Assignment #4

Network Emulation & Dynamic Rate Control

2018 Spring

Yusung Kim

yskim525@skku.edu

Basic Information

- Two programs (Sender & Receiver) based on UDP socket programming
- Run multiple senders with one receiver
- The receiver also acts as a Network Emulator

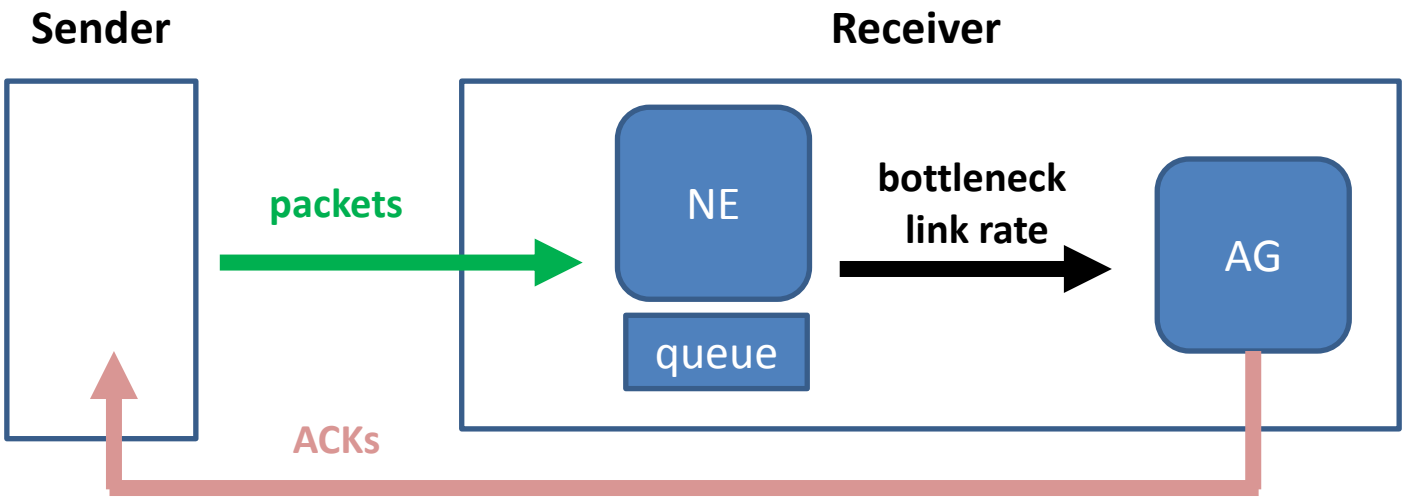
Sender

- When the sender starts, enter an IP address of a receiver, and "initial sending rate"
  - the sending rate unit is the number of packets per second (PPS)
  - a port number of a receiver is 10080
- If the sending rate is 10 PPS, the sender transmits a packet every 1/10 seconds (0.1 seconds).
- The packet size is 1000 bytes which consists of any data.
- Every two seconds, print the following
  - sending rate (the number of sent packets / 2 seconds)
  - goodput (the number of received ACKs / 2 seconds)
  - goodput ratio : goodput / sending rate
- Do not worry about the reliable data delivery in this assignment.

Receiver (1/2)

- A receiver consists of two modules;
  - one module is a network emulator, NE
  - another module is a ACK generator, AG
- When the receiver starts, enter a bottleneck link rate, and queue size for network emulator;
  - For the bottleneck link rate;
    e.g. 10 means forwarding 10 packets per second from NE to AG.
  - For bottleneck queue size,
    e.g. 100 means up to 100 packets can be stored before forwarding to AG.
- The NE module acts as a bottleneck link.
  - The NE module forwards all incoming packets to the AG module as fast as the bottleneck link rate.
  - If the incoming rate is higher than the bottleneck rate, incoming packets should be store in the bottleneck queue.
  - If the bottleneck queue is full, the next incoming packet will be dropped.

Receiver (2/2)



- If the AG module receives a data packet, sends an ACK to the sender.

- Every two seconds, print the following
  - incoming rate (#packets / 2secs) from the sender to NE
  - forwarding rate (#ACKs / 2secs) from NE to AG
  - avg queue occupancy (measure the queue occupancy every 100ms)

The Extension Goals

- Can you get goodput ratio (goodput / sending rate) close to 1
- Can you minimize the bottleneck queue occupancy?
- Is it possible for multiple senders to have a fair bandwidth?

while maximizing the utilization of the bottleneck link.

The Extension Goal Scenario

- Scenario (bottleneck link rate: 30 PPS, queue: 100, and initial sending rate : 30 PPS)
  - start sender1 at time 0 sec
  - start sender2 at time 30 secs
  - start sender3 at time 60 secs
  - stop sender3 at time 90 secs
  - stop sender2 at time 120 secs
  - stop sender1 at time 150 secs

- Make four graphs ;
  - goodputs of the three senders as time goes on
  - goodput ratios of three senders as time goes on
  - forwarding rate at the receiver as time goes on
  - queue occupancy at the receiver as time goes on

Evaluations

- Sender can send packets on the given the rate (20 pts)
  - print three information every 2 seconds
- Receiver can limit the forwarding rate (20 pts)
  - print three information every 2 seconds
- Multiple senders with one receiver (20 pts)
  - whether multiple senders can operate with a receiver (fairness doesn't matter)
- Report file (10 pts)

Evaluations

- The extension goals
  - goodput ratio (goodput / sending rate) gets close to 1 (5 pts)
  - queue occupancy gets as low as possible (5 pts)
  - multiple senders have fair bandwidth (5 pts)

  while maximizing the utilization of the bottleneck link (5 pts)

- If you design the extension goals in a stateless fashion at NE, you will get additional 10 points. **(10 pts)**

Deliverables

- The deadline is 6.10(Sun) 23:59.
  - For delayed submissions, a penalty of -15 points applies every 24 hours. After 72 hours, you get zero points.

- **Report** includes;
  - PDF file format
  - your development environment information in detail (operating systems, languages, compilers/interpreter)
  - how to run sender and receiver programs including the screen capture.
  - how to design this assignment such as data structures and algorithms (how to modify which module such as sender, NE, or AG)
  - **Show graphs if you develop the extension goals.**

- Sender and receiver source codes