shell 脚本编程，生成 TCP 活动状况报告

设计

`netstat --statistics` / `netstat -s`

`netstat -s` 的输出如下:

```
ubuntu@VM-16-12-ubuntu:~$ netstat --statistics
Ip:
    Forwarding: 1
    11624 total packets received
    2 with invalid addresses
    0 forwarded
    0 incoming packets discarded
    11622 incoming packets delivered
    9904 requests sent out
Icmp:
    592 ICMP messages received
    0 input ICMP message failed
    ICMP input histogram:
        echo requests: 592
    592 ICMP messages sent
    0 ICMP messages failed
    ICMP output histogram:
        echo replies: 592
IcmpMsg:
        InType8: 592
        OutType0: 592
```

```
Tcp:
    466 active connection openings
    0 passive connection openings
    1 failed connection attempts
    25 connection resets received
    3 connections established
    10148 segments received
    8451 segments sent out
    7 segments retransmitted
    0 bad segments received
    43 resets sent
Udp:
    869 packets received
    0 packets to unknown port received
    0 packet receive errors
    869 packets sent
    0 receive buffer errors
    0 send buffer errors
UdpLite:
```

```
TcpExt:
    1 resets received for embryonic SYN_RECV sockets
    6 TCP sockets finished time wait in fast timer
    12 delayed acks sent
    6894 packet headers predicted
    1459 acknowledgments not containing data payload received
    183 predicted acknowledgments
    TCPLostRetransmit: 4
    TCPTimeouts: 6
    TCPLossProbes: 1
    TCPBacklogCoalesce: 47
    3 connections reset due to unexpected data
    25 connections reset due to early user close
    TCPRcvCoalesce: 2193
    TCPAutoCorking: 1
    TCPSynRetrans: 1
    TCPOrigDataSent: 1722
    TCPKeepAlive: 6
    TCPDelivered: 2164
    TcpTimeoutRehash: 6
IpExt:
    InMcastPkts: 13
    OutMcastPkts: 2
    InOctets: 53281475
    OutOctets: 986048
    InMcastOctets: 468
    OutMcastOctets: 80
    InNoECTPkts: 44279
MPTcpExt:
```

下面的信息表示 TCP 发送了多少报文，接收了多少报文

```
    10148 segments received
    8451 segments sent out
```

因为 segment 是 TCP 报文的专用名词，因此可以用 `grep` 直接获得对应行的信息

```
ubuntu@VM-16-12-ubuntu:~$ netstat -s | grep -E '[0-9]+ segments received'
    13286 segments received
ubuntu@VM-16-12-ubuntu:~$ netstat -s | grep -E '[0-9]+ segments sent out'
    11366 segments sent out
```

设置两个变量 recvPktTotal, sendPktTotal 分别表示目前为止收到的包的数量和发送的包的数量

```
ubuntu@VM-16-12-ubuntu:~$ recvPktTotal=$(netstat -s | grep -E '[0-9]+ segments received' | grep -o -E '[0-9]+')
ubuntu@VM-16-12-ubuntu:~$ echo $recvPktTotal
18974
ubuntu@VM-16-12-ubuntu:~$ sendPktTotal=$(netstat -s | grep -E '[0-9]+ segments sent out' | grep -o -E '[0-9]+')
ubuntu@VM-16-12-ubuntu:~$ echo $sendPktTotal
16922
```

datetime 输出

```
ubuntu@VM-16-12-ubuntu:~$ date +"%Y-%m-%d %H:%M"
2024-05-26 10:06
```

现在编写 shell 脚本

首先定义三个工具函数

第一个工具函数，获得目前为止收到的 TCP 报文数量

```
# get the total received tcp packet number
getRecvPktTotal() {
        netstat -s | grep -E '[0-9]+ segments received' | grep -o -E '[0-9]+'
        # netstat -s | grep -E '[0-9]+ segments received' | awk '{print $1}'
}
```

第二个工具函数，获得目前为止发送的 TCP 报文数量

```
# get the total sent tcp packet number
getSendPktTotal() {
        netstat -s | grep -E '[0-9]+ segments sent out' | grep -o -E '[0-9]+'
        # netstat -s | grep -E '[0-9]+ segments sent out' | awk '{print $1}'
}
```

第三个工具函数，根据现在收发的报文数量和上一次收发的报文数量的差值，决定输出 ' '，'+' 还是 '-'

```
# based on sum of packet number information set the output flag ' ' or '+' or '-'
diffFlag() {
        if [ "$1" -le 10 ] && [ "$1" -ge 0 ]; then
                echo ' '
        elif [ "$1" -gt 10 ]; then
                echo '+'
        else
                echo '-'
        fi
}
```

循环开始准备

因为第一次输出没有办法与之前比较，因此从循环中剥离出来单独处理

```
prevRecvPktTotal=$(getRecvPktTotal)
prevSendPktTotal=$(getSendPktTotal)

sleep $sleepTime

datetime=$(date +"%Y-%m-%d %H:%M")
recvPktTotal=$(getRecvPktTotal)
sendPktTotal=$(getSendPktTotal)
recvPktCurr=$((recvPktTotal - prevRecvPktTotal))
sendPktCurr=$((sendPktTotal - prevSendPktTotal))
sumPkt=$((recvPktCurr + sendPktCurr))

prevRecvPktTotal=$recvPktTotal
prevSendPktTotal=$sendPktTotal
prevSumPkt=$sumPkt

echo "$datetime $recvPktCurr $sendPktCurr $sumPkt"
```

主循环

先 `sleep $sleepTime` 休眠一段时间，然后获取该段时间收发的报文数量，并与之前对比，最后输出，循环往复

```
while true
do
        sleep $sleepTime

        datetime=$(date +"%Y-%m-%d %H:%M")
        recvPktTotal=$(getRecvPktTotal)
        sendPktTotal=$(getSendPktTotal)
        recvPktCurr=$((recvPktTotal - prevRecvPktTotal))
        sendPktCurr=$((sendPktTotal - prevSendPktTotal))
        sumPkt=$((recvPktCurr + sendPktCurr))

        difference=$((sumPkt - prevSumPkt))
        flag=$(diffFlag "$difference")

        prevRecvPktTotal=$recvPktTotal
        prevSendPktTotal=$sendPktTotal
        prevSumPkt=$sumPkt

        echo "$datetime $recvPktCurr $sendPktCurr $sumPkt $flag"

done
```

测试

将 `sleepTime` 设置为较少的值可以帮助调试

`sleepTime=5` 运行结果如下

```
ubuntu@VM-16-12-ubuntu:~$ ./tcpwatch.sh 5
2024-05-26 11:28 33 29 62
2024-05-26 11:28 9 9 18 -
2024-05-26 11:28 22 23 45 +
2024-05-26 11:28 8 8 16 -
2024-05-26 11:28 23 20 43 +
2024-05-26 11:28 14 15 29 -
2024-05-26 11:28 21 21 42 +
2024-05-26 11:28 8 8 16 -
2024-05-26 11:29 27 25 52 +
2024-05-26 11:29 9 10 19 -
2024-05-26 11:29 19 19 38 +
^C
```

`sleepTime=10` 运行结果如下

```
ubuntu@VM-16-12-ubuntu:~$ ./tcpwatch.sh 10
2024-05-26 11:30 40 38 78
2024-05-26 11:30 37 34 71 -
2024-05-26 11:30 28 29 57 -
2024-05-26 11:30 36 34 70 +
2024-05-26 11:30 26 26 52 -
2024-05-26 11:31 33 31 64 +
2024-05-26 11:31 40 40 80 +
2024-05-26 11:31 38 36 74 -
^C
```

没有提供命令行参数的情况下默认为 60

```
ubuntu@VM-16-12-ubuntu:~$ ./tcpwatch.sh
2024-05-26 11:36 199 199 398
2024-05-26 11:37 209 213 422 +
2024-05-26 11:38 198 200 398 -
2024-05-26 11:39 194 196 390 -
```

源码

```
#! /bin/bash

if [ $# -eq 1 ]; then
        sleepTime=$1
else
        sleepTime=60
fi

# get the total received tcp packet number
getRecvPktTotal() {
        netstat -s | grep -E '[0-9]+ segments received' | grep -o -E '[0-
9]+'
        # netstat -s | grep -E '[0-9]+ segments received' | awk '{print
$1}'
}

# get the total sent tcp packet number
getSendPktTotal() {
        netstat -s | grep -E '[0-9]+ segments sent out' | grep -o -E '[0-
9]+'
        # netstat -s | grep -E '[0-9]+ segments sent out' | awk '{print
$1}'
}

# based on sum of packet number information set the output flag ' ' or '+'
or '-'
diffFlag() {
        if [ "$1" -le 10 ] && [ "$1" -ge 0 ]; then
                echo ' '
        elif [ "$1" -gt 10 ]; then
                echo '+'
        else
                echo '-'
        fi
}

prevRecvPktTotal=$(getRecvPktTotal)
prevSendPktTotal=$(getSendPktTotal)

sleep $sleepTime

datetime=$(date +"%Y-%m-%d %H:%M")
recvPktTotal=$(getRecvPktTotal)
sendPktTotal=$(getSendPktTotal)
recvPktCurr=$((recvPktTotal - prevRecvPktTotal))
sendPktCurr=$((sendPktTotal - prevSendPktTotal))
sumPkt=$((recvPktCurr + sendPktCurr))

prevRecvPktTotal=$recvPktTotal
```

```
prevSendPktTotal=$sendPktTotal
prevSumPkt=$sumPkt

echo "$datetime $recvPktCurr $sendPktCurr $sumPkt"

while true
do
        sleep $sleepTime

        datetime=$(date +"%Y-%m-%d %H:%M")
        recvPktTotal=$(getRecvPktTotal)
        sendPktTotal=$(getSendPktTotal)
        recvPktCurr=$((recvPktTotal - prevRecvPktTotal))
        sendPktCurr=$((sendPktTotal - prevSendPktTotal))
        sumPkt=$((recvPktCurr + sendPktCurr))

        difference=$((sumPkt - prevSumPkt))
        flag=$(diffFlag "$difference")

        prevRecvPktTotal=$recvPktTotal
        prevSendPktTotal=$sendPktTotal
        prevSumPkt=$sumPkt

        echo "$datetime $recvPktCurr $sendPktCurr $sumPkt $flag"

done
```