Configure IOS XR Traffic Controller (XTC)

IOS XR Traffic Controller (XTC) provides stateful path computation element (PCE) functionality by extending the existing IOS-XR PCE functionality with additional capabilities. The current IOS-XR PCE function is part of MPLS-TE which requires an MPLS package. XTC removes this restriction, making PCE functionality part of the base IOS-XR package. XTC is supported on the MPLS data plane and IPv4 control plane.

Note To install XTC, you need to install an instance of Cisco IOS XRv 9000 Router. Refer to the Cisco IOS XRv 9000 Router Installation and Configuration Guide for more information.

About IOS XR Traffic Controller (XTC)

The path computation element (PCE) describes a set of procedures by which a path computation client (PCC) can report and delegate control of head-end tunnels sourced from the PCC to a PCE peer. The PCE peer can request the PCC to update and modify parameters of label switched paths (LSPs) it controls. The stateful model also enables a PCC to allow the PCE to initiate computations allowing the PCE to perform network-wide orchestration.

Note For more information on PCE, PCC, and PCEP, refer to the Path Computation Element section in the *MPLS Configuration Guide for Cisco ASR 9000 Series Routers*.

XTC learns topology information by way of IGP (OSPF or IS-IS) through BGP-LS.

XTC is capable of computing paths using the following methods:

- TE metric—XTC uses the TE metric in its path calculations to optimize latency.
- IGP metric—XTC uses the IGP metric in its path calculations to optimize reachability.
- Disjointness—XTC uses the disjoint policy to compute two list of segments that steer traffic from two source nodes to two destination nodes along disjoint paths. The disjoint paths can originate from the

same head-end or different head-ends. Disjoint level refers to the type of resources that should not be shared by the two computed paths. XTC supports the following disjoint path computations:

- Link – Specifies that links are not shared on the computed paths.
- Node – Specifies that nodes are not shared on the computed paths.
- SRLG – Specifies that links with the same SRLG value are not shared on the computed paths.
- SRLG-node – Specifies that SRLG and nodes are not shared on the computed paths.

When the first request is received with a given disjoint-group ID, a list of segments is computed, encoding the shortest path from the first source to the first destination. When the second request is received with the same disjoint-group ID, information received in both requests is used to compute two disjoint paths: one path from the first source to the first destination, and another path from the second source to the second destination. Both paths are computed at the same time. The shortest lists of segments is calculated to steer traffic on the computed paths.

Configure PCE

This task explains how to configure PCE.

Before You Begin

Optionally install and configure an instance of Cisco IOS XRv 9000 Router.

SUMMARY STEPS

1. **configure**
2. **pce**
3. **address ipv4** *address*
4. **state-sync ipv4** *address*
5. **tcp-buffer** *size*
6. **password** {**clear** | **encrypted**} *password*
7. **segment-routing** {**strict-sid-only** | **te-latency**}
8. **timers**
9. **keepalive** *time*
10. **minimum-peer-keepalive** *time*
11. **reoptimization** *time*
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose
Step 2	**pce** **Example:** `RP/0/RSP0/CPU0:router(config)# pce`	Enables PCE and enters PCE configuration mode.
Step 3	**address ipv4** *address* **Example:** `RP/0/RSP0/CPU0:router(config-pce)# address ipv4 192.168.0.1`	Configures a PCE IPv4 address.
Step 4	**state-sync ipv4** *address* **Example:** `RP/0/RSP0/CPU0:router(config-pce)# state-sync ipv4 192.168.0.3`	Configures the remote peer for state synchronization.
Step 5	**tcp-buffer** *size* **Example:** `RP/0/RSP0/CPU0:router(config-pce)# tcp-buffer 1024000`	Configures the transmit and receive TCP buffer size for each PCEP session, in bytes. The default buffer size is 256000. The valid range is from 204800 to 1024000.
Step 6	**password** {**clear** | **encrypted**} *password* **Example:** `RP/0/RSP0/CPU0:router(config-pce)# password encrypted pwd1`	Enables TCP authentication for all PCEP peers. Any TCP segment coming from the PCC that does not contain a MAC matching the configured password will be rejected. Specify if the password is encrypted or clear text.
Step 7	**segment-routing** {**strict-sid-only** | **te-latency**} **Example:** `RP/0/RSP0/CPU0:router(config-pce)# segment-routing strict-sid-only`	Configures the segment routing algorithm to use strict SID or TE latency. **Note** This setting is global and applies to all LSPs that request a path from this controller.
Step 8	**timers** **Example:** `RP/0/RSP0/CPU0:router(config-pce)# timers`	Enters timer configuration mode.

	Command or Action	Purpose
Step 9	**keepalive** *time* **Example:** `RP/0/RSP0/CPU0:router(config-pce-timers)#` **`keepalive 60`**	Configures the timer value for locally generated keep-alive messages. The default time is 30 seconds.
Step 10	**minimum-peer-keepalive** *time* **Example:** `RP/0/RSP0/CPU0:router(config-pce-timers)#` **`minimum-peer-keepalive 30`**	Configures the minimum acceptable keep-alive timer that the remote peer may propose in the PCEP OPEN message during session establishment. The default time is 20 seconds.
Step 11	**reoptimization** *time* **Example:** `RP/0/RSP0/CPU0:router(config-pce-timers)#` **`reoptimization 30`**	Configures the re-optimization timer. The default timer is 60 seconds.
Step 12	**exit** **Example:** `RP/0/RSP0/CPU0:router(config-pce-timers)#` **`exit`**	Exits timer configuration mode and returns to PCE configuration mode.

Configure the Disjoint Policy (Optional)

This task explains how to configure a disjoint policy on the PCE.

SUMMARY STEPS

1. **disjoint-path**
2. **group-id** *value* **type** {**link** | **node** | **srlg** | **srlg-node**} [**sub-id** *value*]
3. **strict**
4. **lsp** {**1** | **2**} **pcc ipv4** *address* **lsp-name** *lsp_name* [**shortest-path**]

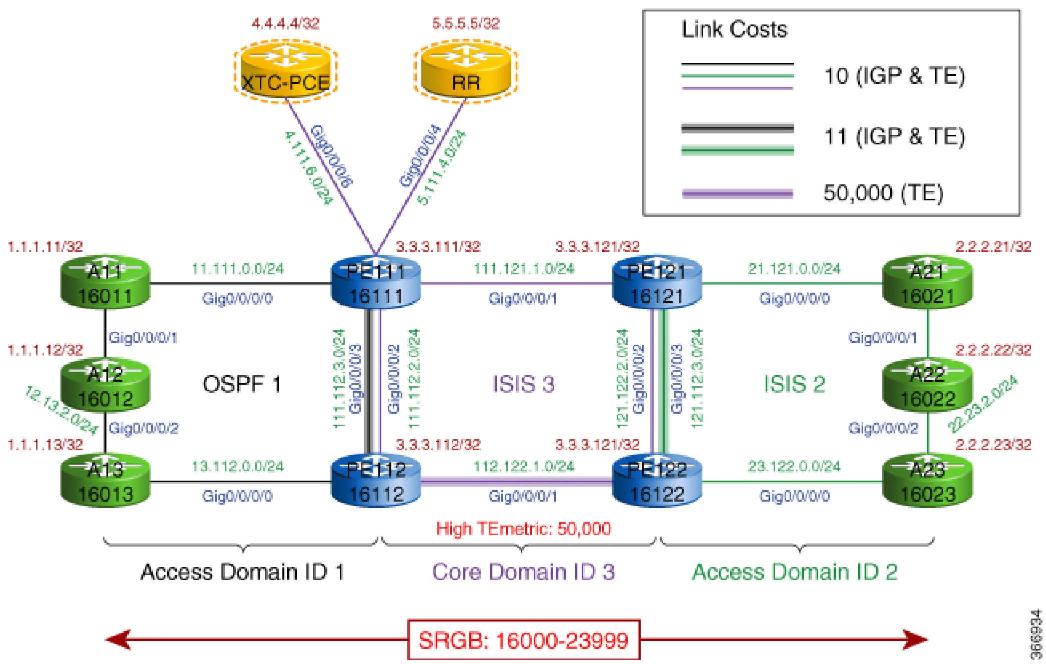
DETAILED STEPS

	Command or Action	Purpose
Step 1	**disjoint-path** **Example:** `RP/0/RSP0/CPU0:router(config-pce)#` **`disjoint-path`**	Enters disjoint configuration mode.
Step 2	**group-id** *value* **type** {**link** | **node** | **srlg** | **srlg-node**} [**sub-id** *value*] **Example:** `RP/0/RSP0/CPU0:router(config-pce-disjoint)#` **`group-id 1 type node sub-id 1`**	Configures the disjoint group ID and defines the preferred level of disjointness (the type of resources that should not be shared by the two paths): • **link—**Specifies that links are not shared on the computed paths. • **node—**Specifies that nodes are not shared on the computed paths. • **srlg—**Specifies that links with the same SRLG value are not shared on the computed paths. • **srlg-node—**Specifies that SRLG and nodes are not shared on the computed paths. If a pair of paths that meet the requested disjointness level cannot be found, then the paths will automatically fallback to a lower level: • If the requested disjointness level is SRLG or node, then link-disjoint paths will be computed. • If the requested disjointness level was link, or if the first fallback from SRLG or node disjointness failed, then the lists of segments encoding two shortest paths, without any disjointness constraint, will be computed.
Step 3	**strict** **Example:** `RP/0/RSP0/CPU0:router(config-pce-disjoint)#` **`strict`**	(Optional) Prevents the automatic fallback behavior of the preferred level of disjointness. If a pair of paths that meet the requested disjointness level cannot be found, the disjoint calculation terminates and no new path is provided. The existing path is not modified.
Step 4	**lsp** {**1** | **2**} **pcc ipv4** *address* **lsp-name** *lsp_name* [**shortest-path**] **Example:** `RP/0/RSP0/CPU0:router(config-pce-disjoint)#` **`lsp 1 pcc ipv4 192.168.0.1 lsp-name rtrA_t1 shortest-path`** `RP/0/RSP0/CPU0:router(config-pce-disjoint)#` **`lsp 2 pcc ipv4 192.168.0.5 lsp-name rtrE_t2`**	Adds LSPs to the disjoint group. The **shortest-path** keyword forces one of the disjoint paths to follow the shortest path from the source to the destination. This option can only be applied to the the first LSP specified.

Configuration Examples for XTC

The XTC configuration examples in this section use the following multi-domain network diagram.

Figure 1: Multi-Domain Network Diagram



The network is divided into three domains. Each node has been assigned a Prefix-SID from the default SRGB. Nodes in different access domains do not have connectivity to each other. XTC-PCE is the centralized PCE for the entire network.

Note Before you begin, configure BGP-LS address family redistribution on the core routers. This example is for PE111; similar configuration must be applied to the other core routers

```
router bgp 1
 bgp router-id 3.3.3.111
 address-family ipv4 unicast
  redistribute ospf 1 route-policy loop
 !
 address-family link-state link-state
 !
 neighbor 4.4.4.4
  remote-as 1
  update-source Loopback0
  address-family link-state link-state
  !
 !
!
route-policy loop
  if destination in (3.3.3.0/32 ge 24) then
    drop
  endif
  if destination in (0.0.0.0/0 ge 32) then
    set aigp-metric igp-cost
    pass
  else
    drop
  endif
end-policy
!
commit
end
```

Setup XTC: Example

Complete these tasks on the PCE to setup and enable XTC.

Task Number	Task Description	Sample Configuration	Details

1	Configure IGP. This step is required for reachability to the BGP-LS peers which use Loopbacks for peering.	`conf t`
		`!`
		`router isis 3`
		 `is-type level-2-only`
		 `net 49.0001.0000.0000.0004.00`
		 `address-family ipv4 unicast`
		  `metric-style wide`
		 `!`
		 `interface Loopback0`
		  `address-family ipv4 unicast`
		  `!`
		 `!`
		 `interface GigabitEthernet0/0/0/6`
		  `point-to-point`
		  `address-family ipv4 unicast`
		  `!`
		 `!`
		`!`
		`commit`
		`end`
			Enabling Segment Routing for IS-IS Protocol
			
			Enabling Segment Routing for OSPF Protocol

2	Configure BGP link-state (BGP-LS). XTC learns topology information through BGP-LS. This step configures BGP to receive the topology information.	`conf t`
		`!`
		`router bgp 1`
		 `bgp router-id 4.4.4.4`
		 `address-family ipv4 unicast`
		  `table-policy fib`
		 `!`
		 `address-family link-state link-state`
		 `!`
		 `neighbor-group epn`
		  `remote-as 1`
		  `update-source Loopback0`
		  `!`
		  `address-family link-state link-state`
		
		  `!`
		 `!`
		 `neighbor 3.3.3.111`
		  `use neighbor-group epn`
		 `!`
		 `neighbor 3.3.3.112`
		  `use neighbor-group epn`
		 `!`
		 `neighbor 3.3.3.121`
		  `use neighbor-group epn`
		 `!`
		 `neighbor 3.3.3.122`
		  `use neighbor-group epn`
		 `!`
		`!`
		`commit`
		`end`
			Configure BGP Link-State

3	Configure static routing for reachability to the PCC clients. The network information is downloaded to the RIB for actual traffic forwarding.		

Task Number	Task Description	Sample Configuration	Details

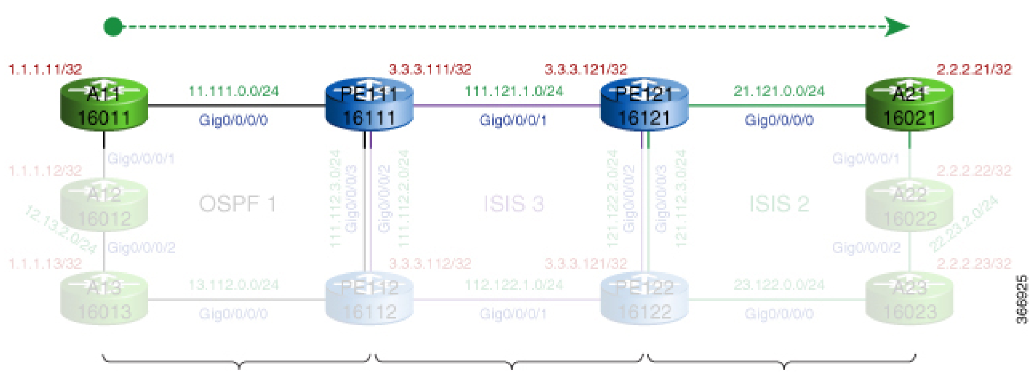
		`conf t`
		`!`
		`router static`
		` address-family ipv4 unicast`
		`  1.1.1.0/24 GigabitEthernet0/0/0/6 4.111.6.111`
		`  2.2.2.0/24 GigabitEthernet0/0/0/6 4.111.6.111`
		`!`
		`commit`
		`end`	

4	Configure PCE server.	`conf t`
		`!`
		`pce`
		` address ipv4 4.4.4.4`
		`!`
		`commit`
		`end`
			Configure PCE, *on page 2*

Centralized Inter-Domain Reachability Optimization: Example

The following example shows how to build a policy from A11 to A21 in which XTC uses the IGP metric in its path calculations. Since these are inter-domain paths without any contiguous IGP between them, there is no path currently from A11 to A21. However, if we add up costs along the links (assuming all links are equal cost), the shortest path would be:

A11 → PE111 → PE121 → A21



Configurations on Node A11

Complete these tasks on A11 to use the IGP metric for path calculations.

Task Number	Task Description	Sample Configuration

1	Configure the PCE session by specifying the XTC server in the **mpls traffic-eng** section. The source IP is typically the Loopback0 interface IP. This is a stateful connection and will stay connected to the XTC node by way of TCP.	<pre>conf t
		!
		mpls traffic-eng
		 pce
		  peer source ipv4 1.1.1.11
		  peer ipv4 4.4.4.4
		  !
		  segment-routing
		  stateful-client
		!
		commit
		end</pre>

2	Configure the tunnel interface and set the metric as **igp** for the PCE computation. When the policy path option attribute is set to use **pce**, XR knows to contact PCE to get the path.	<pre>conf t
		!
		interface tunnel-te100
		 ipv4 unnumbered Loopback0
		 destination 2.2.2.21
		 path-selection
		  metric igp
		 !
		 path-option 1 dynamic pce segment-routing
		!
		commit
		end</pre>

3	Configure static routing to push traffic that is destined to the remote PE Loopback0 interface IP using the newly created SR-TE tunnel interface.	<pre>conf t
		!
		router static
		 address-family ipv4 unicast
		  2.2.2.21/32 tunnel-te100
		 !
		!
		commit
		end</pre>

Verify the Configuration on Node A11

Verify the configuration by checking the status of the peering session with XTC, the policy status and the path computed by XTC along with the SID-list and label-stack, and the routing entry for the destination.

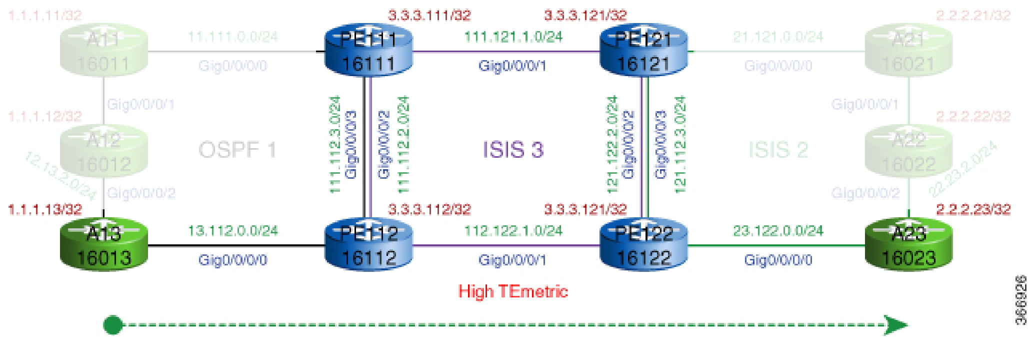
```
RP/0/0/CPU0:A11# show mpls traffic-eng pce peer
        Address   Precedence        State         Learned From
--------------- ------------ ------------ --------------------
        4.4.4.4          255           Up        Static config

RP/0/0/CPU0:A11# show mpls traffic-eng pce tunnels
Tunnel : tunnel-te100
    Destination : 2.2.2.21
    State : up
    Current path option: 1, path learned from PCE 4.4.4.4
    Admin Weight : 21
    Hop Count    : 3

RP/0/0/CPU0:A11# show mpls traffic-eng tunnels 100

Name: tunnel-te100  Destination: 2.2.2.21  Ifhandle:0xf0
  Signalled-Name: A11_t100
```

```
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 1, (Segment-Routing) type dynamic pce (Basis for Setup, path weight 21)
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Fri May 19 19:24:47 2017 (3d19h ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: IGP (interface)
    Path Selection:
      Tiebreaker: Min-fill (default)
      Protection: any (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Tunnel class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    SRLG discovery: Disabled
  History:
    Tunnel has been up for: 3d18h (since Fri May 19 20:43:06 UTC 2017)
    Current LSP:
      Uptime: 3d17h (since Fri May 19 21:23:24 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, has no S2Ls
        Date/Time: Tue May 23 14:23:26 UTC 2017 [00:30:45 ago]
    Prior LSP:
      ID: 2 Path Option: 1
      Removal Trigger: reoptimization completed

  Segment-Routing Path Info (PCE computed path)
    Segment0[Node]: 3.3.3.111, Label: 16111
    Segment1[Node]: 3.3.3.121, Label: 16121
    Segment2[Node]: 2.2.2.21, Label: 16021
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads

RP/0/0/CPU0:A11# show route 2.2.2.21

Routing entry for 2.2.2.21/32
  Known via "static", distance 1, metric 0 (connected)
  Installed May 19 20:43:06.746 for 3d18h
  Routing Descriptor Blocks
    directly connected, via tunnel-te100
      Route metric is 0, Wt is 1
  No advertising protos.
RP/0/0/CPU0:A11#
```

Centralized Inter-Domain TE-Metric Optimization: Example

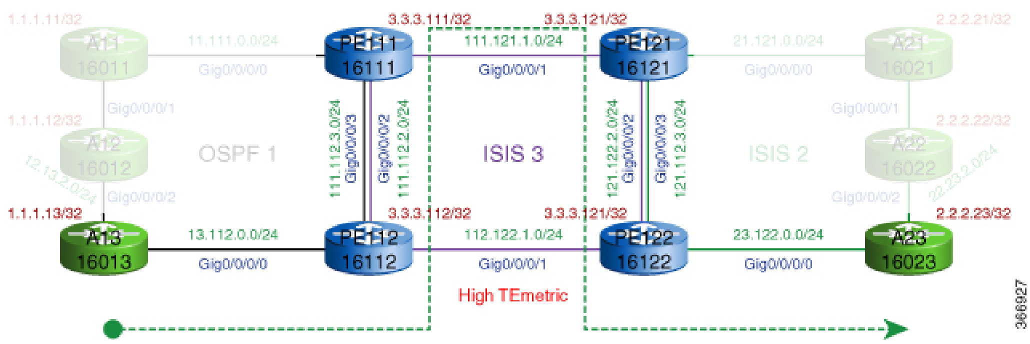
The following example shows how to build a policy from A13 to A23 in which XTC uses the TE metric in its path calculations. Since these are inter-domains paths without any contiguous IGP between them, there is no path currently from A13 to A23. Assuming all links have default TE metric, if we add up the TE metrics along the path, the shortest path would be:

A13 → PE112 → PE122 → A23



However, because the link between PE112 and PE122 has a higher TE metric, the policy is:

A13 → PE112 → PE111 → PE121 → PE122 → A23



Configurations on Node A13

Complete these tasks on A13 to use the TE metric for path calculations.

Task Number	Task Description	Sample Configuration

1	Configure the PCE session by specifying the XTC server in the **mpls traffic-eng** section. The source IP is typically the Loopback0 interface IP. This is a stateful connection and will stay connected to the XTC node by way of TCP.	`conf t`
		`!`
		`mpls traffic-eng`
		` pce`
		`  peer source ipv4 1.1.1.13`
		`  peer ipv4 4.4.4.4`
		`  !`
		`  segment-routing`
		`  stateful-client`
		`!`
		`commit`
		`end`

2	Configure the tunnel interface and set the metric as **te** for the PCE computation. When the policy path option attribute is set to use **pce**, XR knows to contact PCE to get the path.	`conf t`
		`!`
		`interface tunnel-te100`
		` ipv4 unnumbered Loopback0`
		` destination 2.2.2.23`
		` path-selection`
		`  metric te`
		` !`
		` path-option 1 dynamic pce segment-routing`
		`!`
		`commit`
		`end`

3	Configure static routing to push traffic that is destined to the remote PE Loopback0 interface IP using the newly created SR-TE tunnel interface.	`conf t`
		`!`
		`router static`
		` address-family ipv4 unicast`
		`  2.2.2.23/32 tunnel-te100`
		` !`
		`!`
		`commit`
		`end`

Verify the Configuration on Node A13

Verify the configuration on node A13 by checking the status of the peering session with XTC, the policy status and the path computed by XTC along with the SID-list and label-stack, and the routing entry for the destination.

```
RP/0/0/CPU0:A13# show mpls traffic-eng pce peer
        Address   Precedence        State          Learned From
--------------- ------------ ------------ --------------------
        4.4.4.4          255           Up        Static config

RP/0/0/CPU0:A13# show mpls traffic-eng pce tunnels
Tunnel : tunnel-te100
    Destination : 2.2.2.23
    State : up
    Current path option: 1, path learned from PCE 4.4.4.4
    Admin Weight : 21
    Hop Count    : 3

RP/0/0/CPU0:A13# show mpls traffic-eng tunnel 100

Name: tunnel-te100  Destination: 2.2.2.23  Ifhandle:0xb0
```

```
  Signalled-Name: A13_t100
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 1, (Segment-Routing) type dynamic pce (Basis for Setup, path weight 122)
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Fri May 19 19:23:58 2017 (3d20h ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (interface)
    Path Selection:
      Tiebreaker: Min-fill (default)
      Protection: any (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Tunnel class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    SRLG discovery: Disabled
  History:
    Tunnel has been up for: 00:01:17 (since Tue May 23 15:30:48 UTC 2017)
    Current LSP:
      Uptime: 00:01:17 (since Tue May 23 15:30:48 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, has no S2Ls
        Date/Time: Tue May 23 14:23:00 UTC 2017 [01:09:05 ago]
    Prior LSP:
      ID: 3773 Path Option: 1
      Removal Trigger: path tear

  Segment-Routing Path Info (PCE computed path)
    Segment0[Node]: 3.3.3.111, Label: 16111
    Segment1[Node]: 3.3.3.121, Label: 16121
    Segment2[Link]: 121.122.2.121 - 121.122.2.122, Label: 24004
    Segment3[Node]: 2.2.2.23, Label: 16023
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads

RP/0/0/CPU0:A13# show route 2.2.2.23

Routing entry for 2.2.2.23/32
  Known via "static", distance 1, metric 0 (connected)
  Installed May 33 15:30:48.537 for 00:01:17
  Routing Descriptor Blocks
    directly connected, via tunnel-te100
      Route metric is 0, Wt is 1
  No advertising protos.
RP/0/0/CPU0:A13#
```

Centralized LSP Node Disjointness TE-Metric Optimization: Example

The following example shows how to build policies from A11 to A21 and from A13 to A23 using the **node** disjoint-path type. Both policies use the same group ID (1) and source (0.0.0.1) to indicate that the policies are to be grouped together while performing path computations.

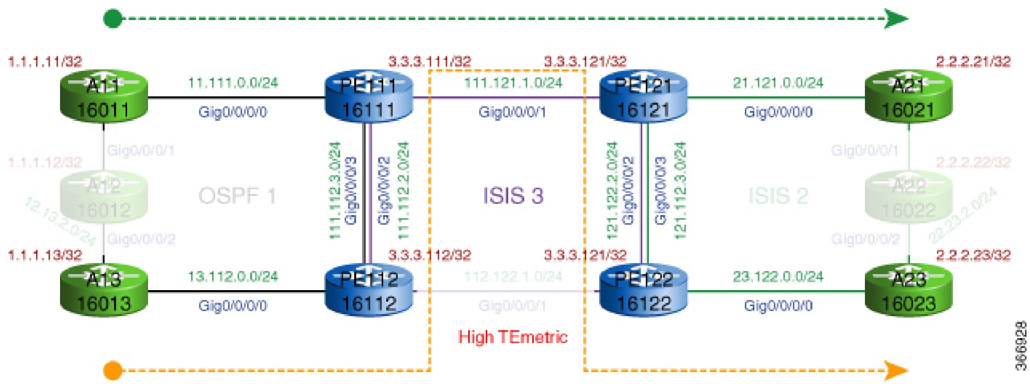
When XTC receives a request for the first policy A11 to A21, it checks the group ID and source. Since no policy exists with the same group ID and source combination, XTC calculates a dynamic path for this policy.

When XTC receives a request for the next policy A13 to A23, it again checks the group ID and source. Since XTC already has an existing policy with the same group ID and source combination, it calculates both policy paths, ensuring that there is no common node in the path. It then returns these paths to the head end routers.

Without the disjoint path feature, the paths of both policies share nodes PE111 and PE121 as the mid-point routers:

A11 → PE111 → PE121 → A21

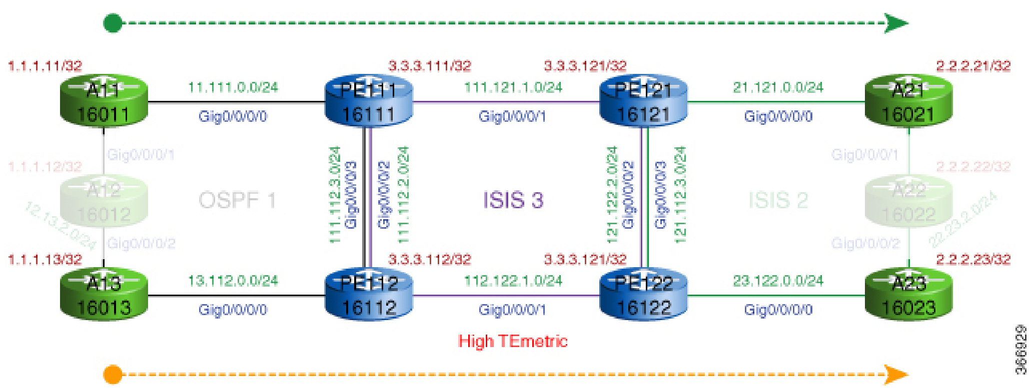
A13 → PE112 → PE111 → PE121 → PE122 → A23



However, using the disjoint-path feature, no nodes are shared in the paths:

A11 → PE111 → PE121 → A21

A13 → PE112 → PE122 → A23



Configurations on Node A11

Complete these tasks on node A11 to use node disjointness for path calculations.

<table>
<tr><th>Task Number</th><th>Task Description</th><th>Sample Configuration</th></tr>
<tr><td>1</td><td>Configure the PCE session by specifying the XTC server in the mpls traffic-eng section. The source IP is typically the Loopback0 interface IP. This is a stateful connection and will stay connected to the XTC node by way of TCP.

For the disjoint paths feature, we create an attribute-set with path-option. Specify pce and configure the disjoint-path type as node, group-id value as 1, and source as 0.0.0.1. The group-id and source are used to identify and group different policies across the network.</td><td><pre>
conf t
!
mpls traffic-eng
 pce
  peer source ipv4 1.1.1.11
  peer ipv4 4.4.4.4
  !
  segment-routing
  stateful-client
  !
 !
 auto-tunnel p2p
  tunnel-id min 1000 max 2000
 !
 attribute-set path-option PO.NODE.DISJ
  pce
   disjoint-path source 0.0.0.1 type node group-id 1
  !
 !
!
commit
end
</pre></td></tr>
<tr><td>2</td><td>Configure the tunnel interface and set the metric as te for the PCE computation.

Configure the path-option to use the attribute-set PO.NODE.DISJ (created in the previous step). This attribute-set also specified pce, so the PCE will provide the disjoint path.</td><td><pre>
conf t
!
interface tunnel-te200
 ipv4 unnumbered Loopback0
 destination 2.2.2.21
 path-selection
  metric te
 !
 path-option 1 dynamic segment-routing attribute-set
PO.NODE.DISJ
!
commit
end
</pre></td></tr>
</table>

Configurations on Node A13

Complete these tasks on node A13 to use node disjointness for path calculations.

<table>
<tr><th>Task Number</th><th>Task Description</th><th>Sample Configuration</th></tr>
<tr><td>1</td><td> Configure the PCE session by specifying the XTC server in the mpls traffic-eng section. The source IP is typically the Loopback0 interface IP. This is a stateful connection and will stay connected to the XTC node by way of TCP. For the disjoint paths feature, we create an attribute-set with path-option. Specify pce and configure the disjoint-path type as node, group-id value as 1, and source as 0.0.0.1. The group-id and source are used to identify and group different policies across the network. </td><td><pre>conf t
!
mpls traffic-eng
 pce
  peer source ipv4 1.1.1.13
  peer ipv4 4.4.4.4
   !
  segment-routing
  stateful-client
   !
 !
 auto-tunnel p2p
  tunnel-id min 1000 max 2000
 !
 attribute-set path-option PO.NODE.DISJ
  pce
   disjoint-path source 0.0.0.1 type node group-id 1
  !
 !
!
commit
end</pre></td></tr>
<tr><td>2</td><td> Configure the tunnel interface and set the metric as te for the PCE computation. Configure the path-option to use the attribute-set PO.NODE.DISJ (created in the previous step). This attribute-set also specified pce, so the PCE will provide the disjoint path. </td><td><pre>conf t
!
interface tunnel-te200
 ipv4 unnumbered Loopback0
 destination 2.2.2.23
 path-selection
  metric te
 !
 path-option 1 dynamic segment-routing attribute-set
PO.NODE.DISJ
!
commit
end</pre></td></tr>
</table>

Verify the Configuration on Node A11

Verify the configuration on node A11 by checking the status of the peering session with XTC, the policy status and the path computed by XTC along with the SID-list and label-stack, and the routing entry for the destination.

```
RP/0/0/CPU0:A11# show mpls traffic-eng pce peer
        Address   Precedence        State        Learned From
--------------- ------------ ------------ --------------------
        4.4.4.4          255           Up       Static config

RP/0/0/CPU0:A11# show mpls traffic-eng pce tunnels
Tunnel : tunnel-te100
    Destination : 2.2.2.21
    State : up
    Current path option: 1, path learned from PCE 4.4.4.4
    Admin Weight : 21
    Hop Count    : 3

RP/0/0/CPU0:A11# show mpls  traffic-eng tunnels 200

Name: tunnel-te200  Destination: 2.2.2.21  Ifhandle:0xd0
```

```
  Signalled-Name: A11_t200
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 1, (Segment-Routing) type dynamic pce (Basis for Setup, path weight 30)
      Path-option attribute: PO%2eNODE%2eDISJ
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Sun Jun  4 19:47:19 2017 (1d05h ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (interface)
    Path Selection:
      Tiebreaker: Min-fill (default)
      Protection: any (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Tunnel class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    SRLG discovery: Disabled
  History:
    Tunnel has been up for: 1d04h (since Sun Jun 04 20:26:26 UTC 2017)
    Current LSP:
      Uptime: 1d04h (since Sun Jun 04 20:46:29 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, has no S2Ls
        Date/Time: Tue Jun 06 00:46:29 UTC 2017 [00:31:51 ago]
    Prior LSP:
      ID: 2 Path Option: 1
      Removal Trigger: reoptimization completed

  Segment-Routing Path Info (PCE computed path)
    Segment0[Link]: 11.111.0.11 - 11.111.0.111, Label: 28107
    Segment1[Node]: 3.3.3.121, Label: 16121
    Segment2[Link]: 21.121.0.121 - 21.121.0.21, Label: 24000
Displayed 1 (of 4) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
RP/0/0/CPU0:A11#
```

Verify the Configuration on Node A13

Verify the configuration on node 13 by checking the status of the peering session with XTC, the policy status and the path computed by XTC along with the SID-list and label-stack, and the routing entry for the destination.

```
RP/0/0/CPU0:A13# show mpls traffic-eng pce peer
        Address    Precedence        State         Learned From
--------------- ------------ ------------ --------------------
        4.4.4.4           255           Up         Static config

RP/0/0/CPU0:A13# show mpls traffic-eng pce tunnels
Tunnel : tunnel-te100
    Destination : 2.2.2.23
    State : up
    Current path option: 1, path learned from PCE 4.4.4.4
    Admin Weight : 21
    Hop Count    : 3

RP/0/0/CPU0:A13# show mpls  traffic-eng  tunnels 200

Name: tunnel-te200  Destination: 2.2.2.23  Ifhandle:0x90
```

```
  Signalled-Name: A13_t200
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 1, (Segment-Routing) type dynamic pce (Basis for Setup, path weight 50020)
      Path-option attribute: PO%2eNODE%2eDISJ
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Sun Jun  4 19:47:48 2017 (1d05h ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (interface)
    Path Selection:
      Tiebreaker: Min-fill (default)
      Protection: any (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Tunnel class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    SRLG discovery: Disabled
  History:
    Tunnel has been up for: 1d04h (since Sun Jun 04 20:26:25 UTC 2017)
    Current LSP:
      Uptime: 00:33:25 (since Tue Jun 06 00:47:21 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, has no S2Ls
        Date/Time: Tue Jun 06 00:47:18 UTC 2017 [00:33:28 ago]
    Prior LSP:
      ID: 60 Path Option: 1
      Removal Trigger: reoptimization completed

  Segment-Routing Path Info (PCE computed path)
    Segment0[Link]: 13.112.0.13 - 13.112.0.112, Label: 28107
    Segment1[Node]: 3.3.3.122, Label: 16122
    Segment2[Link]: 23.122.0.122 - 23.122.0.23, Label: 28096
Displayed 1 (of 2) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
RP/0/0/CPU0:A13#
```

Centralized LSP Link Disjointness TE-Metric Optimization: Example

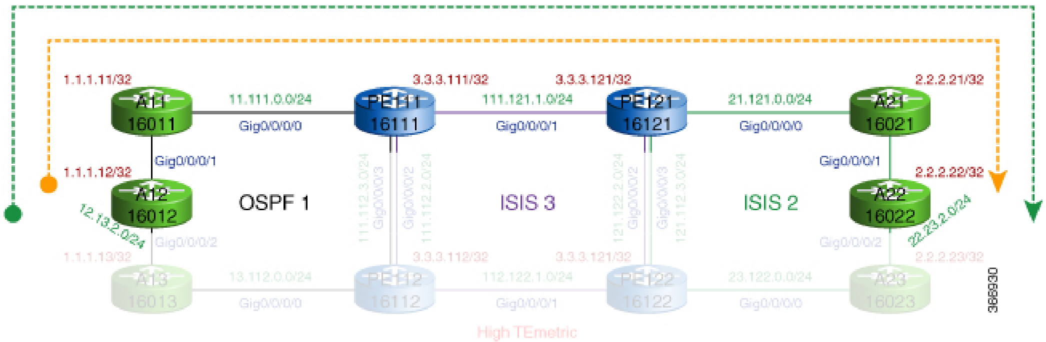
The following example shows how to build two policies from A12 to A22 using the link disjoint-path type. Both policies use the same group ID (2) and source (0.0.0.2) to indicate that the policies are to be grouped together while performing path computations.

When XTC receives a request for the first policy A12 to A22, it checks the group ID and source. Since no policy exists with the same group ID and source combination, XTC calculates a dynamic path for this policy.

When XTC receives a request for the next policy A12 to A23, it again checks the group ID and source. Since XTC already has an existing policy with the same group ID and source combination, it calculates both policy paths, ensuring that there is no common link in the path. It then returns these paths to the head end routers.

Without the disjoint path feature, the dynamic paths of both policies would be the same:

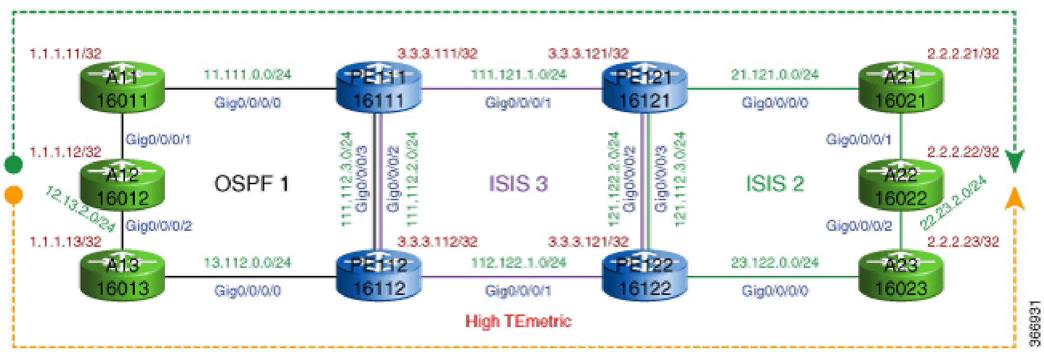
A12 → A11 → PE111 → PE121 → A21 → A22



However, using the disjoint-path feature, no links are shared in the paths:

A12 → A11 → PE111 → PE121 → A21 → A22

A12 → A13 → PE112 → PE122 → A23 → A22



Configurations on Node A12

Complete these tasks on node A12 to use link disjointness for path calculations.

<table>
<tr><th>Task Number</th><th>Task Description</th><th>Sample Configuration</th></tr>
<tr><td>1</td><td>Configure the PCE session by specifying the XTC server in the mpls traffic-eng section. The source IP is typically the Loopback0 interface IP. This is a stateful connection and will stay connected to the XTC node by way of TCP.

For the disjoint paths feature, we create an attribute-set with path-option. Specify pce and configure the disjoint-path type as node, group-id value as 2, and source as 0.0.0.2. The group-id and source are used to identify and group different policies across the network.</td><td><pre>conf t
!
mpls traffic-eng
 pce
  peer source ipv4 1.1.1.12
  peer ipv4 4.4.4.4
   !
  segment-routing
  stateful-client
   !
 !
 auto-tunnel p2p
  tunnel-id min 1000 max 2000
 !
 attribute-set path-option PO.LINK.DISJ
  pce
   disjoint-path source 0.0.0.2 type link group-id 2
  !
 !
!
commit
end</pre></td></tr>
<tr><td>2</td><td>Configure the tunnel interfaces and set the metric as te for the PCE computation.

Configure the path-option to use the attribute-set PO.LINK.DISJ (created in the previous step). This attribute-set also specified pce, so the PCE will provide the disjoint path.</td><td><pre>conf t
!
interface tunnel-te201
 ipv4 unnumbered Loopback0
 destination 2.2.2.22
 path-selection
  metric te
 !
 path-option 1 dynamic segment-routing attribute-set
PO.LINK.DISJ
!
interface tunnel-te202
 ipv4 unnumbered Loopback0
 destination 2.2.2.22
 path-selection
  metric te
 !
 path-option 1 dynamic segment-routing attribute-set
PO.LINK.DISJ
!commit
end</pre></td></tr>
</table>

Verify the Configuration on Node A12

Verify the configuration on node A12 by checking the status of the peering session with XTC, the policy status and the path computed by XTC along with the SID-list and label-stack, and the routing entry for the destination.

```
RP/0/0/CPU0:A12# show mpls traffic-eng pce peer
        Address   Precedence        State       Learned From
--------------- ------------ ------------ --------------------
        4.4.4.4          255           Up        Static config

RP/0/0/CPU0:A12# show mpls  traffic-eng pce tunnels
Tunnel : tunnel-te201
    Destination : 2.2.2.22
```

```
    State : up
    Current path option: 1, path learned from PCE 4.4.4.4
    Admin Weight : 50
    Hop Count    : 5

Tunnel : tunnel-te202
    Destination : 2.2.2.22
    State : up
    Current path option: 1, path learned from PCE 4.4.4.4
    Admin Weight : 50040
    Hop Count    : 5

RP/0/0/CPU0:A12# show mpls traffic-eng tunnels 201

Name: tunnel-te201  Destination: 2.2.2.22  Ifhandle:0x90
  Signalled-Name: A12_t201
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 1, (Segment-Routing) type dynamic pce (Basis for Setup, path weight 50)
      Path-option attribute: PO%2eLINK%2eDISJ
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Sun Jun  4 20:47:49 2017 (1d04h ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (interface)
    Path Selection:
      Tiebreaker: Min-fill (default)
      Protection: any (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Tunnel class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    SRLG discovery: Disabled
  History:
    Tunnel has been up for: 1d04h (since Sun Jun 04 20:49:50 UTC 2017)
    Current LSP:
      Uptime: 00:56:03 (since Tue Jun 06 00:46:39 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, has no S2Ls
        Date/Time: Tue Jun 06 00:46:36 UTC 2017 [00:56:06 ago]
    Prior LSP:
      ID: 59 Path Option: 1
      Removal Trigger: reoptimization completed
  Segment-Routing Path Info (PCE computed path)
    Segment0[Link]: 11.12.1.12 - 11.12.1.11, Label: 28097
    Segment1[Link]: 11.111.0.11 - 11.111.0.111, Label: 28107
    Segment2[Node]: 3.3.3.121, Label: 16121
    Segment3[Link]: 21.121.0. 121 - 21.121.0.21, Label: 24000
    Segment4[Link]: 21.22.1.21 - 21.22.1.22, Label: 28097
Displayed 1 (of 3) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
RP/0/0/CPU0:A12#

RP/0/0/CPU0:A12# show mpls  traffic-eng tunnels  202

Name: tunnel-te202  Destination: 2.2.2.22  Ifhandle:0xb0
  Signalled-Name: A12_t202
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 1, (Segment-Routing) type dynamic pce (Basis for Setup, path weight 50040)
```

```
      Path-option attribute: PO%2eLINK%2eDISJ
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Sun Jun  4 20:47:49 2017 (1d04h ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (interface)
    Path Selection:
      Tiebreaker: Min-fill (default)
      Protection: any (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Tunnel class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    SRLG discovery: Disabled
  History:
    Tunnel has been up for: 1d04h (since Sun Jun 04 20:49:57 UTC 2017)
    Current LSP:
      Uptime: 1d04h (since Sun Jun 04 20:49:57 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, has no S2Ls
        Date/Time: Tue Jun 06 00:46:36 UTC 2017 [00:58:08 ago]
    Prior LSP:
      ID: 3 Path Option: 1
      Removal Trigger: tunnel shutdown

  Segment-Routing Path Info (PCE computed path)
    Segment0[Link]: 12.13.2.12 - 12.13.2.13, Label: 28101
    Segment1[Link]: 13.112.0.13 - 13.112.0.112, Label: 28107
    Segment2[Node]: 3.3.3.122, Label: 16122
    Segment3[Link]: 23.122.0.122 - 23.122.0.23, Label: 28096
    Segment4[Link]: 22.23.2.23 - 22.23.2.22, Label: 28097
Displayed 1 (of 3) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
RP/0/0/CPU0:A12#
```

Centralized LSP SRLG Disjointness TE-Metric Optimization: Example

The following example shows how to build two policies from A11 to A23 using the shared link risk group (SRLG) disjoint-path type. Both policies use the same group ID (3) and source (0.0.0.3) to indicate that the policies are to be grouped together while performing path computations.

Note SRLG disjointness can be considered as SRLG + LINK disjoint, and both conditions must be met to have to feasible paths with no common links.

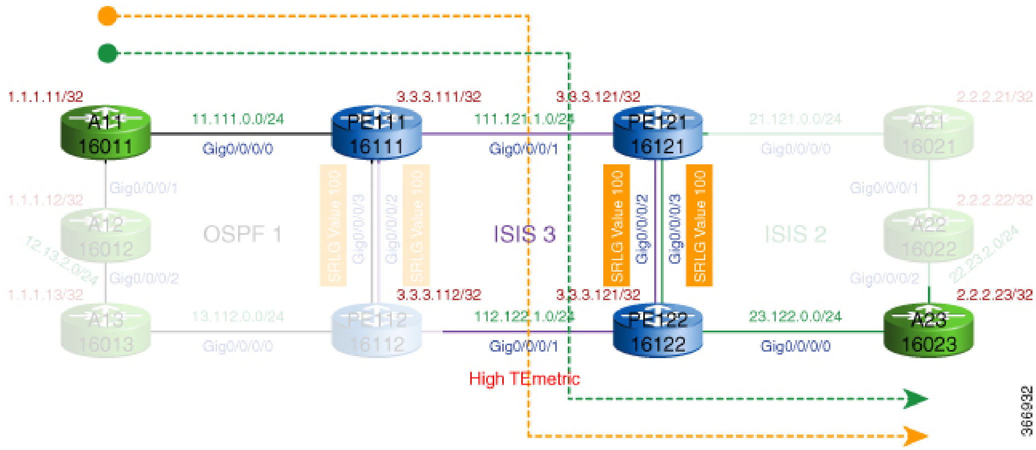
There are four links in the same SRLG: Gig0/0/0/2 and Gig0/0/0/3 between PE111 and PE112, and between PE121 and PE122. The value of these links is set to 100.

When XTC receives a request for the first policy A11 to A23, it checks the group ID and source. Since no policy exists with the same group ID and source combination, XTC calculates a dynamic path for this policy.

When XTC receives a request for the next policy A11 to A23, it again checks the group ID and source. Since XTC already has an existing policy with the same group ID and source combination, it calculates both policy paths, ensuring that there is no link in the same SRLG in the path. It then returns these paths to the head end routers.

If the disjoint path feature is not configured, the dynamic paths of both policies are the same:

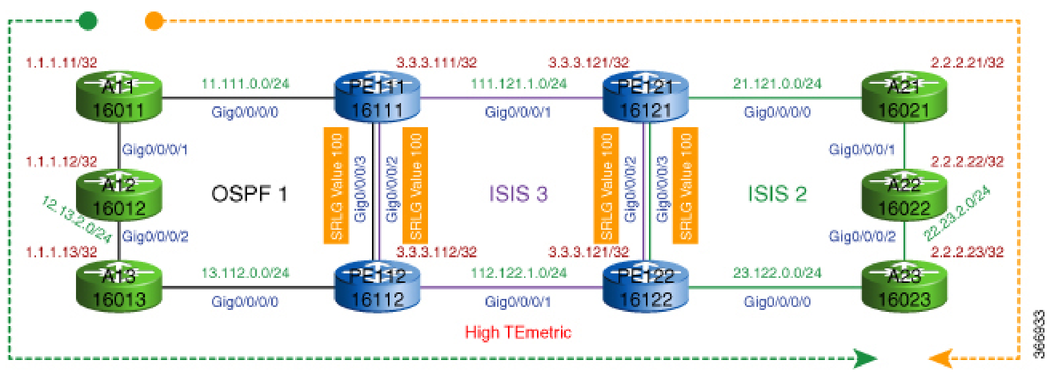
A11 → PE111 → PE121 → PE122 → A23



However, when the disjoint-path feature is configured, no two links use the same SRLG in the paths:

A11 → PE111 → PE121 → A21 → A22 → A23

A11 → A12 → A13 → PE112 → PE122 → A23



Configuration on PE111, PE112, PE121, PE122

Configure SRLG on the nodes in the core (PE111, PE112, PE121, PE122)

<table>
<tr><th>Task Description</th><th>Sample Configuration</th></tr>
<tr><td>Configure SRLG on the nodes in the core so XTC knows which links are in the same group.</td><td><pre>conf t
!
srlg
 interface GigabitEthernet0/0/0/2
  8 value 100
  name CORE_TOP_DOWN
 !
 interface GigabitEthernet0/0/0/3
  8 value 100
  name CORE_TOP_DOWN
 !
!
commit
end</pre></td></tr>
</table>

Configurations on Node A11

Complete these tasks on node A11 to use SRLG disjointness for path calculations.

<table>
<tr><th>Task Number</th><th>Task Description</th><th>Sample Configuration</th></tr>
<tr><td>1</td><td>Configure the PCE session by specifying the XTC server in the mpls traffic-eng section. The source IP is typically the Loopback0 interface IP. This is a stateful connection and will stay connected to the XTC node by way of TCP.

For the disjoint paths feature, we create an attribute-set with path-option. Specify pce and configure the disjoint-path type as srlg, group-id value as 3, and source as 0.0.0.3. The group-id and source are used to identify and group different policies across the network.</td><td><pre>conf t
!
mpls traffic-eng
 pce
  peer source ipv4 1.1.1.11
  peer ipv4 4.4.4.4
  !
  segment-routing
  stateful-client
  !
 !
 auto-tunnel p2p
  tunnel-id min 1000 max 2000
 !
 attribute-set path-option PO.SRLG.DISJ
  pce
   disjoint-path source 0.0.0.3 type srlg group-id 3
  !
 !
!
commit
end</pre></td></tr>
<tr><td>2</td><td>Configure the tunnel interfaces and set the metric as te for the PCE computation.

Configure the path-option to use the attribute-set PO.SRLG.DISJ (created in the previous step). This attribute-set also specified pce, so the PCE will provide the disjoint path.</td><td><pre>conf t
!
interface tunnel-te201
 ipv4 unnumbered Loopback0
 destination 2.2.2.23
 path-selection
  metric te
 !
 path-option 1 dynamic segment-routing attribute-set
PO.SRLG.DISJ
!
interface tunnel-te202
 ipv4 unnumbered Loopback0
 destination 2.2.2.23
 path-selection
  metric te
 !
 path-option 1 dynamic segment-routing attribute-set
PO.SRLG.DISJ
!commit
end</pre></td></tr>
</table>

Verify the Configuration on Node A11

Verify the configuration on node A11 by checking the status of the peering session with XTC, the policy status and the path computed by XTC along with the SID-list and label-stack, and the routing entry for the destination.

```
RP/0/0/CPU0:A11# show mpls traffic-eng pce peer
        Address    Precedence         State         Learned From
--------------- ------------ ------------ --------------------
        4.4.4.4           255            Up         Static config

RP/0/0/CPU0:A11# show mpls traffic-eng pce tunnels
Tunnel : tunnel-te201
    Destination : 2.2.2.23
```

```
    State : up
    Current path option: 1, path learned from PCE 4.4.4.4
    Admin Weight : 50
    Hop Count    : 4

Tunnel : tunnel-te202
    Destination : 2.2.2.23
    State : up
    Current path option: 1, path learned from PCE 4.4.4.4
    Admin Weight : 50040
    Hop Count    : 5

RP/0/0/CPU0:A11# show mpls traffic-eng tunnels 201


Name: tunnel-te201  Destination: 2.2.2.23  Ifhandle:0x170
  Signalled-Name: A11_t201
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 1, (Segment-Routing) type dynamic pce (Basis for Setup, path weight 50)
      Path-option attribute: PO%2eSRLG%2eDISJ
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Tue Jun  6 18:59:07 2017 (1w1d ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (interface)
    Path Selection:
      Tiebreaker: Min-fill (default)
      Protection: any (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Tunnel class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    SRLG discovery: Disabled
  History:
    Tunnel has been up for: 6d21h (since Wed Jun 07 22:25:32 UTC 2017)
    Current LSP:
      Uptime: 00:39:00 (since Wed Jun 14 18:46:36 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, has no S2Ls
        Date/Time: Wed Jun 14 18:46:32 UTC 2017 [00:39:04 ago]
    Prior LSP:
      ID: 149 Path Option: 1
      Removal Trigger: reoptimization completed

  Segment-Routing Path Info (PCE computed path)
    Segment0[Link]: 11.111.0.11 - 11.111.0.111, Label: 28107
    Segment1[Node]: 3.3.3.121, Label: 16121
    Segment2[Link]: 21.121.0.121 - 21.121.0.21, Label: 24000
    Segment3[Node]: 2.2.2.23, Label: 16023
Displayed 1 (of 4) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads

RP/0/0/CPU0:A11# show mpls traffic-eng tunnels 202


Name: tunnel-te202  Destination: 2.2.2.23  Ifhandle:0x110
  Signalled-Name: A11_t202
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected
```

```
    path option 1, (Segment-Routing) type dynamic pce (Basis for Setup, path weight 50040)
      Path-option attribute: PO%2eSRLG%2eDISJ
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Sun Jun  4 19:47:19 2017 (1w2d ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (interface)
    Path Selection:
      Tiebreaker: Min-fill (default)
      Protection: any (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Tunnel class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    SRLG discovery: Disabled
  History:
    Tunnel has been up for: 6d21h (since Wed Jun 07 22:25:32 UTC 2017)
    Current LSP:
      Uptime: 1d20h (since Mon Jun 12 22:41:56 UTC 2017)
    Reopt. LSP:
      Last Failure:
        LSP not signalled, has no S2Ls
        Date/Time: Wed Jun 14 18:46:32 UTC 2017 [00:40:05 ago]
    Prior LSP:
      ID: 59 Path Option: 1
      Removal Trigger: reoptimization completed

  Segment-Routing Path Info (PCE computed path)
    Segment0[Link]: 11.12.1.11 - 11.12.1.12, Label: 28115
    Segment1[Link]: 12.13.2.12 - 12.13.2.13, Label: 28101
    Segment2[Link]: 13.112.0.13 - 13.112.0.112, Label: 28107
    Segment3[Node]: 3.3.3.122, Label: 16122
    Segment4[Link]: 23.122.0.122 - 23.122.0.23, Label: 28096
Displayed 1 (of 4) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
RP/0/0/CPU0:A11#
```