

组网及说明

The image shows a Wireshark packet capture of a TCP connection. The interface is titled 'tcp\_stream eq 0'. The packet list pane shows the following details:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.252.3	192.168.4.9	TCP	74	41967 → 23899 [SYN] Seq=0 Win=65535 Len=0 MSS=1360 SACK_PERM=1 TSval=3618073854 TSecr=0 NS=1824
2	0.000034	192.168.4.9	192.168.252.3	TCP	74	23899 → 41967 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1360 SACK_PERM=1 TSval=3229299781 TSecr=3618073854 MS
3	0.000697	192.168.252.3	192.168.4.9	TCP	66	41967 → 23899 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3618073855 TSecr=3229299781
4	5.652282	192.168.252.3	192.168.4.9	TCP	66	41967 → 23899 [FIN, ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3618079507 TSecr=3229299781
5	5.652344	192.168.4.9	192.168.252.3	TCP	66	23899 → 41967 [FIN, ACK] Seq=1 Ack=2 Win=65536 Len=0 TSval=3229305433 TSecr=3618079507
6	5.863070	192.168.4.9	192.168.252.3	TCP	66	[TCP Retransmission] 23899 → 41967 [FIN, ACK] Seq=1 Ack=2 Win=65536 Len=0 TSval=3229305645 TSecr=3618079507
7	5.869813	192.168.252.3	192.168.4.9	TCP	66	[TCP Retransmission] 41967 → 23899 [FIN, ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3618079724 TSecr=3229299781
8	5.869825	192.168.4.9	192.168.252.3	TCP	78	[TCP Dup ACK 5#1] 23899 → 41967 [ACK] Seq=2 Ack=2 Win=65536 Len=0 TSval=3229305650 TSecr=3618079724 SLE=1 SRE=
9	6.083882	192.168.4.9	192.168.252.3	TCP	66	[TCP Retransmission] 23899 → 41967 [FIN, ACK] Seq=1 Ack=2 Win=65536 Len=0 TSval=3229305865 TSecr=3618079724
12	6.513879	192.168.4.9	192.168.252.3	TCP	66	[TCP Retransmission] 23899 → 41967 [FIN, ACK] Seq=1 Ack=2 Win=65536 Len=0 TSval=3229306295 TSecr=3618079724
13	7.413872	192.168.4.9	192.168.252.3	TCP	66	[TCP Retransmission] 23899 → 41967 [FIN, ACK] Seq=1 Ack=2 Win=65536 Len=0 TSval=3229307105 TSecr=3618079724

在docker的宿主机和容器将网络通信，可以ping通，但建立的ip:port链接timeout或链接丢失，抓包如图：

## 问题描述

### 连接异常网卡信息

```
# ethtool enp3s0f0 Settings for enp3s0f0: Supported ports: [ FIBRE ] Supported link modes: 1000baseT/Full Supported pause frame use: Symmetric Supports auto-negotiation: No Supported FEC modes: Not reported Advertised link modes: 1000baseT/Full Advertised pause frame use: Symmetric Advertised auto-negotiation: No Advertised FEC modes: Not reported Speed: 10000Mb/s Duplex: Full Port: FIBRE PHYAD: 0 Transceiver: internal Auto-negotiation: off Supports Wake-on: d Wake-on: d Current message level: 0x00000007 (7) drv probe link Link detected: yes # ethtool -i enp3s0f0 driver: txgbe version: 1.1.12 firmware-version: 0x00020004 expansion-rom-version: bus-info: 0000:03:00.0 supports-statistics: yes supports-test: yes supports-eeprom-access: yes supports-register-dump: yes supports-priv-flags: no
```

### 连接正常网卡信息

```
# ethtool enp6s0f0 Settings for enp6s0f0: Supported ports: [ FIBRE ] Supported link modes: 1000baseSR/Full Supported pause frame use: Symmetric Supports auto-negotiation: Yes Supported FEC modes: Not reported Advertised link modes: 1000baseSR/Full Advertised pause frame use: No Advertised auto-negotiation: Yes Advertised FEC modes: Not reported Speed: 10000Mb/s Duplex: Full Port: FIBRE PHYAD: 0 Transceiver: internal Auto-negotiation: off Supports Wake-on: d Wake-on: d Current message level: 0x00000007 (7) drv probe link Link detected: yes 正常主机: # ethtool -i enp6s0f0 driver: i40e version: 2.3.2-k firmware-version: 6.01 0x800035cf 1.1747.0 expansion-rom-version: bus-info: 0000:06:00.0 supports-statistics: yes supports-test: yes supports-eeprom-access: yes supports-register-dump: yes supports-priv-flags: yes
```

## 过程分析

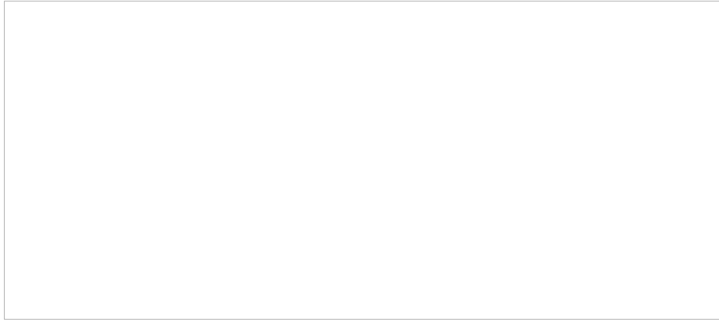
当前虚拟化场景下，虚拟网卡均不会对报文进行IP层的checksum校验。导致请求报文的IP层checksum异常，目的服务器拒绝对其请求进行相应。在网卡硬件本身不支持校验数据包功能之前是由Linux内核读取IP数据包校验的，关闭硬件上的校验和功能后，linux内核仍然会对数据包做校验和，不影响数据安全

网卡开启了Checksum Offload(硬件校验和) 功能，系统将Checksum的计算工作交由网卡去计算，在高速网络交换的情况下可以减轻CPU的工作负荷。

## 解决方法

解决方法:

- 1、 统一关闭虚拟port的tx checksum功能，统一让虚拟机或namespace协议栈去计算L4 checksum  
执行#ethtool -K eth1 tx off
- 2、 修改应用代码支持硬件CSUM功能，统一设置让出物理网卡的包由硬件CSUM  
数据包处理过程说明如下:



COE (Checksum Offload Engine) : 支持硬件checksum 计算和校验

更进一步了解相关信息可参考链接

<https://huataihuang.gitbooks.io/cloud->

[atlas/content/network/packet\\_analysis/tcpdump/udp\\_tcp\\_checksum\\_errors\\_from\\_tcpdump\\_nic\\_hardware\\_offloading.html](https://huataihuang.gitbooks.io/cloud-atlas/content/network/packet_analysis/tcpdump/udp_tcp_checksum_errors_from_tcpdump_nic_hardware_offloading.html)

**checksum说明参考**

<https://datatracker.ietf.org/doc/html/rfc791> #IP协议rfc791说明

<https://datatracker.ietf.org/doc/html/rfc1071> #校验和算法rfc1071说明

<https://www.intel.com/content/dam/doc/manual/pci-pci-x-family-gbe-controllers-software-dev-manual.pdf>

[https://www.wireshark.org/docs/wsug\\_html\\_chunked/ChAdvChecksums.html](https://www.wireshark.org/docs/wsug_html_chunked/ChAdvChecksums.html)

**关闭硬件上的校验和功能，对传输数据本身是没有影响的参考如下:**

<http://docs.gz.ro/node/282>

**如果网卡不支持则，在linux系统里的TCP/IP协议栈来完成数据校验。**

**参考链接:**

<http://docs.gz.ro/tuning-network-cards-on-linux.html>

