ELEC-E7130 - Internet Traffic Measurements and Analysis

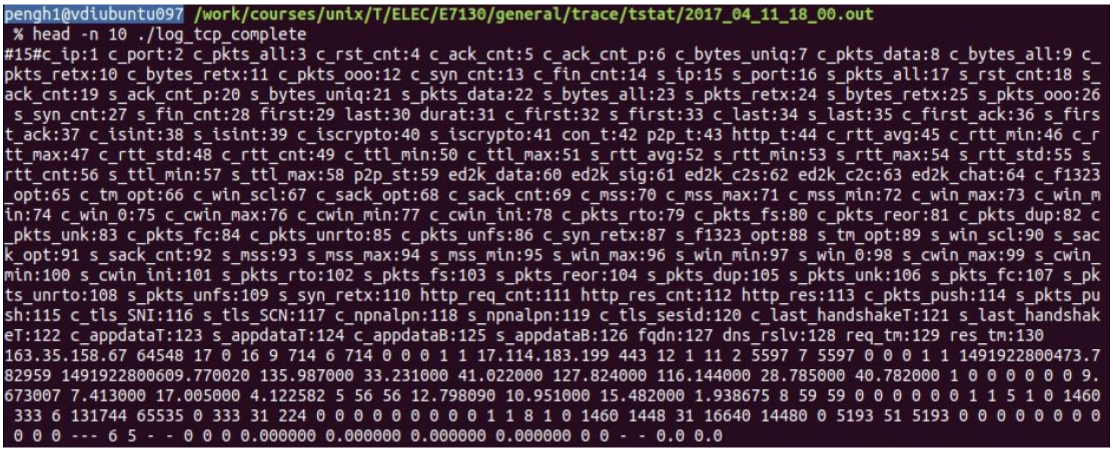
Assignment 1 Learning to use tools for measurement and analysis

Haibi Peng 875552

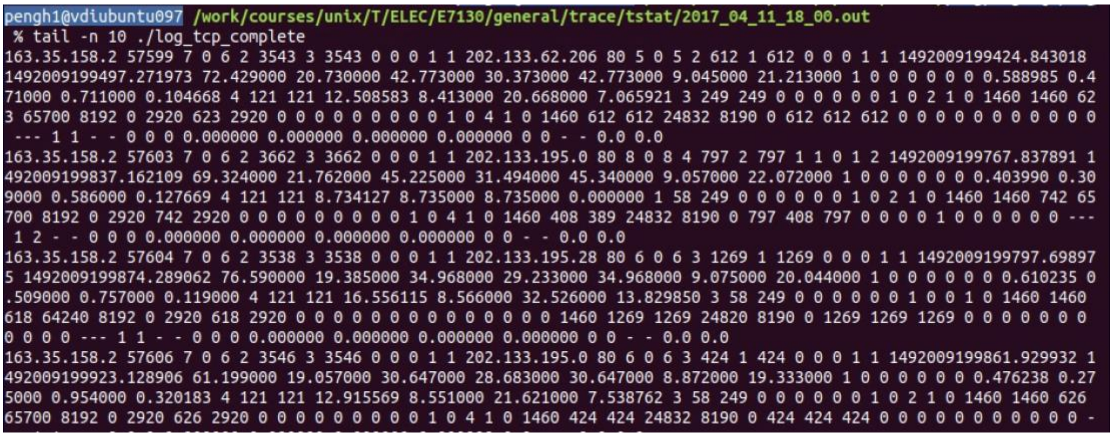
Task 1: Process CSV data on command line

1. How you can peek on file if it is too large to fit into memory?

a. Use *head/tail -n number filename* to view the first/last n lines in the file

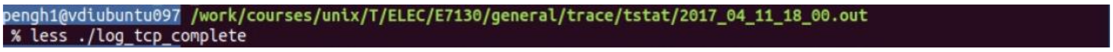


```
pengh1@vdiubuntu097 /work/courses/unix/T/ELEC/E7130/general/trace/tstat/2017_04_11_18_00.out
 % head -n 10 ./log_tcp_complete
#15#c_ip:1 c_port:2 c_pkts_all:3 c_rst_cnt:4 c_ack_cnt:5 c_ack_cnt_p:6 c_bytes_uniq:7 c_pkts_data:8 c_bytes_all:9 c_
pkts_retx:10 c_bytes_retx:11 c_pkts_ooo:12 c_syn_cnt:13 c_fin_cnt:14 s_ip:15 s_port:16 s_pkts_all:17 s_rst_cnt:18 s_
ack_cnt:19 s_ack_cnt_p:20 s_bytes_uniq:21 s_pkts_data:22 s_bytes_all:23 s_pkts_retx:24 s_bytes_retx:25 s_pkts_ooo:26
 s_syn_cnt:27 s_fin_cnt:28 first:29 last:30 durat:31 c_first:32 s_first:33 c_last:34 s_last:35 c_first_ack:36 s_firs
t_ack:37 c_isint:38 s_isint:39 c_iscrypto:40 s_iscrypto:41 con_t:42 p2p_t:43 http_t:44 c_rtt_avg:45 c_rtt_min:46 c_r
tt_max:47 c_rtt_std:48 c_rtt_cnt:49 c_ttl_min:50 c_ttl_max:51 s_rtt_avg:52 s_rtt_min:53 s_rtt_max:54 s_rtt_std:55 s_
rtt_cnt:56 s_ttl_min:57 s_ttl_max:58 p2p_st:59 ed2k_data:60 ed2k_sig:61 ed2k_c2s:62 ed2k_c2c:63 ed2k_chat:64 c_f1323
_opt:65 c_tm_opt:66 c_win_scl:67 c_sack_opt:68 c_sack_cnt:69 c_mss:70 c_mss_max:71 c_mss_min:72 c_win_max:73 c_win_m
in:74 c_win_0:75 c_cwin_max:76 c_cwin_min:77 c_cwin_ini:78 c_pkts_rto:79 c_pkts_fs:80 c_pkts_reor:81 c_pkts_dup:82 c
_pkts_unk:83 c_pkts_fc:84 c_pkts_unrto:85 c_pkts_unfs:86 c_syn_retx:87 s_f1323_opt:88 s_tm_opt:89 s_win_scl:90 s_sac
k_opt:91 s_sack_cnt:92 s_mss:93 s_mss_max:94 s_mss_min:95 s_win_max:96 s_win_min:97 s_win_0:98 s_cwin_max:99 s_cwin_
min:100 s_cwin_ini:101 s_pkts_rto:102 s_pkts_fs:103 s_pkts_reor:104 s_pkts_dup:105 s_pkts_unk:106 s_pkts_fc:107 s_pk
ts_unrto:108 s_pkts_unfs:109 s_syn_retx:110 http_req_cnt:111 http_res_cnt:112 http_res:113 c_pkts_push:114 s_pkts_pu
sh:115 c_tls_SNI:116 s_tls_SCN:117 c_npnalpn:118 s_npnalpn:119 c_tls_sesid:120 c_last_handshakeT:121 s_last_handshak
eT:122 c_appdataT:123 s_appdataT:124 c_appdataB:125 s_appdataB:126 fqdn:127 dns_rslv:128 req_tm:129 res_tm:130
163.35.158.67 64548 17 0 16 9 714 6 714 0 0 0 1 1 17.114.183.199 443 12 1 11 2 5597 7 5597 0 0 0 1 1 1491922800473.7
82959 1491922800609.770020 135.987000 33.231000 41.022000 127.824000 116.144000 28.785000 40.782000 1 0 0 0 0 0 0 9.
673007 7.413000 17.005000 4.122582 5 56 56 12.798090 10.951000 15.482000 1.938675 8 59 59 0 0 0 0 0 0 1 1 5 1 0 1460
 333 6 131744 65535 0 333 31 224 0 0 0 0 0 0 0 0 0 1 1 8 1 0 1460 1448 31 16640 14480 0 5193 51 5193 0 0 0 0 0 0 0 0
 0 0 0 --- 6 5 - - 0 0 0 0.000000 0.000000 0.000000 0.000000 0 0 - - 0.0 0.0
```

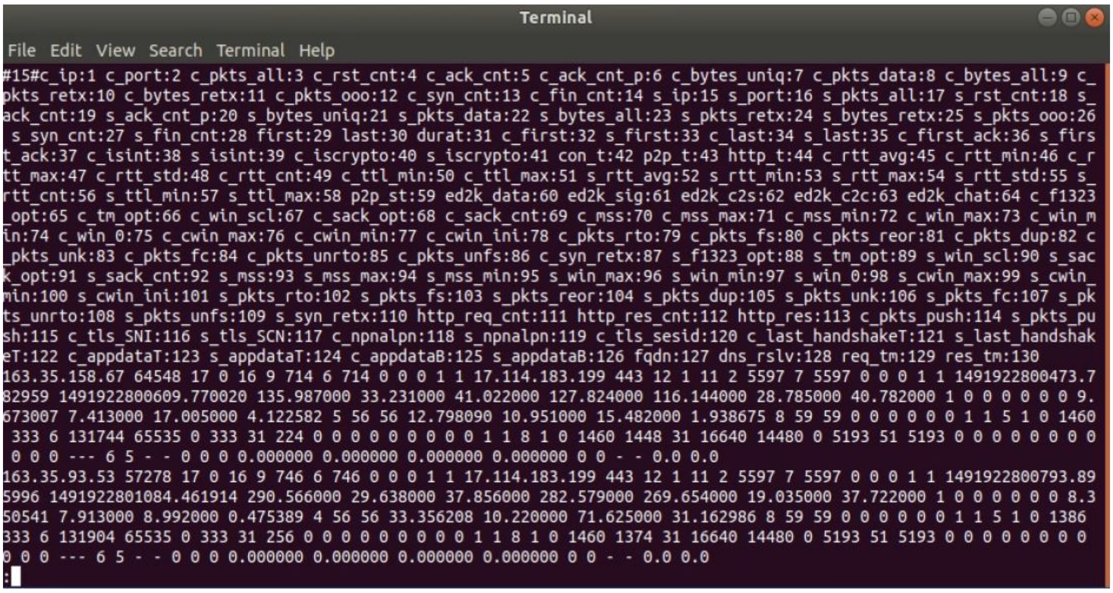


```
pengh1@vdiubuntu097 /work/courses/unix/T/ELEC/E7130/general/trace/tstat/2017_04_11_18_00.out
 % tail -n 10 ./log_tcp_complete
163.35.158.2 57599 7 0 6 2 3543 3 3543 0 0 0 1 1 202.133.62.206 80 5 0 5 2 612 1 612 0 0 0 1 1 1492009199424.843018
1492009199497.271973 72.429000 20.730000 42.773000 30.373000 42.773000 9.045000 21.213000 1 0 0 0 0 0 0 0.588985 0.4
71000 0.711000 0.104668 4 121 121 12.508583 8.413000 20.668000 7.065921 3 249 249 0 0 0 0 0 0 1 0 2 1 0 1460 1460 62
3 65700 8192 0 2920 623 2920 0 0 0 0 0 0 0 0 0 1 0 4 1 0 1460 612 612 24832 8190 0 612 612 612 0 0 0 0 0 0 0 0 0 0 0
 --- 1 1 - - 0 0 0 0.000000 0.000000 0.000000 0.000000 0 0 - - 0.0 0.0
163.35.158.2 57603 7 0 6 2 3662 3 3662 0 0 0 1 1 202.133.195.0 80 8 0 8 4 797 2 797 1 1 0 1 2 1492009199767.837891 1
492009199837.162109 69.324000 21.762000 45.225000 31.494000 45.340000 9.057000 22.072000 1 0 0 0 0 0 0 0.403990 0.30
9000 0.586000 0.127669 4 121 121 8.734127 8.735000 8.735000 0.000000 1 58 249 0 0 0 0 0 0 1 0 2 1 0 1460 1460 742 65
700 8192 0 2920 742 2920 0 0 0 0 0 0 0 0 0 1 0 4 1 0 1460 408 389 24832 8190 0 797 408 797 0 0 0 0 1 0 0 0 0 0 0 ---
 1 2 - - 0 0 0 0.000000 0.000000 0.000000 0.000000 0 0 - - 0.0 0.0
163.35.158.2 57604 7 0 6 2 3538 3 3538 0 0 0 1 1 202.133.195.28 80 6 0 6 3 1269 1 1269 0 0 0 1 1 1492009199797.69897
5 1492009199874.289062 76.590000 19.385000 34.968000 29.233000 34.968000 9.075000 20.044000 1 0 0 0 0 0 0 0.610235 0
.509000 0.757000 0.119000 4 121 121 16.556115 8.566000 32.526000 13.829850 3 58 249 0 0 0 0 0 0 1 0 0 1 0 1460 1460
618 64240 8192 0 2920 618 2920 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1460 1269 1269 24820 8190 0 1269 1269 1269 0 0 0 0 0 0 0
0 0 0 0 --- 1 1 - - 0 0 0 0.000000 0.000000 0.000000 0.000000 0 0 - - 0.0 0.0
163.35.158.2 57606 7 0 6 2 3546 3 3546 0 0 0 1 1 202.133.195.0 80 6 0 6 3 424 1 424 0 0 0 1 1 1492009199861.929932 1
492009199923.128906 61.199000 19.057000 30.647000 28.683000 30.647000 8.872000 19.333000 1 0 0 0 0 0 0 0.476238 0.27
5000 0.954000 0.320183 4 121 121 12.915569 8.551000 21.621000 7.538762 3 58 249 0 0 0 0 0 0 1 0 2 1 0 1460 1460 626
65700 8192 0 2920 626 2920 0 0 0 0 0 0 0 0 0 1 0 4 1 0 1460 424 424 24832 8190 0 424 424 424 0 0 0 0 0 0 0 0 0 0 0 -
```

b. Use *less [OPTIONS] filename* to peek on the file one page at a time



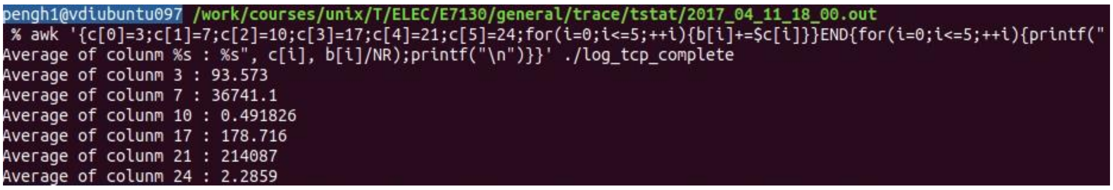
```
pengh1@vdiubuntu097 /work/courses/unix/T/ELEC/E7130/general/trace/tstat/2017_04_11_18_00.out
 % less ./log_tcp_complete
```



```
#15#c_ip:1 c_port:2 c_pkts_all:3 c_rst_cnt:4 c_ack_cnt:5 c_ack_cnt_p:6 c_bytes_uniq:7 c_pkts_data:8 c_bytes_all:9 c_
pkts_retx:10 c_bytes_retx:11 c_pkts_ooo:12 c_syn_cnt:13 c_fin_cnt:14 s_ip:15 s_port:16 s_pkts_all:17 s_rst_cnt:18 s_
ack_cnt:19 s_ack_cnt_p:20 s_bytes_uniq:21 s_pkts_data:22 s_bytes_all:23 s_pkts_retx:24 s_bytes_retx:25 s_pkts_ooo:26
 s_syn_cnt:27 s_fin_cnt:28 first:29 last:30 durat:31 c_first:32 s_first:33 c_last:34 s_last:35 c_first_ack:36 s_firs
t_ack:37 c_isint:38 s_isint:39 c_iscrypto:40 s_iscrypto:41 con_t:42 p2p_t:43 http_t:44 c_rtt_avg:45 c_rtt_min:46 c_r
tt_max:47 c_rtt_std:48 c_rtt_cnt:49 c_ttl_min:50 c_ttl_max:51 s_rtt_avg:52 s_rtt_min:53 s_rtt_max:54 s_rtt_std:55 s_
rtt_cnt:56 s_ttl_min:57 s_ttl_max:58 p2p_st:59 ed2k_data:60 ed2k_sig:61 ed2k_c2s:62 ed2k_c2c:63 ed2k_chat:64 c_f1323
_opt:65 c_tm_opt:66 c_win_scl:67 c_sack_opt:68 c_sack_cnt:69 c_mss:70 c_mss_max:71 c_mss_min:72 c_win_max:73 c_win_m
in:74 c_win_0:75 c_cwin_max:76 c_cwin_min:77 c_cwin_ini:78 c_pkts_rto:79 c_pkts_fs:80 c_pkts_reor:81 c_pkts_dup:82 c
_pkts_unk:83 c_pkts_fc:84 c_pkts_unrto:85 c_pkts_unfs:86 c_syn_retx:87 s_f1323_opt:88 s_tm_opt:89 s_win_scl:90 s_sac
k_opt:91 s_sack_cnt:92 s_mss:93 s_mss_max:94 s_mss_min:95 s_win_max:96 s_win_min:97 s_win_0:98 s_cwin_max:99 s_cwin_
min:100 s_cwin_ini:101 s_pkts_rto:102 s_pkts_fs:103 s_pkts_reor:104 s_pkts_dup:105 s_pkts_unk:106 s_pkts_fc:107 s_pk
ts_unrto:108 s_pkts_unfs:109 s_syn_retx:110 http_req_cnt:111 http_res_cnt:112 http_res:113 c_pkts_push:114 s_pkts_pu
sh:115 c_tls_SNI:116 s_tls_SCN:117 c_npnalpn:118 s_npnalpn:119 c_tls_sesid:120 c_last_handshakeT:121 s_last_handshak
eT:122 c_appdataT:123 s_appdataT:124 c_appdataB:125 s_appdataB:126 fqdn:127 dns_rslv:128 req_tm:129 res_tm:130
163.35.158.67 64548 17 0 16 9 714 6 714 0 0 0 1 1 17.114.183.199 443 12 1 11 2 5597 7 5597 0 0 0 1 1 1491922800473.7
82959 1491922800609.770020 135.987000 33.231000 41.022000 127.824000 116.144000 28.785000 40.782000 1 0 0 0 0 0 0 9.
673007 7.413000 17.005000 4.122582 5 56 56 12.798090 10.951000 15.482000 1.938675 8 59 59 0 0 0 0 0 0 1 1 5 1 0 1460
 333 6 131744 65535 0 333 31 224 0 0 0 0 0 0 0 0 0 1 1 8 1 0 1460 1448 31 16640 14480 0 5193 51 5193 0 0 0 0 0 0 0 0
 0 0 0 --- 6 5 - - 0 0 0 0.000000 0.000000 0.000000 0.000000 0 0 - - 0.0 0.0
163.35.93.53 57278 17 0 16 9 746 6 746 0 0 0 1 1 17.114.183.199 443 12 1 11 2 5597 7 5597 0 0 0 1 1 1491922800793.89
5996 1491922801084.461914 290.566000 29.638000 37.856000 282.579000 269.654000 19.035000 37.722000 1 0 0 0 0 0 0 8.3
50541 7.913000 8.992000 0.475389 4 56 56 33.356208 10.220000 71.625000 31.162986 8 59 59 0 0 0 0 0 0 1 1 5 1 0 1386
333 6 131904 65535 0 333 31 256 0 0 0 0 0 0 0 0 0 1 1 8 1 0 1460 1374 31 16640 14480 0 5193 51 5193 0 0 0 0 0 0 0 0
0 0 0 --- 6 5 - - 0 0 0 0.000000 0.000000 0.000000 0.000000 0 0 - - 0.0 0.0
:
```

2. Averages of columns 3, 7, 10, 17, 21, 24

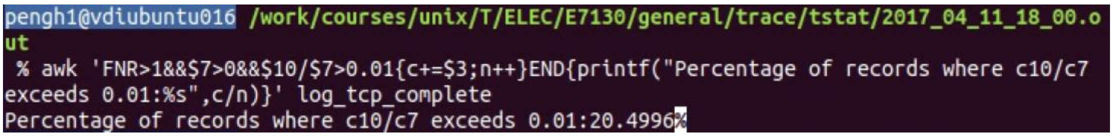
awk '{c[0]=3;c[1]=7;c[2]=10;c[3]=17;c[4]=21;c[5]=24;for(i=0;i<=5;++i){b[i]+=$c[i]}}END{for(i=0;i<=5;++i){printf("Average of colunm %s : %s", c[i], b[i]/NR);printf("\n")}}' ./log_tcp_complete



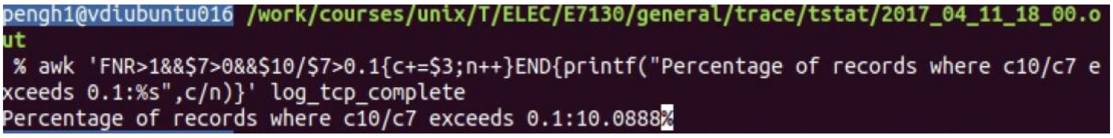
```
pengh1@vdiubuntu097 /work/courses/unix/T/ELEC/E7130/general/trace/tstat/2017_04_11_18_00.out
 % awk '{c[0]=3;c[1]=7;c[2]=10;c[3]=17;c[4]=21;c[5]=24;for(i=0;i<=5;++i){b[i]+=$c[i]}}END{for(i=0;i<=5;++i){printf("
Average of colunm %s : %s", c[i], b[i]/NR);printf("\n")}}' ./log_tcp_complete
Average of colunm 3 : 93.573
Average of colunm 7 : 36741.1
Average of colunm 10 : 0.491826
Average of colunm 17 : 178.716
Average of colunm 21 : 214087
Average of colunm 24 : 2.2859
```

3. Percentage of records where c10/c7 exceeds a) 0.01, b) 0.10, c) 0.20 (c10/c7 means for each line value in column 10 is divided by value in column 7)

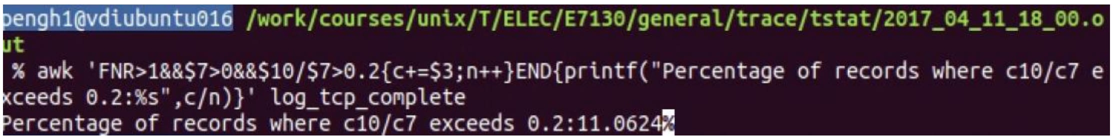
awk 'FNR>1&&$7>0&&$10/$7>0.01{c+=$3;n++}END{printf("Percentage of records where c10/c7 exceeds 0.01:%s",c/n)}' log_tcp_complete



```
pengh1@vdiubuntu016 /work/courses/unix/T/ELEC/E7130/general/trace/tstat/2017_04_11_18_00.o
ut
 % awk 'FNR>1&&$7>0&&$10/$7>0.01{c+=$3;n++}END{printf("Percentage of records where c10/c7
exceeds 0.01:%s",c/n)}' log_tcp_complete
Percentage of records where c10/c7 exceeds 0.01:20.4996%
```

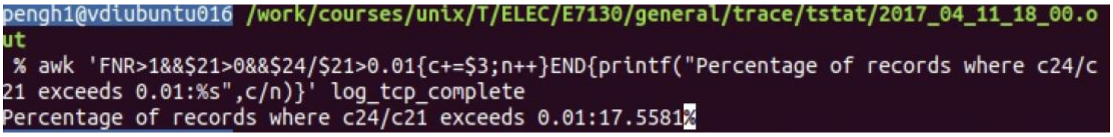


```
pengh1@vdiubuntu016 /work/courses/unix/T/ELEC/E7130/general/trace/tstat/2017_04_11_18_00.o
ut
 % awk 'FNR>1&&$7>0&&$10/$7>0.1{c+=$3;n++}END{printf("Percentage of records where c10/c7 e
xceeds 0.1:%s",c/n)}' log_tcp_complete
Percentage of records where c10/c7 exceeds 0.1:10.0888%
```

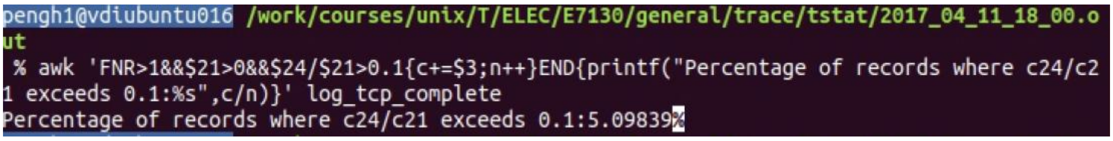


```
pengh1@vdiubuntu016 /work/courses/unix/T/ELEC/E7130/general/trace/tstat/2017_04_11_18_00.o
ut
 % awk 'FNR>1&&$7>0&&$10/$7>0.2{c+=$3;n++}END{printf("Percentage of records where c10/c7 e
xceeds 0.2:%s",c/n)}' log_tcp_complete
Percentage of records where c10/c7 exceeds 0.2:11.0624%
```

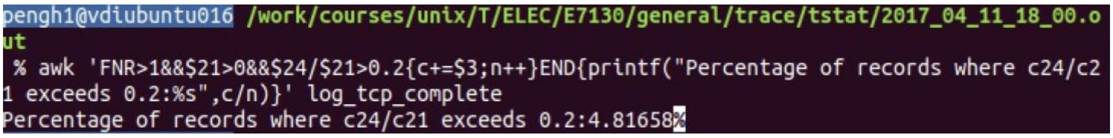
4. Percentage of records where c24/c21 exceeds a) 0.01, b) 0.10, c) 0.20



```
pengh1@vdiubuntu016 /work/courses/unix/T/ELEC/E7130/general/trace/tstat/2017_04_11_18_00.o
ut
 % awk 'FNR>1&&$21>0&&$24/$21>0.01{c+=$3;n++}END{printf("Percentage of records where c24/c
21 exceeds 0.01:%s",c/n)}' log_tcp_complete
Percentage of records where c24/c21 exceeds 0.01:17.5581%
```



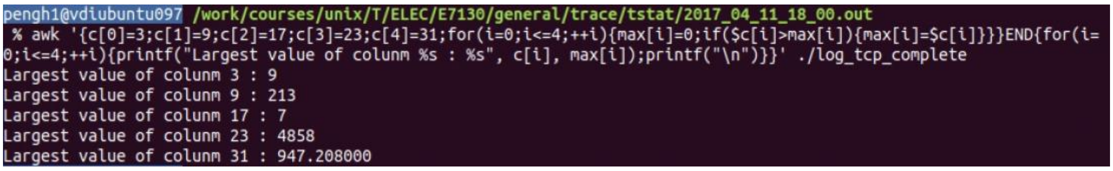
```
pengh1@vdiubuntu016 /work/courses/unix/T/ELEC/E7130/general/trace/tstat/2017_04_11_18_00.o
ut
 % awk 'FNR>1&&$21>0&&$24/$21>0.1{c+=$3;n++}END{printf("Percentage of records where c24/c2
1 exceeds 0.1:%s",c/n)}' log_tcp_complete
Percentage of records where c24/c21 exceeds 0.1:5.09839%
```



```
pengh1@vdiubuntu016 /work/courses/unix/T/ELEC/E7130/general/trace/tstat/2017_04_11_18_00.o
ut
 % awk 'FNR>1&&$21>0&&$24/$21>0.2{c+=$3;n++}END{printf("Percentage of records where c24/c2
1 exceeds 0.2:%s",c/n)}' log_tcp_complete
Percentage of records where c24/c21 exceeds 0.2:4.81658%
```

5. Largest values in columns 3, 9, 17, 23, 31

awk '{c[0]=3;c[1]=9;c[2]=17;c[3]=23;c[4]=31;for(i=0;i<=4;++i){max[i]=0;if($c[i]> max[i]){max[i]=$c[i]}}}END{for(i=0;i<=4;++i){printf("Largest value of colunm %s : %s", c[i], max[i]);printf("\n")}}' ./log_tcp_complete

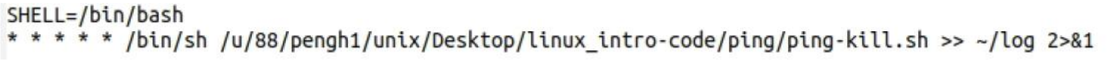


```
pengh1@vdiubuntu097 /work/courses/unix/T/ELEC/E7130/general/trace/tstat/2017_04_11_18_00.out
 % awk '{c[0]=3;c[1]=9;c[2]=17;c[3]=23;c[4]=31;for(i=0;i<=4;++i){max[i]=0;if($c[i]>max[i]){max[i]=$c[i]}}}END{for(i=
0;i<=4;++i){printf("Largest value of colunm %s : %s", c[i], max[i]);printf("\n")}}' ./log_tcp_complete
Largest value of colunm 3 : 9
Largest value of colunm 9 : 213
Largest value of colunm 17 : 7
Largest value of colunm 23 : 4858
Largest value of colunm 31 : 947.208000
```

Task 2: Make a script to produce ping statistics

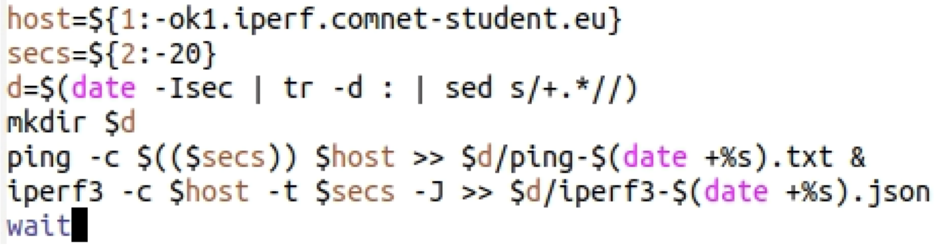
Solution:

Use crontab to control the running frequency of script: *Crontab -e*



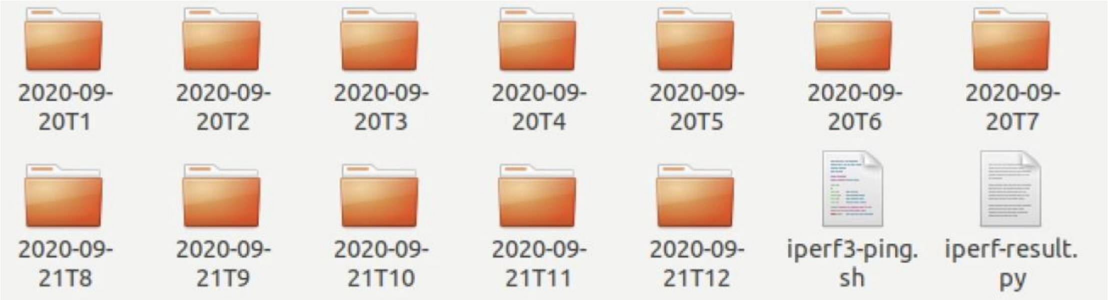
```
SHELL=/bin/bash
* * * * * /bin/sh /u/88/pengh1/unix/Desktop/linux_intro-code/ping/ping-kill.sh >> ~/log 2>&1
```

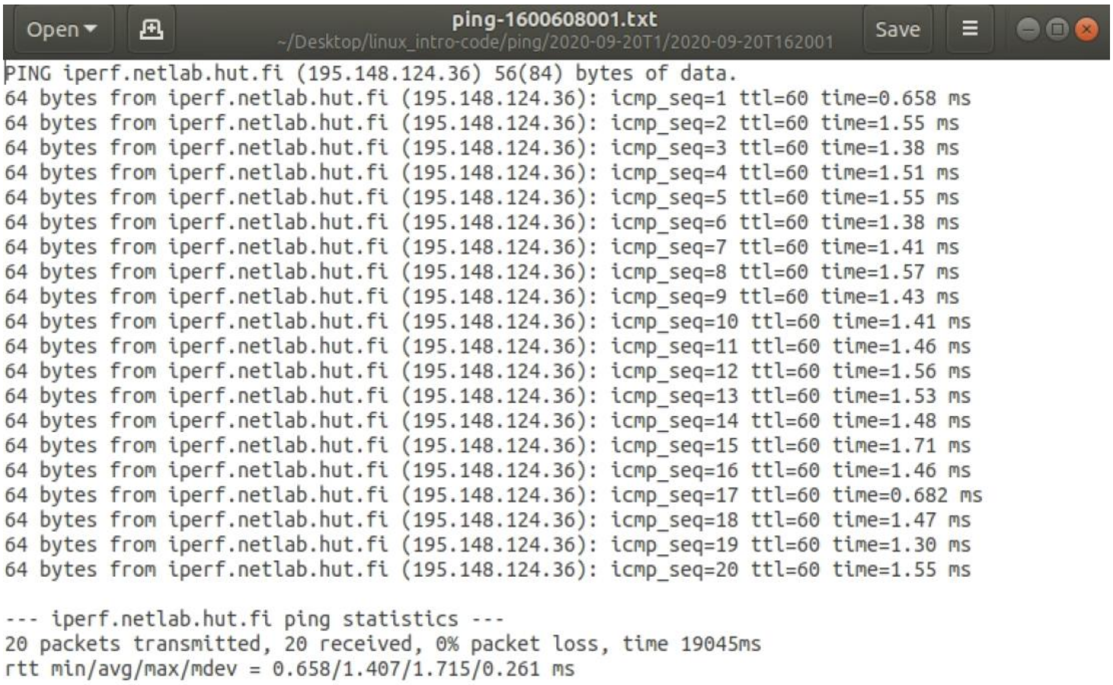
ping-kill.sh:



```
host=${1:-ok1.iperf.comnet-student.eu}
secs=${2:-20}
d=$(date -Isec | tr -d : | sed s/+.*//)
mkdir $d
ping -c $(($secs)) $host >> $d/ping-$(date +%s).txt &
iperf3 -c $host -t $secs -J >> $d/iperf3-$(date +%s).json
wait
```

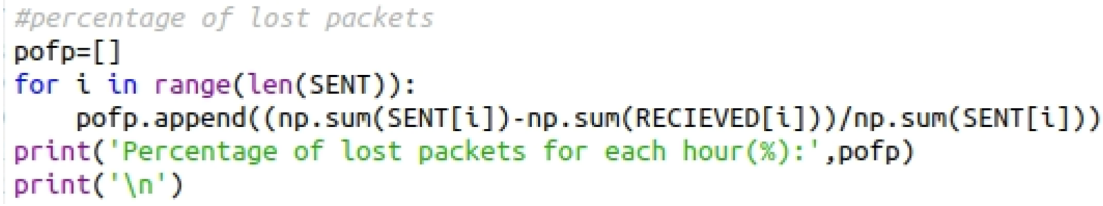
The output data:



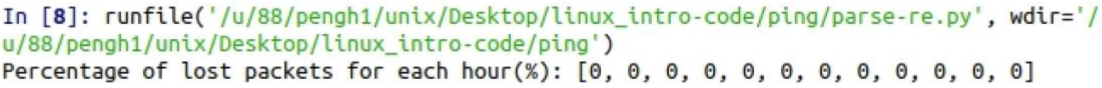

Open
ping-1600608001.txt
~/Desktop/linux_intro-code/ping/2020-09-20T1/2020-09-20T162001
Save

```
PING iperf.netlab.hut.fi (195.148.124.36) 56(84) bytes of data.
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=1 ttl=60 time=0.658 ms
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=2 ttl=60 time=1.55 ms
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=3 ttl=60 time=1.38 ms
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=4 ttl=60 time=1.51 ms
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=5 ttl=60 time=1.55 ms
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=6 ttl=60 time=1.38 ms
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=7 ttl=60 time=1.41 ms
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=8 ttl=60 time=1.57 ms
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=9 ttl=60 time=1.43 ms
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=10 ttl=60 time=1.41 ms
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=11 ttl=60 time=1.46 ms
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=12 ttl=60 time=1.56 ms
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=13 ttl=60 time=1.53 ms
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=14 ttl=60 time=1.48 ms
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=15 ttl=60 time=1.71 ms
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=16 ttl=60 time=1.46 ms
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=17 ttl=60 time=0.682 ms
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=18 ttl=60 time=1.47 ms
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=19 ttl=60 time=1.30 ms
64 bytes from iperf.netlab.hut.fi (195.148.124.36): icmp_seq=20 ttl=60 time=1.55 ms

--- iperf.netlab.hut.fi ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19045ms
rtt min/avg/max/mdev = 0.658/1.407/1.715/0.261 ms
```

1. Percentage of lost packets.

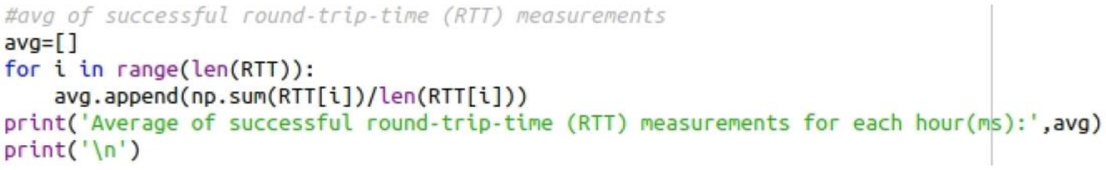


```
#percentage of lost packets
pofp=[]
for i in range(len(SENT)):
     pofp.append((np.sum(SENT[i])-np.sum(RECIEVED[i]))/np.sum(SENT[i]))
print('Percentage of lost packets for each hour(%):',pofp)
print('\n')
```

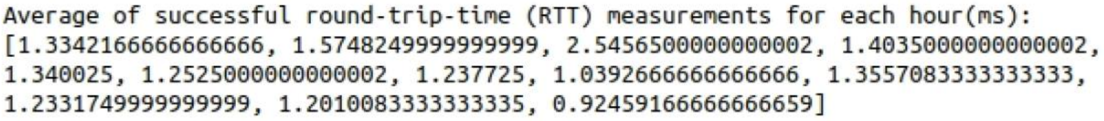


```
In [8]: runfile('/u/88/pengh1/unix/Desktop/linux_intro-code/ping/parse-re.py', wdir='/
u/88/pengh1/unix/Desktop/linux_intro-code/ping')
Percentage of lost packets for each hour(%): [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

2. Average of successful round-trip-time (RTT) measurements.

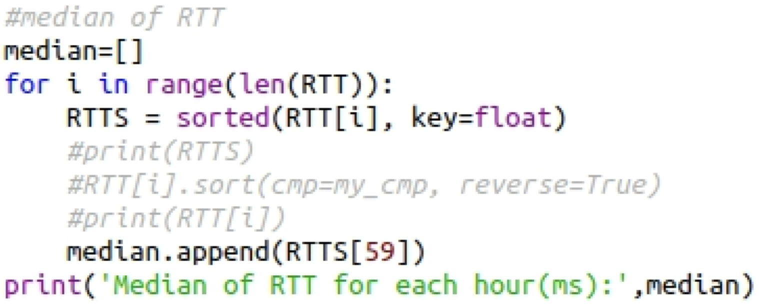


```
#avg of successful round-trip-time (RTT) measurements
avg=[]
for i in range(len(RTT)):
    avg.append(np.sum(RTT[i])/len(RTT[i]))
print('Average of successful round-trip-time (RTT) measurements for each hour(ms):',avg)
print('\n')
```

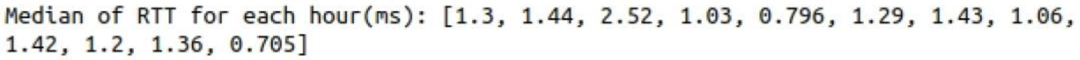


```
Average of successful round-trip-time (RTT) measurements for each hour(ms):
[1.3342166666666666, 1.5748249999999999, 2.5456500000000002, 1.4035000000000002,
1.340025, 1.2525000000000002, 1.237725, 1.0392666666666666, 1.3557083333333333,
1.2331749999999999, 1.2010083333333335, 0.92459166666666659]
```

3. Median of RTT. Lost packet is counted RTT as inf.



```
#median of RTT
median=[]
for i in range(len(RTT)):
    RTTS = sorted(RTT[i], key=float)
    #print(RTTS)
    #RTT[i].sort(cmp=my_cmp, reverse=True)
    #print(RTT[i])
    median.append(RTTS[59])
print('Median of RTT for each hour(ms):',median)
```



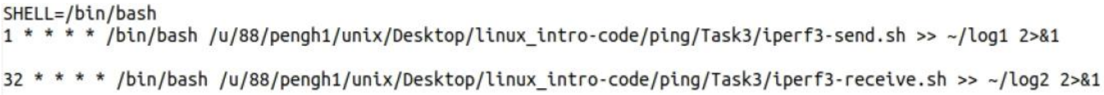
```
Median of RTT for each hour(ms): [1.3, 1.44, 2.52, 1.03, 0.796, 1.29, 1.43, 1.06,
1.42, 1.2, 1.36, 0.705]
```

Task 3: Measure network throughput performance

Solution:

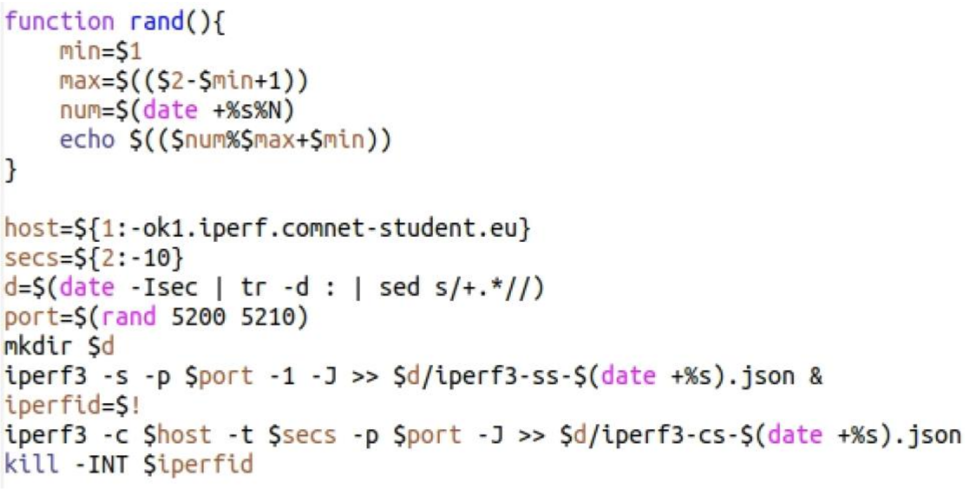
875552%60=32

Use crontab to control the running frequency of script: *Crontab -e*



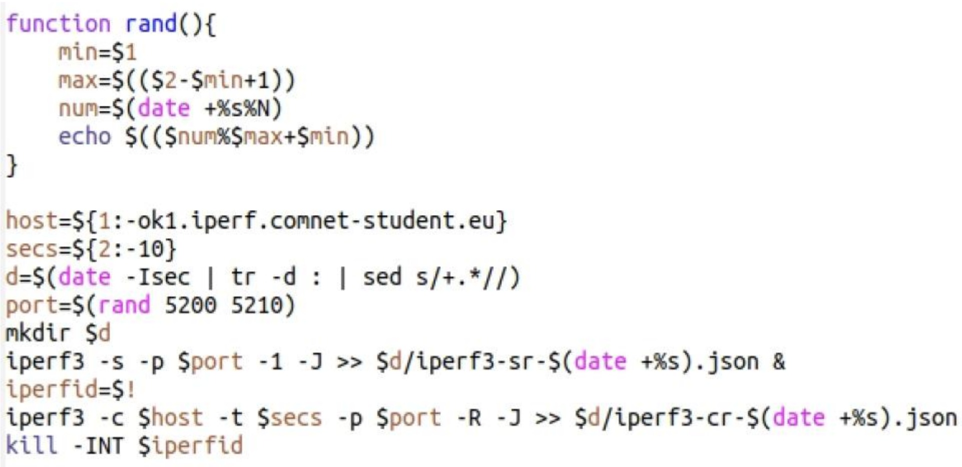
```
SHELL=/bin/bash
1 * * * * /bin/bash /u/88/pengh1/unix/Desktop/linux_intro-code/ping/Task3/iperf3-send.sh >> ~/log1 2>&1

32 * * * * /bin/bash /u/88/pengh1/unix/Desktop/linux_intro-code/ping/Task3/iperf3-receive.sh >> ~/log2 2>&1
```

iperf3-send.sh:



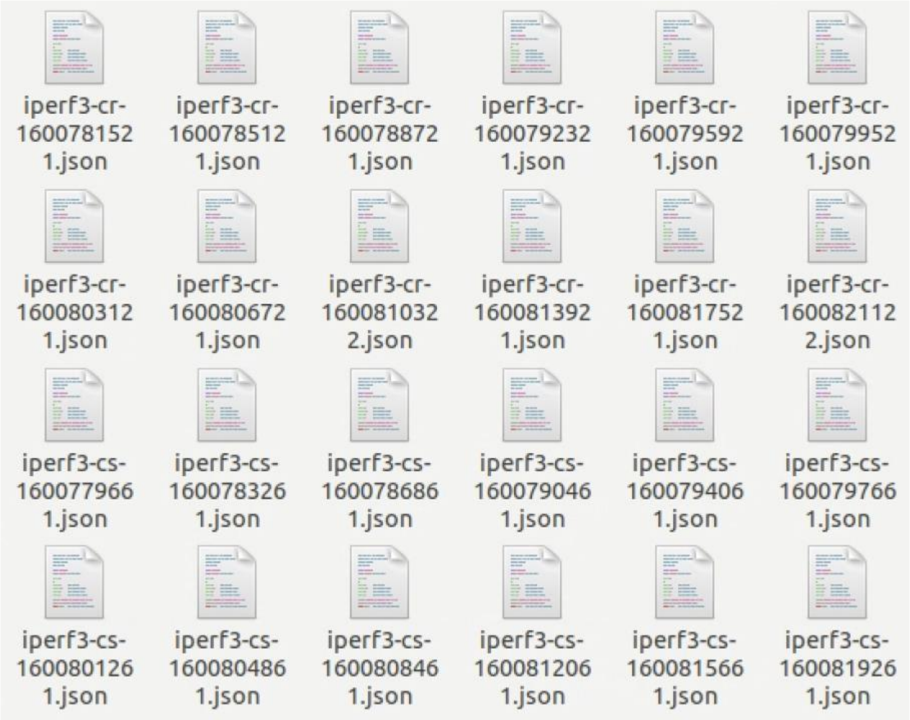
```
function rand(){
    min=$1
    max=$(($2-$min+1))
    num=$(date +%s%N)
    echo $(($num%$max+$min))
}

host=${1:-ok1.iperf.comnet-student.eu}
secs=${2:-10}
d=$(date -Isec | tr -d : | sed s/+.*//)
port=$(rand 5200 5210)
mkdir $d
iperf3 -s -p $port -1 -J >> $d/iperf3-ss-$(date +%s).json &
iperfid=$!
iperf3 -c $host -t $secs -p $port -J >> $d/iperf3-cs-$(date +%s).json
kill -INT $iperfid
```

iperf3-receive.sh:



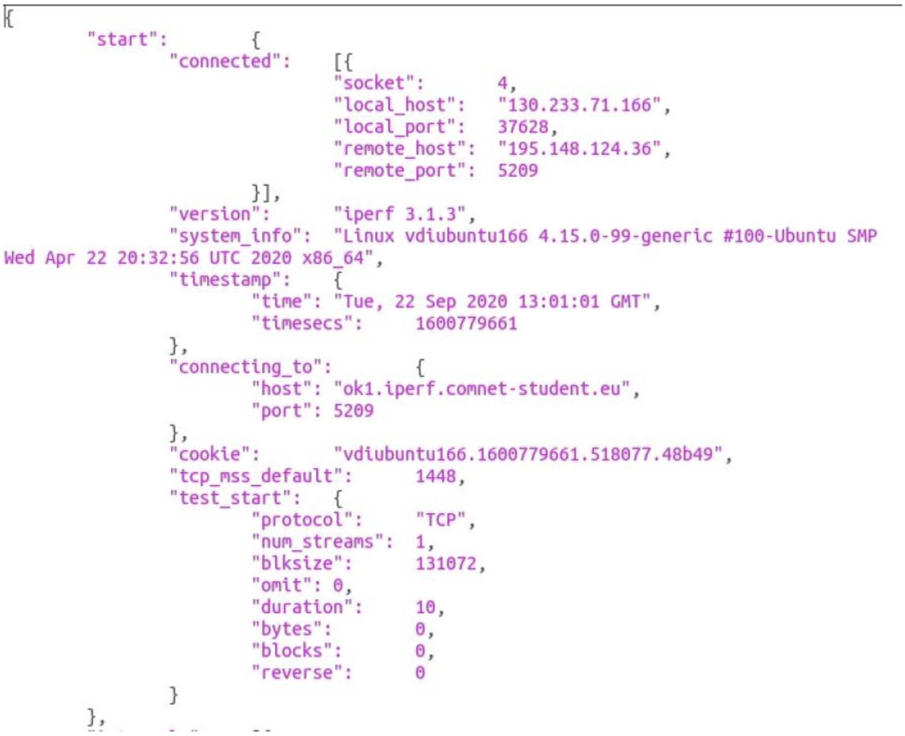
```
function rand(){
    min=$1
    max=$(($2-$min+1))
    num=$(date +%s%N)
    echo $(($num%$max+$min))
}

host=${1:-ok1.iperf.comnet-student.eu}
secs=${2:-10}
d=$(date -Isec | tr -d : | sed s/+.*//)
port=$(rand 5200 5210)
mkdir $d
iperf3 -s -p $port -1 -J >> $d/iperf3-sr-$(date +%s).json &
iperfid=$!
iperf3 -c $host -t $secs -p $port -R -J >> $d/iperf3-cr-$(date +%s).json
kill -INT $iperfid
```

The output data:



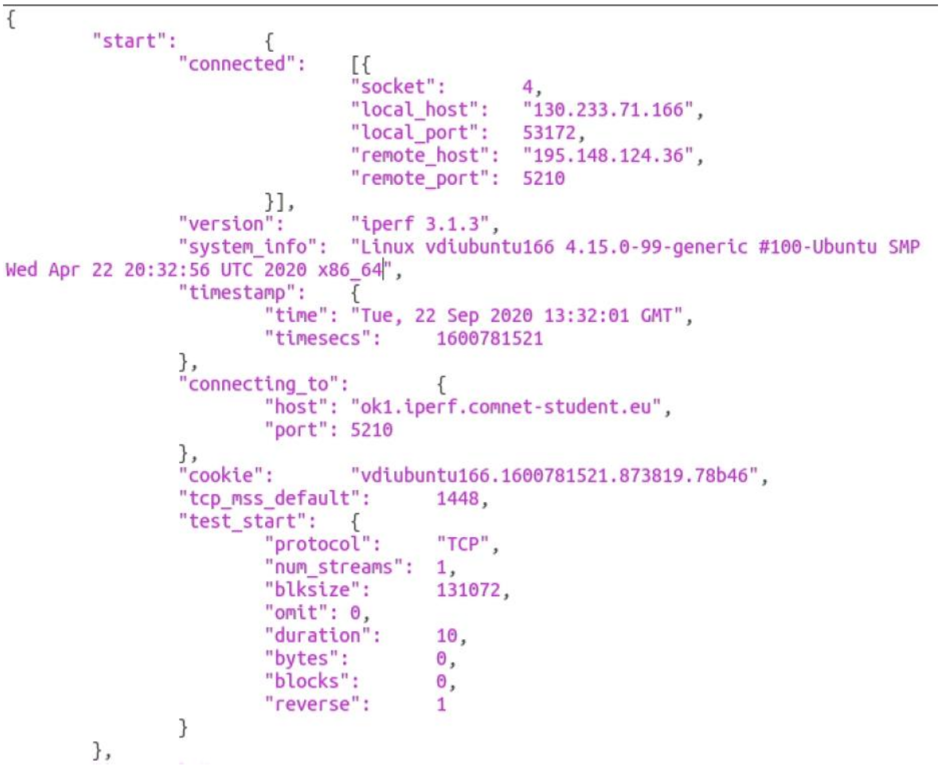
iperf3-cr-1600781521.json
iperf3-cr-1600785121.json
iperf3-cr-1600788721.json
iperf3-cr-1600792321.json
iperf3-cr-1600795921.json
iperf3-cr-1600799521.json
iperf3-cr-1600803121.json
iperf3-cr-1600806721.json
iperf3-cr-1600810322.json
iperf3-cr-1600813921.json
iperf3-cr-1600817521.json
iperf3-cr-1600821122.json
iperf3-cs-1600779661.json
iperf3-cs-1600783261.json
iperf3-cs-1600786861.json
iperf3-cs-1600790461.json
iperf3-cs-1600794061.json
iperf3-cs-1600797661.json
iperf3-cs-1600801261.json
iperf3-cs-1600804861.json
iperf3-cs-1600808461.json
iperf3-cs-1600812061.json
iperf3-cs-1600815661.json
iperf3-cs-1600819261.json

Sending record:



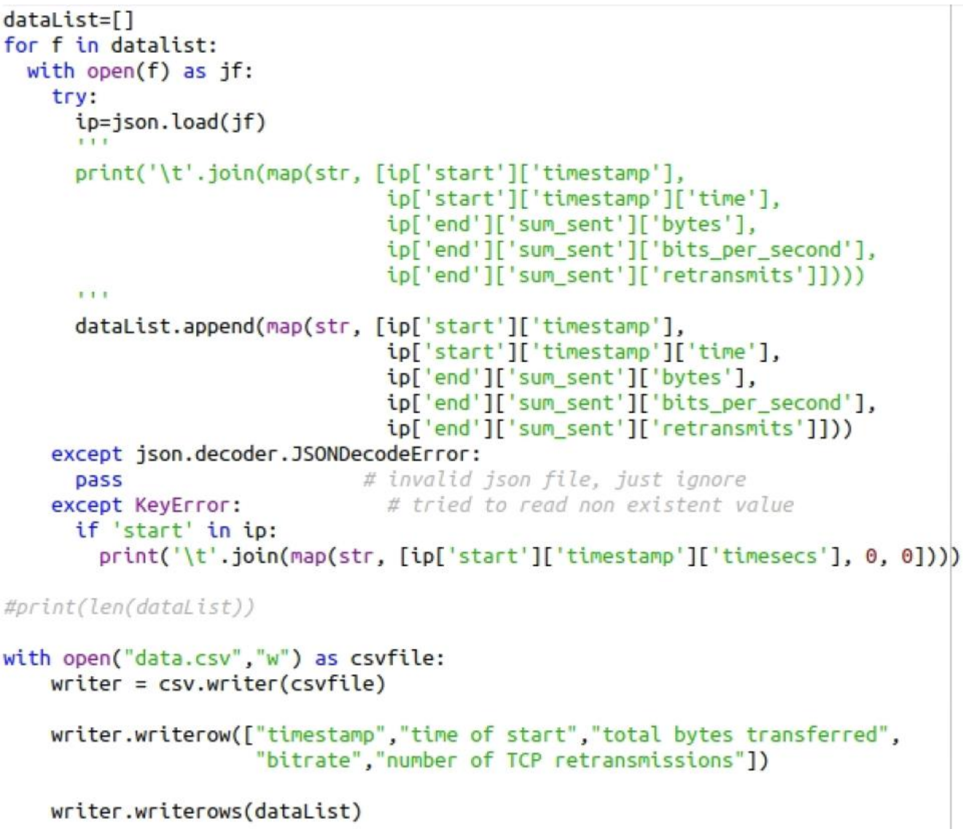
```
{
        "start":        {
                "connected":    [{
                                "socket":       4,
                                "local_host":   "130.233.71.166",
                                "local_port":   37628,
                                "remote_host":  "195.148.124.36",
                                "remote_port":  5209
                        }],
                "version":      "iperf 3.1.3",
                "system_info":  "Linux vdiubuntu166 4.15.0-99-generic #100-Ubuntu SMP
Wed Apr 22 20:32:56 UTC 2020 x86_64",
                "timestamp":    {
                        "time": "Tue, 22 Sep 2020 13:01:01 GMT",
                        "timesecs":     1600779661
                },
                "connecting_to":        {
                        "host": "ok1.iperf.comnet-student.eu",
                        "port": 5209
                },
                "cookie":       "vdiubuntu166.1600779661.518077.48b49",
                "tcp_mss_default":      1448,
                "test_start":   {
                        "protocol":     "TCP",
                        "num_streams":  1,
                        "blksize":      131072,
                        "omit": 0,
                        "duration":     10,
                        "bytes":        0,
                        "blocks":       0,
                        "reverse":      0
                }
        },
```

Receiving record:

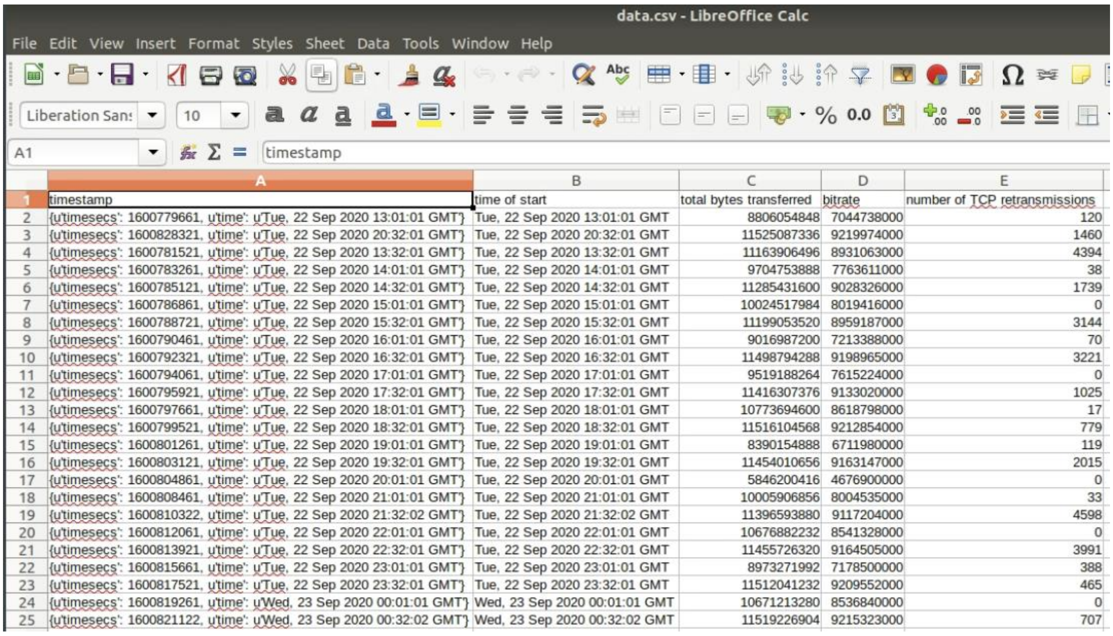


```
{
        "start":        {
                "connected":    [{
                                "socket":       4,
                                "local_host":   "130.233.71.166",
                                "local_port":   53172,
                                "remote_host":  "195.148.124.36",
                                "remote_port":  5210
                        }],
                "version":      "iperf 3.1.3",
                "system_info":  "Linux vdiubuntu166 4.15.0-99-generic #100-Ubuntu SMP
Wed Apr 22 20:32:56 UTC 2020 x86_64",
                "timestamp":    {
                        "time": "Tue, 22 Sep 2020 13:32:01 GMT",
                        "timesecs":     1600781521
                },
                "connecting_to":        {
                        "host": "ok1.iperf.comnet-student.eu",
                        "port": 5210
                },
                "cookie":       "vdiubuntu166.1600781521.873819.78b46",
                "tcp_mss_default":      1448,
                "test_start":   {
                        "protocol":     "TCP",
                        "num_streams":  1,
                        "blksize":      131072,
                        "omit": 0,
                        "duration":     10,
                        "bytes":        0,
                        "blocks":       0,
                        "reverse":      1
                }
        },
```

To produce .csv file:



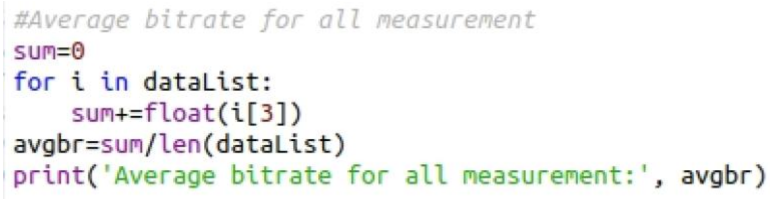
```
dataList=[]
for f in datalist:
  with open(f) as jf:
    try:
      ip=json.load(jf)
      '''
      print('\t'.join(map(str, [ip['start']['timestamp'],
                                 ip['start']['timestamp']['time'],
                                 ip['end']['sum_sent']['bytes'],
                                 ip['end']['sum_sent']['bits_per_second'],
                                 ip['end']['sum_sent']['retransmits']])))
      '''
      dataList.append(map(str, [ip['start']['timestamp'],
                                 ip['start']['timestamp']['time'],
                                 ip['end']['sum_sent']['bytes'],
                                 ip['end']['sum_sent']['bits_per_second'],
                                 ip['end']['sum_sent']['retransmits']]))
    except json.decoder.JSONDecodeError:
      pass                    # invalid json file, just ignore
    except KeyError:            # tried to read non existent value
      if 'start' in ip:
        print('\t'.join(map(str, [ip['start']['timestamp']['timesecs'], 0, 0])))

#print(len(dataList))

with open("data.csv","w") as csvfile:
    writer = csv.writer(csvfile)

    writer.writerow(["timestamp","time of start","total bytes transferred",
                     "bitrate","number of TCP retransmissions"])

    writer.writerows(dataList)
```



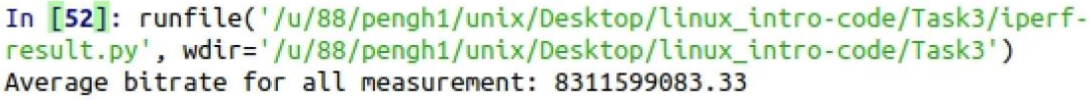
data.csv - LibreOffice Calc
File Edit View Insert Format Styles Sheet Data Tools Window Help
Liberation Sans
10
A1
timestamp

	A	B	C	D	E
1	timestamp	time of start	total bytes transferred	bitrate	number of TCP retransmissions
2	{u'timesecs': 1600779661, u'time': u'Tue, 22 Sep 2020 13:01:01 GMT'}	Tue, 22 Sep 2020 13:01:01 GMT	8806054848	7044738000	120
3	{u'timesecs': 1600828321, u'time': u'Tue, 22 Sep 2020 20:32:01 GMT'}	Tue, 22 Sep 2020 20:32:01 GMT	11525087336	9219974000	1460
4	{u'timesecs': 1600781521, u'time': u'Tue, 22 Sep 2020 13:32:01 GMT'}	Tue, 22 Sep 2020 13:32:01 GMT	11163906496	8931063000	4394
5	{u'timesecs': 1600783261, u'time': u'Tue, 22 Sep 2020 14:01:01 GMT'}	Tue, 22 Sep 2020 14:01:01 GMT	9704753888	7763611000	38
6	{u'timesecs': 1600785121, u'time': u'Tue, 22 Sep 2020 14:32:01 GMT'}	Tue, 22 Sep 2020 14:32:01 GMT	11285431600	9028326000	1739
7	{u'timesecs': 1600786861, u'time': u'Tue, 22 Sep 2020 15:01:01 GMT'}	Tue, 22 Sep 2020 15:01:01 GMT	10024517984	8019416000	0
8	{u'timesecs': 1600788721, u'time': u'Tue, 22 Sep 2020 15:32:01 GMT'}	Tue, 22 Sep 2020 15:32:01 GMT	11199053520	8959187000	3144
9	{u'timesecs': 1600790461, u'time': u'Tue, 22 Sep 2020 16:01:01 GMT'}	Tue, 22 Sep 2020 16:01:01 GMT	9016987200	7213388000	70
10	{u'timesecs': 1600792321, u'time': u'Tue, 22 Sep 2020 16:32:01 GMT'}	Tue, 22 Sep 2020 16:32:01 GMT	11498794288	9198965000	3221
11	{u'timesecs': 1600794061, u'time': u'Tue, 22 Sep 2020 17:01:01 GMT'}	Tue, 22 Sep 2020 17:01:01 GMT	9519188264	7615224000	0
12	{u'timesecs': 1600795921, u'time': u'Tue, 22 Sep 2020 17:32:01 GMT'}	Tue, 22 Sep 2020 17:32:01 GMT	11416307376	9133020000	1025
13	{u'timesecs': 1600797661, u'time': u'Tue, 22 Sep 2020 18:01:01 GMT'}	Tue, 22 Sep 2020 18:01:01 GMT	10773694600	8618798000	17
14	{u'timesecs': 1600799521, u'time': u'Tue, 22 Sep 2020 18:32:01 GMT'}	Tue, 22 Sep 2020 18:32:01 GMT	11516104568	9212854000	779
15	{u'timesecs': 1600801261, u'time': u'Tue, 22 Sep 2020 19:01:01 GMT'}	Tue, 22 Sep 2020 19:01:01 GMT	8390154888	6711980000	119
16	{u'timesecs': 1600803121, u'time': u'Tue, 22 Sep 2020 19:32:01 GMT'}	Tue, 22 Sep 2020 19:32:01 GMT	11454010656	9163147000	2015
17	{u'timesecs': 1600804861, u'time': u'Tue, 22 Sep 2020 20:01:01 GMT'}	Tue, 22 Sep 2020 20:01:01 GMT	5846200416	4676900000	0
18	{u'timesecs': 1600808461, u'time': u'Tue, 22 Sep 2020 21:01:01 GMT'}	Tue, 22 Sep 2020 21:01:01 GMT	10005906856	8004535000	33
19	{u'timesecs': 1600810322, u'time': u'Tue, 22 Sep 2020 21:32:02 GMT'}	Tue, 22 Sep 2020 21:32:02 GMT	11396593880	9117204000	4598
20	{u'timesecs': 1600812061, u'time': u'Tue, 22 Sep 2020 22:01:01 GMT'}	Tue, 22 Sep 2020 22:01:01 GMT	10676882232	8541328000	0
21	{u'timesecs': 1600813921, u'time': u'Tue, 22 Sep 2020 22:32:01 GMT'}	Tue, 22 Sep 2020 22:32:01 GMT	11455726320	9164505000	3991
22	{u'timesecs': 1600815661, u'time': u'Tue, 22 Sep 2020 23:01:01 GMT'}	Tue, 22 Sep 2020 23:01:01 GMT	8973271992	7178500000	388
23	{u'timesecs': 1600817521, u'time': u'Tue, 22 Sep 2020 23:32:01 GMT'}	Tue, 22 Sep 2020 23:32:01 GMT	11512041232	9209552000	465
24	{u'timesecs': 1600819261, u'time': u'Wed, 23 Sep 2020 00:01:01 GMT'}	Wed, 23 Sep 2020 00:01:01 GMT	10671213280	8536840000	0
25	{u'timesecs': 1600821122, u'time': u'Wed, 23 Sep 2020 00:32:02 GMT'}	Wed, 23 Sep 2020 00:32:02 GMT	11519226904	9215323000	707

1. Average bitrate for all measurement.

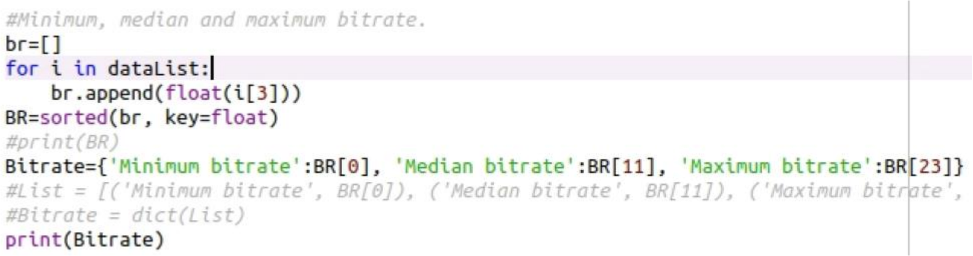


```
#Average bitrate for all measurement
sum=0
for i in dataList:
    sum+=float(i[3])
avgbr=sum/len(dataList)
print('Average bitrate for all measurement:', avgbr)
```

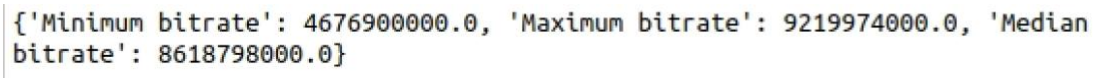


```
In [52]: runfile('/u/88/penghl/unix/Desktop/linux_intro-code/Task3/iperf-result.py', wdir='/u/88/penghl/unix/Desktop/linux_intro-code/Task3')
Average bitrate for all measurement: 8311599083.33
```

2. Minimum, median and maximum bitrate.

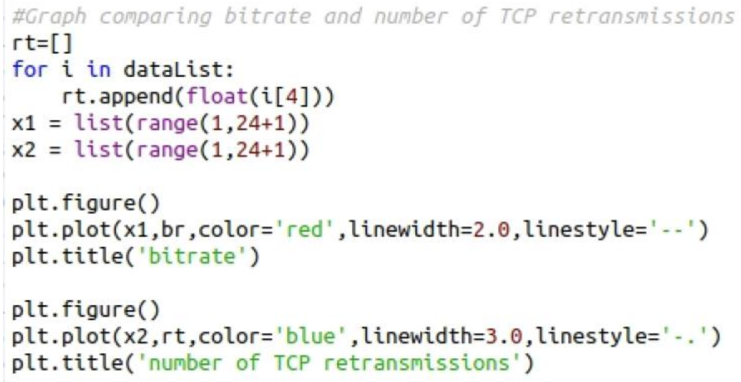


```
#Minimum, median and maximum bitrate.
br=[]
for i in dataList:
    br.append(float(i[3]))
BR=sorted(br, key=float)
#print(BR)
Bitrate={'Minimum bitrate':BR[0], 'Median bitrate':BR[11], 'Maximum bitrate':BR[23]}
#List = [('Minimum bitrate', BR[0]), ('Median bitrate', BR[11]), ('Maximum bitrate',
#Bitrate = dict(List)
print(Bitrate)
```

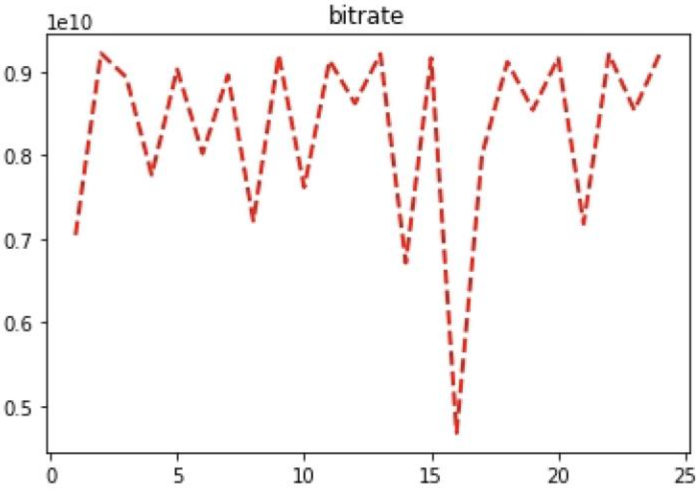


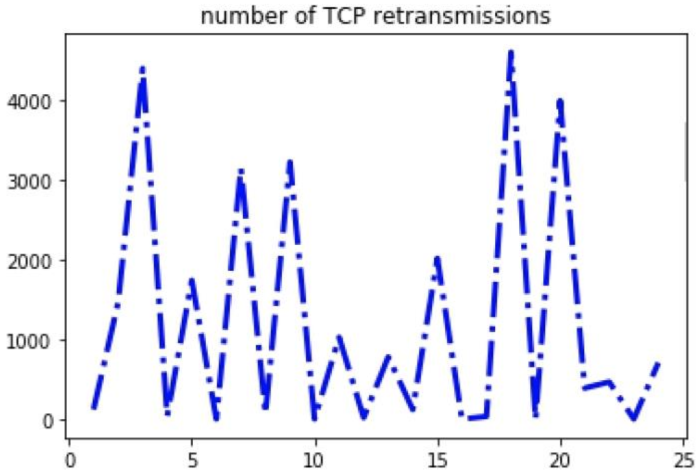
```
{'Minimum bitrate': 4676900000.0, 'Maximum bitrate': 9219974000.0, 'Median
bitrate': 8618798000.0}
```

3. Graph comparing bitrate and number of TCP retransmissions.



```
#Graph comparing bitrate and number of TCP retransmissions
rt=[]
for i in dataList:
    rt.append(float(i[4]))
x1 = list(range(1,24+1))
x2 = list(range(1,24+1))

plt.figure()
plt.plot(x1,br,color='red',linewidth=2.0,linestyle='--')
plt.title('bitrate')

plt.figure()
plt.plot(x2,rt,color='blue',linewidth=3.0,linestyle='-.')
plt.title('number of TCP retransmissions')
```





4. If you used your own computer to run iperf, describe network connectivity. Can you make any conclusions of stability based on data?

The network connectivity remained stable in most of the time but sometimes had fluctuation since the bitrate was comparatively lower and the number of TCP retransmissions were higher, as is seen in the graph.