

TCP-Verbindungen stellen nicht fest, wann Datenverkehr auf asymmetrische Pfade folgt

Inhalt

[Einführung](#)

[Problem](#)

[Topologiediagramm](#)

[Diagnose](#)

[Lösung](#)

[Schlussfolgerung](#)

Einführung

Dieses Dokument beschreibt ein Problem, das auftritt, wenn asymmetrische Pfade für die Weiterleitung von Datenverkehr in einer SD-WAN-Fabric verwendet werden.

Problem

Secure Shell (SSH)-Verbindungen können nicht für host2 (Hostname - edgeclient2) von host1 (Hostname - edgeclient1) eingerichtet werden, aber gleichzeitig funktioniert SSH in umgekehrter Richtung einwandfrei.

```
[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>
```

oder

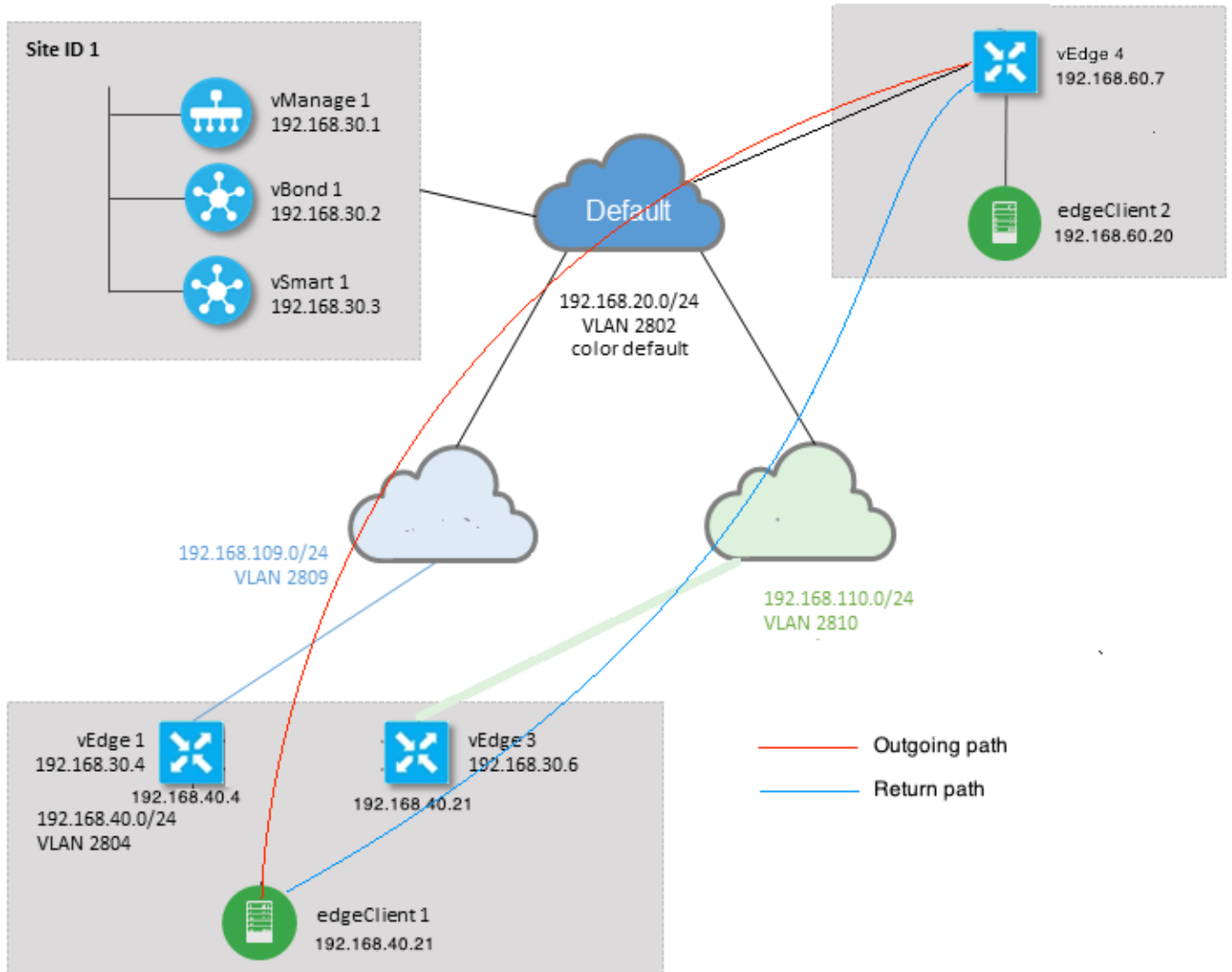
```
[user@edgeclient1 ~]$ ssh user@192.168.60.20
ssh_exchange_identification: Connection closed by remote host
```

Sowohl edgeclient1- als auch edgeclient2-SSH-Daemons und Clients verfügen über zweifelsfrei funktionierende Konfigurationen, und Verbindungen können aus lokalem LAN-Segment erfolgreich hergestellt werden:

```
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 ~]$
```

Alle anderen Transmission Control Protocol (TCP)-Anwendungen haben ähnliche Probleme.

Topologiediagramm



Diagnose

Diese Zugriffskontrolllisten (Access Control Lists, ACLs) wurden in entsprechenden Richtungen auf den serviceseitigen Schnittstellen von vEdge1 und vEdge3 konfiguriert und angewendet:

```
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
!
!

```

Gespiegelte ACL wurde auf vEdge4 angewendet:

```

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
!
!

```

Darüber hinaus wurde die Anwendungstransparenz auf allen vEdge-Routern aktiviert, und die Datenflüsse wurden während der Phase der Einrichtung der SSH-Verbindung überprüft:

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

TIME	EGRESS		INGRESS		TCP							TOTAL
	MIN	MAX	SRC	DEST	SRC	DEST	IP	CNTRL	ICMP	TOTAL		
VPN	SRC IP	DEST IP	PORT	PORT	DSCP	PROTO	INTF	INTF	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
-----
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
-----
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
```

Wie Sie an diesen Ausgängen sehen können, sind ein- und ausgehende Datenflüsse asymmetrisch. edgeclient1 (192.168.40.21) versucht, eine SSH-Sitzung mit edgeclient2 (192.168.60.20) einzurichten, und eingehender Datenverkehr wird über vEdge1 gesendet und Datenverkehr über vEdge3 zurückgegeben. Aus den ACL-Zählern können Sie auch sehen, dass die Anzahl der ein- und ausgehenden Pakete auf vEdge4 nicht mit der Summe in den entsprechenden Richtungen auf vEdge1 und vEdge3 übereinstimmt. Beim Testen mit dem Ping gehen keine Paketverluste ein:

```
[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
```

Außerdem können Sie auch ohne Probleme Dateien über scp/sftp kopieren.

Lösung

Einige DPI-Konfigurationen (Deep Packet Inspection) oder Datenrichtlinien wurden anfänglich vermutet, aber keine dieser Richtlinien wurde aktiviert:

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

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

Schließlich wurde jedoch festgestellt, dass die TCP-Optimierung aktiviert war:

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

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

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

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

Außerdem kann in **debugs** die Meldung `ftm tcptopt CONN_TEARDOWN` angezeigt werden.

```

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

```

Hier sehen Sie ein Beispiel, in dem die TCP-Optimierung ordnungsgemäß funktioniert (CONN_EST-Nachricht ist sichtbar):

```

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

```

Schlussfolgerung

Für die TCP-Optimierung müssen die Datenflüsse symmetrisch sein. Um dieses Problem zu beheben, muss entweder die TCP-Optimierung deaktiviert (**keine VPN 40-TCP-Optimierung**) oder **eine Datenrichtlinie erstellt werden, um zu erzwingen, dass TCP-Datenflüsse in beide Richtungen denselben Pfad verwenden**. Weitere Informationen hierzu finden Sie im [SD-WAN-Designleitfaden](#) im Abschnitt Datenverkehrssymmetrie für DPI, Seite 23.