

Armadillo をルータとして利用し、PC1 から PC2 への RDP・Ping を実施する際の問題について

現在、Armadillo をルータとして使用し、PC1 (Windows 10) から PC2 (Windows) に対して **RDP 接続や ping を行いたい** です。

しかし、PC1 から PC2 (192.168.250.170) への通信ができない状況です。

ネットワーク構成 ↓

PC1 (Windows 10)

- Ethernet: 192.168.100.32
- VPN (OpenVPN クライアント): 10.8.0.1

↓

Internet (LTE回線経由の OpenVPN)

↓

Armadillo (ルータ)

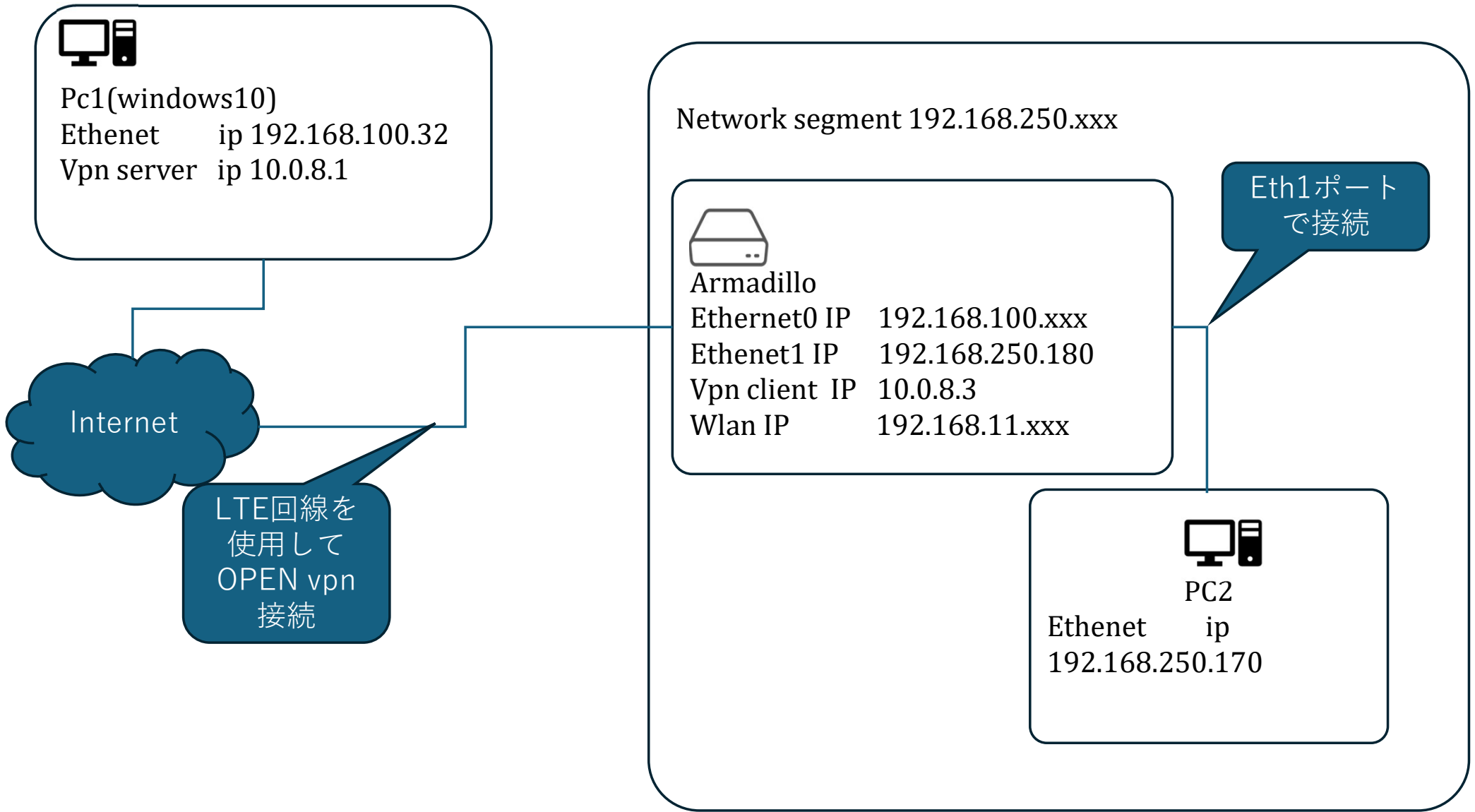
- VPN クライアント: 10.8.0.3
- Ethernet0: 192.168.100.xxx
- Ethernet1: 192.168.250.180

↓

PC2 (Windows)

- Ethernet: 192.168.250.170

次項に図にしています。



実施した確認と結果

1.VPN 経由の ICMP 通信確認

- PC1 (10.8.0.1) から Armadillo (10.8.0.3) へ Ping → OK
- PC1 (10.8.0.1) から PC2 (192.168.250.180) へ Ping → NG
- PC1 から PC2 へ RDP 接続 → NG

2. Armadillo でのパケットキャプチャ

tcpdump -i eth1 icmp → パケットなし

tcpdump -i tun0 icmp → ICMP リクエストとリプライは確認できる

3. iptables 設定

- FORWARD チェイン

Chain FORWARD (policy ACCEPT 2000 packets, 246K bytes)

pkts	bytes	target	prot	opt	in	out	source	destination
0	0	ACCEPT	0	--	tun0	eth1	0.0.0.0/0	0.0.0.0/0
0	0	ACCEPT	0	--	eth1	tun0	0.0.0.0/0	0.0.0.0/0

- NAT (POSTROUTING)

Chain POSTROUTING (policy ACCEPT 3503 packets, 282K bytes)

pkts	bytes	target	prot	opt	in	out	source	destination
0	0	MASQUERADE	0	--	*	eth1	10.8.0.0/24	0.0.0.0/0

実施した確認と結果

P C 1 側 windowsのroute print 出力

```
PS C:\WINDOWS\system32> route print
=====
インターフェイス一覧
15.....Wintun Userspace Tunnel
11...0a 00 27 00 00 0b .....VirtualBox Host-Only Ethernet Adapter
18...00 ff d2 0a e4 4b .....TAP-Windows Adapter V9
20...c8 d9 d2 07 b0 72 .....Intel(R) Ethernet Connection (7) I219-LM
1.....Software Loopback Interface 1
=====

IPv4 ルート テーブル
=====
アクティブ ルート:
ネットワーク宛先      ネットマスク      ゲートウェイ      インターフェイス      メトリ
0.0.0.0                0.0.0.0           192.168.100.201   192.168.100.32        291
10.8.0.0               255.255.255.0     リンク上         10.8.0.1              281
10.8.0.1               255.255.255.255   リンク上         10.8.0.1              281
10.8.0.255             255.255.255.255   リンク上         10.8.0.1              281
127.0.0.0              255.0.0.0         リンク上         127.0.0.1            331
127.0.0.1              255.255.255.255   リンク上         127.0.0.1            331
127.255.255.255        255.255.255.255   リンク上         127.0.0.1            331
192.168.56.0           255.255.255.0     リンク上         192.168.56.1         281
192.168.56.1           255.255.255.255   リンク上         192.168.56.1         281
192.168.56.255         255.255.255.255   リンク上         192.168.56.1         281
192.168.100.0          255.255.255.0     リンク上         192.168.100.32       291
192.168.100.32         255.255.255.255   リンク上         192.168.100.32       291
192.168.100.255        255.255.255.255   リンク上         192.168.100.32       291
192.168.250.0          255.255.255.0     10.8.0.3         10.8.0.1              26
224.0.0.0              240.0.0.0         リンク上         127.0.0.1            331
224.0.0.0              240.0.0.0         リンク上         192.168.56.1         281
224.0.0.0              240.0.0.0         リンク上         10.8.0.1              281
224.0.0.0              240.0.0.0         リンク上         192.168.100.32       291
255.255.255.255        255.255.255.255   リンク上         127.0.0.1            331
255.255.255.255        255.255.255.255   リンク上         192.168.56.1         281
255.255.255.255        255.255.255.255   リンク上         10.8.0.1              281
255.255.255.255        255.255.255.255   リンク上         192.168.100.32       291
=====
固定ルート:
ネットワーク アドレス      ネットマスク      ゲートウェイ アドレス      メトリック
0.0.0.0                    0.0.0.0          192.168.100.201   既定
192.168.250.0              255.255.255.0     10.8.0.3          1
=====
```

質問

PC1 から PC2 (192.168.250.180) へ RDP や ping ができるようにするには、このほか、どのような設定を追加すればよいでしょうか？

Armadillo でルータとして適切に動作させるために追加で確認すべきポイントはありますか？

```
tcpdump -i eth1 icmp
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on eth1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
```

```
0 packets captured  
0 packets received by filter  
0 packets dropped by kernel
```

```
tcpdump -i tun0 icmp
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
```

```
10:43:44.963445 IP 10.8.0.1 > 10.8.0.3: ICMP echo request, id 28, seq 8666, length 40  
10:43:44.963502 IP 10.8.0.3 > 10.8.0.1: ICMP echo reply, id 28, seq 8666, length 40  
10:43:45.980214 IP 10.8.0.1 > 10.8.0.3: ICMP echo request, id 28, seq 8668, length 40  
10:43:45.980280 IP 10.8.0.3 > 10.8.0.1: ICMP echo reply, id 28, seq 8668, length 40  
10:43:47.000555 IP 10.8.0.1 > 10.8.0.3: ICMP echo request, id 28, seq 8670, length 40  
10:43:47.000622 IP 10.8.0.3 > 10.8.0.1: ICMP echo reply, id 28, seq 8670, length 40
```

```
6 packets captured  
6 packets received by filter  
0 packets dropped by kernel
```

```
iptables -L FORWARD -v -n
```

```
Chain FORWARD (policy ACCEPT 2000 packets, 246K bytes)
```

pkts	bytes	target	prot	opt	in	out	source	destination
0	0	ACCEPT	0	--	tun0	eth1	0.0.0.0/0	0.0.0.0/0
0	0	ACCEPT	0	--	eth1	tun0	0.0.0.0/0	0.0.0.0/0

```
iptables -t nat -L -v -n
```

```
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
```

pkts	bytes	target	prot	opt	in	out	source	destination
------	-------	--------	------	-----	----	-----	--------	-------------

```
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
```

pkts	bytes	target	prot	opt	in	out	source	destination
------	-------	--------	------	-----	----	-----	--------	-------------

```
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
```

pkts	bytes	target	prot	opt	in	out	source	destination
------	-------	--------	------	-----	----	-----	--------	-------------

```
Chain POSTROUTING (policy ACCEPT 3503 packets, 282K bytes)
```

pkts	bytes	target	prot	opt	in	out	source	destination
0	0	MASQUERADE	0	--	*	eth1	10.8.0.0/24	0.0.0.0/0