Lab3-Part2 report (BBR)

Joohan Lee(joohanl@usc.edu)

1. Program Structure and Design

(1) ctcp.c and BBR

- In order to incorporate BBR into cTCP, redesigned cTCP.
- Created a new timer function in ctcp_sys_internal.c specifically in do_loop() function so that we can call send_front_segment_in_tx_buffer() at every pacing gap.
- do_loop() calls ctcp_pacing_timer() at every loop, and ctcp_pacing_timer function calls send_front_segment_in_tx_buffer at every pacing gap. Then, send_front_segment_in_tx_buffer function sends a segment if any in the tx buffer.
- Since we need pacing, we don't send pending segments right after ack was received anymore. All data segments will be sent through pacing.
- Every time a segment is sent or an acknowledgement is received, bbr_on_send, bbr_on_ack functions are called respectively.

(2) BBR (ctcp_bbr.c)

- I created bbr_model struct to gather bbr objects and functions in one place for readability and maintanance purpose.
- Each state(connection) has one bbr_model.
- Each transformation_info includes segment and information about transmission and ack. Also, I used size of 8 of windows and each window has time and bw.

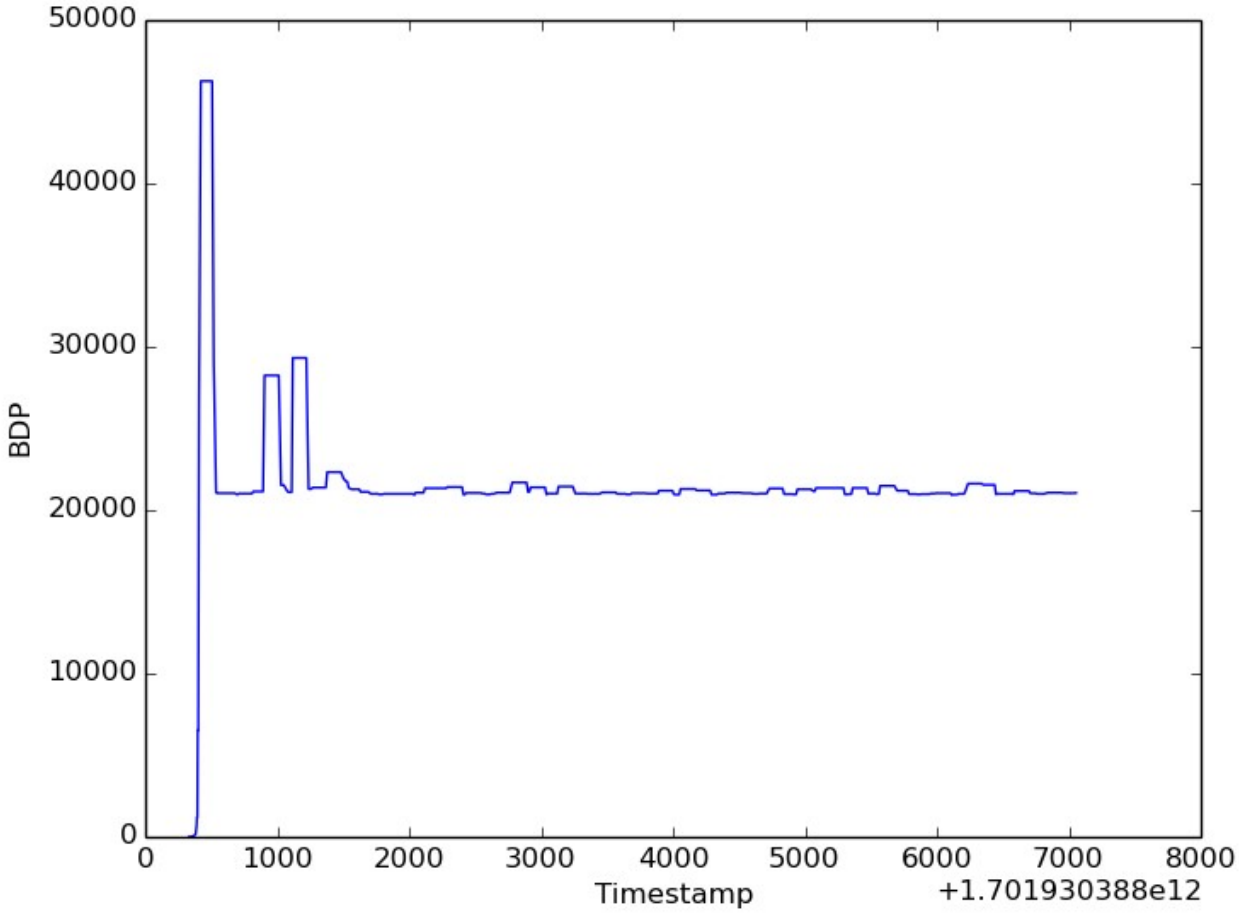
```
struct ctcp_bbr_model {
    ctcp_bbr_t* bbr_object;
    void (*on_send)(ctcp_state_t*, ctcp_transmission_info_t*, ctcp_bbr_t*);
    void (*on_ack)(ctcp_state_t*, ctcp_transmission_info_t*);
};
typedef struct ctcp_bbr_model ctcp_bbr_model_t;
```

```
struct rate_sample {
    uint64_t  prior_mstamp; /* starting timestamp for interval */
    signed int  delivered;      /* number of packets delivered over interval */
    uint16_t is_app_limited;    /* is sample from packet with bubble in pipe? */
};
typedef struct rate_sample ctcp_rs_t;
```

2. Implementation Challenges

- Lots of transitions to each state can happen with various conditions. Designing and tests were extremely difficult. I tried to pack up related objects and functions.
- Detailed measurement of RTT and BW was challenging.

3. BDP Plot



- BDP expressed units of bits.
- This plot was drawn while running bbr_mininet.sh
- In dumbbell topology, all files including binary files can be received properly.

4. Remaining Bugs

- BDP plot always converges, but still, maximum bdp may vary at many different runs of codes.

Others

- From my inspects, when the code is run, bbr_mode tends to transition from [STARTUP->DRAIN->PROBE_BW<->PROBE_RTT]. However, in most case on sending, it is in PROBE_BW and PROBE_RTT rarely happens.