavor）相关行为（Behavior）的值详见表2，COC16相关暂不定义。

表2

SID Flavor 类型	功能描述
---------------	------

None	不增加 Flavor, SID 的默认 Flavor 指示下一个是 128 bits SRv6 SID
COC32	标识下一个是 32bit 的压缩 SID
COC16	标识下一个是 16bit 的压缩 SID

因此当目的地址（Destination Address, DA）中SID携带COC32 Flavor时，表示需要将下一个32bit的压缩SID更新到目的地址。当DA中SID为不携带COC Flavor SID时，表示将“SL--”（SL减一），将下一个128 bit的SID更新到目的地址（SL>0时）。其中，SL可以定位到128比特Container的位置。

5 因此压缩SRv6在SID List中的编码规则包括：

（1）一段压缩SRv6路径的开始由一个128bits的COC32 Flavor SID指示，其包含COC32 Flavor SID。此SID携带了完整的SID信息，包含Common Prefix等信息，可用于与后续的压缩SID恢复完整的下一个SID；

10 （2）压缩路径中间的压缩SID均为携带COC32的压缩SID，指示下一个是32bit的压缩SID；

（3）压缩路径的最后一个压缩SID需为无COC Flavor的32 bits的压缩SID，用于指示压缩路径结束。其在DA中组合成的SRv6 SID按照128bits的SID的处理规则处理。

需要说明的是，最后一个SID为无COC Flavor SID的原因在于：其被更新到目的地址之后，DA中的SID由于无COC Flavor，所以其将被节点按照128 bits的SID来处理，与当前SRv6处理一致。因此它可以起到标识这段压缩路径结束，从32 bits的压缩SID切换到128bits的SID处理的作用。

可以理解的是，这种设计可以很好的支持以下两个场景：

（1）混编场景中压缩路径结束，从32bit切换到128bit的场景；

20 （2）纯压缩场景中，最后一个SID为无COC Flavor，因此最后一跳的处理与当前SRv6一致，更兼容。

此外，在压缩路径中，为了定位下一个压缩SID，可以新增压缩SID的索引（X-SID Index，以下简称XI）来定位其在SID Container中的位置。进

一步地，该索引还可以指示SID是否压缩或者指示SID的SID类型。

定位的基本思路：

1. 下一个压缩SID由SL+XI定位，SL定位到128比特container的位置，XI定位到128位container中具体的X-SID的位置；

5 2. 在混编场景下，如何定位SID长度变化的边界：

 (1) 通过SID的“下一个SID的SID类型”与当前SID的SID类型对比来确认SID的类型是否发生改变；比如：SID的“下一个SID的SID类型”若为32bit，而当前的SID的类型为128bit，则到达了边界；

 (2) 对目的地址中的XI进行相应的改变，以定位下一个SID/X-SID；

10 (3) 若下一个SID的SID类型是32bit，则将下一个32bit的X-SID更新到IPv6目的地址对应的X-SID位置，X-SID可以由 SRH[SL][XI]来定位。

 比如，索引可以放置在目的地址中的压缩的SID之后，示例性地，索引放在128bit的最低2bit位置，参见图10。在发布压缩SID时，需预留空间给索引，其值为0。在转发的过程中，XI的数值是压缩的SID的位置属性，指示其在X-SID Container中的位置。

 比如：通过COC32 Flavor SID指示更新下一个32 bits的压缩SID到目的地址，压缩SID位置可以由SL与XI定位。

 以32 bits的压缩SID为例，节点（Endpoint）收到数据包的处理伪代码如下：

```

20        if IPv6 DA hits a COC32 Flavor SID    //32bit X-SID处理
           if DA.XI==0:                        //下一行的第一个压缩的SID
               SL--;
               DA.XI = 3
           Else                                 //本行的下一个压缩的SID
25                DA.XI--
               DA[CP..CP+31] = SRH[SL][DA.XI]
```

Forward the packet based on new DA

Else //普通SRv6分支，不受影响

SRv6 processing

5 处理流程由COC Flavor SID的触发，操作数据也局限在COC Flavor SID之内，不影响已有SRv6主流程，对已有SID以及SRH的处理毫无影响。格式上兼容SRH，处理上兼容已有SRv6。其中XI在DA的压缩的SID之后。以纯压缩路径+128 bit VPN SID为例，编码与X-SID更新示例如图11所示。

（五）关于混编场景中数据包转发流程。

10 在部署SRv6压缩方案时，需要进行地址规划，然后升级控制面和数据面支持SRv6压缩。

首先，节点通过置位IGP等协议的SRv6 Capabilities TLV的扩展C-flag来标识本节点支持压缩能力。

15 节点实例化可以支持压缩的SRv6 SID，并将通过IGP/BGP/BGP-LS等协议发布到网络中或上送给控制器。节点在实例化SID之后，将会生成对应的SID转发表项。

参见图12，节点k可以发布两套SID，两套SID都携带C-Flag，均支持压缩，一套携带COC32 Flavor，另一套不携带Flavor。比如A:k:1::为绑定到某接口的携带COC32的END.X SID，A:k:2::为绑定到同样接口的不携带COC32 Flavor的END.X SID。根据这个规划，A:1:1::为COC32 Flavor End.X SID，
20 A:4:2::为无COC Flavor的SID。

在发布SID时，需要置位SID A:1:1::、A:2:1::、A:3:1::、A:4:1::、A:4:2::、A:6:1::、A:7:1::、A:8:1::、A:9:1::、A:9:2::以及A:10:10::（格式支持压缩的VPN SID，方便纯压缩案例演示）对应的TLV（如SRv6 End SID sub-TLV、SRv6 End.X SID sub-TLV、SRv6 LAN End.X SID sub-TLV）中的C-flag来标识本SID的格式支持压缩。
25

在发布的SID中，包含有携带COC32 Flavor的可压缩SID，用于标识SID

后面的SID是32的X-SID。比如下例的多个SID A:k:1::。

A:k:1::在发布时还需要携带SID Structure Sub-sub-TLV。假设Common Prefix (A) 的长度是64, X-SID的长度是32 (Node ID长度是16, Function ID长度是16), Argument的长度是32, 则发布的SID的SID Structure Sub-sub-TLV中Locator Block的长度是64, Node ID的长度是16, Function的长度是16, Argument长度为32。

图中A1:: 前缀是不支持压缩SID的前缀, 所以A1::5:1为不可压缩的SID。它在发布时也会携带SID Structure Sub-sub-TLV, 其中Locator Block的长度是64, Node ID的长度是16, Function的长度是48。

设备通过BGP-LS将节点支持压缩的能力, 支持压缩的SID和新增的携带COC32 Flavor的SID上送给控制器, 用于控制器计算路径使用。

控制器算路完成之后, 可以通过BGP SR Policy下发包含压缩SID的SID List, 此时需要在SID list的Segment sub-TLV之间插入描述SID格式的sub-TLV。比如以下示例中的10个SID, 需要在A:2:1::, A:7:1::之前插入SID encoding sub-TLV, Start为Common Prefix的长度(图中为63), End为Common Prefix + 32(图中为95)。还需要在A1::5:1, A:10:10::之前插入SID encoding sub-TLV, 其Start = 0, End = 127。

表3

SID	Block	Node	Function	Arg	Padding	备注
A:k:1::	64	16	16	32	0	可压缩, COC32 Flavor
A:k:2::	64	16	16	32	0	可压缩, None COC Flavor
A1:5::1	64	16	48	0	0	不可压缩, A1 前缀
A:10:10::	64	16	16	32	0	可压缩 VPN SID

示例中, SID List总共包含了10 SIDs:

A:1:1:: END.X (COC32)

A:2:1:: END.X (COC32)

A:3:1:: END.X (COC32)

A:4:2:: END.X

A1::5:1 END.X 不支持压缩的SID

A:6:1:: END.X (COC32)

5 A:7:1:: END.X (COC32)

A:8:1:: END.X (COC32)

A:9:2:: END.X

A:10:10:: END.DT4 VPN SID (不参与压缩)

经过简化 (Reduced) 模式以及压缩编码之后, SID List如图12示例。

10 其中END.X SID A:1:1::未被编码在SID List中。

(1) 当节点1收到数据包时, 目的地址“A:1:1::”在Local SID表中命中到节点本地发布的COC Flavor END.X SID, 此时SRH中SL=5, DA.XI=0, 因此SL--, DA.XI=3, 指向2:1, 将X-SID 2:1更新到DA中, 转发到下一个节点2。此时目的地址为A:2:1::3。

15 (2) 当节点2收到数据包时, 目的地址“A:2:1:3::”在Local SID表中命中匹配到节点本地发布的COC Flavor END.X SID, 此时SL=4, 由于DA.XI为3, 大于0, 所以节点将DA.XI--, 将SRH[SL][DA.XI]指向的下一个X-SID 3:1更新 (或称为复制) 到DA中, 转发到下一个节点。此时目的地址为“A:3:1:2”。

(3) 当节点3收到数据包时, 目的地址“A:3:1:2::”在Local SID表中命中到节点本地发布的COC Flavor END.X SID, 此时SRH中SL=4, DA.XI=2, 大于0, 所以DA.XI--, 指向4:2, 将X-SID 4:2更新到DA中, 转发到下一个节点。此时目的地址为“A:4:2::1”。

(4) 当节点4收到数据包时, 目的地址“A:4:2:1::”在Local SID表中命中到节点本地发布的END.X SID, 因为无COC Flavor, 所以默认指示下一个SID是128bit的SID, 因此, SL=SL-1=3, 将“A1::5:1”复制到DA中进行转发。

(5) 节点5是一个普通SRv6节点, 所以执行普通SRv6转发, 将“A:6:1::”

复制到DA中，转发到下一个节点。

(6) 节点6收到数据包时，目的地址“A:6:1::”在Local SID表中命中到节点本地发布的COC Flavor END.X SID，此时SRH中SL=2，DA.XI=0，所以SL--，DA.XI=3，指向7:1，将X-SID 7:1更新到DA中，转发到下一个节点。

5 此时目的地址为“A:7:1::3”。

可以理解的是“7:1”相当于第一SID，“A:6:1::”相当于第二SID。

(7) 同理，节点7，节点8收到数据包时，处理COC Flavor SID，更新目的地址，转发数据包。

(8) 节点9收到数据包之后，步骤同节点4，由于DA中SID在Local SID
10 表中命中无COC Flavor的SID，所以SL-- = 0，将VPN SID 复制到DA中转发到节点10。

(9) 节点10按照正常VPN SID处理。

(六) 关于纯压缩场景中的数据包转发流程：

参见图13，控制面流程原理，拓扑信息和SID信息同混编案例。在下发
15 SR策略（SR Policy）时，BGP SR Policy的SID list里面需要插入对应的SID Encoding sub-TLV描述格式。以下例为例：

需要在A:2:1::之前插入SID encoding sub-TLV，Start为Common Prefix的长度（图中为63），End为Common Prefix + 32（图中为95）。

示例中，SID List总共包含了10 SIDs：

20 A:1:1:: END.X（COC32）

A:2:1:: END.X（COC32）

A:3:1:: END.X（COC32）

A:4:1:: END.X（COC32）

A:5:1:: END.X（COC32）

25 A:6:1:: END.X（COC32）

A:7:1:: END.X (COC32)

A:8:1:: END.X (COC32)

A:9:1:: END.X (COC32)

A:10:10:: END.DT4 VPN SID (参与压缩)。

5 经过Reduced模式以及压缩编码之后，SID List如图13所示。其中END.X SID A:1:1::未被编码在SID List中。

(1)节点0封装好数据包之后，将数据包发送给下一跳节点1，此时SRH中SL=3，DA.XI=0。SID的Flavor为COC，标识下一个SID封装格式为32bits的X-SID。

10 (2)当节点1收到数据包时，目的地址“A:1:1::”是本地发布的一个格式支持压缩的COC Flavor END.X，指示下一个SID是32 bits的X-SID，而此时的SL=3，DA.XI=0，所以SL--，DA.XI=3，指向2:1，将2:1复制到DA更新X-SID，然后基于新的DA进行查表转发。

可以理解的是，“2:1”相当于第一SID，“1:1”相当于第二SID。

15 (3)后续节点2、3、4、5、6、7、8、9同理，节点基于SL和DA.XI的值将对应X-SID被更新到DA，查表转发。

(4)节点10收到数据包时，DA为“A:10:10::”，其是一个不带COC Flavor的SID（VPN SID本身也无此类Flavor），执行VPN SID处理。

参见图14，本公开实施例还提供一种节点，该节点1400包括：

20 获取模块1401，配置为获取第一SID的第一信息和第二信息，所述第一信息用于指示所述第一SID所在的容器在数据包的SID列表中的位置，所述第二信息用于指示所述第一SID在所述容器中的位置；

处理模块1402，配置为根据所述第一信息和第二信息，得到所述第一SID在SID列表中的位置；

发送模块 1403, 配置为将所述第一SID复制到数据包的地址中, 发送所述数据包。

在一些实施方式中, 获取模块 1401 配置为: 获取第二SID的第三信息和第四信息, 所述第三信息用于指示所述第二SID所在的容器在所述数据包的SID列表中的位置, 所述第四信息用于指示所述第二SID在所述容器中的位置; 根据所述第二SID的第三信息和第四信息, 得到所述第一SID的第一信息和第二信息; 其中, 在所述SID列表中所述第二SID为所述第一SID的一个SID。

在一些实施方式中, 获取模块 1401 配置为: 从所述数据包的SRH中获取所述第二SID的第三信息; 从所述数据包的地址中获取所述第二SID的索引, 所述第二SID的索引包括: 第四信息。

在一些实施方式中, 获取模块 1401 配置为: 根据所述第二SID的第四信息或根据所述第二SID的属性信息, 确定所述第一SID与所述第二SID是否在同一个容器中; 如果所述第一SID与所述第二SID在同一个容器中, 则所述第一SID的第一信息与所述第二SID的第三信息相同, 以及根据所述第二SID的第四信息, 得到所述第一SID的第二信息; 或者, 如果所述第一SID与所述第二SID不在同一个容器中, 则根据所述第二SID的第三信息, 得到所述第一SID的第一信息, 以及根据所述第一SID的SID类型, 确定所述第一SID的第二信息。

在一些实施方式中, 所述第一SID的SID类型是由所述第二SID的属性信息指示的。

在一些实施方式中, 该节点 1400 还包括: 更新模块, 配置为将所述第一SID的第一信息更新到所述数据包的SRH中; 将所述第一SID的第二信息作为所述第一SID的索引更新到所述数据包的地址中。

在一些实施方式中，获取模块 1401 配置为：获取第二SID的第三信息、第四信息和第五信息，所述第三信息用于指示所述第二SID所在的容器在所述数据包的SID列表中的位置，所述第四信息用于指示所述第二SID在所述容器中的位置，所述第五信息用于指示所述第二SID的SID类型和/或指示所述第二SID是否压缩；根据所述第二SID的第三信息、第四信息和第五信息，
5 得到所述第一SID的第一信息和第二信息；其中，在所述SID列表中所述第二SID为所述第一SID的上一个SID。

在一些实施方式中，获取模块 1401 配置为：从所述数据包的SRH中获取所述第二SID的第三信息；从所述数据包的目的地地址中获取所述第二SID
10 的索引，所述第二SID的索引包括：第四信息和第五信息。

在一些实施方式中，获取模块 1401 配置为：根据所述第二SID的SID类型和所述第一SID的SID类型，确定所述第一SID与所述第二SID是否在同一个容器中；如果所述第一SID与所述第二SID在同一个容器中，则所述第一SID的第一信息与所述第二SID的第三信息相同，以及根据所述第二SID的第四信息，得到所述第一SID的第二信息；或者，如果所述第一SID与所述第二SID不在同一个容器中，则根据所述第二SID的第三信息，得到所述第一SID的第一信息，以及根据所述第一SID的SID类型，确定所述第一SID的第二信息。
15

在一些实施方式中，所述容器用于容纳多个压缩SID，所述压缩SID包
20 括：节点标识和所述节点的function的取值。

在一些实施方式中，发送模块 1403 配置为：将所述第一SID复制到数据包的目的地地址中，所述第一SID与所述目的地地址中的公共前缀或地址块组成得到一个新的目的地地址，并根据所述新的目的地地址将所述数据包转发给下一个节点；或者，将所述第一SID和第二信息复制到数据包的目的地地址中，
25 所述第一SID和第二信息与所述目的地地址中的公共前缀或地址块组成得到

一个新的目的地址，并根据所述新的目的地址将所述数据包转发给下一个节点；或者，将所述第一SID和第二信息复制到数据包的目的地址中，将所述数据包转发给下一个节点。

需要说明的是：上述实施例提供的节点在进行数据包处理时，仅以上
5 述各程序模块的划分进行举例说明，实际应用中，可以根据需要而将上述处理分配由不同的程序模块完成，即将节点的内部结构划分成不同的程序模块，以完成以上描述的全部或者部分处理。另外，上述实施例提供的节点与数据包处理的方法实施例属于同一构思，其具体实现过程详见方法实施例，这里不再赘述。

10 参见图15，本公开实施例还提供一种节点，该节点1500包括：收发机1501、处理器1502和用于存储能够在处理器1502上运行的计算机程序的存储器1503；

所述收发机1501在所述处理器1502的控制下发送和接收数据；

所述处理器1502读取所述存储器1503中的计算机程序时，执行本公
15 开前述实施例所述的数据包处理的方法的步骤。

可以理解，存储器1503可以是易失性存储器或非易失性存储器，也可包括易失性和非易失性存储器两者。其中，非易失性存储器可以是只读存储器（Read Only Memory，ROM）、可编程只读存储器（Programmable Read-Only Memory，PROM）、可擦除可编程只读存储器（Erasable
20 Programmable Read-Only Memory，EPROM）、电可擦除可编程只读存储器（Electrically Erasable Programmable Read-Only Memory，EEPROM）、磁性随机存取存储器（Ferromagnetic Random Access Memory，FRAM）、快闪存储器（Flash Memory）、磁表面存储器、光盘、或只读光盘（Compact Disc Read-Only Memory，CD-ROM）；磁表面存储器可以是磁盘存储器或磁带
25 存储器。易失性存储器可以是随机存取存储器（Random Access Memory，

RAM)，其用作外部高速缓存。通过示例性但不是限制性说明，许多形式的RAM可用，例如静态随机存取存储器（Static Random Access Memory，SRAM）、同步静态随机存取存储器（Synchronous Static Random Access Memory，SSRAM）、动态随机存取存储器（Dynamic Random Access Memory，DRAM）、同步动态随机存取存储器（Synchronous Dynamic Random Access Memory，SDRAM）、双倍数据速率同步动态随机存取存储器（Double Data Rate Synchronous Dynamic Random Access Memory，DDRSDRAM）、增强型同步动态随机存取存储器（Enhanced Synchronous Dynamic Random Access Memory，ESDRAM）、同步连接动态随机存取存储器（SyncLink Dynamic Random Access Memory，SLDRAM）、直接内存总线随机存取存储器（Direct Rambus Random Access Memory，DRRAM）。本发明实施例描述的存储器1503旨在包括但不限于这些和任意其它适合类型的存储器。

上述本发明实施例揭示的方法可以应用于处理器1502中，或者由处理器1502实现。处理器1502可能是一种集成电路芯片，具有信号的处理能力。在实现过程中，上述方法的各步骤可以通过处理器1502中的硬件的集成逻辑电路或者软件形式的指令完成。上述的处理器1502可以是通用处理器、数字信号处理器（Digital Signal Processor，DSP），或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。处理器1502可以实现或者执行本发明实施例中的公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者任何常规的处理器等。结合本发明实施例所公开的方法的步骤，可以直接体现为硬件译码处理器执行完成，或者用译码处理器中的硬件及软件模块组合执行完成。软件模块可以位于存储介质中，该存储介质位于存储器1503，处理器1502读取存储器1503中的信息，结合其硬件完成前述方法的步骤。

在示例性实施例中，节点可以被一个或多个应用专用集成电路

(Application Specific Integrated Circuit, ASIC)、DSP、可编程逻辑器件 (Programmable Logic Device, PLD)、复杂可编程逻辑器件 (Complex Programmable Logic Device, CPLD)、现场可编程门阵列 (Field-Programmable Gate Array, FPGA)、通用处理器、控制器、微控制器 (Micro Controller Unit, MCU)、微处理器 (Microprocessor)、或其他电子元件实现, 用于执行前述方法。

在示例性实施例中, 本发明实施例还提供了一种计算机可读存储介质, 例如包括计算机程序的存储器1503, 上述计算机程序可由节点1500的处理器1502执行, 以完成前述方法所述步骤。计算机可读存储介质可以是FRAM、ROM、PROM、EPROM、EEPROM、Flash Memory、磁表面存储器、光盘、或CD-ROM等存储器; 也可以是包括上述存储器之一或任意组合的各种设备。

本公开实施例还提供了一种计算机可读存储介质, 所述计算机可读存储介质上存储有计算机程序, 所述计算机程序被处理器执行时实现本公开实施例前述数据包处理的方法的步骤。

本申请所提供的几个方法或设备实施例中所揭露的特征, 在不冲突的情况下可以任意组合, 得到新的方法实施例或设备实施例。

结合本公开公开内容所描述的方法或者算法的步骤可以硬件的方式来实现, 也可以由在处理器执行软件指令的方式来实现。软件指令可以由相应的软件模块组成, 软件模块可以被存放于RAM、闪存、ROM、EPROM、EEPROM、寄存器、硬盘、移动硬盘、只读光盘或者本领域熟知的任何其它形式的存储介质中。一种示例性的存储介质耦合至处理器, 从而使处理器能够从该存储介质读取信息, 且可向该存储介质写入信息。当然, 存储介质也可以是处理器的组成部分。处理器和存储介质可以携带在ASIC中。另外, 该ASIC可以携带在核心网接口设备中。当然, 处理器和存储介质也

可以作为分立组件存在于核心网接口设备中。

本领域技术人员应该可以意识到，在上述一个或多个示例中，本公开所描述的功能可以用硬件、软件、固件或它们的任意组合来实现。当使用软件实现时，可以将这些功能存储在计算机可读介质中或者作为计算机可读介质上的一个或多个指令或代码进行传输。计算机可读介质包括计算机存储介质和通信介质，其中通信介质包括便于从一个地方向另一个地方传送程序的任何介质。存储介质可以是通用或专用计算机能够存取的任何可用介质。

以上所述的具体实施方式，对本公开的目的、技术方案和有益效果进行了进一步详细说明，所应理解的是，以上所述仅为本公开的具体实施方式而已，并不用于限定本公开的保护范围，凡在本公开的技术方案的基础上，所做的任何修改、等同替换、改进等，均应包括在本公开的保护范围之内。

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

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

功能的装置。

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

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

显然，本领域的技术人员可以对本公开实施例进行各种改动和变型而不脱离本公开的精神和范围。这样，倘若本公开实施例的这些修改和变型属于本公开权利要求及其等同技术的范围之内，则本公开也意图包含这些改动和变型在内。

权利要求书

1. 一种数据包处理的方法，包括：

获取第一段标识 SID 的第一信息和第二信息，所述第一信息用于指示
所述第一 SID 所在的容器在数据包的 SID 列表中的位置，所述第二信息用
5 于指示所述第一 SID 在所述容器中的位置；

根据所述第一信息和第二信息，得到所述第一 SID 在 SID 列表中的位
置；

将所述第一 SID 复制到数据包的地址中，发送所述数据包。

2. 根据权利要求 1 所述的方法，其中，所述获取第一 SID 的第一信息
10 和第二信息，包括：

获取第二 SID 的第三信息和第四信息，所述第三信息用于指示所述第
二 SID 所在的容器在所述数据包的 SID 列表中的位置，所述第四信息用于
指示所述第二 SID 在所述容器中的位置；

根据所述第二 SID 的第三信息和第四信息，得到所述第一 SID 的第一
15 信息和第二信息。

3. 根据权利要求 2 所述的方法，其中，所述获取第二 SID 的第三信息
和第四信息，包括：

从所述数据包的段路由头 SRH 中获取所述第二 SID 的第三信息；

从所述数据包的地址中获取所述第二 SID 的索引，所述第二 SID
20 的索引包括：第四信息。

4. 根据权利要求 2 所述的方法，其中，所述根据所述第二 SID 的第三
信息和第四信息，得到所述第一 SID 的第一信息和第二信息，包括：

根据所述第二 SID 的第四信息或根据所述第二 SID 的属性信息，确定
所述第一 SID 与所述第二 SID 是否在同一个容器中；

25 如果所述第一 SID 与所述第二 SID 在同一个容器中，则所述第一 SID

的第一信息与所述第二 SID 的第三信息相同, 以及根据所述第二 SID 的第四信息, 得到所述第一 SID 的第二信息; 或者,

如果所述第一 SID 与所述第二 SID 不在同一个容器中, 则根据所述第二 SID 的第三信息, 得到所述第一 SID 的第一信息, 以及根据所述第一 SID
5 的 SID 类型, 确定所述第一 SID 的第二信息。

5. 根据权利要求 4 所述的方法, 其中, 所述第一 SID 的 SID 类型是由所述第二 SID 的属性信息指示的。

6. 根据权利要求 1 所述的方法, 其中, 所述方法还包括:

将所述第一 SID 的第一信息更新到所述数据包的 SRH 中;

10 将所述第一 SID 的第二信息作为所述第一 SID 的索引更新到所述数据包的目的地地址中。

7. 根据权利要求 1 所述的方法, 其中, 所述将所述第一 SID 复制到数据包的目的地地址中, 发送所述数据包, 包括:

将所述第一 SID 复制到数据包的目的地地址中, 所述第一 SID 与所述目的
15 的地址中的公共前缀或地址块组成得到一个新的目的地地址, 并根据所述新的目的地地址将所述数据包转发给下一个节点;

或者, 将所述第一 SID 和第二信息复制到数据包的目的地地址中, 所述第一 SID 和第二信息与所述目的地地址中的公共前缀或地址块组成得到一个新的目的地地址, 并根据所述新的目的地地址将所述数据包转发给下一个节点;

20 或者, 将所述第一 SID 和第二信息复制到数据包的目的地地址中, 将所述数据包转发给下一个节点。

8. 根据权利要求 1 所述的方法, 其中, 所述获取第一 SID 的第一信息和第二信息, 包括:

获取第二 SID 的第三信息、第四信息和第五信息, 所述第三信息用于
25 指示所述第二 SID 所在的容器在所述数据包的 SID 列表中的位置, 所述第

四信息用于指示所述第二 SID 在所述容器中的位置，所述第五信息用于指示所述第二 SID 的 SID 类型和/或指示所述第二 SID 是否压缩；

根据所述第二 SID 的第三信息、第四信息和第五信息，得到所述第一 SID 的第一信息和第二信息；

5 其中，在所述 SID 列表中所述第二 SID 为所述第一 SID 的上一个 SID。

9. 根据权利要求 8 所述的方法，其中，所述获取第二 SID 的第三信息、第四信息和第五信息，包括：

从所述数据包的 SRH 中获取所述第二 SID 的第三信息；

10 从所述数据包的目地址中获取所述第二 SID 的索引，所述第二 SID 的索引包括：第四信息和第五信息。

10. 根据权利要求 8 所述的方法，其中，所述根据所述第二 SID 的第三信息、第四信息和第五信息，得到所述第一 SID 的第一信息和第二信息，包括：

15 根据所述第二 SID 的 SID 类型和所述第一 SID 的 SID 类型，确定所述第一 SID 与所述第二 SID 是否在同一个容器中；

如果所述第一 SID 与所述第二 SID 在同一个容器中，则所述第一 SID 的第一信息与所述第二 SID 的第三信息相同，以及根据所述第二 SID 的第四信息，得到所述第一 SID 的第二信息；或者，

20 如果所述第一 SID 与所述第二 SID 不在同一个容器中，则根据所述第二 SID 的第三信息，得到所述第一 SID 的第一信息，以及根据所述第一 SID 的 SID 类型，确定所述第一 SID 的第二信息。

11. 根据权利要求 1 所述的方法，其中，所述容器用于容纳一个标准 SID 或者多个压缩 SID，所述压缩 SID 包括：节点标识和所述节点的功能 function 的取值。

25 12. 一种节点，包括：

获取模块，配置为获取第一 SID 的第一信息和第二信息，所述第一信息用于指示所述第一 SID 所在的容器在数据包的 SID 列表中的位置，所述第二信息用于指示所述第一 SID 在所述容器中的位置；

5 处理模块，配置为根据所述第一信息和第二信息，得到所述第一 SID 在 SID 列表中的位置；

发送模块，配置为将所述第一 SID 复制到数据包的目的地址中，发送所述数据包。

13. 一种节点，包括：收发机、处理器和用于存储能够在处理器上运行的计算机程序的存储器；

10 所述收发机用于在所述处理器的控制下发送和接收数据；

所述处理器用于读取所述存储器中的计算机程序时，执行如权利要求 1 至 11 中任一项所述的数据包处理的方法的步骤。

14. 一种计算机可读存储介质，所述计算机可读存储介质上存储有计算机程序，所述计算机程序被处理器执行时实现包括如权利要求 1 至 11 中任
15 一项所述的数据包处理的方法的步骤。