

DATA PACKET PROCESSING METHOD AND DEVICE

CROSS-REFERENCE TO RELATED APPLICATION

[0001] The disclosure is based upon and claims priority to Chinese patent application No. 202010334603.0, filed on April 24, 2020, the disclosure of which is hereby incorporated by reference in its entirety.

TECHNICAL FIELD

[0002] The present disclosure relates to the technical field of communications, and in particular to a method and device for data packet processing.

BACKGROUND

[0003] In carrier networks, the requirement on the number of Segment Routing (SR) label layers is high. Taking the 5th Generation (5G) bearer network as an example, with the centralized deployment of the 5G core network, the traffic of a base station needs to pass through a metropolitan area network and an Internet Protocol (IP) backbone network. In typical scenarios, in the metropolitan area network, an access ring has 8-10 nodes, a convergence ring has 4-8 nodes, and a core ring also has 4-8 nodes; in the IP backbone network, the traffic also needs to pass through multiple router nodes. At the same time, due to network slicing, highly reliable Service-Level Agreement (SLA) and requirements for manageability and controllability, the carrier network needs to be able to specify an explicit path, and an end-to-end SR tunnel will have 10 hops or more. Therefore, at present, most carriers deploying Multi-Protocol Label Switching (MPLS)-SR are required to support a Segment ID (SID) label of more than 8 layers.

[0004] At present, SRv6 is an SR solution extended based on Internet Protocol Version 6 (IPv6). The SRv6 solution implements forwarding based on a Segment Routing Header (SRH), and has an SID length of 128 bits. The 8-layer SID incurs an overhead of 128 bytes to a data packet. For application payloads with an average length of 256 bytes,

the overhead of SRv6 is more than 1/3, and then the bandwidth utilization is reduced to 67% or less. In the same scenario, the overhead of SR-MPLS is only 32 bytes, and the bandwidth utilization is still 89%. The comparative analysis of the bearing efficiencies of SRv6 and SR-MPLS when the number of SIDs is 1 to 10 is shown in FIG. 1 (only the overheads of the SRH and the SR-MPLS SID are compared simply).

[0005] The increased overhead, on the one hand, causes a decrease in the network utilization, and on the other hand, brings more challenges to support deep data packet load balancing, in-band telemetry, and Network Service Header (NSH).

[0006] In addition, SRv6 deployment will inevitably coexist with SR-MPLS networks. The difference in network utilization may cause a problem of unbalanced interfaces at a network perimeter, which leads to the waste of investment. When an SR-MPLS network interconnects with an SRv6 network domain, considering the case of 100G links, 256-byte data packets, and 8-layer SID, a 100 Gigabit Ethernet (GE) link in an SR-MPLS domain may require two 100GE links to match in an SRv6 domain due to the large difference in link utilization.

[0007] In carrier applications, the SRv6 requires a network chip to insert fields of more than 128 bytes, which are equivalent to the depth of a 32-layer MPLS-SR label, which is beyond the capability of the deployed network chip. If a loopback solution is used inside the chip, the network performance will be greatly reduced and higher delay and jitter will be introduced. In a redesigned network chip, supporting the SRv6 requires further expansion of an internal processing bus bandwidth, which is the key factor of chip cost and power consumption.

[0008] The SRv6 requires the network chip to read a complete SRH at an intermediate node, and then extract and forward the segment to be processed according to a position indicated by a pointer. Compared with the MPLS-SR which only needs to read the outermost label, the introduced complexity further increases a processing delay of the network chip.

[0009] Low power consumption and low delay are the key factors of 5G solutions of the carriers. The increase of power consumption, cost and delay brought by the complexity of SRv6 to the network chip brings challenges to its implementation and application.

[0010] Based on the above analysis, the overhead of the existing SRv6 data packet is high, which increases the complexity of the network chip and makes it difficult to smoothly upgrade. As a result, the SRv6 cannot be rapidly deployed to the carrier networks, and further evolution based on the SRv6 technology is needed.

SUMMARY

[0011] A purpose of the embodiments of the present disclosure is to provide a method and device for data packet processing, so as to solve the problem of high overhead of the existing SRv6 data packet.

[0012] In the first aspect, an embodiment of the present disclosure provides a method for data packet processing, which may include the following operations. First information and second information of a first SID are obtained, the first information being used for indicating a position of a container where the first SID is located in an SID list of a data packet, and the second information being used for indicating a position of the first SID in the container. The position of the first SID in the SID list is obtained according to the first information and the second information. The first SID is copied to a Destination Address (DA) of the data packet, and the data packet is sent.

[0013] In some optional embodiments of the present disclosure, the operation that the first information and the second information of the first SID are obtained may include the following operations. Third information and fourth information of a second SID are obtained, the third information being used for indicating a position of a container where the second SID is located in the SID list of the data packet, and the fourth information being used for indicating a position of the second SID in the container. The first information and the second information of the first SID are obtained according to the third information and the fourth information of the second SID.

[0014] In some optional embodiments of the present disclosure, the operation that the third information and the fourth information of the second SID are obtained may include the following operations. The third information of the second SID is obtained from an SRH of the data packet. The index of the second SID is obtained from the DA of the data packet. The index of the second SID includes the fourth information.

[0015] In some optional embodiments of the present disclosure, the operation that the first information and the second information of the first SID are obtained according to the third information and the fourth information of the second SID may include the following operations. It is determined whether the first SID and the second SID are in the same container according to the fourth information of the second SID or flavor of the second SID. If the first SID and the second SID are in the same container, the first information of the first SID is the same as the third information of the second SID, and the second information of the first SID is obtained according to the fourth information of the second SID; or, if the first SID and the second SID are not in the same container, the first information of the first SID is obtained according to the third information of the second SID, and the second information of the first SID is determined according to an SID type of the first SID.

[0016] In some optional embodiments of the present disclosure, the SID type of the first SID is indicated by the flavor of the second SID.

[0017] In some optional embodiments of the present disclosure, the method may further include the following operations. The first information of the first SID is updated to the SRH of the data packet. The second information of the first SID is updated to the DA of the data packet as the index of the first SID.

[0018] In some optional embodiments of the present disclosure, the operation that the first SID is copied to the DA of the data packet, and the data packet is sent may include the following operation. The first SID is copied to the DA of the data packet, the first SID is combined with a common prefix or an address block in the DA to get a new DA, and the data packet is forwarded to a next endpoint according to the new DA; or, the first SID and the second information are copied to the DA of the data packet, the first SID and the second information are combined with the common prefix or address block in the DA to get a new DA, and the data packet is forwarded to a next endpoint according to the new DA; or, the first SID and the second information are copied to the DA of the data packet, and the data packet is forwarded to the next endpoint.

[0019] In some optional embodiments of the present disclosure, the operation that the first information and the second information of the first SID are obtained may include the following operations. The third information, the fourth information and fifth information

of the second SID are obtained, the third information being used for indicating a position of a container where the second SID is located in the SID list of the data packet, the fourth information being used for indicating a position of the second SID in the container, and the fifth information being used for indicating an SID type of the second SID and/or indicating whether the second SID is compressed. The first information and the second information of the first SID are obtained according to the third information, the fourth information and the fifth information of the second SID. The second SID is an SID immediately before the first SID in the SID list.

[0020] In some optional embodiments of the present disclosure, the operation that the third information, the fourth information and the fifth information of the second SID are obtained may include the following operations. The third information of the second SID is obtained from the SRH of the data packet. The index of the second SID is obtained from the DA of the data packet. The index of the second SID includes the fourth information and the fifth information.

[0021] In some optional embodiments of the present disclosure, the operation that the first information and the second information of the first SID are obtained according to the third information, the fourth information and the fifth information of the second SID may include the following operations. It is determined whether the first SID and the second SID are in the same container according to the SID type of the second SID and the SID type of the first SID. If the first SID and the second SID are in the same container, the first information of the first SID is the same as the third information of the second SID, and the second information of the first SID is obtained according to the fourth information of the second SID; or, if the first SID and the second SID are not in the same container, the first information of the first SID is obtained according to the third information of the second SID, and the second information of the first SID is determined according to the SID type of the first SID.

[0022] In some optional embodiments of the present disclosure, the container is used for containing a standard SID or multiple compressed SIDs. The compressed SID includes a node ID and a value of function of the node.

[0023] In the second aspect, an embodiment of the present disclosure further provides a node, which may include: an obtaining module, a processing module, and a sending

module.

[0024] The obtaining module is configured to obtain the first information and the second information of the first SID, the first information being used for indicating a position of a container where a first SID is located in an SID list of a data packet, and the second information being used for indicating a position of the first SID in the container.

[0025] The processing module is configured to obtain the position of the first SID in the SID list according to the first information and the second information.

[0026] The sending module is configured to copy the first SID to the DA of the data packet, and send the data packet.

[0027] In the third aspect, an embodiment of the present disclosure further provides a node, which may include: a transceiver, a processor, and a memory configured to store a computer program capable of running in the processor.

[0028] The transceiver is configured to send and receive data under control of the processor.

[0029] The processor is configured to execute, when reading the computer program in the memory, steps of the method for data packet processing in the embodiment in the first aspect of the present disclosure.

[0030] In the fourth aspect, an embodiment of the present disclosure further provides a computer-readable storage medium, in which a computer program is stored. When executed by the processor, the computer program implements the steps of the method for data packet processing in the embodiment in the first aspect of the present disclosure.

[0031] In the embodiments of the present disclosure, the SID list includes at least the compressed SID; when the DA of the data packet is updated, the compressed SID is spliced with the common prefix in the current DA to form a new SID, and it continues to perform table lookup and forwarding. In this way, the data packet is compressed, and the header overhead of the data packet is effectively reduced. In addition, the standard SID and the compressed SID may be mixed encoded in the SID list, so as to support mixed arrangement of compressed nodes and common nodes to achieve existed network evolution and smooth upgrade.

BRIEF DESCRIPTION OF THE DRAWINGS

- [0032] By reading the detailed description of preferred implementation modes below, a variety of other advantages and benefits will become clear to those of ordinary skill in the art. The accompanying drawings are only intended to illustrate the purpose of the preferred implementation modes and are not considered as a limitation on the present disclosure. In addition, the same reference marks are used to indicate the same parts throughout the accompanying drawings. In the accompanying drawings:
- [0033] FIG. 1 is a schematic diagram showing the comparative analysis of the bearing efficiencies of SR with different numbers of SIDs when the payload length is 256 bytes (B).
- [0034] FIG. 2 is a schematic diagram of an encapsulation format of a standard SRv6 extension header.
- [0035] FIG. 3 is a flowchart of a method for data packet processing in an embodiment of the present disclosure.
- [0036] FIG. 4 is a schematic diagram of a standard network programmable SRv6 SID format.
- [0037] FIG. 5 is a schematic diagram of a typical network programmable SRv6 Locator.
- [0038] FIG. 6 is a schematic diagram of a compressible SRv6 SID format in an embodiment of the present disclosure.
- [0039] FIG. 7 is a schematic diagram of a compressed 32-bit SID container format in an embodiment of the present disclosure.
- [0040] FIG. 8 is a schematic diagram of a format of mixed encoding a 128-bit SRv6 SID and a 32-bit X-SID in an X-SRH according to an embodiment of the present disclosure.
- [0041] FIG. 9 is a schematic diagram of mixed encoding a 128-bit SRv6 SID and a 32-bit X-SID in an X-SRH according to an embodiment of the present disclosure.

[0042] FIG. 10 is a schematic diagram of XI in an embodiment of the present disclosure.

[0043] FIG. 11 is a schematic diagram of updating an X-SID in an embodiment of the present disclosure.

[0044] FIG. 12 is a schematic diagram of forwarding a data packet in a mixed encoding scenario according to an embodiment of the present disclosure.

[0045] FIG. 13 is a schematic diagram of forwarding a data packet in a pure compression scenario according to an embodiment of the present disclosure.

[0046] FIG. 14 is schematic diagram I of a node according to an embodiment of the present disclosure.

[0047] FIG. 15 is schematic diagram II of a node according to an embodiment of the present disclosure.

DETAILED DESCRIPTION

[0048] SR is a source routing technology, which constitutes path-oriented network architecture based on the concept of Software Defined Network (SDN), supports the multi-level programmable requirements of future networks, and can meet the connection requirements in the application scenarios of super connection and slicing of 5G. SR-MPLS is an SR solution formed based on the current mainstream MPLS forwarding plane.

[0049] SRv6 is an SR solution extended based on IPv6. The IPv6 technology is the main technology of the new generation network. The long-term consideration of the SRv6 based on the IPv6 is the evolution trend of the future network, and the research on the mechanism of the SRv6 technology is a hot spot in the industry.

[0050] The 128-bit SID of standard SRv6 uses the SID of IPv6 address format, and has a routable attribute compared with the SID of MPLS label format, which simplifies the creation of an inter-area path and implements the capability of easily creating an end-to-end path in an IPv6 network. At the same time, the SRv6 SID supports a

programmable capability, can flexibly process network and service functions, and can flexibly meet the requirements of various network and service functions in view of the collaborative support of centralized and distributed control planes, thus adapting to the needs of network and service development.

[0051] The SR guides a data packet to pass through a network by encapsulating a series of instructions through a header node. In SRv6 architecture, an instruction is a 128-bit IPv6 address. FIG. 2 shows an encapsulation format of a standard SRv6 extension header. An IPv6 header includes: a Version field, a Traffic Class field, a Flow Label field, a Payload Length field, a Next Header field, a Hop Limit field, a Source Address field, and a Destination Address field. An SRH extension header includes: a Next Header field, a Hdr Ext len field (indicating the length of the SRH header), a Routing Type field, a Segment Left (SL) field (indicating the number of intermediate nodes that should be accessed before reaching a destination node), a Last Entry field (indicating the index, which contains the last element of a segment list, in the segment list), a Flags field (indicating some IDs of the data packets), a Tag field (for identifying the data packets in the same group), a Segment list field (including the 128-bit IPv6 address), an Optional TLV object field (variable) and a Payload field.

[0052] It can be known from the encapsulation format of the SRv6 extension header that, the encapsulation length of the SRv6 extension header is 40 bytes (IPv6 header)+8 bytes (SRH fixed header)+16×N bytes (segment list). Therefore, as the number of SIDs specified by SRv6 increases, the additional overhead incurred by the encapsulation of the SRv6 extension header increases. To solve the problem, a function of shortening the SRv6 extension header needs to be implemented.

[0053] Although the standard SRv6 has many advantages mentioned above, its disadvantages are equally obvious. The actual deployment of the SRv6 technology in the network currently faces two challenges: first, the overhead of the SRv6 data packet is high, and the bandwidth utilization of a network link is low; the bandwidth utilization is only about 60% in the case of 256-byte packet length and 8-layer SID; second, SRv6 data packet processing has high requirements on a chip, the existing network device has difficulty in supporting deep copying and operation of the SRH header, and the processing of the 128-bit SRH header will reduce the processing efficiency of the existing chip.

[0054] For the problems of the standard SRv6 technology, the inventor found that the SRH of the standard SRv6 may be compressed and optimized, a shorter and simpler X-SID, for example, the typical 32-bit SID, may be used to optimize the SID of the SRv6, and a 32-bit X-SID may be used to represent the standard 128-bit SID and replace its encapsulation in the SRH header. In this way, the overhead of the SRH encapsulation of the standard SRv6 is optimized, and the performance of the SRv6 is greatly optimized while the SRv6 function is supported, and the large-scale deployment of the SRv6 technology is supported.

[0055] The technical solutions in the embodiments of the present application will be described clearly and completely below in combination with the drawings in the embodiments of the present application. It is apparent that the described embodiments are not all embodiments but part of embodiments of the application. All other embodiments obtained by those of ordinary skill in the art based on the embodiments in the present disclosure without creative work shall fall within the scope of protection of the embodiments of the present disclosure.

[0056] In addition, term "include" and any variations thereof in the specification and the claims of the present disclosure are intended to cover non-exclusive inclusions. For example, it is not limited for processes, methods, systems, products or devices containing a series of steps or units to clearly list those steps or units, and other steps or units which are not clearly listed or are inherent to these processes, methods, products or devices may be included instead. In addition, "and/or" used in the specification and the claims indicates at least one of the connected objects, for example, A and/or B indicates three cases, that is, individual A is included, individual B is included, and both A and B exist.

[0057] In the embodiments of the disclosure, the words like "exemplary" or "for example" are used to serve as example, example illustration or explanation. Any embodiments or designs described as "exemplary" or "for example" in the embodiments of the present disclosure shall not be construed as being preferred or superior to other embodiments or designs. More exactly, the purpose of using the word "exemplary" or "for example" is to present related concepts in a specific way.

[0058] The technologies described herein are not limited to 5G systems and subsequent evolved communication systems, and a Long Term Evolution (LTE)/LTE-

Advanced (LTE-A) system, and may also be applied to various wireless communication systems, for example, Code Division Multiple Access (CDMA), Time Division Multiple Access (TDMA), Frequency Division Multiple Access (FDMA), Orthogonal Frequency Division Multiple Access (OFDMA), Single-Carrier Frequency-Division Multiple Access (SC-FDMA), and other systems.

[0059] Terms "system" and "network" are usually used interchangeably. The CDMA system may implement radio technologies such as CDMA2000 and Universal Terrestrial Radio Access (UTRA). UTRA includes Wideband CDMA (WCDMA) and other CDMA variations. The TDMA system may implement radio technologies such as Global System for Mobile Communication (GSM). The OFDMA system may implement radio technologies such as Ultra Mobile Broadband (UMB), Evolution-UTRA (E-UTRA), Institute of Electrical and Electronics Engineers (IEEE) 802.11 (Wireless Fidelity (Wi-Fi)), IEEE 802.16 (World Interoperability for Microwave Access (WiMAX)), IEEE 802.20, and Flash-OFDM. UTRA and E-UTRA are part of Universal Mobile Telecommunications System (UMTS). LTE and more advanced LTE (such as LTE-A) are new UMTS releases using E-UTRA. UTRA, E-UTRA, UMTS, LTE, LTE-A and GSM are described in the documents from the organization named after "3rd Generation Partnership Project (3GPP)". CDMA2000 and UMB are described in the documents from the organization named after "3GPP2". The technologies described herein may be applied not only to the above-mentioned systems and radio technologies but also to other systems and radio technologies.

[0060] Referring to FIG. 3, an embodiment of the present disclosure provides a method for data packet processing. The method may be performed by a node, for example, an SRv6 node. The method may include the following steps.

[0061] At S301, first information and second information of a first SID are obtained, the first information being used for indicating the position of a container where the first SID is located in an SID list of a data packet, and the second information being used for indicating the position of the first SID in the container.

[0062] In some optional embodiments, the first information may also be SL information, which identifies the number of left segments; and the second information may be index information, which may also be referred to as SID Index information and is

not limited in the embodiments of the present disclosure.

[0063] The container (or SID container) may include one or more first SIDs. For example, if the container is 128 bits, the container may contain a standard SID (128-bit SID) or multiple compressed SIDs, that is, the first SID may be a standard SID or a compressed SID. The compressed SID may be a 32-bit compressed SID or a 16-bit compressed SID, which is not limited in the present embodiment.

[0064] Taking the 32-bit compressed SID as an example, a 128-bit container may include: three 32-bit compressed SIDs and one padding for completing 128 bits, or may include two 32-bit compressed SIDs and two paddings for completing 128 bits, or may include one 32-bit compressed SID and three paddings for completing 128 bits, or may include four 32-bit compressed SIDs. It can be understood that the first SID does not include the padding for completing 128 bits.

[0065] Taking that a 128-bit container includes three 32-bit compressed SIDs and one padding for completing 128 bits as an example, it can be padding, 32-bit compressed SID, 32-bit compressed SID, 32-bit compressed SID, and 32-bit compressed SID from left to right, and their respective second information may be 0, 1, 2, and 3.

[0066] Taking that a 128-bit container includes a standard 128-bit SID as an example, there is only one standard SID in the container, and the second information of the standard SID may be null or a default value.

[0067] Taking that the SID list includes five containers as an example, the position of the container in the SID list may be expressed as: SL=0, SL=1, SL=2, SL=3, and SL=4 from bottom to top. It can be understood that the form of expression of "the position of the container in the SID list" is not specifically limited in the embodiments of the present disclosure.

[0068] At S302, the position of the first SID in the SID list is obtained according to the first information and the second information.

[0069] Exemplarily, if the first SID is a compressed SID, a corresponding compressed SID may be located in the SID list according to the SL and index of the compressed SID. If the first SID is a 128-bit SID and the second information may be null, then a corresponding 128-bit SID may be located in the SID list according to the SL of the first

SID.

[0070] At S303, the first SID is copied (or updated) to a DA of the data packet, and the data packet is sent.

[0071] First manner: the first SID is copied to the DA of the data packet, the first SID is combined with a common prefix or an address block in the DA to get a new DA, and the data packet is forwarded to a next endpoint according to the new DA.

[0072] For example, the first SID may be a compressed SID, a corresponding compressed SID may be located in the SID list based on the SL and the index of the compressed SID, the compressed SID is copied to the DA of the data packet and combined with the common prefix or address block to get a new DA, and the data packet is forwarded to the next endpoint according to the new DA.

[0073] Second manner: the first SID and the second information are copied to the DA of the data packet, the first SID and the second information are combined with the common prefix or the address block in the DA to get a new DA, and the data packet is forwarded to the next endpoint according to the new DA.

[0074] For example, the first SID may be a compressed SID, a corresponding compressed SID may be located in the SID list based on the SL and the index of the compressed SID, the compressed SID and the index are copied to the DA of the data packet and combined with the common prefix or the address block to get a new DA, and the data packet is forwarded to the next endpoint according to the new DA.

[0075] Third manner: the first SID and the second information are copied to the DA of the data packet, and the data packet is forwarded to the next endpoint. For example, the first SID is a 128-bit SID, a corresponding 128-bit SID may be located in the SID list based on the SL and the index of the 128-bit SID, the 128-bit SID is copied to the DA of the data packet, and the data packet is forwarded to the next endpoint.

[0076] In some optional embodiments of the present disclosure, S301 may include the following steps.

[0077] At S3011, third information and fourth information of a second SID are obtained, the third information being used for indicating the position of a container where

the second SID is located in the SID list of the data packet, and the fourth information being used for indicating the position of the second SID in the container.

[0078] In some optional embodiments, the third information may also be SL information, which identifies the number of left segments; and the fourth information may also be index information, which is not limited in the embodiments of the present disclosure.

[0079] At S3012, the first information and the second information of the first SID are obtained according to the third information and the fourth information of the second SID.

[0080] In some optional embodiments, the second SID is an SID immediately before the first SID in the SID list, or the second SID is ordered before the first SID in the SID list, and there may be one or more paddings (represented by "0") between the second SID and the first SID. For example, the order in the SID list is the second SID, padding, and the first SID.

[0081] The above "the second SID is an SID immediately before the first SID" may be understood as follows: the first SID is the next SID of the second SID, that is, the node that matches the first SID is the next node of the node that matches the second SID, for example, the node that matches the first SID is N3, and the node that matches the second SID is N2.

[0082] In some optional embodiments, before S3011, it can be determined whether the next SID of the second SID is padding; if not, S3011 is executed; if yes, the first SID is a 128-bit SID by default; in this case, the first information of the first SID may be calculated based on the third information of the second SID, and the second information of the first SID is set to null.

[0083] Exemplarily, it is possible to identify whether the next SID of the second SID is padding based on the flavor of the second SID. That is, if END.X SID (second SID) locally issued by the node is hit in a local SID table, and COC flavor is not configured (the COC flavor is used for indicating the SID type of the next SID), it is possible to identify whether the next SID of the second SID is padding.

[0084] In some optional embodiments of the present disclosure, S3011 may include the following steps.

[0085] At S30111, the third information of the second SID is obtained from the SRH of the data packet.

[0086] At S30112, the index of the second SID is obtained from the DA of the data packet. The index of the second SID includes the fourth information. That is, the index of the second SID may include one or more indicator bits. The format of the index of the second SID is not specifically limited in the embodiments of the present disclosure.

[0087] In some optional embodiments of the present disclosure, S3012 may include the following steps.

[0088] At S30121, it is determined whether the first SID and the second SID are in the same container according to the fourth information or flavor of the second SID.

[0089] In an example, if the fourth information of the second SID is greater than zero, it means that the first SID and the second SID are in the same container; if the fourth information of the second SID is equal to zero, it means that the first SID and the second SID are not in the same container.

[0090] In another example, if the flavor of the second SID does not indicate the SID type of the next SID, it means that the first SID and the second SID are not in the same container. That is, if END.X SID (second SID) locally issued by the node is hit in the local SID table, and COC flavor is not configured (the COC flavor is used for indicating the SID type of the next SID), the next SID (the first SID) is indicated as a 128-bit SID by default, that is, a padding may be included between the second SID and the first SID.

[0091] At S30122, if the first SID and the second SID are in the same container, the first information of the first SID is the same as the third information of the second SID (that is, the SL is the same), and the second information of the first SID is obtained according to the fourth information of the second SID; for example, the fourth information (index) of the second SID is subtracted by one to obtain the second information (index) of the first SID; or,

[0092] if the first SID and the second SID are not in the same container, the first information of the first SID is obtained according to the third information of the second SID; for example, the third information of the second SID is subtracted by one (SL-1) to obtain the first information (SL) of the first SID; and the second information of the first

SID is determined according to the SID type of the first SID.

[0093] Exemplarily, if the SID type of the first SID is 32-bit, the second information of the first SID is "3".

[0094] In some optional embodiments of the present disclosure, the SID type of the first SID may be indicated by the flavor (COC flavor) of the second SID. It can be understood that the flavor of the second SID may be found in the local SID table of the current node, as shown in the table.

Table 1

SID flavor type	Function description
None	Not adding flavor; the default flavor of the SID indicates that the first SID is a 128-bit SRv6 SID
COC32	Identifying that the next SID (the first SID) is a 32-bit compressed SID
COC16	Identifying that the next SID (the first SID) is a 16-bit compressed SID

[0095] In some optional embodiments of the present disclosure, the method may further include the following operations. The first information of the first SID is updated to the SRH of the data packet. The second information of the first SID is updated to the DA of the data packet as the index of the first SID.

[0096] In some optional embodiments of the present disclosure, S301 may include the following steps.

[0097] At S3013, the third information, the fourth information and fifth information of the second SID are obtained, the third information being used for indicating the position of a container where the second SID is located in the SID list of the data packet, the fourth information being used for indicating the position of the second SID in the container, and the fifth information being used for indicating the SID type of the second SID and/or indicating whether the second SID is compressed.

[0098] Exemplarily, the SID types may include 8-bit, 16-bit, 32-bit, 64-bit, or 128-bit.

[0099] In some optional embodiments, the fifth information may include one or more indicator bits.

[00100] At S3014, the first information and the second information of the first SID are obtained according to the third information, the fourth information and the fifth information of the second SID.

[00101] The second SID is an SID immediately before the first SID in the SID list, or the second SID is ordered before the first SID in the SID list, and there may be one or more paddings (represented by "0") included between the second SID and the first SID.

[00102] In some optional embodiments of the present disclosure, S3013 may include the following steps.

[00103] At S30131, the third information of the second SID is obtained from the SRH of the data packet.

[00104] At S30132, the index of the second SID is obtained from the DA of the data packet. The index of the second SID includes the fourth information and the fifth information.

[00105] In some optional embodiments, the index of the second SID may include one or more indicator bits.

[00106] In some optional embodiments of the present disclosure, S3014 may include the following steps.

[00107] At S30141, it is determined whether the first SID and the second SID are in the same container according to the SID type of the second SID and the SID type of the first SID.

[00108] Exemplarily, if the SID type of the second SID is 32-bit and the SID type of the first SID is 32-bit, it is determined that the first SID and the second SID are in the same container; if the SID type of the second SID is 32-bit and the SID type of the first SID is 128-bit, it is determined that the first SID and the second SID are not in the same container.

[00109] At S30142, if the first SID and the second SID are in the same container, the first information of the first SID is the same as the third information of the second SID

(that is, the SL is the same), and the second information of the first SID is obtained according to the fourth information of the second SID; for example, the fourth information of the second SID is subtracted by one (index-1) to obtain the second information of the first SID; or,

[00110] if the first SID and the second SID are not in the same container, the first information of the first SID is obtained according to the third information of the second SID; for example, the third information of the second SID is subtracted by one to obtain the first information of the first SID; and the second information of the first SID is determined according to the SID type of the first SID.

[00111] Exemplarily, if the SID type of the first SID is 32-bit, the second information of the first SID is "3", that is, the index of the first SID is 3.

[00112] In the embodiments of the present disclosure, the container is used for containing a standard SID or multiple compressed SIDs. A compressed SID includes a node ID and a value of a function of the node.

[00113] In the embodiments of the present disclosure, the SID list includes at least a compressed SIDs. When the DA in the data packet is updated, the compressed SID is spliced with the common prefix in the current DA to form a new SID, so as to continue to perform the table lookup and forwarding operations. In this way, the data packet is compressed, and the header overhead of the data packet is effectively reduced.

[00114] In addition, in the embodiments of the present disclosure, the standard SIDs and the compressed SIDs may be mixed encoded in the SID list, so as to support mixed configuration of compressed nodes and common nodes in an SRv6 path to achieve existed network evolution and smooth upgrade. At the same time, in a multi-domain scenario, due to the limited address planning, there is no guarantee that the SIDs of multiple domains have a common prefix, and the SRv6 path across multiple domains can be easily created by mixed encoding.

[00115] An introduction is given below taking a scenario of IPv6 as an example. Of course, it can be understood that the embodiments of the present disclosure may also be applied to the IP technology of other versions (for example, IPv5, IPv7, IPv9, and IPv10), in addition to the IPv6 technology. The method applied to the IP technology of

other versions is similar to that of the IPv6 technology and will not be repeated here.

[00116] In an SRv6 deployment, an address block needs to be distributed through address planning for the allocation of SID. The address block may be called SID space. Therefore, in an SRv6 domain, the SID allocated from the SID space may have a common prefix. In general, all SIDs in the SRH have the same prefix, while this part is redundant. Removing this part of redundant information may effectively reduce the overhead of the SRv6 header.

[00117] Therefore, in the embodiments of the present disclosure, the common prefix of the SID in the SID list is removed, and only the variable compressed SID (or short SID, or X-SID) is carried. When replacing the DA, a new SID is formed by stitching the compressed SID with the common prefix in the current IPv6 DA, to continue the operations of table lookup and forwarding, so as to achieve the compression of SRv6.

[00118] In addition, in the embodiments of the present disclosure, the traditional SRv6 SID and the compressed SID are mixed encoded in the SRv6 (for example, the SRv6 is called X-SRv6), so as to support mixed configuration of compressed nodes and common SRv6 nodes in the SRv6 path, to achieve existed network evolution and smooth upgrade. At the same time, in a multi-domain scenario, due to the limited address planning, there is no guarantee that the SIDs of multiple domains have a common prefix, and the SRv6 path across multiple domains can be easily created by mixed encoding.

[00119] (I) Introduction to the standard 128-bit SRv6 SID format

[00120] SRv6 Segment Identifier is used for identifying an ID of an SRv6 Segment. The standard SRv6 SID is a 128-bit IPv6 address, which usually consists of three parts if supporting SRv6 network programmability is considered, as shown in FIG. 4.

(1) Locator is an ID assigned to a network node in the network topology for routing and forwarding data packets to the node. The locator identifies location information. The route corresponding to the locator will be issued by the node to the network through an Interior Gateway Protocol (IGP) to help other devices forward the data packets to the node that issues the locator. In the SRv6 SID, the length of Locator is variable to adapt to networks of different scales.

(2) Function is used for expressing a forwarding action to be performed by the

instruction, which is equivalent to an operation code of a computer instruction. In SRv6 network programming, different forwarding behaviors are expressed by different functions, such as forwarding a data packet to a specified link, or performing table lookup and forwarding in a specified table.

(3) Arguments (referred to as “Args” for short in the below) is an optional field, and include parameters corresponding to execution of an instruction, and these parameters may include flow, service, or any other relevant information.

[00121] Further, referring to FIG. 5, the locator may be subdivided into B: N. "B" identifies an SRv6 SID block, which is generally assigned to a certain subnet by the carrier and is usually represented by a prefix. "N" represents an ID for distinguishing a node in the subnet. The SIDs in an SRv6 network all have a common prefix B, so the part B of the SIDs in an SID list is repetitive and redundant.

[00122] (II) Introduction to the compressed SRv6 SID format

[00123] It can be seen from the standard SRv6 SID format that, in a typical application that supports SRv6 network programming, the SID format has rules, and the standard SRv6 SID format can be optimized by making full use of the rules of the SID format.

[00124] The embodiments of the present disclosure define a typical compressed 32-bit SRv6 SID format. The compressed SID may consist of a node ID and a function ID in the standard 128-bit SID.

[00125] The standard 128-bit SID format may be referred to as a full SID, which is defined as the standard SRv6 SID. The 32-bit SID is a compressed SID (or X-SID), which is the part of variation of the full SID, as shown in FIG. 6.

[00126] Similarly, a compressed 16-bit SID may also be defined. A description is given below by taking the compressed 32-bit SID as an example. It can be understood that the implementation mode of a compressed 16-bit SID is similar to that of the compressed 32-bit SID and will not be repeated here.

[00127] A conversion relationship between the full SID and the compressed SID may be as follows according to the rules of the SRv6 locator:

[00128] full SID=B (block, namely the common prefix)+N (node ID)+F (function)+A (args) (optional)+padding (optional);

[00129] compressed SID=N (node ID)+F (function).

[00130] So, the compressed SID plus the common prefix/address block can form a full SID.

[00131] To support compression, it is necessary to plan the SRv6 SID format that supports compression. For example, the common prefix is 96 bits, and the compressed SID is 32 bits. For example, the common prefix is 64 bits, the subsequent compressed SID is 32 bits, and the left low 32 bits are 0. In order to identify a specific SRv6 SID as a compressible SID, it is necessary to extend the corresponding control plane protocol to issue the compressible SID, specifically issuing the compression capability and structure information of the SID.

[00132] (III) Introduction to the format of the container that contains the compressed SID and the standard SID

[00133] When the compressed SID and the standard SID are programmed in the SRH SID list, to accurately locate the compressed SID, the compressed SID must be programmed to in 128-bit alignment, that is, four 32-bit compressed SIDs or multiple shortened compressed SIDs of other lengths must be placed in a row of 128 bits. If the row is not fully filled, it is necessary to fill with padding to align to 128 bits.

[00134] For ease of understanding, the concept of SID container is defined in the embodiments of the present disclosure, which is a 128-bit SID container. An SID container may include:

(1) a standard SRv6 SID;

(2) multiple compressed SIDs (X-SIDs), for example, four 32-bit compressed SIDs or eight 16-bit compressed SIDs.

[00135] When a variety of SIDs are mixed encoded, the alignment of 128 bits is required. When the compressed SIDs cannot fully fill 128 bits, it is necessary to fill with padding. Taking the 32-bit compressed SID as an example, the possible format of SID

container is shown in FIG. 7.

[00136] Since multiple types of SID containers may be encoded in the SRv6 SRH, the SRH may be called compressed SRH. It is a new implementation of SRv6, which may be called compressed SRv6. The details of the scheme of the compressed SRv6 on the data plane are described below.

[00137] (VI) Introduction to the compressed SRH of the compressed SRv6 data plane scheme

[00138] The compressed SRH is consistent with the SRH[RFC8754] format without changes to its format and field semantics, but the 128-bit SID and the 32-bit compressed SID can be mixed encoded in the compressed SRH. The format is shown in FIG. 8, including: a Next Header field, an Hdr Ext len field, a Routing Type field, an SL field, a Last Entry field, a Flags field, a Tag field, a X-SID Container field (containing a 128-bit value), and an Optional TLV object field (variable).

[00139] When the 128-bit SID and the 32-bit X-SID are mixed encoded in the X-SRH, an encoding example is shown in FIG. 9.

[00140] A compressed SRv6 path may consist of an SRv6 sub-path and a compressed SRv6 sub-path. The SRv6 sub-path is encoded by the SRv6 SID. The compressed SRv6sub-path consists of a starting compressible 128-bit SRv6 SID and multiple following compressed SIDs. The end of the compressed SID list must be bounded by 128 bits. The next may be a standard 128-bit SRv6 SID or a next compressed SRv6 sub-path with a different prefix.

[00141] In order to identify the start and end of the compressed SRv6 path in the SID list, that is, the boundary between the 128-bit SID and 32-bit SID, it needs to add several types of flavors and issue the corresponding SIDs of the flavors. The values of behaviors related to COC32 flavor can be seen in Table 2. The values of behaviors related to COC16 flavor are not defined temporarily.

Table 2

SID flavor type	Function description
None	Not adding flavor; the default flavor of the SID indicates that the next is

	a 128-bit SRv6 SID
COC32	Identifying that the next SID is a 32-bit compressed SID
COC16	Identifying that the next SID is a 16-bit compressed SID

[00142] Therefore, when the SID in the DA carries the COC32 flavor, it means that the next 32-bit compressed SID needs to be updated to the DA. When the SID in the DA does not carry the COC flavor SID, it means that "SL--" (SL minus 1) and the next 128-bit SID needs to be updated to the DA (when SL>0). The SL may be located to the position of the 128-bit container.

[00143] Therefore, the encoding rules of the compressed SRv6 in the SID list include the followings.

(1) The start of a compressed SRv6 path is indicated by a 128-bit COC32 flavor SID, which contains a COC32 flavor SID. This SID carries the complete SID information, including the common prefix and other information, and may be used for restoring the complete next SID with the subsequent compressed SID.

(2) All the compressed SIDs in the middle of the compressed path are the compressed SIDs carrying COC32, indicating that the next one is a 32-bit compressed SID.

(3) The last compressed SID of the compressed path is a 32-bit compressed SID without a COC flavor for indicating the end of the compressed path. The SRv6 SID combined in the DA is processed according to the processing rules of the 128-bit SID.

[00144] It is to be noted that the last SID has no COC flavor SID because after it is updated to the DA, the SID in the DA does not have the COC flavor, then it will be processed by the node as a 128-bit SID, which is consistent with the processing of the current SRv6. Therefore, it can be used for identifying the end of the compressed path, and switching from the 32-bit compressed SID to the 128-bit SID for processing.

[00145] It can be understood that this design can well support the following two scenarios:

(1) in a mixed encoding scenario, the compressed path ends, and it is switched from the 32-bit SID to the 128-bit SID;

(2) in a pure compression encoding scenario, the last SID has no COC flavor, so the

processing of the last hop is consistent with the current SRv6, which is more compatible.

[00146] Moreover, in the compressed path, in order to locate the next compressed SID, the index of the compressed ID (X-SID Index, referred to as XI for short in the below) may be added to locate the position of the SID in the SID container. Further, the index may also indicate whether the SID is compressed or the SID type of the SID.

[00147] The basic idea of location is as follows:

[00148] First, the next compressed SID is located by $SL+XI$, the SL is located to the position of a 128-bit container, and the XI is located to the position of the X-SID in the 128-bit container.

[00149] Second, in the mixed encoding scenario, how to locate the boundary of the SID with a variable length:

[00150] (1) it is determined whether the SID type has changed by comparing "the SID type of the next SID" of the SID with the SID type of the current SID; for example, if "the SID type of the next SID" of the SID is 32-bit and the type of the current SID is 128-bit, the boundary is reached;

[00151] (2) the XI in the DA is changed correspondingly to locate the next SID/X-SID;

[00152] (3) if the SID type of the next SID is 32-bit, the next 32-bit X-SID is updated to the position of the X-SID corresponding to the IPv6 DA; the X-SID may be located by $SRH[SL][XI]$.

[00153] For example, the index may be placed after the compressed SID in the DA. Exemplarily, the index is placed at 2 Least Significant Bits (LSBs) of 128 bits, as shown in FIG. 10. When the compressed SID is issued, it needs to reserve space for the index, which has a value of 0. During forwarding, the value of XI is the location flavor of the compressed SID, indicating its position in the X-SID container.

[00154] For example, it is indicated through the COC32 flavor SID to update the next 32-bit compressed SID to the DA, and the position of the compressed SID may be located by the SL and the XI.

[00155] Taking the 32-bit compressed SID as an example, processing pseudo-codes of the data packet received by the node (endpoint) are as follows:

if IPv6 DA hits a COC32 flavor SID //32bit X-SID processing

if DA.XI==0: //The first compressed SID in the next line;

SL--;

DA.XI=3

Else //The next compressed SID in the current line

DA.XI--

DA[CP..CP+31]=SRH[SL][DA.XI]

Forward the packet based on new DA

Else //Common SRv6 branch, not affected

SRv6 processing.

[00156] The processing flow is triggered by the COC flavor SID, and operation data is limited to the COC flavor SID. The existing main SRv6 process is not affected, and the processing of the existing SID and SRH is not affected. The format is compatible with the SRH, and the processing is compatible with the existing SRv6. The XI is after the compressed SID of the DA. Taking the pure compressed path+128-bit VPN SID as an example, an example of encoding and X-SID update is shown in FIG. 11.

[00157] (V) Data packet forwarding process in a mixed encoding scenario

[00158] When deploying an SRv6 compression scheme, it needs to perform address planning and then upgrade a control plane and a data plane to support the SRv6 compression.

[00159] First, the node sets the extended C-flag of SRv6 Capabilities TLV of IGP and other protocols to identify that the node supports the compression capability.

[00160] The node instantiates the SRv6 SID that may support compression, and issues

the SID to the network or to a controller through IGP, BGP, BGP-LS and other protocols. After instantiating the SID, the node generates a corresponding SID forwarding table entry.

[00161] Referring to FIG. 12, node k may issue two sets of SIDs; both of the two sets of SIDs carry C-Flag and support compression. One of the two sets of SIDs carries the COC32 flavor, and the other does not carry the flavor. For example, A:k:1:: is an END.X SID carrying the COC32 flavor and bound to an interface, and A:k:2:: is an END.X SID not carrying the COC32 flavor and bound to the same interface. According to the planning, A:1:1:: is a COC32 flavor End.X SID, and A:4:2:: is an SID without COC flavor.

[00162] When issuing the SID, it needs to set the C-flag in the TLV (e.g. SRv6 End SID sub-TLV, SRv6 End.X SID sub-TLV, and SRv6 LAN End.X SID sub-TLV) corresponding to 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:: and A:10:10:: (the formats support the compressed VPN SID, which facilitates the demonstration of a pure compression case) to identify that the format of the SID supports compression.

[00163] The issued SID includes a compressible SID carrying the COC32 flavor to identify that the SID is followed by one or more 32-bit X-SIDs, for example, multiple SIDs A:k:1:: in the following example.

[00164] A:k:1:: also needs to carry SID Structure Sub-sub-TLV when issued. Assuming the length of Common Prefix (A) is 64, the length of X-SID is 32 (the length of Node ID is 16, and the length of Function ID is 16), and the length of Argument is 32, then in the SID Structure Sub-sub-TLV of the issued SID, the length of Locator Block is 64, the length of Node ID is 16, the length of Function is 16, and the length of Argument is 32.

[00165] In the figure, the prefix of A1:: is a prefix that does not support the compressed SID, so A1::5:1 is an incompressible SID. It will also carry the SID Structure Sub-sub-TLV, in which the length of Locator Block is 64, the length of Node ID is 16, and the length of Function is 48.

[00166] A device sends, through BGP-LS, the capability of supporting compression of a node, the SID supporting compression, and the newly added SID carrying the COC32

flavor to a controller for path calculation.

[00167] After completing path calculation, the controller may issue the SID list containing the compressed SID through a BGP SR policy. In this case, a sub-TLV describing the SID format needs to be inserted between the segment sub-TLVs of the SID list. For example, for ten SIDs in the following example, SID encoding sub-TLV needs to be inserted before A:2:1:: and A:7:1::; the start of the SID encoding sub-TLV is the length (63 in the figure) of Common Prefix, and the end of the SID encoding sub-TLV is the length of Common Prefix+32 (95 in the figure). SID encoding sub-TLV also needs to be inserted before A1::5:1 and A:10:10::, the start of the SID encoding sub-TLV is equal to 0, and the end of the SID encoding sub-TLV is equal to 127.

Table 3

SID	Block	Node	Function	Arg	Padding	Notes
A:k:1::	64	16	16	32	0	Compressible, COC32 flavor
A:k:2::	64	16	16	32	0	Compressible, None COC flavor
A1:5::1	64	16	48	0	0	Incompressible, A1 prefix
A:10:10::	64	16	16	32	0	Compressible VPN SID

[00168] In the example, the SID list contains a total of ten 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 which is an SID that does not support compression

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

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

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

A:9:2:: END.X (COC32)

A:10:10:: END.DT4 VPN SID (not participating in compression).

[00169] After reduced-mode and compression encoding, the SID list is shown in FIG. 12. End. X SID A:1:1:: is not encoded in the SID list.

[00170] (1) When node 1 receives a data packet, the DA "A:1:1::" hits the COC flavor END.X SID locally issued by the node in the local SID list; in this case, in the SRH, SL=5 and DA.XI=0, so SL--, and DA.XI=3, which points to 2:1, and then the X-SID 2:1 is updated to the DA and forwarded to the next node 2. At this point, the DA is A:2:1::3.

[00171] (2) When node 2 receives a data packet, the DA "A:2:1:3::" hits and matches the COC flavor END.X SID locally issued by the node in the local SID list; in this case, SL=4, and DA.XI is 3 which is greater than 0, so the node performs DA.XI--, and updates (or copies) the next X-SID 3:1 to which SRH[SL][DA.XI] points to the DA and forwards it to the next node. At this point, the DA is A:3:1::2.

[00172] (3) When node 3 receives a data packet, the DA "A:3:1:2::" hits the COC flavor END.X SID locally issued by the node in the local SID list; in this case, in the SRH, SL=4 and DA.XI=2 which is greater than 0, so DA.XI--, which points to 4:2, and then the X-SID 4:2 is updated to the DA and forwarded to the next node. At this point, the DA is A:4:2::1.

[00173] (4) When node 4 receives a data packet, the DA "A:4:2:1::" hits the END.X SID locally issued by the node in the local SID list; because there is no COC flavor, the next SID is the 128-bit SID by default; therefore, SL=SL-1=3, and "A1::5:1" is copied to the DA for forwarding.

[00174] (5) Node 5 is a common SRv6 node, so common SRv6 forwarding is performed to copy "A:6:1::" to the DA and forward it to the next node.

[00175] (6) When node 6 receives a data packet, the DA "A:6:1::" hits the COC flavor END.X SID locally issued by the node in the local SID list; in this case, in the SRH, SL=2 and DA.XI=0, so SL--, and DA.XI=3, which points to 7:1, and then the X-SID 7:1 is updated to the DA and forwarded to the next node. At this point, the DA is A:7:1::3.

[00176] It can be understood that "7:1" is equivalent to the first SID, and "A:6:1::" is equivalent to the second SID.

[00177] (7) Similarly, when receiving a data packet, node 7 and node 8 process the COC flavor SID, update the DA, and forward the data packet.

[00178] (8) After receiving a data packet, node 9 performs the same procedure as node 4; because the DA hits the SID without the COC flavor in the local SID table, SL-- is equal to 0, and VPN SID is copied to the DA and forwarded to node 10.

[00179] (9) Node 10 performs processing according to the normal VPN SID.

[00180] (VI) Data packet forwarding process in a pure compression scenario

[00181] Referring to FIG. 13, the control plane process principle, topology information and SID information are the same as that of the mixed encoding case. When an SR policy is issued, it is necessary to insert a corresponding SID encoding sub-TLV description format in the SID list of the BGP SR policy. Taking the following as an example:

[00182] the SID encoding sub-TLV needs to be inserted before A:2:1::; the start of the SID encoding sub-TLV is the length (63 in the figure) of Common Prefix, and the end of the SID encoding sub-TLV is the length of Common Prefix+32 (95 in the figure).

[00183] In the example, the SID list contains a total of ten SIDs:

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)

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 (participating in compression).

[00184] After the reduced mode and compression encoding, the SID list is shown in FIG. 13. End. X SID A:1:1:: is not encoded in the SID list.

[00185] (1) After encapsulating a data packet, a node 0 sends the data packet to the next hop node 1; in this case, SL=3 and DA.XI=0 in the SRH. The flavor of SID is COC, which identifies that the encapsulation format of the next SID is the 32-bit X-SID.

[00186] (2) When node 1 receives a data packet, the DA "A:1:1::" is the locally issued COC flavor END.X whose format supports compression, which indicates that the next SID is a 32-bit X-SID; in this case, SL=3, and DA.XI=0, so SL--, and DA.XI=3, which points to 2:1; 2:1 is copied to the DA to update the X-SID, and then table lookup and forwarding are performed based on the new DA.

[00187] It can be understood that "2:1" is equivalent to the first SID, and "1:1" is equivalent to the second SID.

[00188] (3) Similarly, for subsequent nodes 2, 3, 4, 5, 6, 7, 8, and 9, the nodes update the corresponding X-SID to the DA based on the values of SL and DA.XI, and perform table lookup and forwarding.

[00189] (4) When node 10 receives a data packet, the DA is "A:10:10::", which is an SID without COC flavor (the VPN SID itself has no such flavor) and the node performs processing of the VPN SID.

[00190] Referring to FIG. 14, embodiments of the present disclosure further provide a node. The node 1400 may include: an obtaining module 1401, a processing module 1402, and a sending module 1403.

[00191] The obtaining module 1401 is configured to obtain first information and second information of a first SID, the first information being used for indicating a position of a container where the first SID is located in a SID list of a data packet, and the second information being used for indicating a position of the first SID in the container.

- [00192] The processing module 1402 is configured to obtain a position of the first SID in the SID list according to the first information and the second information.
- [00193] The sending module 1403 is configured to copy the first SID to a DA of the data packet, and send the data packet.
- [00194] In some implementation modes, the obtaining module 1401 is configured to: obtain third information and fourth information of a second SID, the third information being used for indicating a position of a container where the second SID is located in the SID list of the data packet, and the fourth information being used for indicating a position of the second SID in the container; and obtain the first information and the second information of the first SID according to the third information and the fourth information of the second SID. The second SID is an SID immediately before the first SID in the SID list.
- [00195] In some implementation modes, the obtaining module 1401 is configured to: obtain the third information of the second SID from an SRH of the data packet; and obtain an index of the second SID from the DA of the data packet. The index of the second SID includes the fourth information.
- [00196] In some implementation modes, the obtaining module 1401 is configured to: determine whether the first SID and the second SID are in the same container according to the fourth information or flavor of the second SID; if the first SID and the second SID are in the same container, the first information of the first SID is the same as the third information of the second SID, and obtain the second information of the first SID according to the fourth information of the second SID; or, if the first SID and the second SID are not in the same container, obtain the first information of the first SID according to the third information of the second SID, and determine the second information of the first SID according to the SID type of the first SID.
- [00197] In some implementation modes, the SID type of the first SID is indicated by the flavor of the second SID.
- [00198] In some implementation modes, the node 1400 may further include: an updating module, which is configured to: update the first information of the first SID to the SRH of the data packet; and update the second information of the first SID to the DA

of the data packet as the index of the first SID.

- [00199] In some implementation modes, the obtaining module 1401 is configured to: obtain the third information, the fourth information and fifth information of the second SID, the third information being used for indicating the position of the container where the second SID is located in the SID list of the data packet, the fourth information being used for indicating the position of the second SID in the container, and the fifth information being used for indicating the SID type of the second SID and/or indicating whether the second SID is compressed; and obtain the first information and the second information of the first SID according to the third information, the fourth information and the fifth information of the second SID. The second SID is an SID immediately before the first SID in the SID list.
- [00200] In some implementation modes, the obtaining module 1401 is configured to: obtain the third information of the second SID from the SRH of the data packet; and obtain the index of the second SID from the DA of the data packet. The index of the second SID includes the fourth information and the fifth information.
- [00201] In some implementation modes, the obtaining module 1401 is configured to: determine whether the first SID and the second SID are in the same container according to the SID type of the second SID and the SID type of the first SID; if the first SID and the second SID are in the same container, the first information of the first SID is the same as the third information of the second SID, and obtain the second information of the first SID according to the fourth information of the second SID; or, if the first SID and the second SID are not in the same container, obtain the first information of the first SID according to the third information of the second SID, and determine the second information of the first SID according to the SID type of the first SID.
- [00202] In some implementation modes, the container is used for containing multiple compressible SIDs. The compressible SID includes the node ID and the value of function of the node.
- [00203] In some implementation modes, the sending module 1403 is configured to: copy the first SID to the DA of the data packet, combine the first SID with the common prefix or address block in the DA to get a new DA, and forward the data packet to a next endpoint according to the new DA; or, copy the first SID and the second information to

the DA of the data packet, combine the first SID and the second information with the common prefix or address block in the DA to get a new DA, and forward the data packet to the next endpoint according to the new DA; or, copy the first SID and the second information to the DA of the data packet, and forward the data packet to the next endpoint.

[00204] It is to be noted that when the node provided by the embodiments performs data packet processing, the apparatus is illustrated only by the division of each program module mentioned above. In practical applications, the above processing may be assigned to different program modules to complete according to needs, that is, the internal structure of the node is divided into different program modules to complete all or part of the processing described above. In addition, the node and the method for data packet processing provided in the above embodiments belong to the same concept, and details about a specific implementation process thereof refer to the method embodiments, and will not be elaborated herein.

[00205] Referring to FIG. 15, an embodiment of the present disclosure further provides a node. The node 1500 may include: a transceiver 1501, a processor 1502, and a memory 1503 configured to store a computer program capable of running in the processor 1502.

[00206] The transceiver 1501 is configured to send and receive data under control of the processor 1502.

[00207] When reading the computer program in the memory 1503, the processor 1502 executes the steps of the method for data packet processing described in the above embodiments of the present disclosure.

[00208] It can be understood that the memory 1503 may be a volatile memory or a nonvolatile memory, or may include both the volatile and nonvolatile memories. The nonvolatile memory may be a Read Only Memory (ROM), a Programmable Read-Only Memory (PROM), an Erasable Programmable Read-Only Memory (EPROM), an Electrically Erasable Programmable Read-Only Memory (EEPROM), a Ferromagnetic Random Access Memory (FRAM), a flash memory, a magnetic surface storage, a compact disc, or a Compact Disc Read-Only Memory (CD-ROM). The magnetic surface memory may be a disk memory or a tape memory. The volatile memory may be a RAM, and is used as an external high-speed cache. It is exemplarily but unlimitedly described

that RAMs in various forms may be adopted, such as a Static Random Access Memory (SRAM), a Synchronous Static Random Access Memory (SSRAM), a Dynamic Random Access Memory (DRAM), a Synchronous Dynamic Random Access Memory (SDRAM), a Double Data Rate Synchronous Dynamic Random Access Memory (DDRSDRAM), an Enhanced Synchronous Dynamic Random Access Memory (ESDRAM), a SyncLink Dynamic Random Access Memory (SLDRAM) and a Direct Rambus Random Access Memory (DRRAM). The memory 1503 described in the embodiment of the disclosure is intended to include, but not limited to, memories of these and any other proper types.

[00209] The method disclosed in the embodiments of the disclosure may be applied to the processor 1502 or implemented by the processor 1502. The processor 1502 may be an integrated circuit chip with a signal processing capability. In an implementation process, the steps of the method may be accomplished by an integrated logic circuit of hardware in the processor 1502 or an instruction in a software form. The processor 1502 may be a universal processor, a Digital Signal Processor (DSP) or another Programmable Logic Device (PLD), a discrete gate or transistor logic device, a discrete hardware component, etc. The processor 1502 may implement or execute each method, step and logical block diagram disclosed in the embodiments of the disclosure. The universal processor may be a microprocessor, any conventional processor or the like. The steps of the method disclosed in combination with the embodiments of the disclosure may be directly embodied to be executed and completed by a hardware decoding processor or executed and completed by a combination of hardware and software modules in the decoding processor. The software module may be located in a storage medium, and the storage medium is located in the memory 1503. The processor 1502 reads information in the memory 1503 and completes the steps of the method in combination with hardware.

[00210] In an exemplary embodiment, the node may be realized by one or more Application Specific Integrated Circuits (ASICs), DSPs, Programmable Logic Devices (PLDs), Complex Programmable Logic Devices (CPLDs), Field-Programmable Gate Array (FPGAs), universal processors, controllers, Micro Controller Units (MCUs), microprocessors, or other electronic components, and is configured to execute the above method.

[00211] In an exemplary embodiment, an embodiment of the present disclosure further provides a computer-readable storage medium, for example, a memory 1503 including a

computer program. The computer program may be executed by a processor 1502 of the node 1500 to complete the steps of the above method. The computer-readable storage medium may be, but is not limited to, an FRAM, an ROM, a PROM, an EPROM, an EEPROM, a flash memory, a magnetic surface storage, a compact disc, or a CD-ROM, and may also be a variety of devices including one or any combination of the above memories.

[00212] An embodiment of the present disclosure further provides a computer-readable storage medium, in which a computer program is stored. When executed by the processor, the computer program implements the steps of the method for data packet processing in the embodiments of the present disclosure.

[00213] The characteristics disclosed in some method or device embodiments provided in the present application may be freely combined without conflicts to obtain new method embodiments or device embodiments.

[00214] The steps of the method or algorithm described in the disclosed content of the present disclosure can be implemented by hardware or by a processor executing software instructions. The software instructions may consist of corresponding software modules. The software modules may be stored in an RAM, a flash memory, an ROM, an EPROM, an EEPROM, a register, a hard disk, a mobile hard disk, a CD-ROM, or any other storage medium known in the field. An exemplary storage medium is coupled to a processor, so that the processor can read information from and write information to the storage medium. Of course, the storage medium can also be a part of the processor. The processor and the storage medium can be carried in an Application Specific Integrated Circuit (ASIC). In addition, the ASIC can be carried in a core network interface device. Of course, the processor and the storage medium can also exist as discrete components in the core network interface device.

[00215] Those skilled in the art may realize that, in one or more abovementioned examples, the functions described in the present disclosure may be realized by hardware, software, firmware or any combination thereof. In case of implementation with the software, these functions are stored in a computer-readable medium or transmitted as one or more instructions or codes in the computer-readable medium. The computer-readable medium includes a computer storage medium and a communication medium, and the

communication medium includes any medium for conveniently transmitting a program from one place to another place. The storage medium may be any available medium accessible for a universal or dedicated computer.

[00216] The specific implementation methods described above further describe in detail the purposes, technical solutions and beneficial effects of the present disclosure. It should be understood that the above is only the specific implementations of the present disclosure and is not intended to limit the scope of protection of the disclosure. Any modification, equivalent replacement, improvement, etc. made on the basis of the technical solutions of the present disclosure shall fall within the scope of protection of the present disclosure.

[00217] Those skilled in the art should understand that the embodiments of the present disclosure may be provided as a method, a system or a program product. Therefore, the embodiment of the disclosure may use form of a pure hardware embodiment, a pure software embodiment, or an embodiment combining software and hardware. Moreover, the embodiment of the disclosure may use form of a program product implemented on one or more computer-available storage media (including, but not limited to, a disk memory, a Compact Disc Read-Only Memory (CD-ROM), and an optical memory) including computer-available program codes.

[00218] The embodiments of the present disclosure are described with reference to flowcharts and/or block diagrams of the method, the device (system) and the program product according to the embodiments of the present disclosure. It is to be understood that each flow and/or block in the flowcharts and/or the block diagrams and combinations of the flows and/or blocks in the flowcharts and/or the block diagrams may be implemented by program instructions. These program instructions can be provided to a processor of a general-purpose computer, a special-purpose computer, an embedded processor or other programmable data processing devices to generate a machine, so that instructions which are executed by the processor of the computer or other programmable data processing devices generate a device which is used for implementing the specified functions in one or more flows of the flowchart and/or one or more blocks of the block diagram.

[00219] These program instructions may further be stored in the computer-readable

memory which can guide the computer or other programmable data processing devices to work in a particular way, so that the instructions stored in the computer-readable memory generate a product including an instruction device. The instruction device implements the specified functions in one or more flows of the flowchart and/or one or more blocks of the block diagram.

[00220] These program instructions may also be loaded on the computer or other programmable data processing devices, so that a series of operation steps are performed on the computer or other programmable data processing devices to generate the processing implemented by the computer, and the instructions executed on the computer or other programmable data processing devices provide the steps for implementing the specified functions in one or more flows of the flowchart and/or one or more blocks of the block diagram.

[00221] It is apparent that those skilled in the art can make various modifications and variations to the embodiments of the present disclosure without departing from the spirit and scope of the present disclosure. Thus, if such modifications and variations of the embodiments of the present disclosure fall within the scope of the appended claims and their equivalents, the present disclosure is also intended to cover the modifications and variations.

CLAIMS

1. A data packet processing method, comprising:

obtaining first information and second information of a first Segment ID (SID), the first information being used for indicating a position of a container where the first SID is located in an SID list of a data packet, and the second information being used for indicating a position of the first SID in the container;

obtaining a position of the first SID in the SID list according to the first information and the second information;

copying the first SID to a Destination Address (DA) of the data packet, and sending the data packet.

2. The method of claim 1, wherein obtaining the first information and the second information of the first SID comprises:

obtaining third information and fourth information of a second SID, the third information being used for indicating a position of a container where the second SID is located in the SID list of the data packet, and the fourth information being used for indicating a position of the second SID in the container;

obtaining the first information and the second information of the first SID according to the third information and the fourth information of the second SID.

3. The method of claim 2, wherein obtaining the third information and the fourth information of the second SID comprises:

obtaining the third information of the second SID from a Segment Routing Header (SRH) of the data packet; and

obtaining an index of the second SID from the DA of the data packet, wherein the

index of the second SID comprises the fourth information.

4. The method of claim 2, wherein obtaining the first information and the second information of the first SID according to the third information and the fourth information of the second SID comprises:

determining whether the first SID and the second SID are in the same container according to the fourth information of the second SID or flavor of the second SID; and

in response to the first SID and the second SID being in the same container, the first information of the first SID being the same as the third information of the second SID, obtaining the second information of the first SID according to the fourth information of the second SID; or,

in response to the first SID and the second SID not being in the same container, obtaining the first information of the first SID according to the third information of the second SID, and determining the second information of the first SID according to an SID type of the first SID.

5. The method of claim 4, wherein the SID type of the first SID is indicated by the flavor of the second SID.

6. The method of claim 1, further comprising:

updating the first information of the first SID to an SRH of the data packet;

updating the second information of the first SID to the DA of the data packet as an index of the first SID.

7. The method of claim 1, wherein copying the first SID to the DA of the data packet, and sending the data packet comprises:

copying the first SID to the DA of the data packet, combining the first SID with a common prefix or an address block in the DA to get a new DA, and forwarding the data packet to a next endpoint according to the new DA;

or,

copying the first SID and the second information to the DA of the data packet, combining the first SID and the second information with the common prefix or the address block in the DA to get a new DA, and forwarding the data packet to a next endpoint according to the new DA;

or,

copying the first SID and the second information to the DA of the data packet, and forwarding the data packet to the next endpoint.

8. The method of claim 1, wherein obtaining the first information and the second information of the first SID comprises:

obtaining third information, fourth information and fifth information of the second SID, the third information being used for indicating a position of a container where the second SID is located in the SID list of the data packet, the fourth information being used for indicating a position of the second SID in the container, and the fifth information being used for indicating at least one of: an SID type of the second SID, or whether the second SID is compressed;

obtaining the first information and the second information of the first SID according to the third information, the fourth information and the fifth information of the second SID;

wherein the second SID is an SID immediately before the first SID in the SID list.

9. The method of claim 8, wherein obtaining the third information, the fourth information and the fifth information of the second SID comprises:

obtaining the third information of the second SID from the SRH of the data packet;

obtaining an index of the second SID from the DA of the data packet, wherein the index of the second SID comprises the fourth information and the fifth information.

10. The method of claim 8, wherein obtaining the first information and the second information of the first SID according to the third information, the fourth information and the fifth information of the second SID comprises:

determining whether the first SID and the second SID are in the same container according to the SID type of the second SID and an SID type of the first SID; and

in response to the first SID and the second SID being in the same container, the first information of the first SID being the same as the third information of the second SID, obtaining the second information of the first SID according to the fourth information of the second SID; or,

in response to the first SID and the second SID not being in the same container, obtaining the first information of the first SID according to the third information of the second SID, and determining the second information of the first SID according to the SID type of the first SID.

11. The method of claim 1, wherein the container is used for containing a standard SID or multiple compressed SIDs; the compressed SID comprises a node ID and a value of function of the node.

12. The method of claim 1, wherein the first SID is a compressed SID, and an index of the first SID is placed after the compressed SID in the DA.

13. The method of claim 12, wherein the index of the first SID is placed at Least Significant Bits (LSBs) of the DA.

14. A node, comprising:

an obtaining module, configured to obtain first information and second information of a first Segment ID (SID), the first information being used for indicating a position of a container where the first SID is located in an SID list of a data packet, and the second information being used for indicating a position of the first SID in the container;

a processing module, configured to obtain a position of the first SID in the SID list according to the first information and the second information; and

a sending module, configured to copy the first SID to a Destination Address (DA) of the data packet, and send the data packet.

15. A node, comprising: a transceiver, a processor and a memory configured to store a computer program capable of running in the processor;

the transceiver is configured to send and receive data under control of the processor;

the processor is configured to execute, when reading the computer program in the memory, steps in the method for data packet processing of any one of claims 1 to 13.

16. A computer-readable storage medium, in which a computer program is stored, wherein the computer program, when executed by a processor, executes steps in the method for data packet processing of any one of claims 1 to 13.