Problem 1.5 Performance evaluation

Plots

Method C with `pp.au`

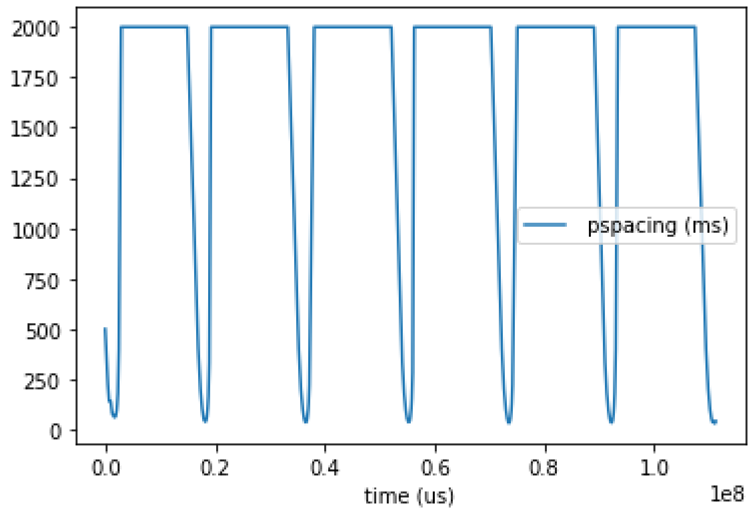
On one server:

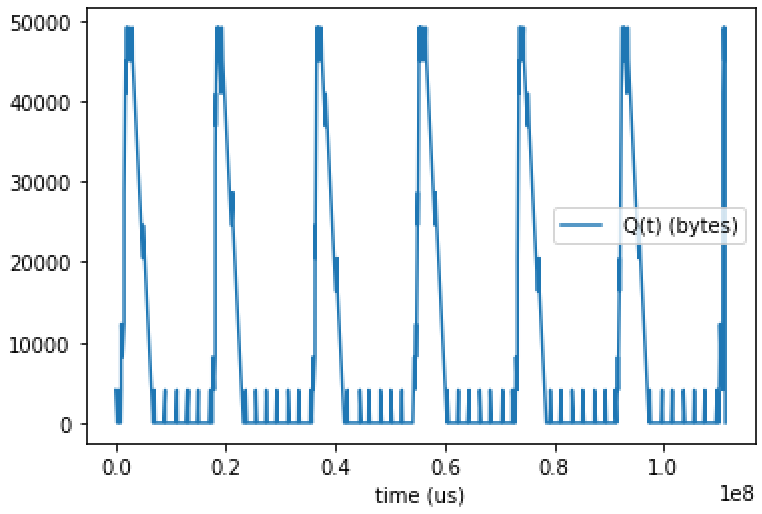
```
./audiosrv 128.10.112.135 22222 2 logsrv
```

One one client:

```
./audiocli 128.10.112.135 22222 pp.au 4096 12 6 2 0 logcli
```

- Block size: 4096 bytes
- Buffer size: 48 KB (12 * 4096 bytes)
- Target buffer size: 24KB (6 * 4096 bytes)
- Initial influx rate: 2 packets/s
- Method C
    - Epsilon: 0.5





Method D with `pp.au`

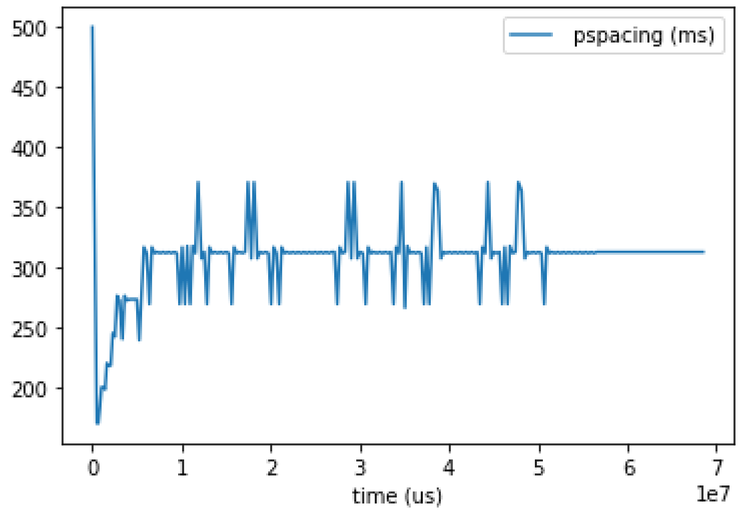
On one server:

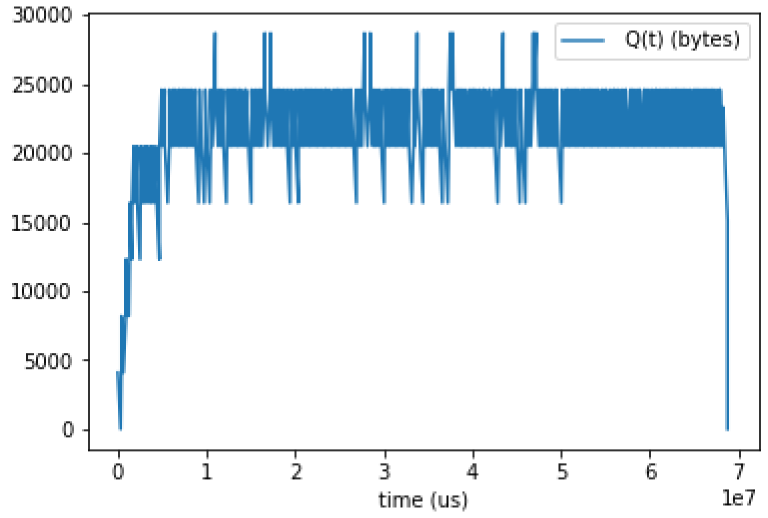
```
./audiosrv 128.10.112.135 22222 2 logsrv
```

One one client:

```
./audiocli 128.10.112.135 22222 pp.au 4096 12 6 2 1 logcli
```

- Block size: 4096 bytes
- Buffer size: 48 KB (12 * 4096 bytes)
- Target buffer size: 24KB (6 * 4096 bytes)
- Initial influx rate: 2 packets/s
- Method D
    - Epsilon: 0.5
    - Beta: 1.1





Method C with `kj.au`

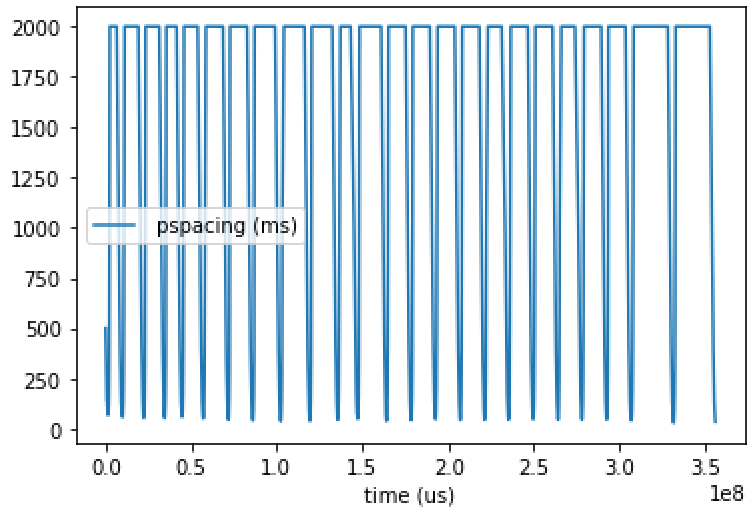
On one server:

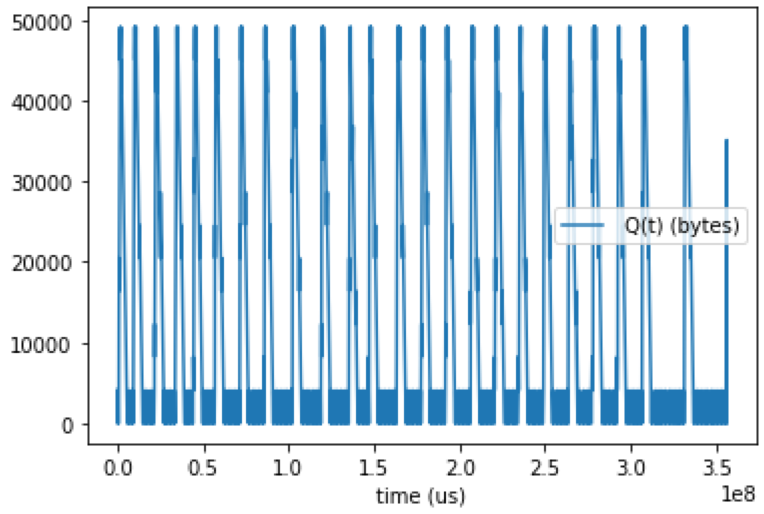
```
./audiosrv 128.10.112.135 22222 2 logsrv
```

One one client:

```
./audiocli 128.10.112.135 22222 kj.au 4096 12 6 2 0 logcli
```

- Block size: 4096 bytes
- Buffer size: 48 KB (12 * 4096 bytes)
- Target buffer size: 24KB (6 * 4096 bytes)
- Initial influx rate: 2 packets/s
- Method C
    - Epsilon: 0.5





Method D with `kj.au`

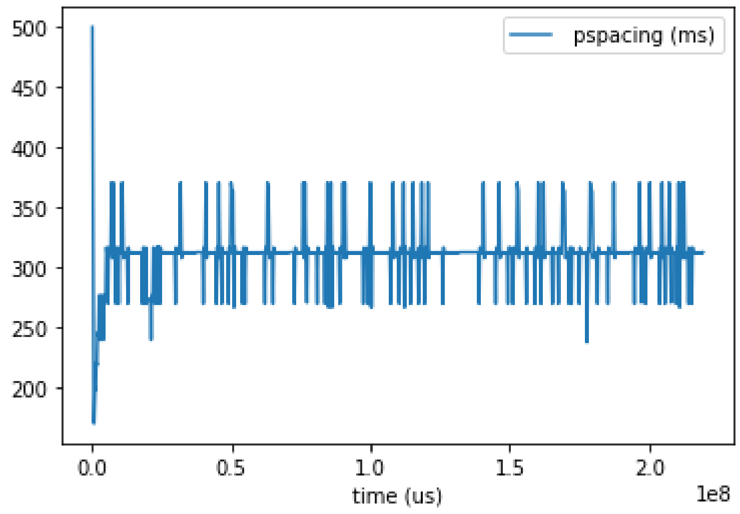
On one server:

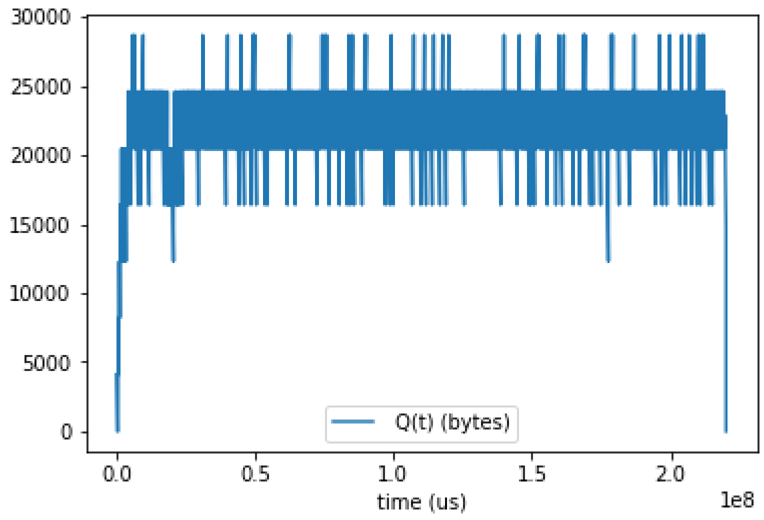
```
./audiosrv 128.10.112.135 22222 2 logsrv
```

One one client:

```
./audiocli 128.10.112.135 22222 kj.au 4096 12 6 2 1 logcli
```

- Block size: 4096 bytes
- Buffer size: 48 KB (12 * 4096 bytes)
- Target buffer size: 24KB (6 * 4096 bytes)
- Initial influx rate: 2 packets/s
- Method D
    - Epsilon: 0.5
    - Beta: 1.1





Analysis

According to the plots, the influx rate (`pspacing`) and buffer load are unstable with congestion control method C for both files. This causes the audio playing intermittently as the buffer gets empty or full periodically.

On the other hand, with method D, the `pspacing` and buffer load are both stable though some fluctuations occur due to the network status. Thus, the audio plays smoothly without getting the buffer empty or full.

Bonus Problem

Definition of Method E

To improving the method D, we decide to implement the derivative term of the PID controller. The formula is derived as follows.

$$\lambda(t) = K_p e(t) + K_i \int_0^\tau e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$

We can differentiate both sides of the equation.

$$\frac{d}{dt}\lambda(t) = -K_p(\lambda(t) - \gamma) + K_i(Q^* - Q(t)) - K_d \frac{d}{dt}\lambda(t)$$

Assuming *Kd* is always larger or equal to zero, we can transpose the derivative term and finally get the *lambda(t + 1)*.

$$\lambda(t+1) = \lambda(t) + \frac{-K_p(\lambda(t) - \gamma) + K_i(Q^* - Q(t))}{(1 + K_d)}$$

Plots

Method E with `pp.au`

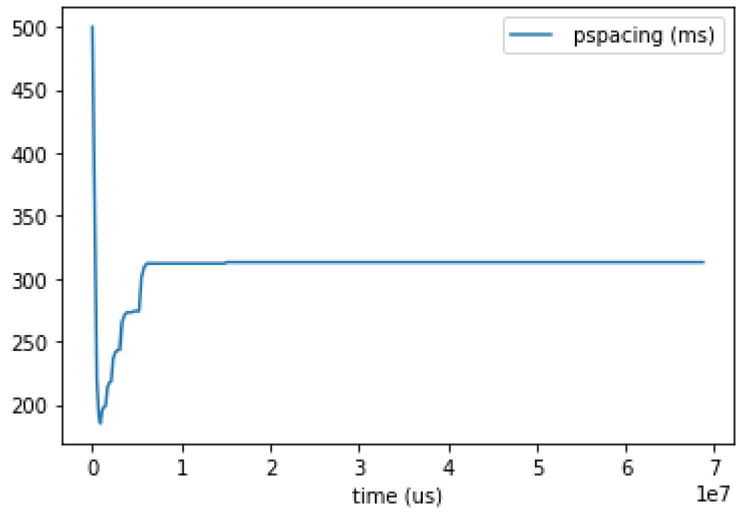
On one server:

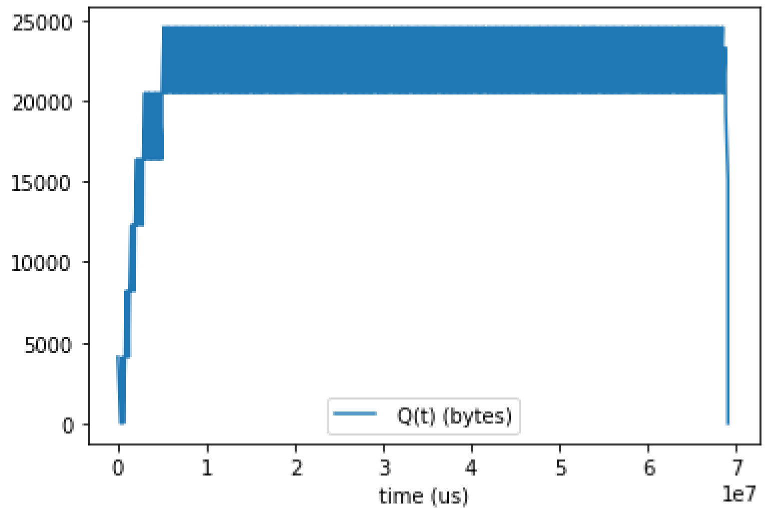
```
./audiosrv 128.10.112.135 22222 2 logsrv
```

One one client:

```
./audiocli 128.10.112.135 22222 pp.au 4096 12 6 2 1 logcli
```

- Block size: 4096 bytes
- Buffer size: 48 KB (12 * 4096 bytes)
- Target buffer size: 24KB (6 * 4096 bytes)
- Initial influx rate: 2 packets/s
- Method E
    - Epsilon: 0.5
    - Beta: 1.1
    - Alpha: 0.5





Method E with `kj.au`

On one server:

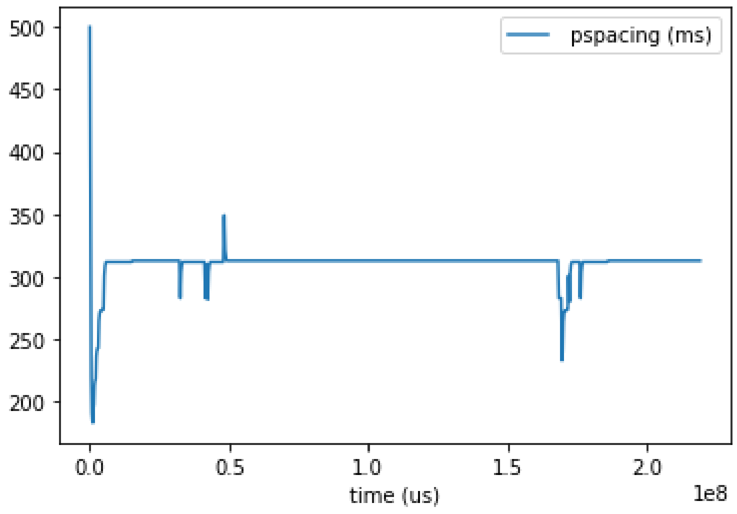
```
./audiosrv 128.10.112.135 22222 2 logsrv
```

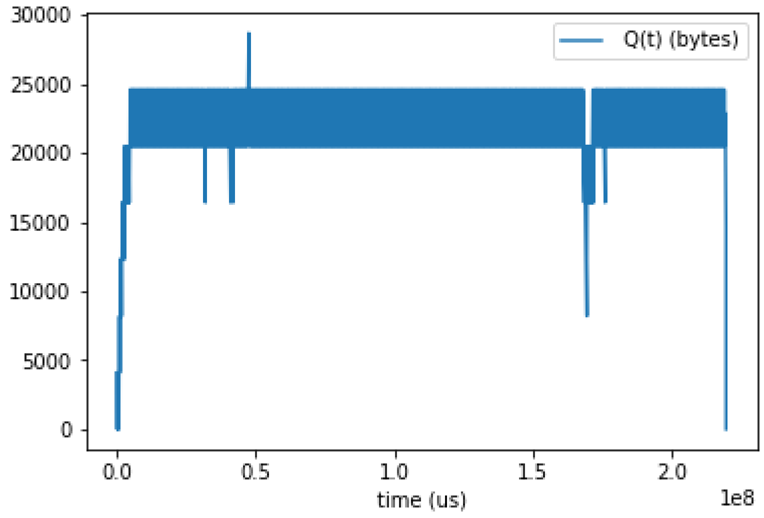
One one client:

```
./audiocli 128.10.112.135 22222 kj.au 4096 12 6 2 1 logcli
```

- Block size: 4096 bytes

- Buffer size: 48 KB (12 * 4096 bytes)
- Target buffer size: 24KB (6 * 4096 bytes)
- Initial influx rate: 2 packets/s
- Method E
    - Epsilon: 0.5
    - Beta: 1.1
    - Alpha: 0.5





Analysis

After apply the "D" term of the PID formula, the plots indicate that the transmission is more stable compared to the plots of method D. Though there are several unstable events possibility caused by the network status at the time, the overall congestion control performance is better than other methods.