

Check2 实验报告

王昕浩 221502005

实验测试结果如下。

```
wexther@wxh: ~/minnow$ ./scripts/tun.sh start 144
wexther@wxh: ~/minnow$ cmake --build build --target check_webget
Test project /home/wexther/minnow/build
  Start 1: compile with bug-checkers
1/2 Test #1: compile with bug-checkers ..... Passed    0.24 sec
  Start 2: t_webget
2/2 Test #2: t_webget ..... Passed    3.34 sec

100% tests passed, 0 tests failed out of 2

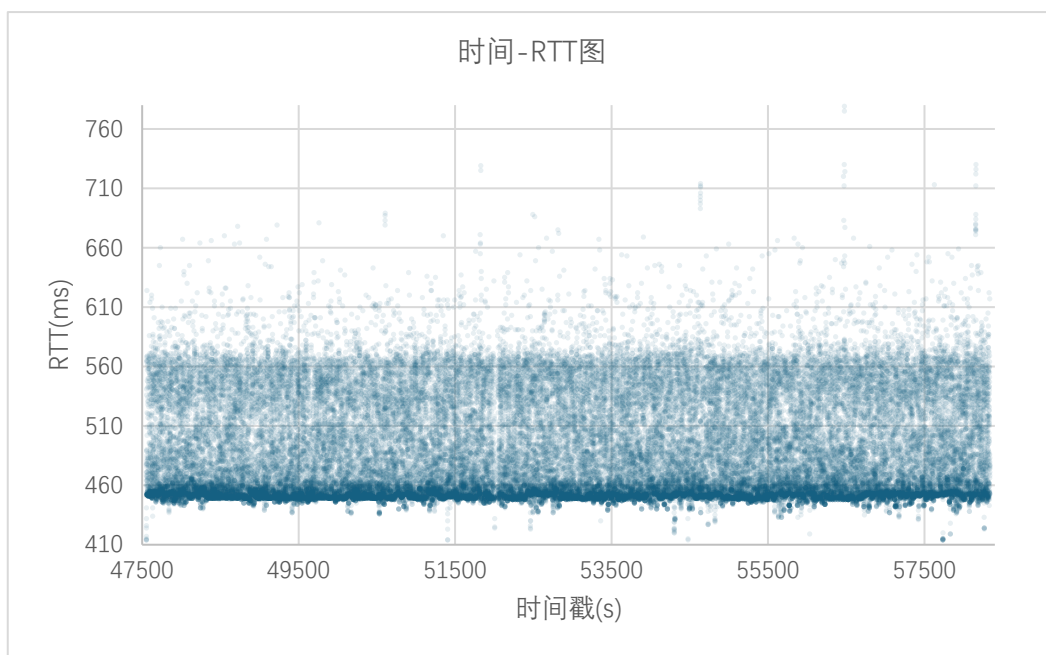
Total Test time (real) = 3.58 sec
Built target check_webget
wexther@wxh: ~/minnow$ ./build/apps/webget cs144.keithw.org /nph-hashier/xyzzzy
Function called: get_URL(cs144.keithw.org, /nph-hashier/xyzzzy)
DEBUG: minnow connecting to 104.196.238.229:80...
DEBUG: minnow successfully connected to 104.196.238.229:80.
DEBUG: minnow inbound stream from HTTP/1.1 200 OK
Content-type: text/plain

7SmXqWkrLKzVBCEalbSPqBcvs11Pw263K7x4Wv3JckI
DEBUG: minnow waiting for clean shutdown... 104.196.238.229:80 finished cleanly.
DEBUG: minnow outbound stream to 104.196.238.229:80 finished (0 seqnos still in flight).
DEBUG: minnow outbound stream to 104.196.238.229:80 has been fully acknowledged.
DEBUG: minnow TCP connection finished cleanly.
done.
wexther@wxh: ~/minnow$
```

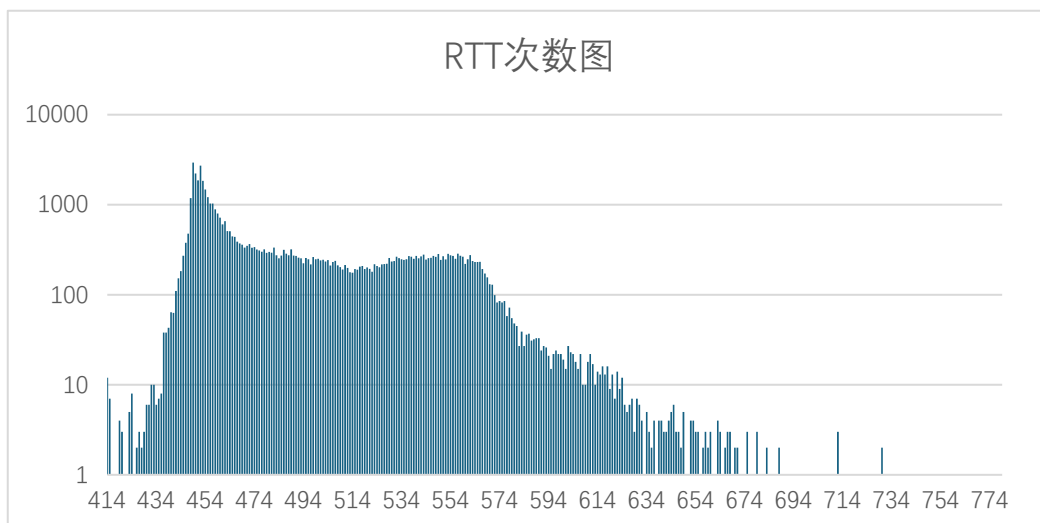
Analyzing data

1. 总共有 53515 条 ping 被发送，53257 条收到回复，送达率 $53257/53515 \approx 99.518\%$ 。
2. 最长的连续成功 ping 是从 icmp_seq=24398 到 icmp_seq=47227 的共 22830 条 ping。
3. 最长的连续丢包是从 icmp_seq=22076 到 icmp_seq=22255 的共 180 条 ping。
4. 以频率估计概率，
当第 N 条收到回复时，第 N+1 条收到回复的概率为 $(53256-8)/53256 \approx 99.985\%$ 。
当第 N 条未收到回复时，第 N+1 条收到回复的概率为 $8/258 \approx 3.101\%$ 。
当第 N 条收到回复时，下一条收到回复的概率较总体概率略微增加，当第 N 条未收到回复时，下一条收到回复的概率较总体概率下降巨大。从此，我有较大把握认为丢包事件相互之间不独立，是突发性的 (bursty)。
5. 最小的 RTT 为 414ms。
6. 最大的 RTT 为 779ms。
7. 如下图，其中为图表可读性，将每个数据点的透明度设置为 90%，时间戳仅截取

后五位。

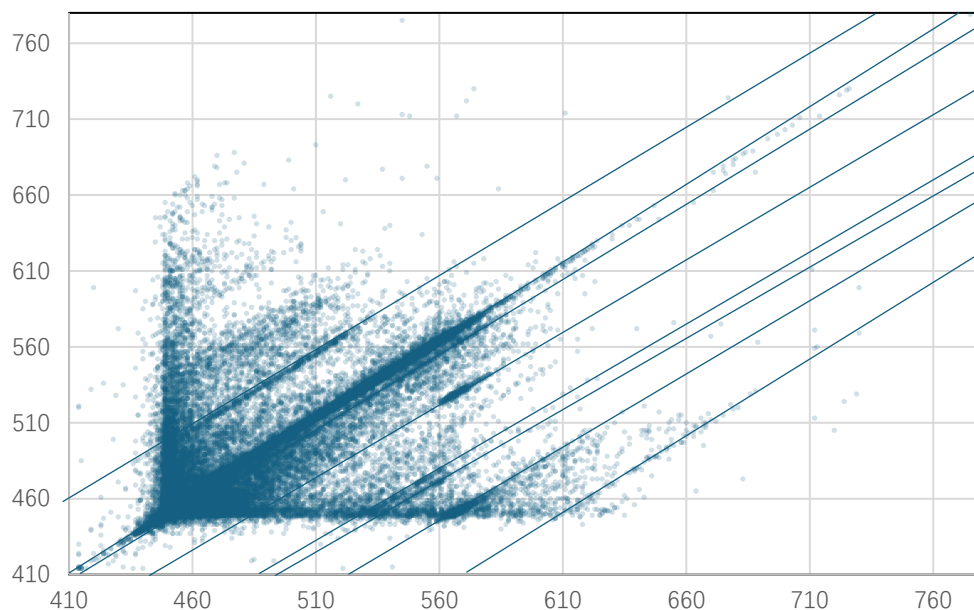


8. 如下图，其中为图表可读性，纵坐标设置为对数分布。该图近似泊松分布。



9. 如下图，其中为图表可读性，将每个数据点的透明度设置为 80%。整个图表像一只燕子。从图可以看出本次 RTT 值与下次 RTT 值近乎线性相关。我认为在特定时刻，本次 RTT 与下次 RTT 成线性相关；在不同时刻，相关系数相同而回归直线截距不同。我在下图中描绘出了一族平行直线，大部分数据都沿着该族直线分布。

第N次RTT-第N+1次RTT图



10. 从此可以得到结论，丢包率与上个包是否丢失有关，本次 RTT 值也与上次 RTT 值相关。我认为这些都是因为当前丢包率与 RTT 值与当前网络拥塞情况相关，而网络拥塞情况可以用上次丢包概率与 RTT 值反应。因此网络行为和我预想的一样。令我意外的是第 N 次 RTT-第 N+1 次 RTT 图中数据不仅沿一条直线分布，而是沿一族多条平行直线分布。