

拥塞控制测试的说明

由于传输过程中发送数据包的数量, 丢包情况的出现存在随机因素, 拥塞控制将在最终的线下验收环境进行现场人工验收

- **测试思路**

一方面, 使用抓包程序记录客户端发向服务端的数据包, 绘制发送的数据包**SeqNum随时间变化**的曲线, 从而展示TCP的慢启动、拥塞避免、快速恢复、超时等特性

另一方面, 使用server.event.log中记录的拥塞窗口变化情况, 直接绘制**拥塞窗口随时间变化**的曲线, 从而直接展示TCP的慢启动、拥塞避免、快速恢复、超时等特性

- **测试方法**

用于拥塞控制测试的文件包括三个可执行文件, test_congestion、gen_graph_seq、gen_graph_win

test_congestion囊括了完整的测试过程, gen_graph_seq用于绘制SeqNum随时间变化的曲线, gen_graph_win用于绘制拥塞窗口随时间变化的曲线

在终端输入如下指令, 运行可执行文件test_congestion即可进行拥塞控制的测试

```
/vagrant/tju_tcp/test/test_congestion
```

默认的带宽、延迟、延迟波动、丢包率分别是100Mbps、300ms、50、10%, 可以通过参数传入自定义设置, 例如

```
/vagrant/tju_tcp/test/test_congestion 100 300 20 50
```

表示设置测试的网络环境为带宽100Mbps、延迟300ms、延迟波动20、丢包率50%

其中, test_congestion内部的测试过程如下:

```
=> 设置网络环境（包括丢包率10%、延迟300ms、带宽100Mbps等）
=> 使用tcpdump在服务端开启抓包进程, 记录从客户端发向服务端的数据包
=> 运行双端的测试程序
    (/vagrant/tju_tcp/test/rdt_server和/vagrant/tju_tcp/test/rdt_client)
=> 等待双方通讯60s后, 停止服务端抓包
=> 调用gen_graph_seq, 根据抓包结果
=> 调用gen_graph_win, 根据日志文件绘制图像
```

- **测试结果**

测试结束之后, 在/vagrant/tju_tcp/test/中查看绘制的图像:

```
SeqNum_VS_Time.png: SeqNum随时间变化的曲线
CongestionWindowSize_VS_Time.png: 拥塞窗口随时间变化的曲线
ReceiveWindowSize_VS_Time.png: 接收方接收窗口随时间变化的曲线
SendWindowSize_VS_Time.png: 发送窗口随时间变化的曲线
AllWindowSize_VS_Time.png: 拥塞窗口、接收方接收窗口、发送窗口随时间变化的曲线（该曲线可用于观察拥塞窗口、接收方接收窗口、发送窗口三者随时间之间的变化趋势以及三者之间的关系，辅助流量控制的观测）
```

- 注:

- 每次运行都会覆盖之前的图像，注意备份
- 由于测试存在随机性，可以进行多次测试得到满意的结果
- gen_graph_seq 默认使用 /home/vagrant/server.pcap 文件进行绘图，可以通过传入pcap文件路径，用 pcap_file_path 对应的pcap的文件绘制序列号随时间变化的曲线；

gen_graph_win 默认使用 /vagrant/tju_tcp/test/server.event.log 文件进行绘图，可以通过传入log文件路径，用 log_file_path 对应的日志文件绘制各窗口随时间变化的曲线

```
/vagrant/tju_tcp/test/gen_graph_seq pcap_file_path  
/vagrant/tju_tcp/test/gen_graph_win log_file_path
```