

Enabling Segment Routing Flexible Algorithm

Table 1: Feature History

Feature Name	Release Information	Feature Description
IS-IS Flexible Algorithm Include Affinity Support	Cisco IOS XE Bengaluru 17.6.1	This feature supports "include-any" and "include-all" affinities in IS-IS. Prior to Cisco IOS XE Bengaluru 17.6.1 release, only Flexible Algorithm affinity "exclude-any" was supported.
Segment Routing Flexible Algorithm with OSPF	Cisco IOS XE Bengaluru 17.5.1	This feature allows you to configure Segment Routing Flexible Algorithm with OSPF. Flexible Algorithm with OSPF supports metric minimization and avoidance, multi-plane, delay metric with rounding, and ODN with auto-steering.

Feature Name	Release Information	Feature Description
Segment Routing Flexible Algorithm support for TI-LFA uLoop Avoidance, SID Leaking, and ODN with Auto-Steering	Cisco IOS XE Bengaluru 17.4.1	This feature allows you to compute Loop Free Alternate (LFA) paths, TI-LFA backup paths, and Microloop Avoidance paths for a particular Flexible Algorithm using the same constraints as the calculation of the primary paths for such Flexible Algorithms, for IS-IS. See Calculation of Flexible Algorithm Path, on page 4 Inter-area leaking of Flexible Algorithm SIDs and prefixes and selectively filtering the paths that are installed to the MFI are also supported. See Flexible Algorithm Prefix-SID Advertisement, on page 4 and Installation of Forwarding Entries for Flexible Algorithm Paths, on page 10
Segment Routing Flexible Algorithm with IS-IS	Cisco IOS XE Amsterdam 17.3.1	This feature allows you to customize IGP shortest path computation according to your needs. You can assign custom SR prefix-SIDs to forward the packets beyond link-cost-based SPF. As a result, a traffic engineered path is automatically computed by the IGP to any destination reachable by the IGP.

Segment Routing Flexible Algorithm allows operators to customize IGP shortest path computation according to their own needs. An operator can assign custom SR prefix-SIDs to realize forwarding beyond link-cost-based SPF. As a result, Flexible Algorithm provides a traffic engineered path automatically computed by the IGP to any destination reachable by the IGP.

The SR architecture associates prefix-SIDs to an algorithm which defines how the path is computed. Flexible Algorithm allows for user-defined algorithms where the IGP computes paths based on a user-defined combination of metric type and constraint.

This document describes the IS-IS and OSPF extensions to support Segment Routing Flexible Algorithm on an MPLS data-plane.

- Prerequisites for Flexible Algorithm, on page 3
- Restrictions for Flexible Algorithm, on page 3
- Building Blocks of Segment Routing Flexible Algorithm, on page 3
- Flexible Algorithm Configurations, on page 10
- Verifying the Flexible Algorithm Configuration, on page 16

Prerequisites for Flexible Algorithm

Segment routing must be enabled on the router before the Flexible Algorithm functionality is activated.

Restrictions for Flexible Algorithm

- A maximum of four IS-IS flexible algorithm sessions are supported.

Building Blocks of Segment Routing Flexible Algorithm

This section describes the building blocks that are required to support the SR Flexible Algorithm functionality in IS-IS and OSPF.

Flexible Algorithm Definition

Many possible constraints may be used to compute a path over a network. Some networks are deployed with multiple planes. A simple form of constraint may be to use a particular plane. A more sophisticated form of constraint can include some extended metric, like delay, as described in [RFC7810]. Even more advanced case could be to restrict the path and avoid links with certain affinities. Combinations of these are also possible. To provide a maximum flexibility, the mapping between the algorithm value and its meaning can be defined by the user. When all the routers in the domain have the common understanding what the particular algorithm value represents, the computation for such algorithm is consistent and the traffic is not subject to looping. Here, since the meaning of the algorithm is not defined by any standard, but is defined by the user, it is called as Flexible Algorithm.

Effective Cisco IOS XE Bengaluru 17.6.1, the following affinity types are supported in IS-IS and OSPF:

- Include-any—Includes a link when calculating a path, if at least one link color is same as the defined color under IGP.
- Exclude-any—Excludes a link when calculating a path, if at least one link color is same as the defined color under IGP.
- Include-all—Includes a link when calculating a path, only if each link color is same as the defined color under IGP.

Flexible Algorithm Support Advertisement

An algorithm defines how the best path is computed by IGP. Routers advertise the support for the algorithm as a node capability. Prefix-SIDs are also advertised with an algorithm value and are tightly coupled with the algorithm itself.

An algorithm is a one octet value. Values from 128 to 255 are reserved for user defined values and are used for Flexible Algorithm representation.

Flexible Algorithm Definition Advertisement

To guarantee the loop free forwarding for paths computed for a particular Flexible Algorithm, all routers in the network must share the same definition of the Flexible Algorithm. This is achieved by dedicated router(s) advertising the definition of each Flexible Algorithm. Such advertisement is associated with the priority to make sure that all routers will agree on a single and consistent definition for each Flexible Algorithm.

Definition of Flexible Algorithm includes:

- Metric type
- Affinity constraints

To enable the router to advertise the definition for the particular Flexible Algorithm, **advertise-definition** command is used. At least one router in the area, preferably two for redundancy, must advertise the Flexible Algorithm definition. Without the valid definition being advertised, the Flexible Algorithm will not be functional.

Flexible Algorithm Prefix-SID Advertisement

To forward traffic on a Flexible Algorithm specific path, all routers participating in the Flexible Algorithm install a MPLS labeled path for the Flexible Algorithm specific prefix-SID. This Flexible Algorithm specific prefix-SID is advertised for the prefix. Only prefixes for which the Flexible Algorithm specific Prefix-SID is advertised, is subject to Flexible Algorithm specific forwarding.

Inter-Area Leaking

Effective Cisco IOS XE Bengaluru 17.4.1, Flexible Algorithm SIDs and prefixes are leaked between IS-IS areas. However, only the prefixes that are reachable by Layer1 or Layer2 routers are leaked. Similarly, only SIDs that are reachable in a given Flexible Algorithm are leaked.

For example, consider a prefix P:

- that originated in Layer1 and leaked in to Layer2
- has SID value = 128 in Flexible Algorithm 128, and SID value = 129 in Flexible Algorithm 129
- for which Layer1 path exist only for SID value = 128, but not for SID value = 129

As a result of the above conditions, only SID 128 is leaked from Layer1 to Layer2 and not SID 129.

Calculation of Flexible Algorithm Path

A router may compute path for multiple Flexible Algorithms. A router must be configured to support particular Flexible Algorithm before it can compute any path for such Flexible Algorithm. A router must have a valid definition of the Flexible Algorithm before such Flexible Algorithm is used.

When computing the shortest path tree for particular Flexible Algorithm:

- All nodes that do not advertise support for such Flexible Algorithm are pruned from the topology.
- If the Flexible Algorithm definition includes affinities that are excluded, then all links for which any of such affinities are advertised are pruned from the topology.

- Router uses the metric that is part of the Flexible Algorithm definition. If the metric is not advertised for the particular link, that link is pruned from the topology.

Effective Cisco IOS XE Bengaluru 17.4.1, for IS-IS, Loop Free Alternate (LFA) paths, TI-LFA backup paths, and Microloop Avoidance paths for a Flexible Algorithm are computed using the same constraints as the calculation of the primary paths for such Flexible Algorithm. These paths use Prefix-SIDs advertised specifically for such Flexible Algorithm to enforce a backup or microloop avoidance path.

OSPF Flexible Algorithm TI-LFA

Table 2: Feature History

Feature Name	Release Information	Feature Description
OSPF Flexible Algorithm (Ph2): Topology-Independent Loop-Free Alternate (TI-LFA) Path	Cisco IOS XE Bengaluru 17.6.1	This feature allows you to configure the Loop-Free Alternate (LFA) and TI-LFA backup or repair paths for a Flexible Algorithm. The backup path is computed based on the constraints and metrics of the primary path. Prior to Cisco IOS XE Bengaluru 17.6.1, OSPF Flexible Algorithm supported only the primary path.

Effective Cisco IOS XE Bengaluru 17.6.1, for OSPFv2, OSPF Flexible Algorithm supports Loop Free Alternate (LFA) paths and Topology-Independent Loop-Free Alternate (TI-LFA) backup or repair paths. The constraints and metrics of the primary path applies for the computation of the backup path.

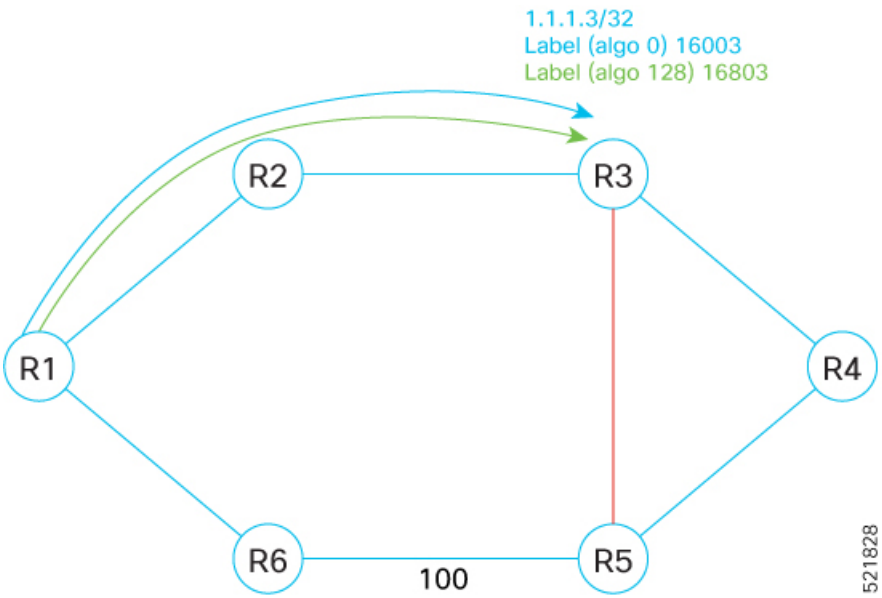
OSPF Flexible Algorithm TI-LFA Computation

- The TI-LFA algorithm applies to the topology that you have included in the Flexible Algorithm.
- The backup path is expressed with the prefix-SIDs and optimized based on the Flexible Algorithm.

Examples

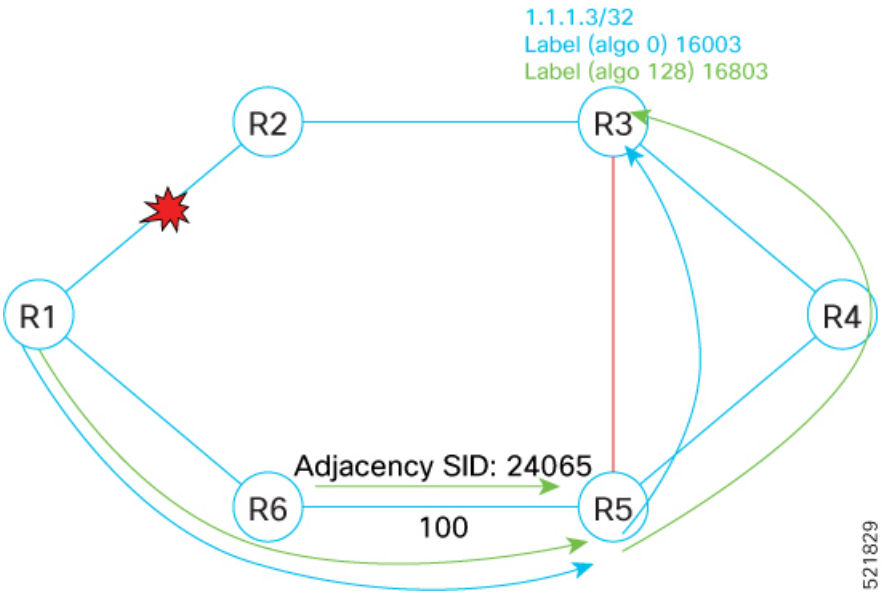
The following examples show how the Flexible Algorithm calculates the primary and TI-LFA backup paths:

Figure 1: Primary Path Per Flexible Algorithm



For the prefix 1.1.1.3/32, Algo 0 is assigned the prefix-SID 16003 and Algo 128 is assigned the prefix-SID 16803. Each prefix-SID is related to an algorithm. If a node advertises participation in a Flexible Algorithm, it also advertises a prefix-SID. All routers participate in Algo 128, which is defined as minimum IGP metric and avoid red affinity. The default IGP link metric is 1:10. The primary path for R1 to R3 is through R2.

Figure 2: TI-LFA Backup Path Per Flexible Algorithm



Assume that the link between R1 and R2 is down. The label 24065 is the adjacency SID for the link between R6 and R5. The backup path for Algo 0 is <24065, 16003> through R6.

The backup path for Algo 128 is <24065, 16803> through R6. Therefore, the usage of Algo 128 prefix-SID 16803 ensures that the Algo 128 backup path also avoids the red affinity and uses R1→R6→R5→R4→R3.

Configuring OSPF Flexible Algorithm TI-LFA

Procedure

Step 1 Configure the Flexible Algorithm TI-LFA functionality.

```
Router(config)#router ospf 1
Router(config-router)#router-id 255.255.255.255
Router(config-router)#nsr
Router(config-router)#segment-routing area 0 mpls
Router(config-router)#segment-routing mpls
Router(config-router)#segment-routing prefix-sid-map advertise-local
Router(config-router)#fast-reroute per-prefix enable prefix-priority low
Router(config-router)#fast-reroute per-prefix ti-lfa
Router(config-router)#flex-algo 129
Router(config-ospf-fa)#fast-reroute disable
Router(config-ospf-fa)#metric-type te-metric
Router(config-ospf-fa)#affinity include-all
Router(config-ospf-fa-aff)#name GREEN
Router(config-ospf-fa-aff)#name RED
Router(config-ospf-fa-aff)#affinity-map
Router(config-ospf-aff-map)#name GREEN bit-position 130
Router(config-ospf-aff-map)#name RED bit-position 129
Router(config-ospf-aff-map)#end
```

Note
You can use the **fast-reroute disable** command to disable TI-LFA for a particular Flexible Algorithm.

Step 2 Attach the OSPF Flex-Algo to the SR-TE policy to steer traffic.

```
Router(config)#segment-routing traffic-eng
Router(config-srte)#policy FLEXALGO1
Router(config-srte-policy)#color 1 end-point 6.6.6.6
Router(config-srte-policy)#candidate-paths
Router(config-srte-policy-path)#preference 1
Router(config-srte-policy-path-pref)#constraints
Router(config-srte-policy-path-pref-constr)#segments
Router(config-srte-policy-path-pref-constr-seg)#dataplane mpls
Router(config-srte-policy-path-pref-constr-seg)#algorithm 129
Router(config-srte-policy-path-pref-constr-seg)#!
Router(config-srte-policy-path-pref-constr-seg)#!
Router(config-srte-policy-path-pref-constr-seg)#dynamic
Router(config-srte-policy-path-pref-dyn)#end
```

Verifying the OSPF Flexible Algorithm TI-LFA

Verify the OSPF Flexible Algorithm TI-LFA:

Procedure

Step 1 Use the **show segment-routing traffic-eng policy name** command to verify the IP address of the endpoint, the algorithm attached, tunnel ID, and the event history.

```
Router#show segment-routing traffic-eng policy name <FLEXALGO1> [detail]

Name: FLEXALGO1 (Color: 1 End-point: 6.6.6.6)
  Owners : CLI
  Status:
    Admin: up, Operational: up for 00:00:04 (since 05-07 15:26:18.320)
  Candidate-paths:
    Preference 1 (CLI):
      Constraints:
        Algorithm: 129---Algo 129 is attached
      Dynamic (active)
          17666 [Prefix-SID, 6.6.6.6]
  Attributes:
    Binding SID: 16
      Allocation mode: dynamic
      State: Programmed
  Tunnel ID: 65536 (Interface Handle: 0x21)
Per owner configs:
    CLI
      Binding SID: dynamic
  Stats:
    5 minute output rate 0 bits/sec, 0 packets/sec
    Packets: 0  Bytes: 0

  Event history:
    Timestamp                  Client                  Event type              Context:
Value
    ---------                  ------                  ----------              -------:
-----
    05-05 11:50:51.019         CLI                     Policy created          Name: CLI
    05-05 11:50:51.089         CLI                     Set colour              Colour: 1
    05-07 15:25:35.625         CLI                     Set colour              Colour: 1
    05-07 15:25:35.625         CLI                     Set end point           End-point:
 6.6.6.6
    05-07 15:25:35.640         FH Resolution           Policy state UP         Status:
PATH RESOLVED CP: 1
    05-07 15:25:36.303         FH Resolution           REOPT triggered         Status:
REOPTIMIZED CP: 1
    05-07 15:26:18.321         CLI                     Policy ADMIN DOWN       shutdown:
FLEXALGO1
    05-07 15:26:19.173         CLI                    Policy state DOWN       no shutdown:
 FLEXALGO1
    05-07 15:26:19.177         FH Resolution           Policy state UP         Status:
PATH RESOLVED CP: 1
```

Step 2 Use the **show ip ospf topology route algorithm** command to display the path information of the route computed based on route calculation. The command displays details such as the attached label, the algorithm ID, SID, total metric, primary path (label), and the backup path (label and TI-LFA tunnel).

```
Router#show ip ospf topology route algorithm 129 6.6.6.6/32 detail
 Route Table of OSPF-1 with router ID 255.255.255.255 (VRF global)
```

```
 Algorithm 129

  Route entry for 6.6.6.6/32, Metric 102, SID 666, Label 17666
   Priority : Medium
   Flags : New

     Route type : Intra
     Flags: Inuse
     Prefix Contrib Algo 129 SID 666
      Total Metric : 102
      Contrib Flags : Inuse, Reachable
      SID Flags : Exp Null, PHP off, Index, Global, Node, Valid
Path: 102.0.0.1, from 4.4.4.4, via GigabitEthernet0/5/1--------------primary path
       Out Label  : 17666
       Weight     : 0

 Backup path: TI-LFA, Repair-List: P node: 5.5.5.5 Label: 17555
                                          Backup Tunnel: MPLS-SR-Tunnel7----------TILFA
 Tunnel
              101.0.0.1, from 6.6.6.6, via BDI1001,
              Out Label: 17666
              Attributes: Metric: 203, LC Disjoint, Node Protect, Interface Disjoint, SRLG
 Disjoint

      Path: 122.0.0.1, from 4.4.4.4, via BDI1022
       Out Label  : 17666
       Weight     : 0

           Backup path: TI-LFA, Repair-List: P node:          5.5.5.5 Label: 17555
                                             Backup Tunnel: MPLS-SR-Tunnel7
              101.0.0.1, from 6.6.6.6, via BDI1001,
              Out Label: 17666
              Attributes: Metric: 203, Node Protect, Interface Disjoint, SRLG Disjoint
```

Step 3 Use the **show ip ospf topology fast-reroute tunnel** command to display details such as the flex-algo TI-LFA tunnels, interface, next hop, mid/end points, and label.

```
Router#show ip ospf topology fast-reroute tunnel

            OSPF Router with ID (255.255.255.255) (Process ID 1)

Tunnel                Interface         Next Hop         Mid/End Point    Label
------------------------------------------------------------------------------
MPLS-SR-Tunnel7       BD1001            101.0.0.1        5.5.5.5          17555
MPLS-SR-Tunnel8       BD1001            101.0.0.1        7.7.7.7          17777
```

Step 4 Use the **show mpls infrastructure lfd lte** <*flex-algo label*> command to view the next hop outgoing interface/IP address.

```
RSP2-12#show mpls infrastructure lfd lte 17111
17111 [table 0]
  ldm: LTE Broker, OSPF
  flags: nsf | ver2 | multi-bind | srlbls (0x1B)
  plist: NONIP (0x6A3B5FB8) 0-11.11.11.11/32-33028
  output chain:
    label 17111-(local:17111)
    TAG adj out of GigabitEthernet0/5/1, addr 102.0.0.1 7F083B00
```

Step 5 Use the **show ip ospf topology summary** command to display information such as number of areas, algos, nodes, the algo label, and the metric type.

show ip ospf [*process-id*] [*area-id*] **topology summary** [**algorithm** *algorithm-number*]

```
Router#show ip ospf topology summary
        Process OSPF-1

 Instance global
  Router ID       : 255.255.255.255
  Number of Areas : 1
  Number of Algos : 1
  Max Path count  : 4
  Route count     : 10
  SR Global Block : 17000 - 23999
   Area 0
    Number of Nodes : 11
     Algo 129
      FAD Advertising Router : 199.199.199.199
      FAD Area ID : 0
       Algo Type   : 0
       Metric Type : 1
       Include Any Affinity : 0x00000000 00000000 00000000 00000000 00000006
```

Installation of Forwarding Entries for Flexible Algorithm Paths

Flexible Algorithm path to any prefix must be installed in the forwarding using the Prefix-SID that was advertised for such Flexible Algorithm. If the Prefix-SID for Flexible Algorithm is not known, such Flexible Algorithm path is not installed in forwarding for such prefix..

Only MPLS to MPLS entries are installed for a Flexible Algorithm path. No IP to IP or IP to MPLS entries are installed. These follow the native IPG paths computed based on the default algorithm and regular IGP metrics.

Effective Cisco IOS XE Bengaluru 17.4.1, you can selectively filter the paths that are installed to the MFI by using the configuration command **distribute-list** *filter name* **in**. See Configuring Selective Path Filtering for configuration example. This feature is only supported for IS-IS Flexible Algorithm.

Flexible Algorithm Configurations

This section describes various configurations that are required to support the SR Flexible Algorithm functionality.

The following IS-IS and OSPF configuration sub-mode is used to configure Flexible Algorithm:

flex-algo *algorithm number*

algorithm number —value from 128 to 255

Commands under Flexible Algorithm Configuration Mode

The following commands are used to configure Flexible Algorithm definition under the flex-algo sub-mode:

- To set the metric delay:
  - In IS-IS

    metric-type delay

Note By default the regular IGP metric is used. If delay metric is enabled, the advertised delay on the link is used as a metric for Flexible Algorithm computation.

- In OSPF

```
metric-type {delay | te-metric | igp-metric}
```

- To set the affinity:
  - In IS-IS

```
affinity

[exclude-any | include-any | include-all]
name affinity-name
```

  - In OSPF

```
affinity {exclude-any | include-any | include-all}
name affinity-name
```

    affinity-name—name of the affinity-map

- To set the priority in both IS-IS and OSPF:

```
priority priority value
```

  priority value—priority used during the Flexible Algorithm definition election.

The following command is used to enable advertisement of the Flexible Algorithm definition in IS-IS and OSPF:

```
advertise-definition
```

Commands for Affinity Configuration

The following commands are used for defining the affinity-map. Affinity-map associates the name with the particular bit positions in the Extended Admin Group bitmask.

- In IS-IS:

```
affinity-map affinity name bit-position bit number
```

- In OSPF:

```
affinity-map
 name affinity name bit-position bit number
```

  - *affinity name*—name of the affinity-map.
  - *bit number*—bit position in the Extended Admin Group bitmask.

The following command is used to associate the affinity with an interface:

- IS-IS

```
isis affinity flex-algo
name affinity-name
```

- OSPF

```
ip ospf affinity flex-algo
name affinity-name
```

affinity-name—name of the affinity-map

Command for Prefix-SID for Flexible Algorithm Configuration

The following command is used to advertise prefix-SID for flexible algorithm in IS-IS and OSPF:

```
segment-routing mpls
connected-prefix-sid-map
address-family ipv4 [ algorithmalgorithm-number | strict-spf]
prefix/sid [index | absolute] sid value range range value
```

Configuring IS-IS Flexible Algorithm

The following is an example of how to configure the IS-IS flexible algorithm.

```
router isis 1
 net 49.0002.0000.0001.00
 is-type level-1
 metric-style wide
 log-adjacency-changes
 nsf cisco
 distribute link-state
 segment-routing mpls
 segment-routing prefix-sid-map advertise-local
 affinity-map blue bit-position 8
 affinity-map green bit-position 201
 affinity-map red bit-position 65

 fast-reroute per-prefix level-1 all
 fast-reroute tie-break level-1 node-protecting 100
 fast-reroute tie-break level-1 srlg-disjoint 50
 fast-reroute ti-lfa level-1
 fast-reroute ti-lfa level-2
 microloop avoidance segment-routing
 microloop avoidance rib-update-delay 10000

flex-algo 129
  advertise-definition
  metric-type delay
  priority 120
  affinity
   exclude-any
    name red
   !
 bfd all-interfaces
 mpls traffic-eng router-id Loopback1
 mpls traffic-eng level-1
 mpls traffic-eng level-2
```

Use the **fast-reroute disable** command to disable TI-LFA.

Configuring OSPF Flexible Algorithm

The following is an example of how to configure OSPF flexible algorithm.

```
interface Gi0/0/0
 ip ospf affinity flex-algo
  name red
 !
interface Ethernet0/1
 ip ospf affinity flex-algo
  name blue
  name red
 !
interface Ethernet0/2
 ip ospf affinity flex-algo
  name blue
 !

router ospf 1
 router-id 255.255.255.255
 nsr
 segment-routing area 0 mpls
 segment-routing mpls
 segment-routing prefix-sid-map advertise-local
 fast-reroute per-prefix enable prefix-priority low ------ Enables TI-LFA for prefix
 fast-reroute per-prefix ti-lfa --------- Enables TI-LFA for all the flex algo
flex-algo 129
fast-reroute disable ------ Optional command to disable TI-LFA on a particular Flex algo

 affinity-map
  name red bit-position 65
  name blue bit-position 8
  name green bit-position 201
 !
 flex-algo 128
  affinity exclude-any
   name red
   name blue
!
  affinity include-any
   name green
  !
 !
 flex-algo 129
  affinity exclude-any
   name green
  !
  affinity include-all
   name red
   name blue
  !
  advertise-definition
  priority 20
  metric-type delay
 !
policy FLEXALGO1
  color 1 end-point 11.11.11.11
  candidate-paths
   preference 1
    constraints
     segments
      dataplane mpls
      algorithm 129 ------ Attach OSPF flexible algorithm into SR-TE policy to steer traffic
```

```
 !
!
dynamic
```

Configuring Prefix-SID

```
segment-routing mpls
set-attributes
  address-family ipv4
   sr-label-preferred
  exit-address-family
 global-block 17000 23999
connected-prefix-sid-map
  address-family ipv4
   10.0.0.1/32 index 18 range 1
  exit-address-family
  address-family ipv4 algorithm 129
   10.0.0.1/32 index 38 range 1
  exit-address-family
```

Configuring SRTE-ODN Assosiation

```
segment-routing traffic-eng
 on-demand color 100
  authorize
  candidate-paths
   preference 100
    constraints
     segments
      dataplane mpls
      algorithm 129
     !
    !
    dynamic
     metric
      type delay
    !
   !
```

Configuring the Interface for Flexible Algorithm

```
interface GigabitEthernet0/0/6
 ip address 11.11.11.1 255.255.255.0
 ip router isis 1
 mpls ip
 mpls traffic-eng tunnels
 bfd template pw_bfd
 isis network point-to-point
 isis affinity flex-algo
  name red
  !
```

Configuring BGP

```
router bgp 100
```

```
 bgp router-id 10.0.0.1
 bgp log-neighbor-changes
 bgp graceful-restart
 neighbor 2.2.2.2 remote-as 100
 neighbor 2.2.2.2 ha-mode sso
 neighbor 2.2.2.2 update-source Loopback1
 !
 address-family ipv4
  neighbor 2.2.2.2 activate
 exit-address-family
 !
address-family vpnv4
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community both
  neighbor 2.2.2.2 route-map BGP_TE_MAP out
 exit-address-family
 !
 address-family ipv4 vrf SR
  redistribute connected
  neighbor 132.1.1.1 remote-as 101
  neighbor 132.1.1.1 activate
 exit-address-family
!
```

Configuring Selective Path Filtering

The following is an example of how you can selectively filter the paths that are installed to the MFI.

```
R1-ASR900-2011(config)#show run int lo1
Building configuration...

Current configuration : 95 bytes
!
interface Loopback1
 ip address 10.0.0.1 255.255.255.255
 ip router isis 1
 isis tag 111
end

On R6-node:
++++++++++++
!
route-map block deny 10
 match tag 111
!
route-map block permit 100
!

router isis 1
!
 flex-algo 135
  advertise-definition
  metric-type delay
  priority 120
  affinity
   exclude-any
    name green
   !
  !
 !
distribute-list route-map block in
 bfd all-interfaces
 mpls traffic-eng router-id Loopback1
```

```
 mpls traffic-eng level-1
 mpls traffic-eng level-2
!
R6-RSP3-2038(config)#show isis rib flex-algo 129


IPv4 local RIB for IS-IS process 1

IPV4 unicast topology base (TID 0, TOPOID 0x0) =================
Repair path attributes:
    DS - Downstream, LC - Linecard-Disjoint, NP - Node-Protecting
    PP - Primary-Path, SR - SRLG-Disjoint


Flex-algo 129

10.0.0.1/32  prefix attr X:0 R:0 N:1  source router id: 10.0.0.1  SID index 38
  [115/L1/13] via 45.45.45.1(BDI1) R4-RSP3-2036.00-00, from 10.0.0.1, tag 111,
     LSP 4/1/156(156), prefix attr: X:0 R:0 N:1
     Source router id: 10.0.0.1
      Prefix-SID index: 38, R:0 N:1 P:0 E:0 V:0 L:0
     label: 17038
  [115/L1/13] via 48.48.48.1(BDI2) R5-ASR900-2012.00-00, from 10.0.0.1, tag 111,
     LSP 5/1/156(156), prefix attr: X:0 R:0 N:1
     Source router id: 10.0.0.1
      Prefix-SID index: 38, R:0 N:1 P:0 E:0 V:0 L:0
     label: 17038

3.3.3.3/32  prefix attr X:0 R:0 N:1  source router id: 3.3.3.3  SID index 200 - Bound
  [115/L1/12] via 45.45.45.1(BDI1) R4-RSP3-2036.00-00, from 3.3.3.3, tag 0
     LSP 4/2/150(150), prefix attr: X:0 R:0 N:1
     Source router id: 3.3.3.3
      Prefix-SID index: 200, R:0 N:1 P:0 E:0 V:0 L:0
     label: 17200
     repair path: 48.48.48.1 (BDI2) metric: 14 (NP,SR)
       local LFA
       label: 17200
       repair source: R2-RSP3-2015, LSP 2

4.4.4.4/32  prefix attr X:0 R:0 N:1  source router id: 4.4.4.4  SID index 221 - Bound
  [115/L1/12] via 48.48.48.1(BDI2) R5-ASR900-2012.00-00, from 4.4.4.4, tag 0
     LSP 5/3/151(151), prefix attr: X:0 R:0 N:1
     Source router id: 4.4.4.4
      Prefix-SID index: 221, R:0 N:1 P:0 E:0 V:0 L:0
     label: 17221
     repair path: 10.0.0.1 (MPLS-SR-Tunnel9) metric: 3 (DS,NP,SR)
       next-hop: 45.45.45.1 (BDI1)

R6-RSP3-2038(config)#show mpls for lab 17038 <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< no output from
MFI
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
R6-RSP3-2038(config)#
```

Verifying the Flexible Algorithm Configuration

Use the command **show isis flex-algo** *value* to view all the information regarding IS-IS flexible algorithm.

```
show isis flex-algo 129
Tag 1:
IS-IS Flex-Algo Database
 Flex-Algo count: 7
```

```
Flex-Algo 129:
    IS-IS Level-1
      Definition Priority: 222
      Definition Source: R2-RSP3-2015.00, (Local)
      Definition Equal to Local: Yes
      Definition Metric Type: Delay
      Definition Flex-Algo Prefix Metric: No
      Disabled: No
      Microloop Avoidance Timer Running: No
   Local Priority: 222
   FRR Disabled: No
   Microloop Avoidance Disabled: No
```

Use the command **show isis rib flex-algo** *value* to view all the IS-IS local RIB information.

```
show isis rib flex-algo 129
IPv4 local RIB for IS-IS process 1

IPV4 unicast topology base (TID 0, TOPOID 0x0) =================
Repair path attributes:
    DS - Downstream, LC - Linecard-Disjoint, NP - Node-Protecting
    PP - Primary-Path, SR - SRLG-Disjoint

Flex-algo 129

10.0.0.1/32  prefix attr X:0 R:0 N:1  source router id: 10.0.0.1  SID index 38 - Bound
  [115/L1/113] via 11.11.11.1(GigabitEthernet0/4/6) R1-ASR920-2011.00-00, from 10.0.0.1,
tag 0
     LSP 6/6/351(351), prefix attr: X:0 R:0 N:1
     Source router id: 10.0.0.1
      Prefix-SID index: 38, R:0 N:1 P:0 E:0 V:0 L:0
     label: implicit-null
     repair path: 20.20.20.2 (GigabitEthernet0/4/7) metric: 117 (DS,SR)
       local LFA
       label: implicit-null
       repair source: R1-ASR920-2011, LSP 6

2.2.2.2/32  prefix attr X:0 R:0 N:1  source router id: 2.2.2.2  SID index 39 - Bound
  [115/L1/24] via 13.13.13.2(GigabitEthernet0/1/5) R4-RSP3-2036.00-00, from 2.2.2.2, tag 0
     LSP 2/3/345(345), prefix attr: X:0 R:0 N:1
     Source router id: 2.2.2.2
      Prefix-SID index: 39, R:0 N:1 P:0 E:0 V:0 L:0
     label: 17039
     repair path: 4.4.4.4 (MPLS-SR-Tunnel4) metric: 170 (DS,NP,SR)
       next-hop: 20.20.20.2 (GigabitEthernet0/4/7)
       TI-LFA node/SRLG-protecting, SRLG-protecting
       SRGB: 17000, range: 7000 prefix-SID index: 39, R:0 N:1 P:0 E:0 V:0 L:0
       label: 17039
       P node: R3-RSP2-2013[4.4.4.4], label: 17221
       repair source: R6-RSP3-2038, LSP 3

4.4.4.4/32  prefix attr X:0 R:0 N:1  source router id: 4.4.4.4  SID index 221 - Bound
  [115/L1/172] via 13.13.13.2(GigabitEthernet0/1/5) R4-RSP3-2036.00-00, from 4.4.4.4, tag
0
     LSP 2/7/24(24), prefix attr: X:0 R:0 N:1
     Source router id: 4.4.4.4
      Prefix-SID index: 221, R:0 N:1 P:0 E:0 V:0 L:0
     label: 17221
     repair path: 20.20.20.2 (GigabitEthernet0/4/7) metric: 184 (DS,NP,SR)
       local LFA
       label: 17221
       repair source: R3-RSP2-2013, LSP 7
```

```
5.5.5.5/32  prefix attr X:0 R:0 N:1  source router id: 5.5.5.5  SID index 222 - Bound
  [115/L1/17] via 13.13.13.2(GigabitEthernet0/1/5) R4-RSP3-2036.00-00, from 5.5.5.5, tag 0
     LSP 2/2/347(347), prefix attr: X:0 R:0 N:1
     Source router id: 5.5.5.5
      Prefix-SID index: 222, R:0 N:1 P:0 E:0 V:0 L:0
     label: implicit-null
     repair path: 4.4.4.4 (MPLS-SR-Tunnel4) metric: 170 (DS,SR)
       next-hop: 20.20.20.2 (GigabitEthernet0/4/7)
       TI-LFA SRLG-protecting
       SRGB: 17000, range: 7000 prefix-SID index: 222, R:0 N:1 P:0 E:0 V:0 L:0
       label: 17222
       P node: R3-RSP2-2013[4.4.4.4], label: 17221
       repair source: R4-RSP3-2036, LSP 2

6.6.6.6/32  prefix attr X:0 R:0 N:1  source router id: 6.6.6.6  SID index 333 - Bound
  [115/L1/122] via 13.13.13.2(GigabitEthernet0/1/5) R4-RSP3-2036.00-00, from 6.6.6.6, tag
0
     LSP 2/4/351(351), prefix attr: X:0 R:0 N:1
     Source router id: 6.6.6.6
      Prefix-SID index: 333, R:0 N:1 P:0 E:0 V:0 L:0
     label: 17333
     repair path: 4.4.4.4 (MPLS-SR-Tunnel4) metric: 170 (DS,NP,SR)
       next-hop: 20.20.20.2 (GigabitEthernet0/4/7)
       TI-LFA node/SRLG-protecting, SRLG-protecting
       SRGB: 17000, range: 7000 prefix-SID index: 333, R:0 N:1 P:0 E:0 V:0 L:0
       label: 17333
       P node: R3-RSP2-2013[4.4.4.4], label: 17221
       repair source: R5-ASR920-2012, LSP 4
```

Use the command **show isis topo flex-algo** *value* for information regarding the IS-IS paths to intermediate systems.

```
show isis topo flex-algo 129
Tag 1:
IS-IS TID 0 paths to level-1 routers
Flex-algo 129
System Id            Metric     Next-Hop             Interface   SNPA
920_1                3          RSP2_2               Gi0/15/0    e8ed.f3b8.f804
RSP3_R1              **
RSP2_1               2          RSP2_2               Gi0/15/0    e8ed.f3b8.f804
RSP3_R2              **
RSP2_2               1          RSP2_2               Gi0/15/0    e8ed.f3b8.f804
RSP3_R3              --
```

Use the command **show isis fast-reroute ti-lfa tunnel** for information regarding the IS-IS TI-LFA tunnels.

```
show isis fast-reroute ti-lfa tunnel
Tag null:
Fast-Reroute TI-LFA Tunnels:
Tunnel  Interface  Next Hop         End Point        Label     End Point Host
Tag 1:
Fast-Reroute TI-LFA Tunnels:

Tunnel  Interface  Next Hop         End Point        Label     End Point Host

MP2     Gi0/0/6    12.12.12.2       2.2.2.2          17019     RSP3_R3

MP5     Gi0/0/5    11.11.11.2       2.2.2.2          17019     RSP3_R3

MP3     Gi0/0/6    12.12.12.2       6.6.6.6          17333     RSP2_2

                                    2.2.2.2          16        RSP3_R3

MP9     Gi0/0/5    11.11.11.2       2.2.2.2          17039     RSP3_R3
```

```
MP1     Gi0/0/6    12.12.12.2       6.6.6.6          20333     RSP2_2
                                    2.2.2.2          16        RSP3_R3
MP6     Gi0/0/5    11.11.11.2       2.2.2.2          17049     RSP3_R3
```

Use the command **show isis flex-algo** [*flex-algorithm-id*] to display the configured flexible algorithm and the defined parameters.

```
R1#show isis flex-algo 255
Tag Ring#1:
IS-IS Flex-Algo Database
Flex-Algo count: 1

Flex-Algo 255:
    IS-IS Level-1
      Definition Priority: 11
      Definition Source: PE23.00
      Definition Equal to Local: Yes
      Definition Metric Type: IGP
      Definition Include-any Affinity:
       0x00000000 0x00000000 0x00000002
      Definition Flex-Algo Prefix Metric: No
      Disabled: No
      Microloop Avoidance Timer Running: No
    IS-IS Level-2
      Definition Priority: 11
      Definition Source: PE23.00
      Definition Equal to Local: Yes
      Definition Metric Type: IGP
      Definition Include-any Affinity:
       0x00000000 0x00000000 0x00000002
      Definition Flex-Algo Prefix Metric: No
      Disabled: No
      Microloop Avoidance Timer Running: No
   Local Priority: 11
   FRR Disabled: No
   Microloop Avoidance Disabled: No
```

Use the command **show ip ospf topology summary internal** to display instance level and area level information.

show ip ospf [*process-id*] [*area-id*] **topology summary** [**algorithm***algorithm-number*] [**internal**]

```
R1#show ip ospf topology summary internal
        Process OSPF-10

 Instance global
  Router ID       : 10.0.0.1
  Number of Areas : 2
  Number of Algos : 2
  Max Path count  : 4
  Route count     : 4
  Algo Status
   Algo 128 : In-use
   Algo 129 : In-use
  Route Work Queue Sizes
   Critical : 0, High     : 0
   Medium   : 0, Low      : 0
  SR Global Block : 16000 - 24000
   Area
    Number of Nodes : 8
Algo 128
```

```
      FAD Advertising Router : 10.0.0.1
      FAD Area ID : 0
       Algo Type   : 0
       Metric Type : 0
       Include Any Affinity : 0x00000002
      SPF count : 10
      SPF Trigger Flags : Valid, Explicit FAD
     Algo 129
      FAD Advertising Router : 10.0.0.1
      FAD Area ID : 0
       Algo Type   : 0
       Metric Type : 0
       Exclude Affinity : 0x00000002
      SPF count : 11
      SPF Trigger Flags : Valid, Explicit FAD
   Area
    Number of Nodes : 2
     Algo 128
      FAD Advertising Router : 10.0.0.1
      FAD Area ID : 1
       Algo Type   : 0
       Metric Type : 0
       Include Any Affinity : 0x00000002
      SPF count : 4
      SPF Trigger Flags : Valid, Explicit FAD
     Algo 129
      FAD Advertising Router : 10.0.0.1
      FAD Area ID : 1
       Algo Type   : 0
       Metric Type : 0
       Exclude Affinity : 0x00000002
      SPF count : 4
      SPF Trigger Flags : Valid, Explicit FAD
```

Use the command **show ip ospf topology** to display the node and link information compiled from the LSAs.

show ip ospf [*process-id*] [*area-id*] **topology** [**algorithm***algorithm-number*] [**node-id** *router-id*] [**link-address***link-address*] [**detail**]

```
R1#show ip ospf topology
        Process OSPF-10

  Instance : global
   Router ID : 10.0.0.1
    Area :  (8 nodes)
      Node : 1.2.0.2 (pseudo) (2 links)
        Link : 10.0.0.1 0.0.0.0 Transit
        Link : 1.1.1.2 0.0.0.0 Transit
      Node : 10.0.0.1 (root) (3 links) ABR
        Algos supported: 128, 129
        Flex Algo Definition: 128
        Flex Algo Definition: 129
        Link : 1.1.1.6 0.0.0.2 Point-to-point
        Link : 1.1.1.6 1.6.1.1 Point-to-point
        Link : 1.2.0.2 1.2.0.1 Transit
      Node : 1.1.1.2  (3 links)
        Algos supported: 128
        Link : 1.1.1.3 2.3.0.2 Point-to-point
        Link : 1.1.1.54 2.5.0.2 Point-to-point
        Link : 1.2.0.2 1.2.0.2 Transit
      Node : 1.1.1.3  (2 links)
        Algos supported: 128
        Link : 1.1.1.2 2.3.0.3 Point-to-point
        Link : 1.1.1.4 3.4.0.3 Point-to-point
      Node : 1.1.1.4  (3 links) ABR, ASBR
```

```
        Algos supported: 128, 129
        Link : 1.1.1.3 3.4.0.4 Point-to-point
        Link : 1.1.1.9 0.0.0.3 Point-to-point
        Link : 1.1.1.54 4.5.0.4 Point-to-point
      Node : 1.1.1.6  (4 links)
        Algos supported: 129
        Link : 10.0.0.1 0.0.0.2 Point-to-point
        Link : 10.0.0.1 1.6.1.6 Point-to-point
        Link : 1.1.1.54 5.6.0.6 Point-to-point
        Link : 1.1.1.54 5.6.1.6 Point-to-point
      Node : 1.1.1.9  (1 links) ABR
        Link : 1.1.1.4 0.0.0.3 Point-to-point
      Node : 1.1.1.54  (4 links)
        Algos supported: 129
        Link : 1.1.1.2 2.5.0.5 Point-to-point
        Link : 1.1.1.4 4.5.0.5 Point-to-point
        Link : 1.1.1.6 5.6.0.5 Point-to-point
        Link : 1.1.1.6 5.6.1.5 Point-to-point
    Area :  (2 nodes)
      Node : 10.0.0.1 (root) (1 links) ABR
        Algos supported: 128, 129
        Flex Algo Definition: 128
        Flex Algo Definition: 129
        Link : 1.1.1.8 1.8.0.1 Point-to-point
      Node : 1.1.1.8  (1 links)  ASBR
        Link : 10.0.0.1 1.8.0.8 Point-to-point

R1#show ip ospf topo node-id 10.0.0.1 de
        Process OSPF-10

  Instance : global
   Router ID : 10.0.0.1
    Area :  (8 nodes)
      Node : 10.0.0.1 (root) (3 links) ABR
       No. of algos : 2
        Algos supported: 128, 129
       SRGB List:
        16000 - 24000
       Oper Flags : 0x00
       Contributing LSAs : Router, Router Info

        Flex Algo Definition: 128
         Algo Type   : 0
         Metric Type : 0
         Priority    : 10
         Include Any Affinity : 0x00000002
         Oper Flags  : 0x00

        Flex Algo Definition: 129
         Algo Type   : 0
         Metric Type : 0
         Priority    : 128
         Exclude Affinity : 0x00000002
         Oper Flags  : 0x00

        Link : 1.1.1.6 0.0.0.2 Point-to-point
         Metric     : 10
         Remote link: 10.0.0.1 0.0.0.2 Point-to-point
         Oper Flags : 0x00
         Link Attributes:
          LSID     : 8.0.0.2
          Local-Remote Interface ID : (2, 2)
          SR TE Metric : 40
         Contributing LSAs : Router, Ext. Link
```

```
     Link : 1.1.1.6 1.6.1.1 Point-to-point
      Metric     : 10
      Subnet mask: 255.255.255.0
      Remote link: 10.0.0.1 1.6.1.6 Point-to-point
      Oper Flags : 0x00
      Link Attributes:
       LSID     : 8.0.0.5
       Local-Remote Interface ID : (5, 5)
       Remote IPv4 Address: 1.6.1.6
       SR TE Metric : 40
       Number of SRLGs  : (2)
         [1]:  1           [2]:  6
      Contributing LSAs : Router, Ext. Link

     Link : 1.2.0.2 1.2.0.1 Transit
      Metric     : 10
      Remote link: 10.0.0.1 0.0.0.0 Transit
      Oper Flags : 0x00
      Link Attributes:
       LSID     : 8.0.0.3
       SR TE Metric : 40
       Ext. Admin Group: 0x00000002
      Contributing LSAs : Router, Ext. Link

Area :  (2 nodes)
  Node : 10.0.0.1 (root) (1 links) ABR
   No. of algos : 2
    Algos supported: 128, 129
   SRGB List:
    16000 - 24000
   Oper Flags : 0x00
   Contributing LSAs : Router, Router Info

    Flex Algo Definition: 128
     Algo Type   : 0
     Metric Type : 0
     Priority    : 10
     Include Any Affinity : 0x00000002
     Oper Flags  : 0x00

    Flex Algo Definition: 129
     Algo Type   : 0
     Metric Type : 0
     Priority    : 128
     Exclude Affinity : 0x00000002
     Oper Flags  : 0x00

    Link : 1.1.1.8 1.8.0.1 Point-to-point
     Metric     : 10
     Subnet mask: 255.255.255.0
     Remote link: 10.0.0.1 1.8.0.8 Point-to-point
     Oper Flags : 0x00
     Link Attributes:
      LSID     : 8.0.0.4
      Local-Remote Interface ID : (4, 4)
      Remote IPv4 Address: 1.8.0.8
      SR TE Metric : 20
      Min Delay    : 499
     Contributing LSAs : Router, Ext. Link
```

Use the command **show ip ospf topology prefix** display the node and prefix information compiled from the LSAs.

show ip ospf [*process-id*] [*area-id*] **topology prefix** [**algorithm***algorithm-number*] [**node-id** *router-id*] [*prefix/mask*] [**detail**]

```
R1#show ip ospf topology prefix
        Process OSPF-10

  Instance : global
   Router ID : 10.0.0.1
    Area :  (8 nodes)
      Node : 1.2.0.2 (pseudo) (2 links)
      Node : 10.0.0.1 (root) (3 links) ABR
        Algos supported: 128, 129
        Flex Algo Definition: 128
        Flex Algo Definition: 129
      Node : 1.1.1.2  (3 links)
        Algos supported: 128
      Node : 1.1.1.3  (2 links)
        Algos supported: 128
        Prefix : 1.1.1.34/32
      Node : 1.1.1.4  (3 links) ABR, ASBR
        Algos supported: 128, 129
        Prefix : 1.1.1.4/32
        Prefix : 1.1.1.34/32
        Prefix : 1.1.1.45/32
      Node : 1.1.1.6  (4 links)
        Algos supported: 129
      Node : 1.1.1.9  (1 links) ABR
      Node : 1.1.1.54  (4 links)
        Algos supported: 129
        Prefix : 1.1.1.54/32
    Area :  (2 nodes)
      Node : 10.0.0.1 (root) (1 links) ABR
        Algos supported: 128, 129
        Flex Algo Definition: 128
        Flex Algo Definition: 129
      Node : 1.1.1.8  (1 links)  ASBR

R1#show ip ospf topo prefix 1.1.1.4/32 de
        Process OSPF-10

  Instance : global
   Router ID : 10.0.0.1
    Area :  (8 nodes)
      Node : 1.2.0.2 (pseudo) (2 links)
       No. of algos : 0
       Oper Flags : 0x00
       Contributing LSAs : Network

      Node : 10.0.0.1 (root) (3 links) ABR
       No. of algos : 2
        Algos supported: 128, 129
       SRGB List:
        16000 - 24000
       Oper Flags : 0x00
       Contributing LSAs : Router, Router Info

        Flex Algo Definition: 128
         Algo Type   : 0
         Metric Type : 0
         Priority    : 10
         Include Any Affinity : 0x00000002
         Oper Flags  : 0x00

        Flex Algo Definition: 129
```

```
    Algo Type   : 0
    Metric Type : 0
    Priority    : 128
    Exclude Affinity : 0x00000002
    Oper Flags  : 0x00

Node : 1.1.1.2  (3 links)
 No. of algos : 1
  Algos supported: 128
 SRGB List:
  16000 - 24000
 Oper Flags : 0x00
 Contributing LSAs : Router, Router Info

Node : 1.1.1.3  (2 links)
 No. of algos : 1
  Algos supported: 128
 SRGB List:
  16000 - 24000
 Oper Flags : 0x00
 Contributing LSAs : Router, Router Info

Node : 1.1.1.4  (3 links) ABR, ASBR
 No. of algos : 2
  Algos supported: 128, 129
 SRGB List:
  16000 - 30001
 Oper Flags : 0x00
 Contributing LSAs : Router, Router Info

  Prefix : 1.1.1.4/32
   Metric : 1
   Route Type : Intra

    Algo 128 SID 132
     Total Metric : 31
     Contrib Flags : Inuse, Reachable
     SID Flags : Exp Null, PHP off, Index, Global, Node, Valid

    Algo 129 SID 133
     Total Metric : 31
     Contrib Flags : Inuse, Reachable
     SID Flags : Exp Null, PHP off, Index, Global, Node, Valid

Node : 1.1.1.6  (4 links)
 No. of algos : 1
  Algos supported: 129
 SRGB List:
  16000 - 24000
 Oper Flags : 0x00
 Contributing LSAs : Router, Router Info

Node : 1.1.1.9  (1 links) ABR
 No. of algos : 0
 SRGB List:
  16000 - 24000
 Oper Flags : 0x00
 Contributing LSAs : Router, Router Info

Node : 1.1.1.54  (4 links)
 No. of algos : 1
  Algos supported: 129
 SRGB List:
  16000 - 24000
```

```
      Oper Flags : 0x00
      Contributing LSAs : Router, Router Info

    Area :  (2 nodes)
      Node : 10.0.0.1 (root) (1 links) ABR
       No. of algos : 2
        Algos supported: 128, 129
       SRGB List:
        16000 - 24000
       Oper Flags : 0x00
       Contributing LSAs : Router, Router Info

        Flex Algo Definition: 128
         Algo Type   : 0
         Metric Type : 0
         Priority    : 10
         Include Any Affinity : 0x00000002
         Oper Flags  : 0x00

        Flex Algo Definition: 129
         Algo Type   : 0
         Metric Type : 0
         Priority    : 128
         Exclude Affinity : 0x00000002
         Oper Flags  : 0x00

      Node : 1.1.1.8  (1 links)  ASBR
       No. of algos : 0
       SRGB List:
        16000 - 24000
       Oper Flags : 0x00
       Contributing LSAs : Router, Router Info
```

Use the command **show ip ospf topology path** display the path information of nodes computed by SPF.

show ip ospf [*process-id*] [*area-id*] **topology path** [**algorithm***algorithm-number*] [**node-id** *router-id*] [**detail**]

```
R1#show ip ospf topology path
       Process OSPF-10

  Instance global
   Router ID : 10.0.0.1
    Area
      Node 1.1.1.2
        Algo 128, Distance 10
          1.2.0.2, from 1.1.1.2, via Ethernet0/1
      Node 1.1.1.3
        Algo 128, Distance 20
          1.2.0.2, from 1.1.1.2, via Ethernet0/1
      Node 1.1.1.4  ABR, ASBR
        Algo 128, Distance 30
          1.2.0.2, from 1.1.1.2, via Ethernet0/1
        Algo 129, Distance 30
          1.1.1.6, from 1.1.1.6, via Ethernet0/0
          1.6.1.6, from 1.1.1.6, via Ethernet0/3
      Node 1.1.1.6
        Algo 129, Distance 10
          1.1.1.6, from 1.1.1.6, via Ethernet0/0
          1.6.1.6, from 1.1.1.6, via Ethernet0/3
      Node 1.1.1.54
        Algo 129, Distance 20
          1.1.1.6, from 1.1.1.6, via Ethernet0/0
          1.6.1.6, from 1.1.1.6, via Ethernet0/3
```

```
R1#show ip ospf topo path node-id 1.1.1.4 de
        Process OSPF-10

  Instance global
   Router ID : 10.0.0.1
    Area
      Node 1.1.1.4  ABR, ASBR
        Algo 128, Distance 30
         Flags : Reachable

          1.2.0.2, from 1.1.1.2, via Ethernet0/1
           Weight : 0

        Algo 129, Distance 30
         Flags : Reachable

          1.1.1.6, from 1.1.1.6, via Ethernet0/0
           Weight : 0

          1.6.1.6, from 1.1.1.6, via Ethernet0/3
           Weight : 0
```

Use the command **show ip ospf topology route** display the path information of routes computed based on route calculation.

show ip ospf [*process-id*] [*area-id*] **topology route** [**algorithm***algorithm-number*] [**prefix/mask**] [**detail**]

```
R1#show ip ospf topology route
 Route Table of OSPF-10 with router ID 10.0.0.1 (VRF global)

  1.1.1.4/32
    Algo 128, Metric 31, SID 132, Label 16132
      1.2.0.2, from 1.1.1.2, via Ethernet0/1
    Algo 129, Metric 31, SID 133, Label 16133
      1.1.1.6, from 1.1.1.6, via Ethernet0/0
      1.6.1.6, from 1.1.1.6, via Ethernet0/3
  1.1.1.34/32
    Algo 128, Metric 21, SID 43, Label 16043
      1.2.0.2, from 1.1.1.2, via Ethernet0/1
  1.1.1.45/32
    Algo 129, Metric 31, SID 4294967295, Label 1048577
      1.1.1.6, from 1.1.1.6, via Ethernet0/0
      1.6.1.6, from 1.1.1.6, via Ethernet0/3
  1.1.1.54/32
    Algo 129, Metric 21, SID 45, Label 16045
      1.1.1.6, from 1.1.1.6, via Ethernet0/0
      1.6.1.6, from 1.1.1.6, via Ethernet0/3

R1#show ip ospf topo route 1.1.1.4/32 de
 Route Table of OSPF-10 with router ID 10.0.0.1 (VRF global)

  Route entry for 1.1.1.4/32
   Priority : Medium
   Flags : New

    Algo 128, Metric 31, SID 132, Label 16132
     Route type : Intra
     Flags: Inuse
     Prefix Contrib Algo 128 SID 132
      Total Metric : 31
      Contrib Flags : Inuse, Reachable
      SID Flags : Exp Null, PHP off, Index, Global, Node, Valid
```

```
  Path: 1.2.0.2, from 1.1.1.2, via Ethernet0/1
   Out Label  : 16132
   Weight     : 0

Algo 129, Metric 31, SID 133, Label 16133
 Route type : Intra
 Flags: Inuse
 Prefix Contrib Algo 129 SID 133
  Total Metric : 31
  Contrib Flags : Inuse, Reachable
  SID Flags : Exp Null, PHP off, Index, Global, Node, Valid

  Path: 1.1.1.6, from 1.1.1.6, via Ethernet0/0
   Out Label  : 16133
   Weight     : 0

  Path: 1.6.1.6, from 1.1.1.6, via Ethernet0/3
   Out Label  : 16133
   Weight     : 0
```