# Check2实验报告

王昕浩 221502005

## Analyzing data

1. 总共有53515条ping被发送，53257条收到回复，送达率53257/53515≈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≈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≈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。
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。