Authors: M. Xu   B. Wang
        Tsinghua University   Tsinghua University
        T. Wang
        Beijing University of Posts and Telecommunications
        J. Wu
        Tsinghua University

Segment Routing in Two Dimensional IP Routing

Abstract

Segment Routing (SR) allows a headend node to steer traffic into a
Segment Routing Policy (SR Policy), which represents the routing
path by matching the destination address and the corresponding
Binding Segment Identifier (BSID). However, determining the target
SR Policy only based on destination aspect is incapable for demands
for higher dimensional routing. Two Dimensional IP (TwoD-IP) routing
is an Internet routing architecture that makes forwarding decisions
based on source and destination addresses. TwoD-IP routing can
easily express a routing policy between host to host, or network to
network, and have much lower storage and calculation consumption
compared to the higher dimensional routing.

In this document, we extend SR to support TwoD-IP routing,
illustrate some typical scenarios of SR with TwoD-IP routing to
express the advantage of extending SR to support TwoD-IP routing,
and describe the mechanism of how TwoD IP routing protocol
cooperates with SR.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

Internet-Drafts are draft documents valid for a maximum of six
months and may be updated, replaced, or obsoleted by other documents
at any time. It is inappropriate to use Internet-Drafts as reference
material or to cite them other than as "work in progress."

This Internet-Draft will expire on 5 March 2022.

Copyright Notice

Table of Contents

1. Introduction

Segment routing (SR) [RFC8402] allows a headend node to steer a packet flow into an Segment Routing Policy (SR Policy), which represents the routing path. So that the administrator can specific forwarding path on headend node [I-D.ietf-spring-segment-routing-policy].

The headend node can steers packets into an SR Policy either by matching the destination address or routing policy. However, due to the increasing demands of higher dimensional routing like Multi-homing or Source Related Policy Routing, only directs packets based on destination aspect is limited under those scenarios. Moreover, directing packets into SR Policy by routing policy, which can match other fields in packets like port and source address, needs many access in memory. Considering the high-speed ternary content-addressable memory (TCAM) based solution for routers is limited by its low capacity, simply adding one more dimension on routing policy can easily become undeployable on TCAM-based solution.

To achieve higher Dimensional routing objectives, we introduce Two Dimensional-IP (TwoD-IP) routing protocol. [I-D.xu-rtgwg-twod-ip-routing] The TwoD-IP routing architecture can easily express a routing policy between host to host, or network to network, and have much lower storage and calculation consumption compared to higher dimensional routing.

In this document, we extend Segment Routing to support Two Dimensional IP(TwoD-IP) routing to enriches the routing abilities, describe how they cooperate to achieve higher dimensional routing like Multi-homing.

To extend Segment Routing to support Two Dimensional IP(TwoD-IP) routing, modification of the data plane and control plane is necessary. In data plane, the TwoD-IP routing protocol needs a TwoD-IP forwarding table to stores the source and destination address information. In control plane, the TwoD-IP routing protocol leverages OSPFv3 Router Information(RI) LSA to broadcast configuration and the SR Policy Dynamic Path Computation to compute optimal forwarding path under setting configuration. With these modifications, the headend node will be able to make forwarding decisions base on both source and destination aspects without damaging existing SR architecture.

2. Terminology

Terminology used in this document:

  *SR: Segment Routing.

*TwoD-IP routing protocol: Two Dimensional IP routing protocol,
    which makes routing decisions based on both destination and
    source IP addresses.

   *SID: Segment Identifier.

   *SRv6: Segment Routing over IPv6 data plane.

   *SR Policy: a framework that enables instantiation of an ordered
    list of segments on a node for implementing a source routing
    policy with a specific intent for traffic steering from that
    node.

3. Benefit of extend Segment routing to support TwoD-IP routing

   This section lists two scenarios which can benefit from TwoD-IP
   routing protocol with Segment Routing technology. Illustrating that
   TwoD-IP routing protocol can seamless deployment with SR and combine
   their advantage to achieve users demands of higher dimensional
   routing.

3.1. Multi-homing

   Even though Segment Routing is able to steer flows to the
   destination in different way, it is still limited to process the
   source-related routing scenario like Multi-homing.

   Multi-homing[RFC4177] is prevalent among ISPs for better traffic
   distribution and reliability. In this case, a network could be
   connected to multiple upstream ISPs, Hosts are assigned multiple
   addresses, one for each ISP. The network is responsible for
   delivering packets from Hosts to the export exit router that is
   connected to the corresponding upstream ISP.

   For example, in Figure 1, a multi-homed site is connected to two
   ISPs: ISP1 and ISP2. ISP1 has a prefix P1, and ISP2 has a prefix P2.
   A host connect to the multi-homed site has two addresses, address A
   that assigned from ISP1, and address B that assigned from ISP2. the
   multi-homed site should deliver the traffic from A towards the
   Internet to ISP1, and deliver the traffic from B towards the
   Internet to ISP2.

```
                  +--------------------+
                  |                    |
                  |      Internet      |
                  |                    |
                  +--+---------------+-+
                     |               |
                     |  l3           | l4
                     |               |
             +------+----+        +--+--------+
             |   ISP1    |        |   ISP2    |
             | Prefix P1 |        | Prefix P2 |
             +--------+--+        +-+---------+
                      |             |
                      | l1          | l2
                 +--+-----------+--+
                 |                 |
                 | Multi-homed Site |        +---------+
                 |                 +--------+  Host   |
                 +-----------------+        +---------+
                                        ISP1 assign address: A
                                        ISP2 assign address: B
```
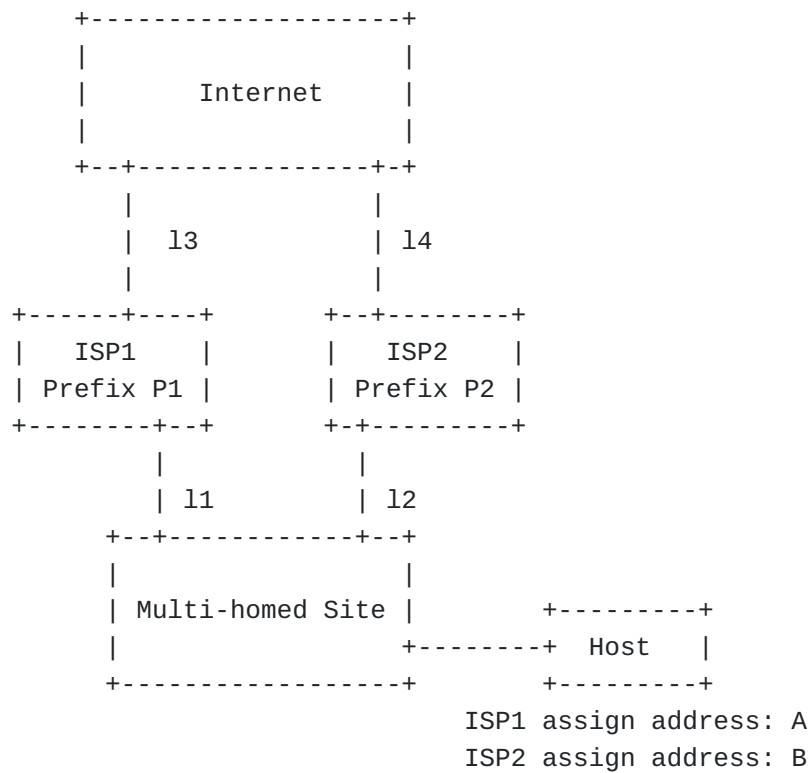
                Figure 1: Multi-homing scenario

   Although SR can assign different forwarding paths to different ISP
   for an incoming packet, it still lacks the ability to classify the
   packets toward the same destination address with different source
   addresses A and B. With the help of TwoD-IP and Segment Routing, the
   administrator can easily implement Multi-homing by modifying the
   headend node that receives packets from Host, which means that
   administrator does not need to concern about the intermediate node.
   After extending SR to support TwoD-IP routing, the headend node can
   routing packet based on both source and destination address, guides
   packets from Host through the optimal path to the corresponding ISP.

3.2. Source Related Policy Routing

   In this scenario, an ingress edge node wants to forward packets with
   the same destination address through different kind of paths in
   order to achieve source related demand.

   For example, in Figure 2, assume a network has four routers: a, b, c
   and d, c has a service(such as authentication or encipherer). The
   operator wants a packet from a to d to be delivered to service c
   first and then node c will forward the processed packet to it's
   destination d.

```
                              +---------+
                     +------+    c    +-----+
                     |      +(service)+     |
                     |       +---------+    |
                     |                      |
   +------+          +--+---+           +---+--+
   |  a   +----------+  b   +--------------+  d  |
   +------+          +------+           +------+
```

                   Figure 2: TwoD-IP routing for Service

   In Segment Routing Architecture, we can allocate a Service segment
   associated with node c's service.[I-D.ietf-spring-sr-service-
   programming] and can be integrated as part of an SR Policy
   P1(Headend node: b, color, Endpoint: d) of Segment-List < c > . But
   SR Policy steers packets to a specific SR Policy only when this
   packet's destination matching corresponding entry, which means
   headend node can't only steers packets from a to SR Policy P1.

   But with TwoD-IP routing, headend node b can easily steer packets
   matching destination of b and source of a to SR Policy P1, then the
   packet will be delivered to service c first and then node c will
   forward the processed packet to it's destination d.

4. Framework

   The mechanism of how we combine TwoD IP routing and Segment Routing
   can be separated into two parts: data plane and control plane.

4.1. Data Plane

   TCAM resources are future limited to the higher dimensional routing,
   including TwoD-IP routing. [I-D.xu-rtgwg-twod-ip-routing] propose a
   novel forwarding table design to increase the TCAM resource
   utilizatioin significantly. The data plane of extending SR to
   support TwoD-IP routing introduces this Two-D forwarding table
   design into the original SR data plane.

   Considering Segment Routing leverages the source routing paradigm,
   administrator just need to deploy TwoD-IP forwarding table in
   headend node, which makes implementing TwoD-IP routing much easier.
   Different with traditional destination-based FIB, each entry in the
   TwoD-IP forwarding table is a 3-tuple: {destination address, source
   address, next hop} [I-D.xu-rtgwg-twod-ip-routing]

   The next hop in the TwoD-IP forwarding table should steer the packet
   into the associated SR Policy, and the SR architecture has provided
   an identifier to do that: Binding segment/BSID. The headend of an SR
   Policy binds a SID(BSID) to its policy, so the next hop in TwoD-IP
   forwarding table should be a BSID, which is associated with the SR

Policy that corresponding to the entry's <destination address, source address>pairs.

So the TwoD-IP forwarding table tuple is {destination address, source address, BSID}, when a packet arrive, the headend node can extract both destination and source addresses from the packet, then lookup the Two-IP forwarding table, and output a matched entry. Finally, headend node will forward the packet to the matched entry's next hop, which is a BSID associated with a SR Policy, to get the dynamic candidate path that steers it to the destination.

Segment Routing can be instantiated on two data planes: SR over MPLS (SR-MPLS) and SR over IPv6 (SRv6). Different data plane has different BSID format, in SR-MPLS, SID are an MPLS label or an index into an MPLS label space, in SRv6, SID is an IPv6 address. So even we could deploy TwoD-IP routing on both two data planes, we still deploy TwoD-IP routing with SR over IPv6(SRv6), so that the next hop elements in TwoD-IP forwarding table will be SRv6 BSID which is an IPv6 address.

4.2. Control Plane

Within TwoD-IP routing, the control plane is concerned with user demands, it focus on providing services that are related with source addresses. They need to collect demands from users, and compute the routing table to satisfy these demands. And Segment Routing support many control plane. So we can extend one of them to achieve configuration exchange and optimal forwarding path computation objects..

  *TwoD-IP configuration exchange: TwoD-IP routing protocol focus on transforming users demand for destination and source address pairs to forwarding action, which mean that we have one more dimension of source address information to exchange along with headend node, and the forwarding configurations corresponding to the destination and source address pairs.

  *TwoD-IP forwarding path computation: We leverage the SR Policy Dynamic Path Computation to achieve forwarding path computation, transferring our demand for <destination, source> pair to optimization object and constraint source format which can specify a dynamic candidate path of SR Policy, then the dynamic candidate path can be computed by either the headend or a Path Computation Element (PCE). So TwoD-IP routing protocol doesn't need to concern about how the candidate path is computed and what the path really is.

5. Advertisement of TwoD-IP configuration

TwoD-IP configuration needs to be installed in the headend node so
that the headend node can creates the TwoD-IP forwarding table entry
and the SR Policy with dynamic candidate path corresponding it. The
TwoD-IP configuration can be installed in headend node manually or
automatically by advertising of TwoD-IP configurations between
nodes.

A TwoD-IP configuration contains three parts: destination
addresses,source addresses and the user demands of the <destination,
source> pairs, including optimization objective and constraints.

5.1. TwoD-IP configuration TLV

The configurations of TwoD-IP routing is within the TwoD-IP
configuration TLV, the TwoD-IP configuration TLV is a TLV of OSPFv3
Router Information(RI) LSA, TwoD-IP configuration information will
be broadcast by letting An OSPF router advertising an OSPFv3 RI LSA
that include the TwoD-IP configuration TLV.

Because TwoD-IP routing is deployed in SR over IPv6, the OSPF router
will exchange router information with OSPFv3 Router Information(RI)
LSA.

The router may advertise multiple instances of TwoD-IP configuration
TLV within the OSPFv3 Router Information(RI) LSA - one for each of
the TwoD-IP configuration needs to be advertised.

The format of the TwoD-IP configuration TLV is the same as the
format used by[RFC3630]. The variable TLV section consists of one or
more nested TLV tuples. The format of each TLV is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Sub-TLVs                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
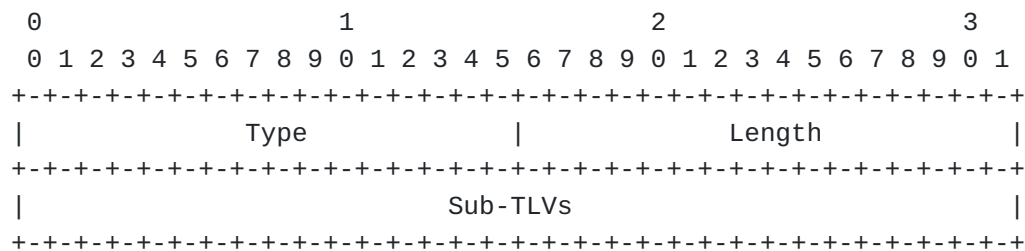
         Figure 3. TwoD-IP configuration TLV Format

Where:

Type is TBD

Length: 16 bit field. The total length of the value portion(Sub-TLVs) of the TLV

Sub-TLVs: Each TwoD-IP configuration TLV has four Sub-TLVs: Optimization Object Sub-TLV, Constraint Sub-TLV, destination and source address Sub-TLV. There could be multiple Sub-TLV of them to specify the dynamic candidate path of SR Policy and the <destination address, source address> pairs associated the SR Policy.

5.2. Optimization Object Sub-TLV

The Optimization and the constraint is basically refer to [I-D.filsfils-spring-sr-policy-considerations].

The format of Optimization Object Sub-TLV is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Type              |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Reserved               | Flags |     T       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Optimization Object Sub-TLV Format

Where:

Type: 16 bit field. The value is 1 for this type.

Length: 16 bit field. The total length of the value portion of the TLV.

Reserved (20 bits): This field MUST be set to zero on transmission and MUST be ignored on receipt.

Flags(4 bits): Identify the optimization objective, The following flags are defined:

```
 0 1 2 3
+-+-+-+-+
|M|N|   |
+-+-+-+-+
```
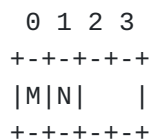
Where:

   *M flag: Min-Metric - requests computation of a solution
    Segment-List optimized for a selected metric.

   *N flag: Min-Metric with margin and maximum number of SIDs -
    Min Metric with two changes: a margin of by which two paths
    with similar metrics would be considered equal, a
    constraint on the max number of SIDs in the Segment-List.

T (Type - 8 bits): Specifies the metric type. Three values are
currently defined:

   *T=1: IGP metric

   *T=2: TE metric

   *T=3: Hop Counts

5.3. Constraints Sub-TLV

The constrains we use in TwoD-IP routing is Inclusion and/or
exclusion some node/service identified as a IPv6 address.

The format of Optimization Object Sub-TLV is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Type              |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Reserved          | Flags |        T              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          variable                             |
|                            ...                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
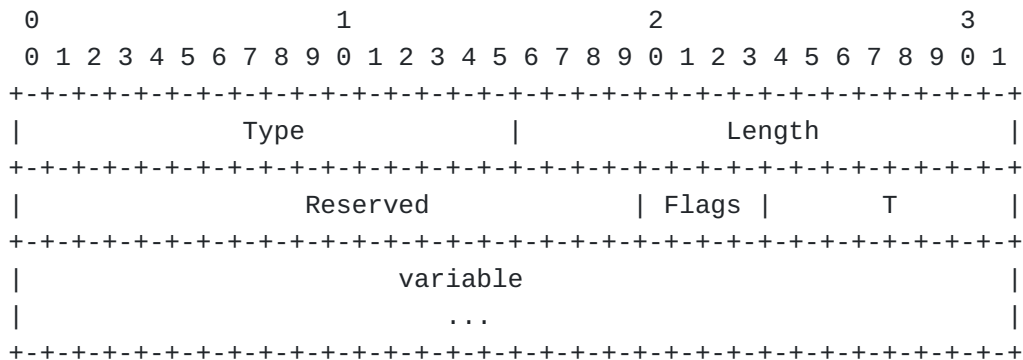
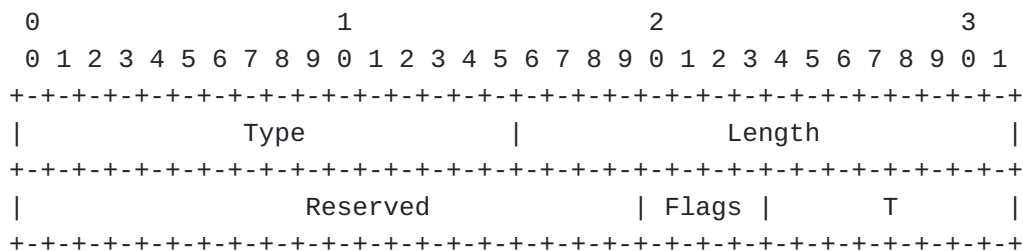Figure 5: Constraints Sub-TLV Format

Where:

Type: 16 bit field. The value is 2 for this type.

Length: 16 bit field. The total length of the value portion of
the TLV.

Reserved (20 bits): This field MUST be set to zero on
transmission and MUST be ignored on receipt.

Flags(4 bits): Identify the optimization objective, The following
flags are defined:

```
 0 1 2 3
+-+-+-+-+
|I|E|D| |
+-+-+-+-+
```

Where:

    *I flag: Inclusion

    *E flag: Exclusion

    *D flag: Diversity to another service instance (e.g., link,
     node)

T (Type - 8 bits): Specifies the metric type. Two values are
currently defined:

    *T=1: TE affinity

    *T=2: IPv6 address(can be a SRv6 SID)

variable: 128 bit field. Corresponding the variable defined in T.

5.4. destination/source address Sub-TLV

A TwoD-IP routing demand corresponding a <destination, source> pair,
so the Optimization object and constraint define a TwoD-IP demand
and naturally need to bind a <destination, source> pair. The pair's
information is carried in destination/source address Sub-TLV, and
it's format is:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Type               |              Length           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | PrefixLength   |    Reserved    |         PrefixNumbers          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         Address Prefix1                         |
    |                              ...                                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         Address Prefix2                         |
    |                              ...                                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                              ...                                |
    |                                                                 |
```

Figure 6: destination/source address Sub-TLV Format

Where:

   Type: 16 bit field. The value is 3 for destination address, 4 for
   source address.

   Length: 16 bit field. The total length of the value
   portion(addresses information) of the TLV.

   PrefixLength: 8 bit field. The length of IPv6 address. The IPv6
   address information is transferring in Prefix format in order to
   reduce packet length and all the addresses needed to transferring
   is group by same prefix length.

   Reserved (8 bits): This field MUST be set to zero on transmission
   and MUST be ignored on receipt.

   PrefixNumbers: 16 bit field. The number of address prefix in the
   length of PrefixLength.

   Address Prefix: The address prefix in the length of PrefixLength
   and will be padded with 0 to fit the multiple 32 bit length.

6. TwoD-IP forwarding path computation

   The procedure of transforming the TwoD-IP configuration to a
   candidate path has two steps: setting up the SR Policy Dynamic
   candidate path and TwoD-IP forwarding table entry creation.

6.1. Setting up the SR Policy Dynamic candidate path

   The TwoD-IP configuration contains the Optimization object and
   constraint information, when the headend node receive TwoD-IP
```

configuration information(manually or automatically), it will
extracts the Optimization object and constraint information to
generate a SR Policy which enable Dynamic candidate path.

An SR Policy is identified through the tuple<headend, color,
endpoint>,[I-D.ietf-spring-segment-routing-policy] so the headend
node will extract the first destination address in TwoD-IP
configuration TLV - destination address Sub-TLV as the endpoint,
dynamic assign a color value to distinguish the SR Policy from
others with the same endpoint. And the candidate path associated to
the SR Policy is a dynamic candidate path which expresses an
optimization objective and a set of constraints that extract from
the TwoD-IP configuration TLV - Optimization Object Sub-TLV and
Constraint Sub-TLV. Then the headend node(or with the help of a PCE)
computes the solution Segment-List that solve the optimization
problem and also match our TwoD-IP routing demand.

## 6.2.  TwoD-IP forwarding table entry creation

After Setting up the SR Policy dynamic candidate path, the active
dynamic candidate path will be defined with a BSID which can steers
the packets to the dynamic candidate path, and installs the BSID-
keyed entry in the TwoD-IP forwarding table.

The TwoD-IP forwarding table has two dimensions: destination and
source, so TwoD-IP forwarding table has two index: destination
address and source address, a pair of <destination address, source
address> determines a action(next hop) which is a BSID.

The headend node receives a TwoD-IP configuration TLV, generates a
SR Policy depends on it, and also will creates <destination address,
source address> pairs in TwoD-IP forwarding table as a index, and
installs their action(next hop) with the BSID that define the
dynamic candidate path generated with the same TwoD-IP configuration
TLV informations.

## 7.  TwoD-IP forwarding table Design

Traditional forwarding table only supports making forwarding
decisions based on destination IP addresses even in Segment Routing
architecture. TwoD-IP routing needs a new forwarding table structure
that supports making forwarding decisions based on both destination
and source IP addresses.

Considering the compatibility with Segment Routing architecture,
There's no need to modify the FIB structure directly. we imitate the
way SR Policy process interacting with the FIB process to install
our TwoD IP process in data plane. we just need to install in FIB
with destination address that in the active TwoD-IP <destination,

source> pair with next-hop = TwoD IP routing process which maintain a TwoD IP Forwarding table.

The TwoD-IP forwarding table structure is the same as [I-D.xu-rtgwg-twod-ip-routing] called FIST which is designed to avoid forwarding table explosion with a new source dimension.

In conclusion, TwoD IP routing procedure maintains a TwoD-IP forwarding table separate from the headend node's FIB and installs FIB entries with destination addresses in the chosen <destination address, source address> pairs with next-hop = TwoD IP routing interface.

When a headend node receives a packet with destination matching the destination address of TwoD IP routing pairs, headend node steers the packet to TwoD IP routing process to the TwoD-IP forwarding table, then the headend node will lookup the TwoD-IP forwarding table base on the packet's both destination and source address extracted from the packet and output a matched entry. Finally, headend node will forward the packet to the next hop associated with the matched entry, which is a BSID, after steering the packet to the BSID, the headend node will insert the Segment-List of the dynamic candidate path of the BSID to the packet head so that it can be forwarded through this path

SR Policy process interacts with the RIB/FIB process to install an active SR Policy in the data plane.[I-D.filsfils-spring-sr-policy-considerations] TwoD IP routing protocol doesn't change the FIB structure so Segment Routing's advanced features like On-Demand Next Hop will still work. (we allocate higher authority to TwoD IP routing process to install and modify FIB entries to avoid conflicting)

## 8.  Security Considerations

This document does not introduce additional security requirements and mechanisms.

## 9.  IANA Considerations

Some newly designed TwoD-IP routing protocols may need new protocol numbers assigned by IANA.

## 10.  References

### 10.1.  Normative References

[RFC2119]  Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/

RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8402]  Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L.,
           Decraene, B., Litkowski, S., and R. Shakir, "Segment
           Routing Architecture", RFC 8402, DOI 10.17487/RFC8402,
           July 2018, <https://www.rfc-editor.org/info/rfc8402>.

## 10.2.  Informative References

[I-D.filsfils-spring-sr-policy-considerations]
           Filsfils, C., Talaulikar, K., Krol, P., Horneffer, M.,
           and P. Mattes, "SR Policy Implementation and Deployment
           Considerations", Work in Progress, Internet-Draft, draft-
           filsfils-spring-sr-policy-considerations-07, 4 April
           2021, <https://www.ietf.org/archive/id/draft-filsfils-
           spring-sr-policy-considerations-07.txt>.

[I-D.ietf-spring-segment-routing-policy]
           Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A.,
           and P. Mattes, "Segment Routing Policy Architecture",
           Work in Progress, Internet-Draft, draft-ietf-spring-
           segment-routing-policy-13, 28 May 2021, <https://
           www.ietf.org/archive/id/draft-ietf-spring-segment-
           routing-policy-13.txt>.

[I-D.ietf-spring-sr-service-programming]
           Clad, F., Xu, X., Filsfils, C., Bernier, D., Li, C.,
           Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W.,
           and S. Salsano, "Service Programming with Segment
           Routing", Work in Progress, Internet-Draft, draft-ietf-
           spring-sr-service-programming-05, 10 September 2021,
           <https://www.ietf.org/archive/id/draft-ietf-spring-sr-
           service-programming-05.txt>.

[I-D.xu-rtgwg-twod-ip-routing] Xu, M., Wu, J., Yang, S., and D.
           Wang, "Two Dimensional IP Routing Architecture", Work in
           Progress, Internet-Draft, draft-xu-rtgwg-twod-ip-
           routing-00, 4 March 2012, <https://www.ietf.org/archive/
           id/draft-xu-rtgwg-twod-ip-routing-00.txt>.

[RFC3630]  Katz, D., Kompella, K., and D. Yeung, "Traffic
           Engineering (TE) Extensions to OSPF Version 2", RFC 3630,
           DOI 10.17487/RFC3630, September 2003, <https://www.rfc-
           editor.org/info/rfc3630>.

[RFC4177]  Huston, G., "Architectural Approaches to Multi-homing for
           IPv6", RFC 4177, DOI 10.17487/RFC4177, September 2005,
           <https://www.rfc-editor.org/info/rfc4177>.

**Authors' Addresses**

Mingwei Xu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing

Phone: +86-10-6278-5822
Email: xmw@cernet.edu.cn

Bo Wang
Tsinghua University
Beijing

Email: wangbo2019@tsinghua.edu.cn

Tingfeng Wang
Beijing University of Posts and Telecommunications
Beijing
P.R. China

Email: wangtingfeng@bupt.edu.cn

Jianping
Tsinghua University
Beijing
P.R. China

Email: jianping@cernet.edu.cn