

Dépannage de la détection du transfert bidirectionnel vEdge et des problèmes de connexions du plan de données

Table des matières

[Introduction](#)

[Conditions préalables](#)

[Exigences](#)

[Composants utilisés](#)

[Informations du plan de contrôle](#)

[Vérifier les propriétés locales du contrôle](#)

[Vérifier les connexions de contrôle](#)

[protocole de gestion de recouvrement](#)

[Vérifier que les TLOC OMP sont annoncés à partir des vEdge](#)

[Vérifier que vSmart reçoit et annonce les TLOC](#)

[Détection de transfert bidirectionnel](#)

[Comprendre la commande show bfd sessions](#)

[Commande show tunnel statistics](#)

[Liste d'accès](#)

[Traduction d'adresses réseau](#)

[Comment utiliser les outils stun-client pour détecter les cartes et les filtres NAT.](#)

[Types NAT pris en charge pour les tunnels de plan de données « Envoi » utilisés dans l'interface de ligne de commande](#)

[Pare-Feu](#)

[Sécurité](#)

[Problèmes du FAI avec le trafic marqué DSCP](#)

[Debug BFD](#)

[Utiliser Packet-Trace pour capturer les paquets BFD \(version 20.5 et ultérieure\)](#)

[Informations connexes](#)

Introduction

Ce document décrit les problèmes de connexion du plan de données vEdge après une connexion du plan de contrôle, mais aucune connectivité du plan de données entre les sites.

Conditions préalables

Exigences

Cisco recommande de connaître la **Cisco Software Defined Wide Area Network (SDWAN)** solution.

Composants utilisés

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques. Ce document est axé sur les plates-formes vEdge.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes. Pour les routeurs Cisco Edge (routeurs Cisco IOS® XE en mode contrôleur) , veuillez lire .

Informations du plan de contrôle

Vérifier les propriétés locales du contrôle

Afin de vérifier l'état des **Wide Area Network (WAN)** interfaces sur un vEdge, utilisez la commande, **show control local-properties wan-interface-list**.

Dans cette sortie, vous pouvez voir le document RFC 4787 **Network Address Translation (NAT) Type**.

Lorsque le serveur vEdge se trouve derrière un périphérique NAT (pare-feu, routeur, etc.), des adresses IPv4 publiques et privées, des **User Datagram Protocol (UDP)** ports sources publics et privés sont utilisés pour construire les tunnels du plan de données.

Vous pouvez également rechercher l'état de l'interface du tunnel, la couleur et le nombre maximal de connexions de contrôle configurées.

```
vEdge1# show control local-properties wan-interface-list NAT TYPE: E -- indicates End-point independent mapping A -- indicates Address-port dependent
```

Avec ces données, vous pouvez identifier certaines informations sur la façon dont les tunnels de données doivent être construits et sur les ports que vous pouvez vous attendre (du point de vue des routeurs) à utiliser lorsque vous formez les tunnels de données.

Vérifier les connexions de contrôle

Il est important de s'assurer que la couleur qui ne forme pas les tunnels du plan de données a une connexion de contrôle établie avec les contrôleurs dans la superposition.

Sinon, le serveur vEdge n'envoie pas les **Transport Locator (TLOC)** informations au serveur vSmart via **Overlay Management Protocol (OMP)**.


Vous pouvez vérifier s'il est opérationnel à l'aide de la **show control connections** commande et rechercher l' **connect** état.

```
vEdge1# show control connections PEER PEER CONTROLLER PEER PEER PEER SITE DOMAIN PEER PRIV PEER PUB GROUP TYPE PROT SYS
```

Si l'interface (qui ne forme pas de tunnels de données) essaie de se connecter, résolvez-la avec un démarrage réussi des connexions de contrôle via cette couleur.

Vous pouvez également définir le **max-control-connections 0** dans l'interface sélectionnée sous la section d'interface du tunnel.

```
vpn 0 interface ge0/1 ip address 10.20.67.10/24 tunnel-interface encapsulation ipsec color mpls restrict max-control-connections 0 no allow-service bgp all
```

 **Remarque** : vous pouvez parfois utiliser la **no control-connections** commande pour atteindre le même objectif. Cependant, cette commande n'établit pas un nombre maximal de connexions de contrôle. Cette commande est déconseillée depuis la version 15.4 et n'est pas utilisée sur un logiciel plus récent.

protocole de gestion de recouvrement

Vérifier que les TLOC OMP sont annoncés à partir des vEdge

Les TLOC OMP ne peuvent pas être envoyés, car l'interface tente d'établir des connexions de contrôle via cette couleur et n'est pas en mesure d'atteindre les contrôleurs.

Vérifiez si la couleur (que les tunnels de données) envoie le TLOC pour cette couleur particulière à vSmarts.


Utilisez la commande **show omp tlocs advertised** afin de vérifier les TLOC qui sont envoyés aux homologues OMP.

Exemple : Couleurs **mpls** et **gold**. Aucune TLOC n'est envoyée à vSmart pour les mpls couleur.

```
vEdge1# show omp tlocs advertised C -> chosen I -> installed Red -> redistributed Rej -> rejected L -> looped R -> resolved S -> stale Ext -> extranet Stg
```

Exemple : Couleurs **mpls** et **gold**. TLOC est envoyé pour les deux couleurs.

```
vEdge2# show omp tlocs advertised C -> chosen I -> installed Red -> redistributed Rej -> rejected L -> looped R -> resolved S -> stale Ext -> extranet Stg
```

 **Remarque** : pour toute information de plan de contrôle générée localement, le champ "**FROM PEER**" est défini sur 0.0.0.0. Lorsque vous recherchez des informations d'origine locale, assurez-vous de les faire correspondre en fonction de cette valeur.

Vérifier que vSmart reçoit et annonce les TLOC

Les TLOC sont désormais annoncés à la vSmart. Confirmez qu'il reçoit les TLOC de l'homologue correct et l'annonce à l'autre serveur vEdge.

Exemple : vSmart reçoit les TLOC de 10.1.0.2 vEdge1.

<#root>

```
vSmart1# show omp tlocs received
```

```
C -> chosen I -> installed
```

```
Red -> redistributed Rej -> rejected L -> looped
```

```
R -> resolved
```

```
S -> stale Ext -> extranet Stg -> staged Inv -> invalid PUBLIC PRIVATE ADDRESS PSEUDO PUBLIC PRIVATE P
```

```
10.1.0.2 mpls ipsec 10.1.0.2 C,I,R 1 10.20.67.20 12386 10.20.67.20 12386 :: 0 :: 0 -
```

```
10.1.0.2 blue ipsec 10.1.0.2 C,I,R 1 198.51.100.187 12406 10.19.146.2 12406 :: 0 :: 0 -
```

```
10.1.0.30 mpls ipsec 10.1.0.30 C,I,R 1 10.20.67.30 12346 10.20.67.30 12346 :: 0 :: 0 - 10.1.0.30 gold
```

Si vous ne voyez pas les TLOC ou si vous voyez d'autres codes ici, vérifiez les points suivants :

```
<#root>
```

```
vSmart-vIPtela-MEX# show omp tlocs received
```

```
C -> chosen
```

```
I -> installed
```

```
Red -> redistributed
```

```
Rej -> rejected
```

```
L -> looped
```

```
R -> resolved
```

```
S -> stale Ext -> extranet Stg -> staged
```

```
Inv -> invalid
```

```
PUBLIC PRIVATE ADDRESS PSEUDO PUBLIC PRIVATE PUBLIC IPV6 PRIVATE IPV6 BFD FAMILY TLOC IP COLOR ENCAP F
```

```
10.1.0.2 mpls ipsec 10.1.0.2 C,I,R 1 10.20.67.20 12386 10.20.67.20 12386 :: 0 :: 0 -
```

```
10.1.0.2 blue ipsec 10.1.0.2 Rej,R,Inv 1 198.51.100.187 12406 10.19.146.2 12406 :: 0 :: 0 -
```

```
10.1.0.30 mpls ipsec 10.1.0.30 C,I,R 1 10.20.67.30 12346 10.20.67.30 12346 :: 0 :: 0 - 10.1.0.30 gold
```

Vérifiez qu'aucune stratégie ne bloque les TLOC.

show run policy control-policy - recherchez toute liste de contrôle de l'accès rejetant vos TLOC comme **advertised** ou **received** dans le vSmart.

```
<#root>
```

```
vSmart1(config-policy)# sh config policy lists tloc-list SITE20
```

```
tloc 10.1.0.2 color blue encap ipsec
```

```
!! control-policy SDWAN
```

```
sequence 10 match tloc tloc-list SITE20 ! action reject ---->
```

here we are rejecting the TLOC 10.1.0.2,blue,ipsec !! default-action accept !

```
apply-policy
```

```
site-list SITE20
```

```
control-policy SDWAN in ----->
```

the policy is applied to control traffic coming IN the vSmart, it will filter the tlocs before adding i



Remarque : si une TLOC est **Rejected** ou **Invalid**, elle n'est pas annoncée aux autres arêtes.

Assurez-vous qu'une stratégie ne filtre pas le TLOC lorsqu'il est annoncé à partir de la vSmart. Vous pouvez voir que le TLOC est reçu sur le vSmart, mais vous ne le voyez pas sur l'autre vEdge.

Exemple 1 : vSmart avec TLOC dans C, I, R.

```
<#root>
```

```
vSmart1# show omp tlocs
```

```
C -> chosen I -> installed
```

```
Red -> redistributed Rej -> rejected L -> looped
```

```
R -> resolved
```

```
S -> stale Ext -> extranet Stg -> staged Inv -> invalid PUBLIC PRIVATE ADDRESS PSEUDO PUBLIC PRIVATE P
```

```
10.1.0.2 mpls ipsec 10.1.0.2 C,I,R 1 10.20.67.20 12386 10.20.67.20 12386 :: 0 :: 0 - 10.1.0.2 blue ipse
```

```
10.1.0.30 mpls ipsec 10.1.0.30 C,I,R 1 10.20.67.30 12346 10.20.67.30 12346 :: 0 :: 0 - 10.1.0.30 gold
```

Exemple 2 : vEdge1 ne voit pas la TLOC de couleur bleue provenant de vEdge2. Il ne voit que le TLOC MPLS.

```
<#root>
```

```
vEdge1# show omp tlocs C -> chosen I -> installed Red -> redistributed Rej -> rejected L -> looped R -> resolved S -> stale Ext -> extranet Stg -> staged I
```

```
10.1.0.2 mpls ipsec 10.1.0.3 C,I,R 1 10.20.67.20 12386 10.20.67.20 12386 :: 0 :: 0 up
```

```
10.1.0.30 mpls ipsec 10.1.0.3 C,I,R 1 10.20.67.30 12346 10.20.67.30 12346 :: 0 :: 0 up 10.1.0.30 gold
```

Lorsque vous vérifiez la stratégie, vous pouvez voir pourquoi le TLOC n'apparaît pas sur le vEdge1.

```
<#root>
```

```
vSmart1# show running-config policy policy lists tloc-list SITE20
```

```
tloc 10.1.0.2 color blue encap ipsec
! site-list SITE10 site-id 10 !! control-policy SDWAN sequence 10 match tloc
tloc-list SITE20
! action reject !! default-action accept !
apply-policy
site-list SITE10
control-policy SDWAN out
!
!
```

Détection de transfert bidirectionnel

Comprendre la commande [show bfd sessions](#)

Voici les éléments clés à rechercher dans le résultat :

```
<#root>
```

```
vEdge-2# show bfd sessions SOURCE TLOC REMOTE TLOC DST PUBLIC DST PUBLIC DETECT TX SYSTEM IP SITE ID STATE COLOR COLO
10.1.0.5 10 down blue gold 10.19.146.2 203.0.113.225 4501 ipsec 7 1000 NA 7
10.1.0.30 30 up blue gold 10.19.146.2 192.0.2.129 12386 ipsec 7 1000 0:00:00:22 2 10.1.0.4 40 up blue
10.1.0.4 40 up mpls mpls 10.20.67.10
```


- **SYSTEM IP**: homologue ip-système
- **SOURCE and REMOTE TLOC COLOR**: cette fonction est utile pour savoir ce que la TLOC est censée recevoir et envoyer.
- **SOURCE IP**: il s'agit de l'adresse IP **private** source. Si vous êtes derrière une NAT, cette information est affichée ici (elle peut être vue avec l'utilisation de **show control local-properties <wan-interface-list>**).
- **DST PUBLIC IP**: il s'agit de la destination utilisée par le serveur vEdge pour former le **Data Plane** tunnel, qu'il soit ou non derrière la fonction NAT. (Exemple : des serveurs vEdge directement connectés à Internet ou des **Multi-Protocol Label Switching (MPLS)** liaisons)
- **DST PUBLIC PORT** Port NAT public utilisé par le serveur vEdge afin de former le **Data Plane** tunnel vers le serveur vEdge distant.
- **TRANSITIONS**: nombre de fois où l'état de la session BFD a changé, de **NA** à **UP** et vice versa.

Commande show tunnel statistics

Le **show tunnel statistics** peut afficher des informations sur les tunnels du plan de données. Vous pouvez déterminer si vous envoyez ou recevez des paquets pour un tunnel IPSEC particulier entre les vEdge.

Cela peut vous aider à comprendre si des paquets arrivent à chaque extrémité et à isoler les problèmes de connectivité entre les noeuds.

Dans l'exemple, lorsque vous exécutez la commande plusieurs fois, vous pouvez remarquer un incrément ou aucun incrément dans le **tx-pkts** ou le **rx-pkts**.

 **Conseil** : si votre compteur pour tx-pkts s'incrémente, vous transmettez des données à l'homologue. Si votre rx-pkts n'est pas incrémenté, cela signifie que les données ne sont pas reçues de votre homologue. Dans ce cas, vérifiez l'autre extrémité et confirmez si tx-pkts s'incrémente.

<#root>

TCP vEdge2# show tunnel statistics

```
TUNNEL SOURCE DEST TUNNEL MSS PROTOCOL SOURCE IP DEST IP PORT PORT SYSTEM IP LOCAL COLOR REMOTE COLOR MTU tx-
ipsec 172.16.16.147 10.88.244.181 12386 12406 10.1.0.5 public-internet default 1441 38282 5904968 38276
ipsec 172.16.16.147 10.152.201.104 12386 63364 10.1.0.0 public-internet default 1441 33421 5158814 334
```

TUNNEL	SOURCE IP	DEST IP	SOURCE PORT	DEST PORT	SYSTEM IP	LOCAL COLOR	REMOTE COLOR	MTU
ipsec	172.16.16.147	10.88.244.181	12386	12406	10.1.0.5	public-internet	default	1441
ipsec	172.16.16.147	10.152.201.104	12386	63364	10.1.0.0	public-internet	default	1441
ipsec	172.16.16.147	10.152.204.31	12386	58851	10.1.0.7	public-internet	public-internet	1441
ipsec	172.24.90.129	10.88.244.181	12426	12406	10.1.0.5	biz-internet	default	1441
ipsec	172.24.90.129	10.152.201.104	12426	63364	10.1.0.0	biz-internet	default	1441
ipsec	172.24.90.129	10.152.204.31	12426	58851	10.1.0.7	biz-internet	public-internet	1441

Une autre commande utile est **show tunnel statistics bfd** qui peut être utilisée pour vérifier le nombre de paquets BFD envoyés et reçus dans un tunnel de plan de données particulier :

```
vEdge1# show tunnel statistics bfd BFD BFD BFD BFD BFD BFD PMTU PMTU PMTU PMTU TUNNEL SOURCE DEST ECHO TX ECHO RX BFD
```

Liste d'accès

Une liste d'accès est une étape utile et nécessaire après avoir examiné le **show bfd sessions** résultat.

Maintenant que les adresses IP et les ports privés et publics sont connus, vous pouvez créer une correspondance **Access Control List (ACL)**

avec les adresses SRC_PORT, DST_PORT, SRC_IP et DST_IP.

Cela peut aider à vérifier les messages BFD envoyés et reçus.

Vous trouverez ici un exemple de configuration d'une liste de contrôle d'accès :

```
policy access-list checkbfd-out sequence 10 match source-ip 192.168.0.92/32 destination-ip 198.51.100.187/32 source-port 12426 destination-port 12426 !
default-action accept
!
access-list checkbfd-in sequence 20 match source-ip 198.51.100.187/32 destination-ip 192.168.0.92/32 source-port 12426 destination-port 12426 ! action a
vpn 0
interface ge0/0
access-list checkbfd-in in
access-list checkbfd-out out
!
!
!
```

Dans l'exemple, cette liste de contrôle d'accès utilise deux séquences. La séquence 10 correspond aux messages BFD qui sont envoyés de ce vEdge à l'homologue. La séquence 20 fait le contraire.

Elle est comparée aux ports source (**Private**) et de destination (**Public**). Si le serveur vEdge utilise la fonction NAT, assurez-vous de vérifier les ports source et de destination appropriés.

Pour vérifier les résultats sur chaque compteur de séquence, émettez la commande **show policy access-list counters <access-list name>**

```
vEdge1# show policy access-list-counters NAME COUNTER NAME PACKETS BYTES ----- checkbfd bfd-out-t
```

Traduction d'adresses réseau

Comment utiliser les outils stun-client pour détecter les cartes et les filtres NAT.

Si vous avez effectué toutes les étapes et que vous êtes derrière la NAT, l'étape suivante consiste à identifier le **UDP NAT Traversal (RFC 4787) Map and Filter** comportement.

Cet outil est utilisé pour découvrir l'adresse IP externe vEdge locale lorsque cette adresse est située derrière un périphérique NAT.

Cette commande obtient un mappage de port pour le périphérique et découvre éventuellement des propriétés sur la NAT entre le périphérique local et un serveur (serveur public : exemple de serveur stun google).



Remarque : Pour plus d'informations, consultez : [Docs Viptela - STUN Client](#)

<#root>


```
vEdge1# tools stun-client vpn 0 options "--mode full --localaddr 192.168.12.100 12386 --verbosity 2 stunclient --mode full --localaddr 192.168.12.100 stun.l.google.com in VPN 0 Binding test: success
Local address: 192.168.12.100:12386
Mapped address: 203.0.113.225:4501
Behavior test: success
```

Nat behavior: Address Dependent Mapping

Filtering test: success

Nat filtering: Address and Port Dependent Filtering


Dans les versions plus récentes du logiciel, la syntaxe peut être différente par bit :

<#root>

```
vEdge1# tools stun-client vpn 0 options "--mode full --localaddr 192.168.12.100 --localport 12386 --verb
```

Dans cet exemple, vous effectuez un test de détection NAT complet avec l'utilisation du port source UDP 12386 sur le serveur Google STUN.

Le résultat de cette commande vous donne le comportement NAT et le type de filtre NAT basé sur RFC 4787.

 **Remarque** : lorsque vous utilisez **tools stun**, n'oubliez pas d'autoriser le service STUN dans l'interface du tunnel, sinon il ne fonctionne pas. À utiliser **allow-service stun** afin de laisser passer les données étourdissantes.

<#root>

```
vEdge1# show running-config vpn 0 interface ge0/0 vpn 0 interface ge0/0 ip address 10.19.145.2/30 ! tunnel-interface encapsulation ipsec color gold max-
allow-service stun
! no shutdown ! !
```

Cette figure illustre le mappage entre la terminologie STUN (NAT Full-Cone) et RFC 4787 (NAT Behavioral for UDP).

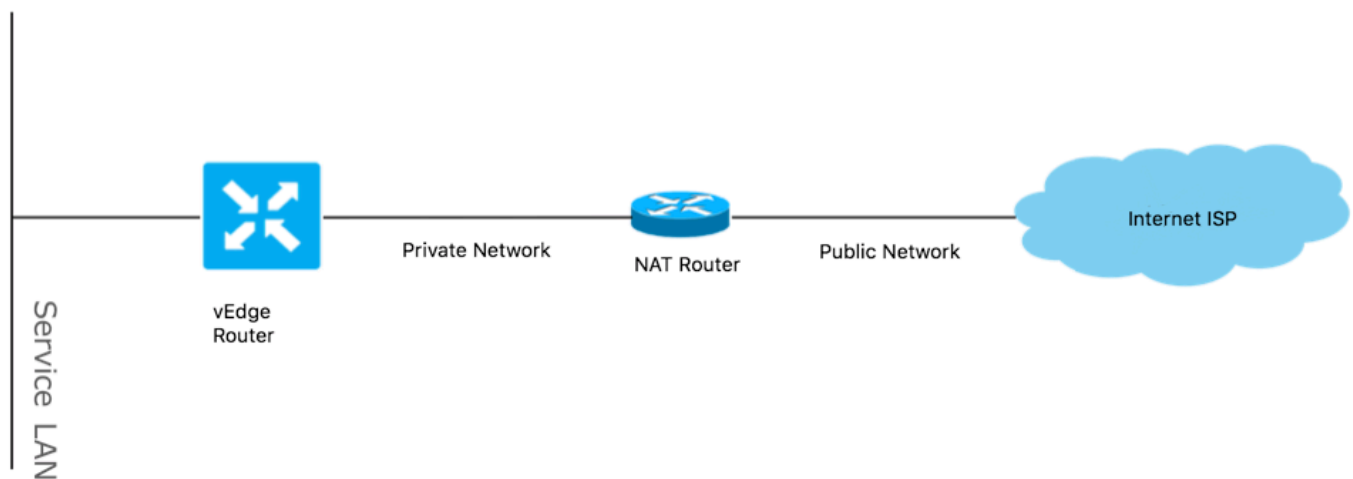
NAT Traversal Mapping Between used Viptela Terminologies		
STUN RFC 3489 Terminology	RFC 4787 Terminology	
	Mapping Behavior	Filtering Behavior
Full-cone NAT	Endpoint-Independent Mapping	Endpoint-Independent Filtering
Restricted Cone NAT	Endpoint-Independent Mapping	Address-Dependent Filtering
Port-Restricted Cone NAT	Endpoint-Independent Mapping	Address and Port-Dependent Filtering
Symmetric NAT	Address-and(or) Port-Dependent Mapping	Address-Dependent Filtering
		Address and Port-Dependent Filtering

Types NAT pris en charge pour les tunnels de plan de données « Envoi » utilisés dans l'interface de ligne de commande

Dans la plupart des cas, vos couleurs publiques comme biz-internet ou public-internet peuvent être directement attachés à l'internet.

Dans d'autres cas, il existe un périphérique NAT derrière l'interface WAN vEdge et le fournisseur d'accès Internet réel.

De cette manière, le vEdge peut avoir une adresse IP privée et l'autre périphérique (routeur, pare-feu, etc.) peut être le périphérique avec les adresses IP publiques.



Si vous avez un type NAT incorrect, il peut s'agir de l'une des raisons les plus courantes qui ne permettent pas la formation de tunnels du plan de données. Voici les types NAT pris en charge.

NAT Traversal Support		
Source	Destination	Supported (YES/NO)
Full-Cone NAT	Full-cone NAT	Yes
Full-Cone NAT	Restricted Cone NAT	Yes
Full-Cone NAT	Port-Restricted Cone NAT	Yes
Full-Cone NAT	Symmetric NAT	Yes
Restricted Cone NAT	Full-cone NAT	Yes
Restricted Cone NAT	Restricted Cone NAT	Yes
Restricted Cone NAT	Port-Restricted Cone NAT	Yes
Restricted Cone NAT	Symmetric NAT	Yes
Port-Restricted Cone NAT	Full-cone NAT	Yes
Port-Restricted Cone NAT	Restricted Cone NAT	Yes
Port-Restricted Cone NAT	Port-Restricted Cone NAT	Yes
Port-Restricted Cone NAT	Symmetric NAT	No
Symmetric NAT	Full-cone NAT	Yes
Symmetric NAT	Restricted Cone NAT	yes
Symmetric NAT	Port-Restricted Cone NAT	No
Symmetric NAT	Symmetric NAT	No

Pare-Feu

Si vous avez déjà vérifié la NAT et son non dans les types **Source** et **Destination** non pris en charge, il est possible qu'un pare-feu bloque les ports utilisés pour former les **Data Plane** tunnels.

Assurez-vous que ces ports sont ouverts dans le pare-feu pour les connexions du plan de données : **vEdge to vEdge Data Plane**:

UDP 12346 à 13156

Pour les connexions de contrôle de vEdge aux contrôleurs :

UDP 12346 à 13156

TCP 23456 à 24156

Assurez-vous d'ouvrir ces ports afin d'obtenir une connexion réussie des tunnels du plan de données.

Lorsque vous vérifiez les ports source et de destination utilisés pour les tunnels du plan de données, vous pouvez utiliser **show tunnel statistics** ou **show bfd sessions | tab** mais pas **show bfd sessions**.

Il n'affiche aucun port source, uniquement les ports de destination, comme vous pouvez le voir :

```
vEdge1# show bfd sessions SOURCE TLOC REMOTE TLOC DST PUBLIC DST PUBLIC DETECT TX SYSTEM IP SITE ID STATE COLOR COLOR
```



Remarque : pour plus d'informations sur les ports de pare-feu SD-WAN utilisés, cliquez [ici](#).

Sécurité

Si vous constatez que le compteur ACL augmente en entrée et en sortie, vérifiez plusieurs itérations **show system statistics diff** and ensure there are no drops.

<#root>

```
vEdge1# show policy access-list-counters NAME COUNTER NAME PACKETS BYTES -----
```

```
checkbfd bfd-out-to-dc1-from-br1 55 9405
```

```
bfd-in-from-dc1-to-br1 54 8478
```

Dans cette sortie, **rx_replay_integrity_drops** augmentez à chaque itération de la **show system statistics diff** command.

```
<#root>
```

```
vEdge1#show system statistics diff
```

```
rx_pkts : 5741427  
ip_fwd : 5952166  
ip_fwd_arp : 3  
ip_fwd_to_egress : 2965437  
ip_fwd_null_mcast_group : 26  
ip_fwd_null_nhop : 86846  
ip_fwd_to_cpu : 1413393  
ip_fwd_from_cpu_non_local : 15  
ip_fwd_rx_ipsec : 1586149  
ip_fwd_mcast_pkts : 26  
rx_bcast : 23957  
rx_mcast : 304  
rx_mcast_link_local : 240  
rx_implicit_acl_drops : 12832  
rx_ipsec_decap : 21  
rx_spi_ipsec_drops : 16
```

```
rx_replay_integrity_drops : 1586035
```

```
port_disabled_rx : 2
rx_invalid_qtags : 212700
rx_non_ip_drops : 1038073
pko_wred_drops : 3
bfd_tx_record_changed : 23
rx_arp_non_local_drops : 19893
rx_arp_reqs : 294
rx_arp_replies : 34330
arp_add_fail : 263
tx_pkts : 4565384
tx_mcast : 34406
port_disabled_tx : 3
tx_ipsec_pkts : 1553753
tx_ipsec_encap : 1553753
tx_pre_ipsec_pkts : 1553753
tx_pre_ipsec_encap : 1553753
tx_arp_replies : 377
tx_arp_reqs : 34337
tx_arp_req_fail : 2
bfd_tx_pkts : 1553675
bfd_rx_pkts : 21
bfd_tx_octets : 264373160
bfd_rx_octets : 3600
bfd_pmtu_tx_pkts : 78
bfd_pmtu_tx_octets : 53052
rx_icmp_echo_requests : 48
rx_icmp_network_unreach : 75465
rx_icmp_other_types : 47
tx_icmp_echo_requests : 49655
tx_icmp_echo_replies : 48
tx_icmp_network_unreach : 86849
tx_icmp_other_types : 7
vEdge1# show system statistics diff
```

```
rx_pkts : 151
ip_fwd : 157
ip_fwd_to_egress : 75
ip_fwd_null_nhop : 3
ip_fwd_to_cpu : 43
ip_fwd_rx_ipsec : 41
rx_bcast : 1
```

```
rx_replay_integrity_drops : 41
```

```
rx_invalid_qtags : 7
rx_non_ip_drops : 21
rx_arp_non_local_drops : 2
tx_pkts : 114
tx_ipsec_pkts : 40
tx_ipsec_encap : 40
tx_pre_ipsec_pkts : 40
tx_pre_ipsec_encap : 40
tx_arp_reqs : 1
```

```
bfd_tx_pkts : 40
bfd_tx_octets : 6800
tx_icmp_echo_requests : 1
vEdge1# show system statistics diff
```

```
rx_pkts : 126
ip_fwd : 125
ip_fwd_to_egress : 58
ip_fwd_null_nhop : 3
ip_fwd_to_cpu : 33
ip_fwd_rx_ipsec : 36
rx_bcast : 1
rx_implicit_acl_drops : 1
```

```
rx_replay_integrity_drops : 35
```

```
rx_invalid_qtags : 6
rx_non_ip_drops : 22
rx_arp_replies : 1
tx_pkts : 97
tx_mcast : 1
tx_ipsec_pkts : 31
tx_ipsec_encap : 31
tx_pre_ipsec_pkts : 31
tx_pre_ipsec_encap : 31
bfd_tx_pkts : 32
bfd_tx_octets : 5442
rx_icmp_network_unreach : 3
tx_icmp_echo_requests : 1
tx_icmp_network_unreach : 3
vEdge1# show system statistics diff
```

```
rx_pkts : 82
ip_fwd : 89
ip_fwd_to_egress : 45
ip_fwd_null_nhop : 3
ip_fwd_to_cpu : 24
ip_fwd_rx_ipsec : 22
rx_bcast : 1
rx_implicit_acl_drops : 1
```

```
rx_replay_integrity_drops : 24
```

```
rx_invalid_qtags : 2
rx_non_ip_drops : 14
```

```
rx_arp_replies : 1
tx_pkts : 62
tx_mcast : 1
tx_ipsec_pkts : 24
tx_ipsec_encap : 24
tx_pre_ipsec_pkts : 24
tx_pre_ipsec_encap : 24
tx_arp_reqs : 1
bfd_tx_pkts : 23
bfd_tx_octets : 3908
rx_icmp_network_unreach : 3
tx_icmp_echo_requests : 1
tx_icmp_network_unreach : 3
vEdge1# show system statistics diff
```

```
rx_pkts : 80
ip_fwd : 84
ip_fwd_to_egress : 39
ip_fwd_to_cpu : 20
ip_fwd_rx_ipsec : 24
```

```
rx_replay_integrity_drops : 22
```

```
rx_invalid_qtags : 3
rx_non_ip_drops : 12
tx_pkts : 66
tx_ipsec_pkts : 21
tx_ipsec_encap : 21
tx_pre_ipsec_pkts : 21
tx_pre_ipsec_encap : 21
bfd_tx_pkts : 21
bfd_tx_octets : 3571
```

Commencez par effectuer une analyse **request security ipsec-rekey** sur le serveur vEdge. Ensuite, passez en revue plusieurs itérations de **show system statistics diff** et voyez si vous voyez toujours **rx_replay_integrity_drops**.

Si c'est le cas, vérifiez votre configuration de sécurité.

```
vEdge1# show running-config security security
ipsec
authentication-type sha1-hmac ah-sha1-hmac
!
```

Problèmes du FAI avec le trafic marqué DSCP

Par défaut, tout le trafic de contrôle et de gestion du routeur vEdge vers les contrôleurs transite par des connexions DTLS ou TLS et est marqué avec une valeur DSCP de CS6 (48 décimales).

Pour le trafic des tunnels de data center, les routeurs vEdge utilisent l'encapsulation IPsec ou GRE pour s'envoyer mutuellement du trafic de données.

Pour la détection des défaillances du plan de données et la mesure des performances, les routeurs s'envoient périodiquement des paquets BFD.

Ces paquets BFD sont également marqués avec une valeur DSCP de CS6 (48 décimales).

Du point de vue d'ISP, ce type de trafic est considéré comme du trafic UDP avec la valeur DSCP CS6, car les routeurs vEdge et les contrôleurs SD-WAN copient par défaut le DSCP qui marque l'en-tête IP externe.

Voici à quoi cela peut ressembler si tcpdump s'exécute sur le routeur ISP de transit :

```
14:27:15.993766 IP (tos 0xc0, ttl 64, id 44063, offset 0, flags [DF], proto UDP (17), length 168) 192.168.109.5.12366 > 192.168.20.2.12346: [udp sum ok]
```

Comme on peut le voir ici, tous les paquets sont marqués avec l'octet TOS 0xc0 également appelé champ DS (qui est égal à 192 décimal, ou 110 000 00 en binaire).

Les 6 premiers bits d'ordre haut correspondent à la valeur 48 en décimal (CS6) des bits DSCP.

Les 2 premiers paquets de la sortie correspondent à un tunnel de plan de contrôle et les 2 paquets restants, à un trafic de tunnel de plan de données.

En fonction de la longueur du paquet et de la marque TOS, il peut conclure avec une grande confiance qu'il s'agissait de paquets BFD (directions RX et TX). Ces paquets sont également marqués avec CS6.

Parfois, certains fournisseurs de services (en particulier les fournisseurs de services MPLS L3 VPN/MPLS L2 VPN) maintiennent différents SLA et peuvent gérer une classe de trafic différente en fonction des marques DSCP.

Par exemple, si vous disposez d'un service haut de gamme pour hiérarchiser le trafic voix et de signalisation DSCP EF et CS6.

Puisque le trafic prioritaire est presque toujours réglementé, même si la bande passante totale d'une liaison ascendante n'est pas dépassée, pour ce type de perte de paquets de trafic peut être vu et par conséquent les sessions BFD peuvent également être instables.

Il a été constaté dans certains cas que si la file d'attente prioritaire dédiée sur le routeur du fournisseur de services est saturée, vous ne voyez aucune perte pour le trafic normal (par exemple, lorsque vous exécutez une simple **requête ping** à partir du routeur vEdge).

En effet, ce trafic est marqué avec la valeur DSCP par défaut 0 comme on peut le voir ici (octet TOS) :

```
15:49:22.268044 IP (tos 0x0, ttl 62, id 0, offset 0, flags [DF], proto UDP (17), length 142) 192.168.110.5.12366 > 192.168.109.7.12346: [no csum] UDP,
```

Mais en même temps, vos sessions BFD battent :

```
show bfd history DST PUBLIC DST PUBLIC RX TX SYSTEM IP SITE ID COLOR STATE IP PORT ENCAP TIME PKTS PKTS DEL -----
```

Et voici que **nping** est pratique pour dépanner :

```
vedge2# tools nping vpn 0 options "--tos 0x0c --icmp --icmp-type echo --delay 200ms -c 100 -q" 192.168.109.7 Nping in VPN 0 Starting Nping 0.6.47 (ht
```


Debug BFD

Si une étude plus approfondie est requise, exécutez le débogage de BFD sur le routeur vEdge.

Forwarding Traffic Manager (FTM) est responsable des opérations BFD sur les routeurs vEdge et donc vous avez besoin de **debug ftm bfd**.

Toute la sortie de débogage est stockée dans un **/var/log/tmplog/vdebug** fichier et si vous souhaitez que ces messages soient affichés sur la console (comme le **terminal monitor** comportement de Cisco IOS), vous pouvez utiliser **monitor start /var/log/tmplog/vdebug**.

Afin d'arrêter la journalisation, vous pouvez utiliser **monitor stop /var/log/tmplog/vdebug**

Voici comment le résultat recherche une session BFD qui s'arrête en raison du délai d'attente (le TLOC distant avec l'adresse IP 192.168.110.6 n'est plus accessible) :

```
log:local7.debug: May 7 16:23:09 vedge2 FTMD[674]: bfdmgr_session_update_state[1008]: BFD-session TNL 192.168.110.5:12366->192.168.110.6:123
```

Un autre débogage utile pour activer est **Tunnel Traffic Manager (TTM) events** debug is **debug ttm events**.

Voici à quoi ressemble l' **BFD DOWN** événement du point de vue de TTM :

```
log:local7.debug: May 7 16:58:19 vedge2 TTMD[683]: ttm_debug_announcement[194]: Received TTM Msg LINK_BFD, Client: ftmd, AF: LINK log:loc
```

Utiliser Packet-Trace pour capturer les paquets BFD (version 20.5 et ultérieure)

Un autre outil utile introduit dans les versions 20.5.1 et ultérieures du logiciel est packet-trace for vEdge.

Étant donné que la session BFD utilise les mêmes ports standard, généralement 12346, il est plus simple de filtrer en fonction de l'adresse IP de l'homologue.

Exemple :

```
vedge# show bfd sessions SOURCE TLOC REMOTE TLOC DST PUBLIC DST PUBLIC DETECT TX SYSTEM IP SITE ID STAT
```

La commande packet-trace doit être configurée :

```
vedge# debug packet-trace condition ingress-if ge0/0 vpn 0 source-ip 192.168.29.39
vedge# debug packet-trace condition start
vedge# debug packet-trace condition stop
```

Les résultats peuvent être affichés à l'aide des commandes show indiquées ci-dessous. Pour les paquets entrants, il y a un indicateur 'isBFD' qui est défini sur '1' (true) pour le trafic BFD.

```
vedge# show packet-trace statistics
packet-trace statistics 0
source-ip          192.168.29.39
source-port        12346
destination-ip     192.168.16.29
destination-port   12346
source-interface   ge0_0
destination-interface loop0.1
```

```

decision          FORWARD
duration          25
packet-trace statistics 1
source-ip         192.168.29.39
source-port       12346
destination-ip    192.168.16.29
destination-port  12346
source-interface  ge0_0
destination-interface loop0.1
decision          FORWARD
duration          14
packet-trace statistics 2
source-ip         192.168.29.39
source-port       12346
destination-ip    192.168.16.29
destination-port  12346
source-interface  ge0_0
destination-interface loop0.1
decision          FORWARD
duration          14

```

```
vedge# show packet-trace detail 0
```

```

=====
Pkt-id          src_ip(ingress_if)          dest_ip(egress_if)          Duration          Decision
=====
0              192.168.29.39:12346 (ge0_0)  192.168.16.29:12346 (loop0.1)  25 us            FORWARD
INGRESS_PKT:
00 50 56 84 79 be 00 50 56 84 3c b5 08 00 45 c0 00 96 ab 40 40 00 3f 11 e0 c1 c0 a8 1d 27 c0
a8 10 1d 30 3a 30 3a 00 82 00 00 a0 00 01 02 00 00 0e 3f 4b 65 07 bc 61 03 38 71 93 53 58
88 d8 08 41 95 7c 1a ff 8b cc b4 d0 d8 61 44 40 67 cc 1a 01 fd 1f c4 45 95 ea 7e 15 c9 08
2e b6 63 84 00
EGRESS_PKT:
a1 5e fe 11 00 00 00 00 00 00 00 00 00 00 04 00 0c 04 00 41 01 02 00 00 00 00 00 00 00 00
00 00 00 00 00 00 04 00 00 00 00 00 00 00 02 00 3a 30 3a 30 1d 10 a8 c0 00 00 00 00 00 00
00 00 00 00 00 00 01 00 00 00 27 1d a8 c0 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00
a4 00 01 00 00
Feature Data
-----
TOUCH : fp_proc_packet
core_id: 2
DSCP: 48
-----
TOUCH : fp_proc_packet2
core_id: 2
DSCP: 48
-----
TOUCH : fp_ip_forward
core_id: 2
DSCP: 48
-----
TOUCH : fp_ipsec_decrypt
core_id: 2
DSCP: 48
-----
FP_TRACE_FEAT_IPSEC_DATA:
src_ip : 192.168.29.39
src_port : 3784
dst_ip : 192.168.16.29
dst_port : 3784
isBFD : 1
core_id: 2
DSCP: 48

```

```

-----
TOUCH : fp_send_pkt
core_id: 2
DSCP: 48
-----
TOUCH : fp_hw_x86_pkt_free
core_id: 2
DSCP: 48
-----
TOUCH : fp_proc_remote_bfd_
core_id: 2
DSCP: 48
-----
TOUCH : BFD_ECHO_REPLY
core_id: 2
DSCP: 48
-----
TOUCH : fp_hw_x86_pkt_free
core_id: 2
DSCP: 48

```

Les paquets BFD sortants sont capturés de la même manière. Ces résultats identifient le type spécifique, qu'il s'agisse d'une requête d'écho ou d'une réponse.

```

vedge# debug packet-trace condition vpn 0 destination-ip 192.168.29.39
vedge# debug packet-trace condition start
vedge# debug packet-trace condition stop

```

```

vedge# show packet-trace statistics
packet-trace statistics 0
source-ip          192.168.16.29
source-port        3784
destination-ip     192.168.29.39
destination-port   3784
source-interface   loop0.0
destination-interface ge0_0
decision           FORWARD
duration           15
packet-trace statistics 1
source-ip          192.168.16.29
source-port        3784
destination-ip     192.168.29.39
destination-port   3784
source-interface   loop0.0
destination-interface ge0_0
decision           FORWARD
duration           66
packet-trace statistics 2
source-ip          192.168.16.29
source-port        3784
destination-ip     192.168.29.39
destination-port   3784
source-interface   loop0.0
destination-interface ge0_0
decision           FORWARD
duration           17

```

```

vedge# show packet-trace details 0
=====

```

Pkt-id	src_ip(ingress_if)	dest_ip(egress_if)	Duration	Decision
0	192.168.16.29:3784 (loop0.0)	192.168.29.39:3784 (ge0_0)	15 us	FORWARD

```

INGRESS_PKT:
45 c0 00 4f 00 00 40 00 ff 11 cc 48 c0 a8 10 1d c0 a8 1d 27 0e c8 0e c8 00 3b 00 00 80 c0 07
00 00 00 00 01 00 00 00 01 00 0f 42 40 00 0f 42 40 00 0f 42 40 01 00 0c 01 00 00 1d 3b b1
c9 89 d7 03 00 0f c0 a8 10 1d 30 3a c0 a8 1d 27 30 3a a3 96 07 3b 47 1c 60 d1 d5 76 4c 72
78 1f 9a 0d 00
EGRESS_PKT:
00 50 56 84 3c b5 00 50 56 84 79 be 08 00 45 c0 00 96 ab 40 40 00 3f 11 e0 c1 c0 a8 10 1d c0
a8 1d 27 30 3a 30 3a 00 82 00 00 a0 00 01 01 00 00 5c 3d 88 9a c7 28 23 1b e6 18 ea fe 73
1b b9 e3 79 bf d9 f4 72 41 96 c1 47 07 44 56 77 5a a2 fb 43 59 c1 97 59 47 62 21 77 d4 f4
47 8b 30 b0 00
Feature Data
-----
TOUCH : fp_send_bfd_pkt
core_id: 0
DSCP: 48
-----
TOUCH : BFD_ECHO_REPLY
core_id: 0
DSCP: 48
-----
TOUCH : fp_ipsec_loopback_f
core_id: 0
DSCP: 48
-----
TOUCH : fp_send_pkt
core_id: 0
DSCP: 48
-----
TOUCH : fp_ip_forward
core_id: 2
DSCP: 48
-----
TOUCH : fp_send_ip_packet
core_id: 2
DSCP: 48
-----
TOUCH : fp_send_pkt
core_id: 2
DSCP: 48
-----
TOUCH : fp_hw_x86_pkt_free
core_id: 2
DSCP: 48
vedge# show packet-trace details 1

```

Pkt-id	src_ip(ingress_if)	dest_ip(egress_if)	Duration	Decision
1	192.168.16.29:3784 (loop0.0)	192.168.29.39:3784 (ge0_0)	66 us	FORWARD

```

INGRESS_PKT:
45 c0 00 56 00 00 40 00 ff 11 cc 41 c0 a8 10 1d c0 a8 1d 27 0e c8 0e c8 00 42 00 00 80 c0 07
00 00 00 00 01 00 00 00 01 00 0f 42 40 00 0f 42 40 00 0f 42 40 01 00 0c 00 00 00 1d b8 35
a8 09 88 03 00 0f c0 a8 10 1d 30 3a c0 a8 1d 27 30 3a 04 00 07 01 00 05 a6 38 ff 7e 06 1e
da 23 19 d5 00
EGRESS_PKT:
00 50 56 84 3c b5 00 50 56 84 79 be 08 00 45 c0 00 9d ab 40 40 00 3f 11 e0 ba c0 a8 10 1d c0
a8 1d 27 30 3a 30 3a 00 89 00 00 a0 00 01 01 00 00 5c 3e 2d 3b 9e 81 aa 10 26 54 7f 47 5c
d8 81 4f 23 2e 3c 39 1e 94 b2 f4 fb a4 ba 98 54 73 99 8f 2e 95 d7 69 fb 91 41 96 93 03 5b
a4 e4 e8 82 00
Feature Data

```

```
-----  
TOUCH : fp_send_bfd_pkt  
core_id: 0  
DSCP: 48  
-----  
TOUCH : BFD_ECHO_REQUEST  
core_id: 0  
DSCP: 48  
-----  
TOUCH : fp_ipsec_loopback_f  
core_id: 0  
DSCP: 48  
-----  
TOUCH : fp_send_pkt  
core_id: 0  
DSCP: 48  
-----  
TOUCH : fp_ip_forward  
core_id: 2  
DSCP: 48  
-----  
TOUCH : fp_send_ip_packet  
core_id: 2  
DSCP: 48  
-----  
TOUCH : fp_send_pkt  
core_id: 2  
DSCP: 48  
-----  
TOUCH : fp_hw_x86_pkt_free  
core_id: 2  
DSCP: 48
```

Informations connexes

- [Assistance et documentation techniques - Cisco Systems](#)

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.