

# CSC 361 Lab Session 9

Boyang Yu

Junnan Lu

Dept. of Computer Science  
University of Victoria

Fall 2014

# Agenda

- 1 TCP Congestion Control
- 2 P2: Network Impairment Emulation through TC
- 3 Network Address Translation (NAT)

# Congestion control mechanism

Review questions:

- What is the goal of TCP congestion control?
- What is the TCP congestion control mechanism?
- What is *Slow Start*?
- What is *the Congestion Avoidance*?

# Experiment configuration

Please download trace files: `tcp-retran-t.cap`

Experiment configuration:

- Sender: 192.168.1.1
- Receiver: 192.168.1.100 port 5001
- TCP traffic generator: **ttcp**
- Network emulator: Traffic Control (tc):

# Congestion window

Please open trace file: “tcp-retran-t.cap”

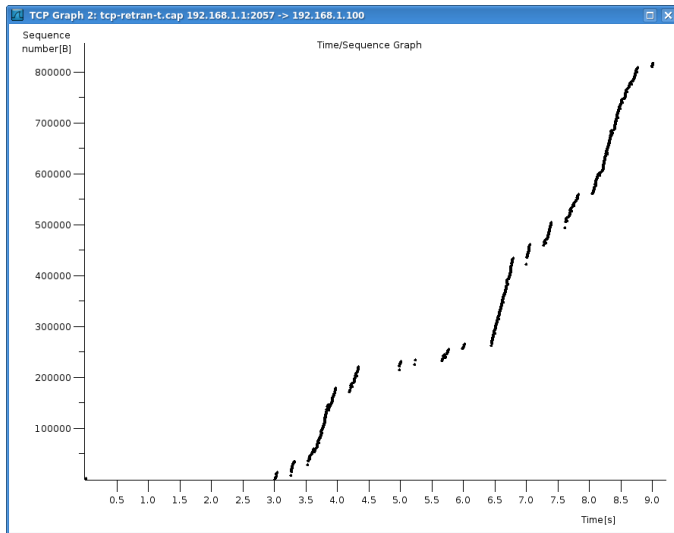
- MSS: 1460 bytes (packet #1 and #2, TCP Options)
- Slow start: initial cwnd == 2 MSS (tcp.seq == 1 and tcp.seq == 1461)
- Congestion signals:
  - ▶ Retransmission timer timeout: cwnd → 1 MSS, “Slow Start”.  
E.g. packet #6, cwnd == 1 MSS, packet #7 to #9, cwnd == 2 MSS
  - ▶ 3 duplicated ACKs: cwnd → 1/2 cwnd, “Fast recovery”

# Fast retransmission/recovery

- Apply the “tcp.analysis.retransmission” display filter
- Find fast retransmission packet #66 (tcp.seq==40881)
- Duplicate ACKs (tcp.ack==40881):  
packet #59, #60, #63, #64

# The impact of retransmission on sending rate

## The TCP Time/Sequence Graph (Stevens)



# TCP Competing with UDP

Please download and open: tcp-cong-s.cap

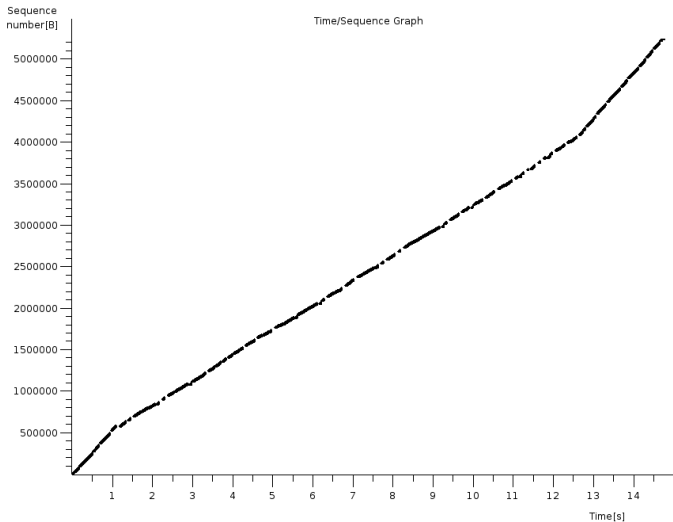
Experiment configuration

- TCP Sender: 192.168.1.1
- TCP Receiver: 192.168.1.100 port 5001
- UDP Sender: 192.168.1.1
- UDP Receiver: 192.168.1.100 port 5002
- The UDP stream started 1 seconds after the TCP stream started



# TCP Competing with UDP (Cont.)

## The TCP Time/Sequence Graph (Stevens)



# TCP Competing with UDP (Cont.)

## Observations:

- The TCP stream started from 0 second, lasted 14.779 seconds
- The UDP stream started from 1.10768 second, lasted 11.5109 seconds
- Use Conversations dialogue to analyze the streams
- Use TCP Stream Graph to analyze the TCP stream

# Network impairment emulation through TC

- Change the traffic control settings of your **router**

## Add

```
tc qdisc add dev br0 root netem delay 200ms 50ms 25% loss 10% duplicate 5%
```

- Show the current settings

## Show

```
tc qdisc show dev br0
```

- Return to the normal settings

## Remove

```
tc qdisc del dev br0 root
```

# Address translation for outgoing packets

Please download trace files: nat-int.cap, nat-ext.cap

Experiment settings:

- Internal network address: 192.168.1.0/24
- External network address: 10.10.1.1/32
- Traffic generator: ping  
ping -l 192.168.1.100 10.10.1.100

Outgoing packets:

- Out from pc: 192.168.1.100 → 10.10.1.100 (ICMP Echo Request)
- After NAT: 10.10.1.1 → 10.10.1.100 (ICMP Echo Request)

# Address translation for incoming packets

Incoming packets:

- Out from pc: 10.10.1.100 → 10.10.1.1 (ICMP Echo Reply)
- After NAT: 10.10.1.100 → 192.168.1.100 (ICMP Echo Reply)