

Las conexiones TCP no pueden establecer cuándo el tráfico sigue rutas asimétricas

Contenido

[Introducción](#)

[Problema](#)

[Diagrama de topología](#)

[Diagnóstico](#)

[Solución](#)

[Conclusión](#)

Introducción

Este documento describe el problema que surge cuando se utilizan trayectos asimétricos para el reenvío de tráfico en el entramado SD-WAN.

Problema

Las conexiones Secure Shell (SSH) no se pueden establecer en host2 (nombre de host - edgeclient2) desde host1 (nombre de host - edgeclient1), pero al mismo tiempo SSH funciona correctamente en dirección inversa.

```
[root@edgeclient2 user]# ssh user@192.168.40.21
user@192.168.40.21's password:
Last login: Sun Feb 10 13:26:32 2019 from 192.168.60.20
[user@edgeclient1 ~]$
```

```
[root@edgeclient1 user]# ssh user@192.168.60.20
<nothing happens after that>
```

or

```
[user@edgeclient1 ~]$ ssh user@192.168.60.20
ssh_exchange_identification: Connection closed by remote host
```

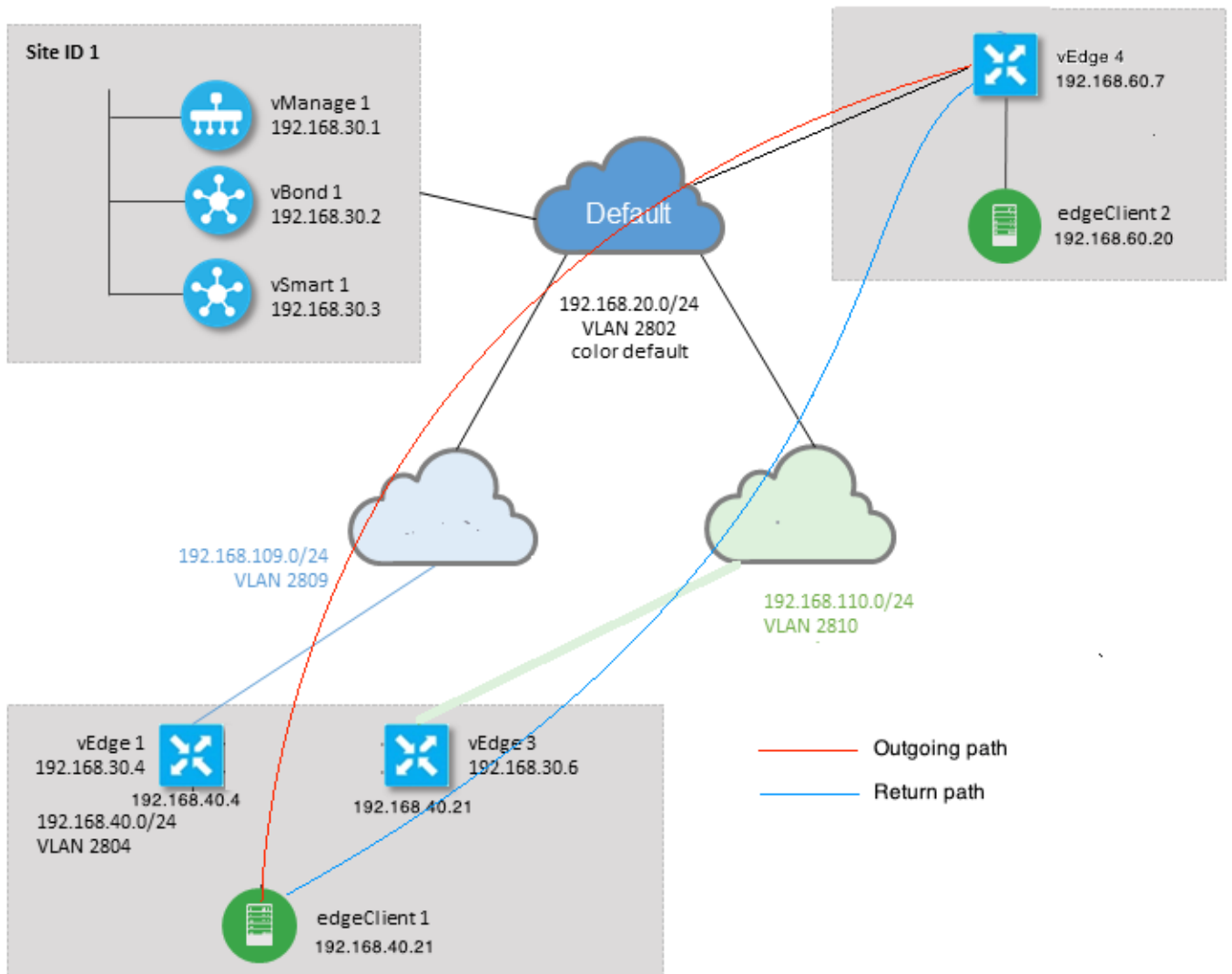
Tanto los demonios SSH de edgeclient1 como los de edgeclient2 y los clientes tienen configuraciones y conexiones buenas conocidas que se pueden establecer con éxito desde el segmento LAN local:

```
vedge4# request execute vpn 40 ssh user@192.168.60.20
user@192.168.60.20's password:
Last login: Sun Feb 10 13:28:23 2019 from 192.168.60.7
[user@edgeclient2 ~]$
```

Todas las demás aplicaciones de protocolo de control de transmisión (TCP) tienen problemas

similares.

Diagrama de topología



Diagnóstico

Estas listas de control de acceso (ACL) se configuraron y aplicaron en las direcciones correspondientes en las interfaces del lado de los servicios de vEdge1 y vEdge3:

```
policy
access-list SSH_IN
sequence 10
match
source-ip 192.168.40.21/32
destination-ip 192.168.60.20/32
!
action accept
count SSH_IN
!
!
default-action accept
!
access-list SSH_OUT
```

```

sequence 10
match
  source-ip      192.168.60.20/32
  destination-ip 192.168.40.21/32
!
action accept
  count SSH_OUT
!
!
default-action accept
!
!

```

La ACL duplicada se aplicó en vEdge4:

```

policy
access-list SSH_IN
sequence 10
match
  source-ip      192.168.60.20/32
  destination-ip 192.168.40.21/32
!
action accept
  count SSH_IN
!
!
default-action accept
!
access-list SSH_OUT
sequence 10
match
  source-ip      192.168.40.21/32
  destination-ip 192.168.60.20/32
!
action accept
  count SSH_OUT
!
!
default-action accept
!
!

```

También se habilitó la visibilidad de la aplicación en todos los routers vEdge y se comprobaron los flujos durante la fase de establecimiento de la conexión SSH:

```
vedgel# show app cflowd flows | tab ; show policy access-list-counters
```

TIME	EGRESS		INGRESS	SRC	DEST	IP	TCP		ICMP	TOTAL		
	MIN	MAX					CNTRL	INTF				
TOTAL	SRC	IP	DEST	PORT	PORT	DSCP	PROTO	BITS	OPCODE	NHOP	IP	PKTS
BYTES	LEN	LEN	START	TIME	EXP	NAME	NAME	NAME				
40	192.168.40.21	192.168.60.20	47866	22	0	6	24	0		192.168.109.7	3	
227	66	87	Sun Feb 17 14:13:25 2019	34		ge0/0	ge0/1					

```

          COUNTER
NAME      NAME      PACKETS  BYTES
-----
SSH_IN    SSH_IN    3         227
SSH_OUT   SSH_OUT   2         140

```

```
vedge3# show app cflowd flows | tab ; show policy access-list-counters
```

```

                                     TCP
TIME      EGRESS  INGRESS
TOTAL    MIN    MAX
VPN      SRC IP      DEST IP      SRC  DEST      IP      CNTRL  ICMP
BYTES   LEN    LEN  START TIME  PORT  PORT    DSCP  PROTO  BITS  OPCODE  NHOP IP      PKTS
                                     EXPIRE  NAME    NAME
-----
40      192.168.60.20  192.168.40.21  22   47866  0     6     18    0     192.168.40.21  8
480     60     60   Sun Feb 17 14:14:08 2019  51     ge0/1  ge0/0

```

```

          COUNTER
NAME      NAME      PACKETS  BYTES
-----
SSH_IN    SSH_IN    0         0
SSH_OUT   SSH_OUT   7         420

```

```
vedge4# show app cflowd flows | tab ; show policy access-list-counters
```

```

                                     TCP
TIME      EGRESS  INGRESS
TOTAL    TOTAL  MIN    MAX
VPN      SRC IP      DEST IP      SRC  DEST      IP      CNTRL  ICMP
BYTES   LEN    LEN  START TIME  PORT  PORT    DSCP  PROTO  BITS  OPCODE  NHOP IP      PKTS
                                     EXPIRE  NAME    NAME
-----
40      192.168.40.21  192.168.60.20  47866  22     0     6     2     0     192.168.60.20  4
240     60     60   Sun Feb 17 14:17:44 2019  37     ge0/2  ge0/0
40      192.168.60.20  192.168.40.21  22     47866  0     6     18    0     192.168.110.6  8
592     74     74   Sun Feb 17 14:17:44 2019  49     ge0/0  ge0/2

```

```

          COUNTER
NAME      NAME      PACKETS  BYTES
-----
SSH_IN    SSH_IN    8         592
SSH_OUT   SSH_OUT   4         240

```

Como puede ver en estas salidas, los flujos entrantes y salientes son asimétricos. edgeclient1 (192.168.40.21) está intentando establecer una sesión SSH con edgeclient2 (192.168.60.20) y el tráfico entrante viene a través de vEdge1 y el retorno de tráfico a través de vEdge3. Desde los contadores de ACL también puede ver que el número de paquetes entrantes y salientes en vEdge4 no coincide con la suma en las direcciones correspondientes en vEdge1 y vEdge3. Al mismo tiempo, no hay pérdida de paquetes al probar con ping:

```
[root@edgeclient1 user]# ping -f 192.168.60.20 -c 10000
PING 192.168.60.20 (192.168.60.20) 56(84) bytes of data.
```

```

--- 192.168.60.20 ping statistics ---
10000 packets transmitted, 10000 received, 0% packet loss, time 3076ms
rtt min/avg/max/mdev = 0.128/0.291/6.607/0.623 ms, ipg/ewma 0.307/0.170 ms

```

```
[root@edgeclient2 user]# ping -f 192.168.40.21 -c 10000
PING 192.168.40.21 (192.168.40.21) 56(84) bytes of data.
```

```
--- 192.168.40.21 ping statistics ---
10000 packets transmitted, 10000 received, 0% packet loss, time 3402ms
rtt min/avg/max/mdev = 0.212/0.318/2.766/0.136 ms, ipg/ewma 0.340/0.327 ms
```

También recopila que SSH funciona correctamente en dirección inversa y que los archivos se pueden copiar sobre scp/sftp también sin ningún problema.

Solución

Inicialmente se sospechó que se habían realizado algunas configuraciones de inspección profunda de paquetes (DPI) o políticas de datos, pero no se activó ninguna de ellas:

```
vedge3# show policy from-vsmart
% No entries found.
```

```
vedge1# show policy from-vsmart
% No entries found.
```

Pero finalmente se descubrió que la optimización de TCP estaba habilitada:

```
vedge1# show app tcp-opt active-flows
```

RX	UNOPT	PROXY	SRC	DEST		EGRESS	INGRESS	
VPN	SRC IP	DEST IP	PORT	PORT	START TIME	INTF	INTF	TX
BYTES	TCP STATE	REASON	IDENTITY			NAME	NAME	BYTES
40	192.168.40.21	192.168.60.20	47868	22	Sun Feb 17 14:18:13 2019	ge0_0	ge0_1	314
0	In-progress	-	Client-Proxy					

```
vedge1# show app tcp-opt expired-flows
```

TX	RX	UNOPT	PROXY	SRC	DEST					
TIMESTAMP	VPN	SRC IP	DEST IP	PORT	PORT	START TIME				END
TIME		BYTES	BYTES	TCP STATE	REASON	IDENTITY	DELETE	REASON		
1549819969608	40	192.168.40.21	192.168.60.7	22	56612	Sun Feb 10 18:32:49 2019	Sun			Sun
Feb 10 18:36:03 2019		5649	4405	Optimized	-	Server-Proxy	CLOSED			
1549820055487	40	192.168.40.21	192.168.60.7	22	56613	Sun Feb 10 18:34:15 2019	Sun			Sun
Feb 10 19:07:46 2019		5719	4669	Optimized	-	Server-Proxy	CLOSED			
1550408210511	40	192.168.40.21	192.168.60.20	47862	22	Sun Feb 17 13:56:50 2019	Sun			Sun
Feb 17 13:56:58 2019		401	0	Optimized	-	Client-Proxy	STATE-TIMEOUT			
1550408981634	40	192.168.40.21	192.168.60.20	47864	22	Sun Feb 17 14:09:41 2019	Sun			Sun
Feb 17 14:09:49 2019		401	0	Optimized	-	Client-Proxy	STATE-TIMEOUT			
1550409205399	40	192.168.40.21	192.168.60.20	47866	22	Sun Feb 17 14:13:25 2019	Sun			Sun
Feb 17 14:13:33 2019		227	0	Optimized	-	Client-Proxy	STATE-TIMEOUT			
1550409493042	40	192.168.40.21	192.168.60.20	47868	22	Sun Feb 17 14:18:13 2019	Sun			Sun
Feb 17 14:18:21 2019		401	0	Optimized	-	Client-Proxy	STATE-TIMEOUT			

Además, en debugs `ftm tcpopt` se puede ver el mensaje `CONN_TEARDOWN`.

```

vedge1# show log /var/log/tmplog/vdebug tail "-f"
local7.debug: Feb 17 13:56:50 vedge1 FTMD[662]: ftm_tcptopt_flow_add[268]: Created new tcpflow :-
vrid-3 192.168.40.21/47862 192.168.60.20/22
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_send_conn_tear_down[388]: Trying to
pack and send the following message to TCPD
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_send_conn_tear_down[408]: Sending
following CONN_TD msg
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_send_conn_tear_down[413]:
192.168.40.21:47862->192.168.60.20:22; vpn:40; syn_seq_num:4172167164; identity:0; cport_prime:0
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_msgq_tx[354]: Transferring size = 66
bytes data
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_send_conn_tear_down[416]: Successfully
sent conn_td msg to TCPD
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcptopt_propagate_tear_down[1038]: Sent
CONN_TEARDOWN msg to tcpd for existing tcpflow :- vrid-3 192.168.40.21/47862 192.168.60.20/22 ;
identity:CLIENT_SIDE_PROXY . Send Successful !
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcptopt_append_expired_err_flow_tbl[958]:
Appending flow vrid-3 192.168.40.21/47862 192.168.60.20/22 to the expired flow table at Sun Feb
17 13:56:58 2019
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcptopt_append_expired_err_flow_tbl[980]:
Appending flow vrid-3 192.168.40.21/47862 192.168.60.20/22 to the error flow table at Sun Feb
17 13:56:58 2019
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcptopt_flow_delete[293]: Removing tcpflow :-
vrid-3 192.168.40.21/47862 192.168.60.20/22
local7.debug: Feb 17 13:56:58 vedge1 TCPD[670]: handle_upstream_connect[538]: Error - BP NULL
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_msg_decode[254]: FTM-TCPD: Received
FTM_TCPD__PB_FTM_TCPD_MSG__E_MSG_TYPE__CONN_CLOSED msg
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_handle_conn_closed[139]: FTM-TCPD:
Received CONN_CLOSED for following C->S
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_handle_conn_closed[150]:
192.168.40.21:47862->192.168.60.20:22; vpn:40; syn_seq_num:4172167164; identity:0;
cport_prime:47862; bind_port:0
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_handle_conn_closed[184]: FTM-TCPD:
Could not find entry in FT for following flow
local7.debug: Feb 17 13:56:58 vedge1 FTMD[662]: ftm_tcpd_handle_conn_closed[185]: vrid-3
192.168.40.21/47862 192.168.60.20/22

```

Y aquí puede ver un ejemplo cuando la optimización de TCP funciona correctamente (se puede ver el mensaje CONN_EST):

```

vedge3# show log /var/log/tmplog/vdebug tail "-f -n 0"
local7.debug: Feb 17 15:41:13 vedge3 FTMD[657]: ftm_tcpd_msg_decode[254]: FTM-TCPD: Received
FTM_TCPD__PB_FTM_TCPD_MSG__E_MSG_TYPE__CONN_CLOSED msg
local7.debug: Feb 17 15:41:13 vedge3 FTMD[657]: ftm_tcpd_handle_conn_closed[139]: FTM-TCPD:
Received CONN_CLOSED for following C->S
local7.debug: Feb 17 15:41:13 vedge3 FTMD[657]: ftm_tcpd_handle_conn_closed[150]:
192.168.40.21:47876->192.168.60.20:22; vpn:40; syn_seq_num:2779178897; identity:0;
cport_prime:47876; bind_port:0
local7.debug: Feb 17 15:41:15 vedge3 FTMD[657]: ftm_tcpd_msg_decode[258]: FTM-TCPD: Received
FTM_TCPD__PB_FTM_TCPD_MSG__E_MSG_TYPE__CONN_EST msg
local7.debug: Feb 17 15:41:15 vedge3 FTMD[657]: ftm_tcpd_handle_conn_est[202]: FTM-TCPD:
Received CONN_EST for following C->S
local7.debug: Feb 17 15:41:15 vedge3 FTMD[657]: ftm_tcpd_handle_conn_est[213]:
192.168.40.21:47878->192.168.60.20:22; vpn:40; syn_seq_num:2690847868; identity:0;
cport_prime:47878; bind_port:0
local7.debug: Feb 17 15:41:15 vedge3 FTMD[657]: ftm_tcptopt_flow_add[268]: Created new tcpflow :-
vrid-3 192.168.40.21/47878 192.168.60.20/22

```

Conclusión

La optimización de TCP requiere que los flujos sean simétricos, por lo que para resolver este problema, la optimización de TCP debe ser inhabilitada (**no vpn 40 tcp-Optimization**) o se debe **crear una política de datos para obligar a los flujos TCP a tomar el mismo trayecto en ambas direcciones**. Puede encontrar más información al respecto en la sección [Guía de diseño de SD-WAN](#) Simetría de tráfico para DPI, página 23.