Mécanisme de convergence de SR-TE Policy-based Explicit-Path avec protection de noeud TI-LFA

Table des matières

Introduction

Ce document décrit la protection de noeud pour le chemin principal explicite par Topology Independent (TI) - Loop-Free Alternative (LFA) et la solution utilisant le chemin Segment Routing (SR) - Traffic Engineering (TE) avec la métrique SR-TE et l'algorithme flexible Open Shortest Path First (OSPF) .

Problème

Cette section explique les exigences des réseaux XYZ, les contraintes de conception et la raison pour laquelle le chemin de sauvegarde TI-LFA ne parvient pas à protéger toute défaillance de noeud intermédiaire pour un chemin principal explicitement défini.

Exigences

Selon les réseaux XYZ, voici les exigences de leur conception de réseau nouvelle génération :

1. Le chemin du trafic principal doit être explicitement défini et contrôlé par la politique SR-TE (admin), mais pas par la métrique IGP.

2. En cas de défaillance d'une liaison ou d'un noeud, le trafic doit converger vers un chemin de

secours en moins de 50 ms avec un réseau à échelle zéro.

Si vous regardez la Figure 1, une stratégie SR-TE a été configurée de bout en bout au niveau du noeud source PE1, PE3 étant le noeud de destination.

Les configurations SR-TE et OSPF sont les suivantes :

<#root>

```
segment-routing


 traffic-eng
   !
  !


segment-list PrimaryPath1


   index 10 mpls adjacency 10.1.11.0
--> First Hop (P1 node) of the explicit-path


   index 20 mpls adjacency 10.1.3.1
-->


Second Hop (P3 node) of the explicit-path


   index 30 mpls adjacency 10.3.13.1
--> Third Hop (PE3 node) of the explicit-path


  !
  policy POL1
   source-address ipv4 11.11.11.11
--> Source Node of the explicit-path


   color 10 end-point ipv4 33.33.33.33
--> Destination Node of the explicit-path


   candidate-paths


preference 100          --> Secondary Path taken care of dynamically by IGP TI-LFA


     dynamic
      metric
       type igp
      !
     !
    !
```

```
preference 200


     explicit segment-list PrimaryPath1
```

--> Primary Explicit-Path of the SR-TE policy

```
     !
    !


router ospf CORE


 nsr
distribute link-state
 log adjacency changes
 router-id 11.11.11.11
 segment-routing mpls
 nsf cisco
 microloop avoidance segment-routing
 max-metric router-lsa on-startup 360
 area 0


interface Bundle-Ether111
```

--> Primary Explicit-Path Interface

```
   authentication null
   network point-to-point
   fast-reroute per-prefix
   fast-reroute per-prefix ti-lfa enable
```

--> Enabling TI-LFA on the primary interface

```
   fast-reroute per-prefix tiebreaker node-protecting index 100
   fast-reroute per-prefix tiebreaker srlg-disjoint index 200
   prefix-suppression
  !


interface Bundle-Ether211
```

--> Secondary Dynamic Path Interface

```
   authentication null
   network point-to-point
   fast-reroute per-prefix
   fast-reroute per-prefix ti-lfa enable
```

--> Enabling TI-LFA on the secondary interface

```
   fast-reroute per-prefix tiebreaker node-protecting index 100
   fast-reroute per-prefix tiebreaker srlg-disjoint index 200
   prefix-suppression
  !
  interface Loopback80
```

```
   passive enable
   prefix-sid index 32130
```

--> Enabling Node SID on the loopback interface

```
  !
 !
```

Cette configuration est un exemple de méthode pour configurer une politique SR-TE basée sur un chemin explicite. Il existe également d'autres méthodes. Et sous OSPF, il est observé que TI-LFA est activé.

Cependant, avec la combinaison de fonctionnalités SR-TE et OSPF, il est démontré dans les travaux pratiques avec la politique de chemin explicite SR-TE que OSPF TI-LFA n'est pas en mesure de se courber et d'installer un chemin de sauvegarde post-convergence de bout en bout (PE1 à PE3) du chemin principal explicite SR-TE pour les scénarios de défaillance de noeud intermédiaire, comme illustré à la Figure 2. Par conséquent, le temps de convergence de la protection du trafic dépasse largement les 50 ms en cas de panne du noeud P1 ou P3.

Un exemple simple a été choisi pour expliquer le problème :

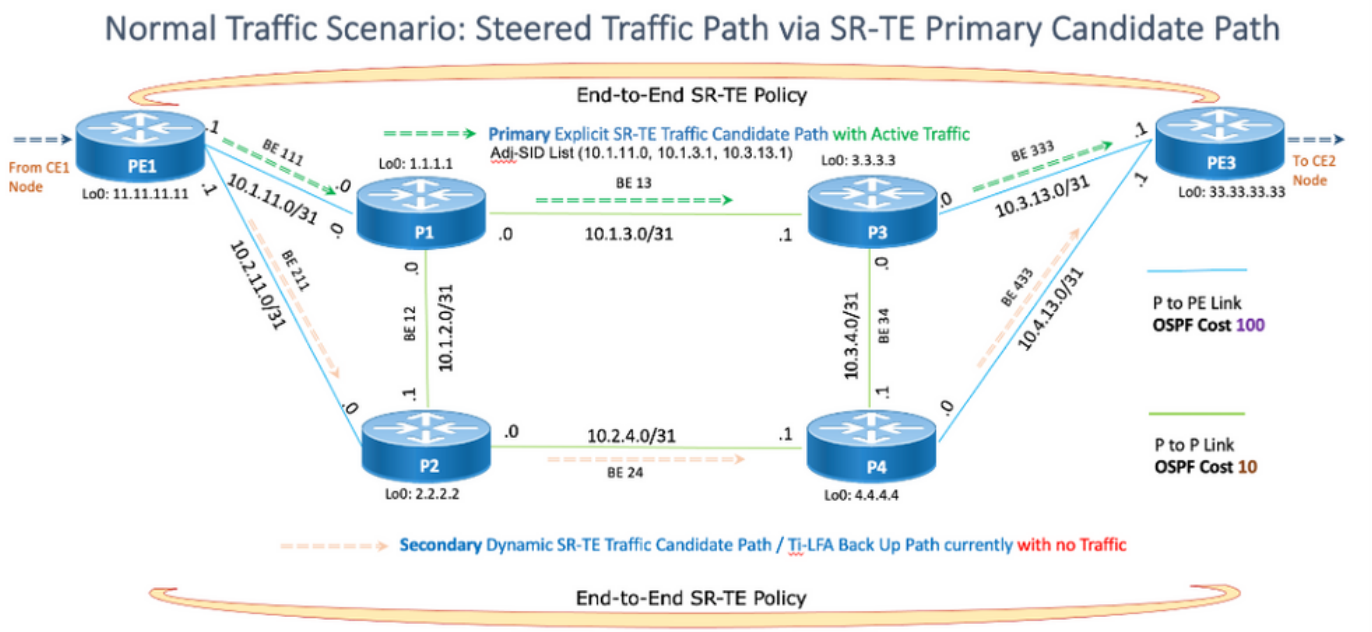

Figure 1 : Scénario de trafic normal

Comme dans la Figure 1, le noeud source du trafic est PE1 et le noeud de destination est PE3. Ici, si nous configurons une politique de chemin explicite SR-TE où il y a un besoin administratif d'envoyer le trafic via le chemin de trafic primaire explicite PE1> P1> P3> PE3.

Dans cette situation, si nous configurons un chemin SR-TE explicite via PE1> P1 > P3> PE3, alors en cas de défaillance du noeud, comme illustré à la Figure 2., TI-LFA n'est pas en mesure de protéger le scénario de défaillance du noeud, mais ne peut protéger que le scénario de défaillance de la liaison. Le scénario de défaillance de liaison a été traité en détail dans le

document de référence [Convergence of SR-TE Explicit-Path for Link Protection.]

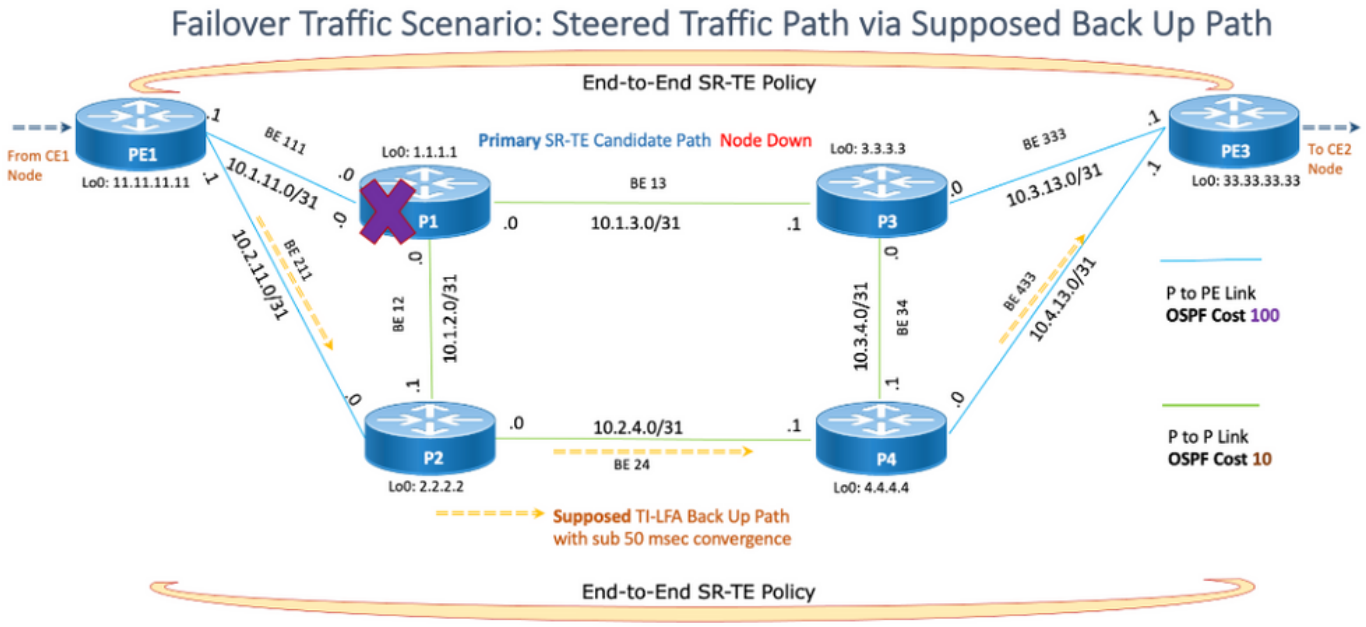

Figure 2 : Scénario de trafic de basculement

Pourquoi le chemin de sauvegarde TI-LFA ne protège pas les pannes de noeuds intermédiaires

Une fois configuré sous OSPF, le TI-LFA pointe par défaut vers le SID du noeud de destination pour calculer et installer le chemin de sauvegarde dans le plan de données.

Mais pour ce scénario et cette configuration d'ensemble de fonctionnalités, la couverture TI-LFA du noeud source au noeud de destination ne fonctionne pas, ou en d'autres termes, le chemin de sauvegarde TI-LFA ne parvient pas à protéger toute défaillance de noeud intermédiaire inférieure à 50 ms pour le chemin principal explicitement défini.

L'analyse montre que l'algorithme de calcul du chemin de sauvegarde TI-LFA prend le premier saut/noeud suivant du chemin explicite comme point d'extrémité de destination au lieu du noeud de destination réel et calcule le chemin de sauvegarde en essayant de protéger uniquement le premier saut/noeud suivant, par exemple, le noeud P1 comme dans la Figure 2. Par conséquent, TI-LFA n'est pas en mesure de calculer et d'installer un chemin de secours pour protéger le noeud d'extrémité ou de destination réel, par exemple, le noeud PE3.

Par conséquent, il n'est pas en mesure d'assurer une protection de bout en bout dans un délai inférieur à 50 ms de convergence pour le noeud de destination réel PE3 en cas de défaillance d'un noeud intermédiaire dans un chemin de trafic principal explicitement défini.

Une autre façon de l'examiner est dans la Figure 1., si vous configurez le noeud P3 comme tronçon suivant dans le chemin explicite, alors TI-LFA peut fournir une protection de moins de 50 ms pour la défaillance du noeud P1 et vice versa. Mais la protection de noeud ne peut pas se produire pour ce noeud particulier qui est défini comme l'un des sauts explicites du chemin explicite de bout en bout.

Solution

Cette section se concentre sur les points pour les scénarios explicites spécifiques au chemin principal :

Comment le chemin de sauvegarde TI-LFA protège désormais toute panne de noeud intermédiaire en dessous de la convergence de 50 ms

Une solution éprouvée et testée consiste à apporter des fonctionnalités/modifications supplémentaires au scénario afin de permettre à TI-LFA de prendre en charge les moins de 50 ms de convergence pendant le scénario de défaillance du noeud ainsi que la défaillance de la liaison. Cette solution a été choisie en fonction des exigences du réseau XYZ, comme indiqué dans la section Problème.

Décomposition des étapes de la solution

1. Explicit-Path est nécessaire, mais la métrique IGP ne peut pas être utilisée conformément à la condition.

2. Par conséquent, une autre métrique (SR-TE metric) est utilisée pour orienter le trafic sur un certain chemin sans spécifier les sauts explicites.

3. OSPF Flex-Algo est utilisé pour envoyer du trafic au noeud de destination (à l'aide d'un SID de noeud Flex-Algo distinct accessible via l'algorithme Flex Algo) via la topologie qui utilise la métrique SR-TE.

3. Après l'ajout de l'algorithme OSPF Flex-Algo, TI-LFA peut fonctionner normalement, car il peut maintenant protéger le SID du noeud de destination réel.

Compréhension Différents composants de la solution

Caractéristique De Chemin Explicite

Étant donné que, selon l'une des exigences, la métrique IGP ne peut pas être utilisée pour le contrôle explicite du chemin principal, la caractéristique rationalisée explicite du chemin SR-TE principal est contrôlée via la métrique TE configurée en outre sous les interfaces SR-TE (sous le routage de segment) pour tous les noeuds, y compris le noeud PE de tête de réseau, jusqu'au PE de destination distant. Leurs métriques SR-TE sont à leur tour utilisées par OSPF Flex Algo pour créer un chemin explicite dans le cadre du paradigme flex-algo.

Mesure SR-TE sous Segment Routing at PE1 :

<#root>

```
segment-routing
 global-block 100000 299999
 traffic-eng
```

interface Bundle-Ether111

metric 10

--> SR-TE Metric of BE111 is less that BE211, so it is a more preferred explicit path given that rest of

!

interface Bundle-Ether211

metric 100

!
logging
policy status
!

policy er100_to_er102 --> SR-TE policy defined

source-address ipv4 11.11.11.11.

--> Source Node of the explicit-path

color 150 end-point ipv4 33.33.33.33

--> Destination Node of the explicit-path

autoroute
force-sr-include
include all
!

candidate-paths

preference 200

dynamic --> Here that the primary path is configured as dynamic but it is the SR-TE metric defined ab

make it fixed or explicit

!
constraints
segments

sid-algorithm 128. --> Primary SR-TE path is configured with constraint as Flex-Algo 128 with no explic

the backup path implicitly ensuring sub 50 msec of convergence

```
      !
     !
```

Commande show au noeud PE1 :

<#root>

```
P/0/RP0/CPU0:PE1#
```

show segment-routing traffic-eng policy

```
Fri Feb  3 10:25:24.716 UTC

SR-TE policy database
---------------------
```

Color: 150, End-point: 33.33.33.33 --> Color and Endpoint Loopback IP address of PE3

```
  Name: srte_c_150_ep_33.33.33.33
  Status:
    Admin: up  Operational: up for 04:57:30 (since Feb  3 05:27:54.774)
  Candidate-paths:
```

Preference: 200

```
 (configuration) (active)
```

--> Preference of 200 as configured under SR-TE policy

```
      Name: er100_to_er102
      Requested BSID: dynamic
      Constraints:
```

Prefix-SID Algorithm: 128   --> Attached to Flex-Algo 128 as configured under SR-TE policy

Protection Type: protected-preferred   --> Protected Primary Path

```
        Maximum SID Depth: 12
      Dynamic (valid)
```

Metric Type: TE

```
,   Path Accumulated Metric: 0
```

--> Metric Type is SR-TE metric

```
133138

 [Prefix-SID: 33.33.33.33, Algorithm: 128].

--> Node SID of destination node PE3 with index 33138


   Attributes:
    Binding SID: 24010
    Forward Class: Not Configured
    Steering labeled-services disabled: no
    Steering BGP disabled: no
    IPv6 caps enable: yes
    Invalidation drop enabled: no
```

OSPF Flex-Algo

Aperçu:

Segment Routing Flexible Algorithm permet aux opérateurs de personnaliser le calcul du plus court chemin IGP en fonction de leurs propres besoins. Un opérateur peut attribuer des SID de préfixe SR personnalisés pour effectuer le transfert au-delà du SPF basé sur le coût de la liaison. Par conséquent, Flexible Algorithm fournit un chemin d'ingénierie de trafic automatiquement calculé par l'IGP vers n'importe quelle destination accessible par l'IGP.

Pour offrir une flexibilité maximale, le mappage entre la valeur de l'algorithme et sa signification peut être défini par l'utilisateur. Lorsque tous les routeurs du domaine ont une compréhension commune de ce que représente la valeur d'algorithme particulière, le calcul d'un tel algorithme est cohérent et le trafic n'est pas soumis à une boucle. Ici, comme la signification de l'algorithme n'est définie par aucune norme, mais est définie par l'utilisateur, on parle d'algorithme flexible.

Selon le paradigme de routage OSPF, de nombreuses contraintes possibles peuvent être utilisées pour calculer un chemin sur un réseau. Certains réseaux sont déployés avec un seul plan IGP et d'autres avec plusieurs plans IGP. Pour un réseau donné, sous chaque processus OSPF, il existe par défaut Flex-Algo 0 avec une forme simple de contrainte, par exemple, la métrique OSPF.

Cependant, en gardant à l'esprit les exigences spécifiques, une forme plus sophistiquée de contrainte est utilisée ici qui inclut des paramètres étendus comme TE-metric (Multiple Flex-Algo numbers range from 128 to 255). Dans Cisco IOS® XR 7.3.2, cette métrique TE doit être configurée sous la section d'ingénierie de trafic SR-TE mais est utilisée par OSPF Flex-Algo pour le calcul de chemin explicite.

TI-LFA calcule le chemin de secours et maintient le plan de données prêt en cas de défaillance du chemin principal et commute le trafic avec un temps de convergence inférieur à 50 ms pour un réseau à échelle zéro.

Configuration:

OSPF Flex-Algo est configuré sous Router OSPF et annoncé sur le réseau. L'algorithme OSPF flex-algo et la métrique TE prennent en charge ensemble le chemin explicite et les moins de 50

ms de convergence. La configuration de Flex-Algo sous OSPF crée une topologie OSPF virtuelle et aide TI-LFA à calculer un chemin de sauvegarde de bout en bout à l'avance pour une paire de points d'extrémité source-destination, ce qui garantit à son tour moins de 50 secondes de convergence en cas de défaillance du chemin principal.

Configuration OSPF sur PE1 :

<#root>

```
router ospf CORE
 nsr
 distribute link-state
 log adjacency changes
 router-id 11.11.11.11
 segment-routing mpls
 nsf cisco
 microloop avoidance segment-routing
 max-metric router-lsa on-startup 360
 area 0
  interface Bundle-Ether111
   cost 10000
   authentication null
   network point-to-point
   fast-reroute per-prefix
   fast-reroute per-prefix ti-lfa enable
   fast-reroute per-prefix tiebreaker node-protecting index 100
   fast-reroute per-prefix tiebreaker srlg-disjoint index 200
   prefix-suppression
  !
  interface Bundle-Ether211
   cost 10000
   authentication null
   network point-to-point
   fast-reroute per-prefix
   fast-reroute per-prefix ti-lfa enable
   fast-reroute per-prefix tiebreaker node-protecting index 100
   fast-reroute per-prefix tiebreaker srlg-disjoint index 200
   prefix-suppression
  !
  interface Loopback80
   passive enable
   prefix-sid index 32130


prefix-sid algorithm 128 index 33130     --> Assigning different Node SIDs to different Flex Algo to kee


   prefix-sid algorithm 129 index 34130     --> Assigning different Node SIDs to different Flex Algo to


  !
 !

flex-algo 128         --> Defining OSPF Flex Algo which creates a virtual topology and enables TI-LFA to


  metric-type te-metric
  advertise-definition
 !
```

```
flex-algo 129.       --> One or more than one Flex Algo can be defined based on the requirement


  metric-type delay
  advertise-definition
 !
!
```

Configuration OSPF sur PE3 :

<#root>

```
router ospf CORE


 nsr
 distribute link-state
 log adjacency changes
 router-id 33.33.33.33
 segment-routing mpls
 nsf cisco
 microloop avoidance segment-routing
 max-metric router-lsa on-startup 360
 area 0
  interface Bundle-Ether111
   cost 10000
   authentication null
   network point-to-point
   fast-reroute per-prefix
   fast-reroute per-prefix ti-lfa enable
   fast-reroute per-prefix tiebreaker node-protecting index 100
   fast-reroute per-prefix tiebreaker srlg-disjoint index 200
   prefix-suppression
  !
  interface Bundle-Ether211
   cost 10000
   authentication null
   network point-to-point
   fast-reroute per-prefix
   fast-reroute per-prefix ti-lfa enable
   fast-reroute per-prefix tiebreaker node-protecting index 100
   fast-reroute per-prefix tiebreaker srlg-disjoint index 200
   prefix-suppression
  !
  interface Loopback80
   passive enable
   prefix-sid index 32138


prefix-sid algorithm 128 index 33138   --> Node SID assigned for OSPF Flex-Algo 128 which is shown above


   prefix-sid algorithm 129 index 34138   --> Assigning different Node SIDs to different Flex Algo to ke


  !
 !

flex-algo 128.         --> Defining OSPF Flex Algo which creates a virtual topology and enables TI-LFA t
```

```
metric-type te-metric --> Metric type te-metric



advertise-definition  --> To enable the router to advertise the definition for the particular Flexible A
 command is used

 !
flex-algo 129


--> Additional Flex Algo definition (if needed)



metric-type delay     --> Metric type delay


  advertise-definition
 !
!
```

Résumé de la solution

Pour résumer, les métriques SR-TE aident à naviguer dans le trafic via le chemin explicite SR-TE désigné, puisque la métrique IGP ne peut pas être utilisée. En ajoutant une couche du plan de contrôle virtuel, le protocole OSPF Flex-Algo aide le protocole TI-LFA à assurer une convergence inférieure à 50 ms du trafic du chemin explicite principal vers le chemin de sauvegarde TI-LFA précalculé. Cela se produit puisque seul le SID du noeud de destination est annoncé pour permettre à TI-LFA de déterminer le noeud de destination réel et ainsi protéger les deux noeuds intermédiaires (P1 et P3) entre une paire de noeuds source-destination du chemin primaire explicite PE1> P1 > P3> PE3. Le chemin de sauvegarde protégé dynamiquement adhérant à moins de 50 ms de convergence avec échelle zéro, dans ce cas, est PE1> P2 > P4> PE3.

Logiciels utilisés

Le logiciel utilisé pour tester et valider la solution est Cisco IOS® XR 7.3.2

Informations connexes

- Partie 1. Convergence de SR-TE Explicit-Path pour la protection des liaisons
- Assistance technique et téléchargements Cisco

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.