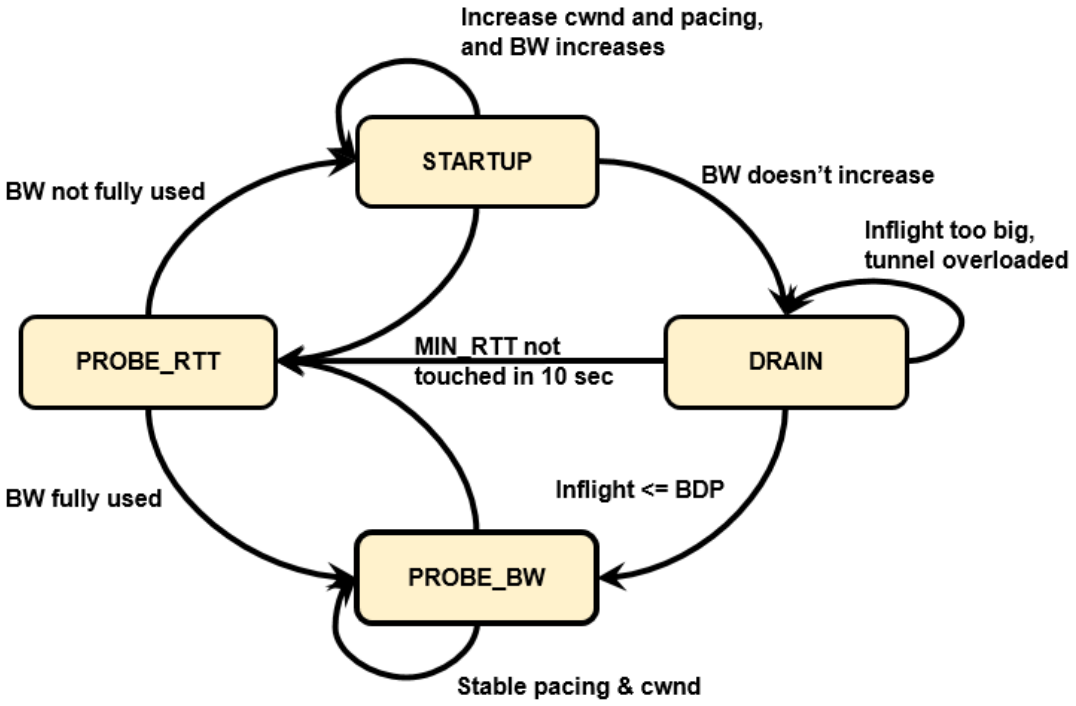
Report_2 for Lab 4

Yifan Wu

1. High level design of BBR (state machine)



2. Detailed Implementation

(1) Add ctcp_bbr.c and ctcp_bbr.h, which contains struct and functions for bbr.
(2) Modify some functions in ctcp.c to introduce bbr, and some attributes of ctcp_state in ctcp_h (adding bbr struct to it).
(3) Modify arrtibutes of ctcp_segment in ctcp_sys.h, to fit the need of bbr
(4) Comment some code in ctcp_sys_internal.c, in order to deal with some issue, which will be mentioned below.

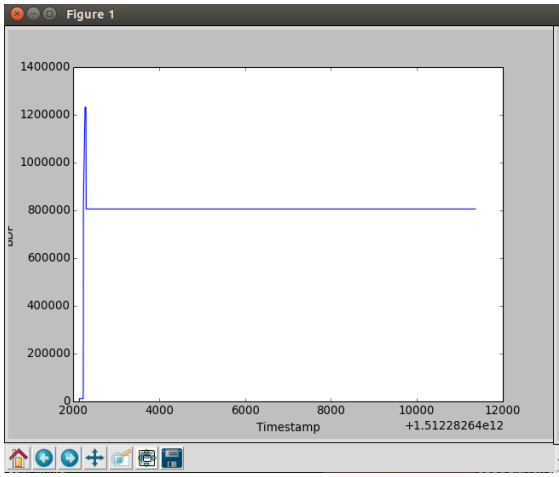
3. Implementation Challenges

(1) At first the bbr kernel code is hard to read, because there are a lot calls from other kernel files, which is hard to track. But when combined with the state machine and ignore trivial stuff of the kernel code, it becomes easier.
(2) It's not an easy job to add bbr to ctcp directly. So when trying to combine them together, I modified some attributes in the original struct.
(3) When I try to transfer binary files to server, it shows some weird behavior, which is the transferred file is sometimes one or several bytes larger or smaller than the original file. I tried

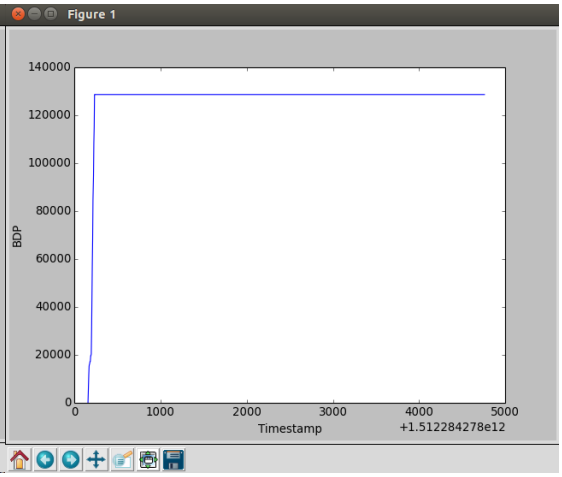
to compare the files carefully and look up via Google, finally it seems that's because of the ctcp_sys_internal.c, which modifies "\n" to "\r\n" in conn_input() (about line 740). So I just comment these codes and it works fine, and I think that's reasonable.

4. Testing Cases

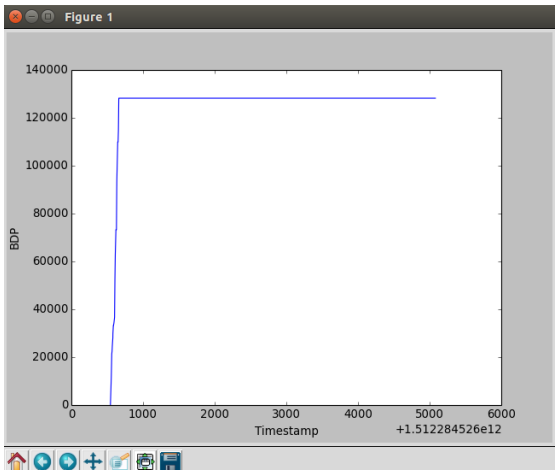
For testing, I tried topologies offered in the lab description, and the results are below:



1. Simple topology



2. Line topology



3. Dumbbell topology

5. Remaining Bugs

(1) The graph is not perfect, which doesn't show clear phase change in the graph.
(2) ProbeRTT state is temporarily missing, so for large files (several decades or hundreds MB), there will be no state change is the graph, which means the state is stable in ProbeBW.