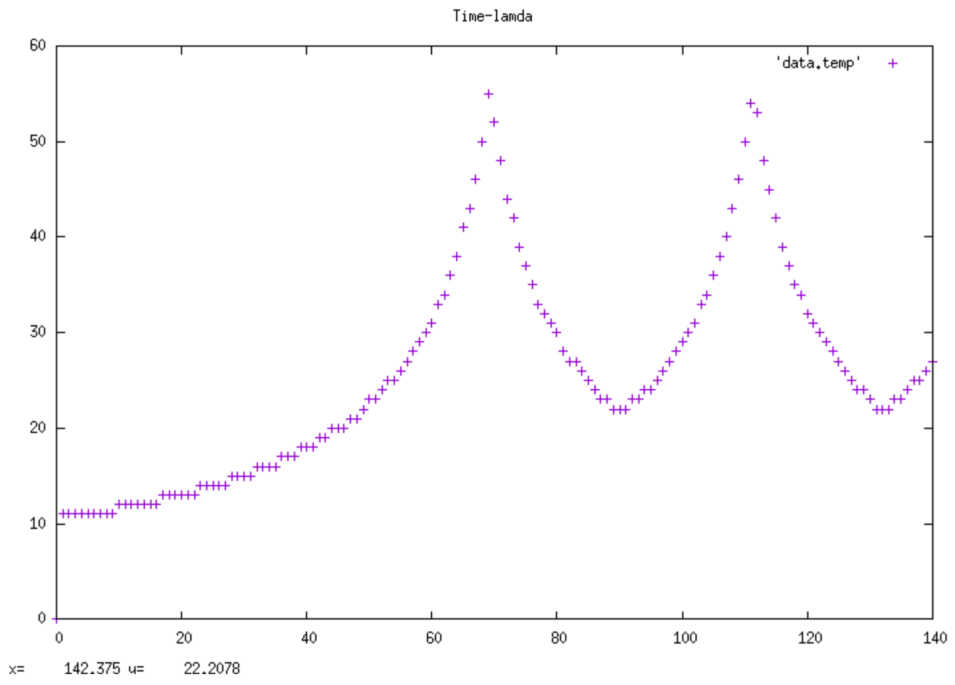
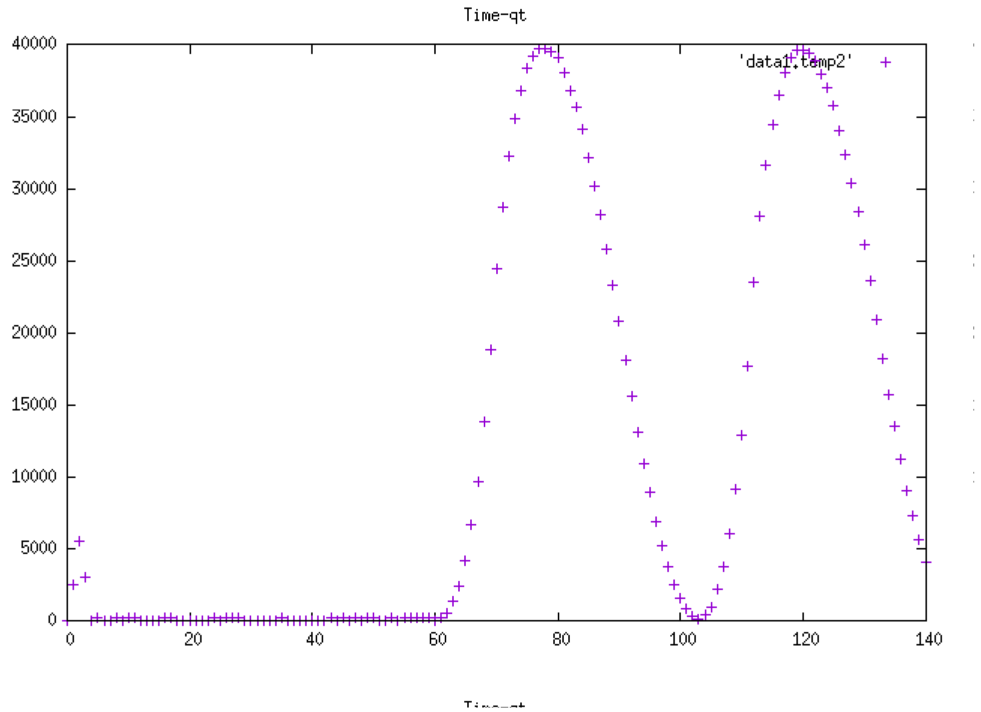
P1

Method 1

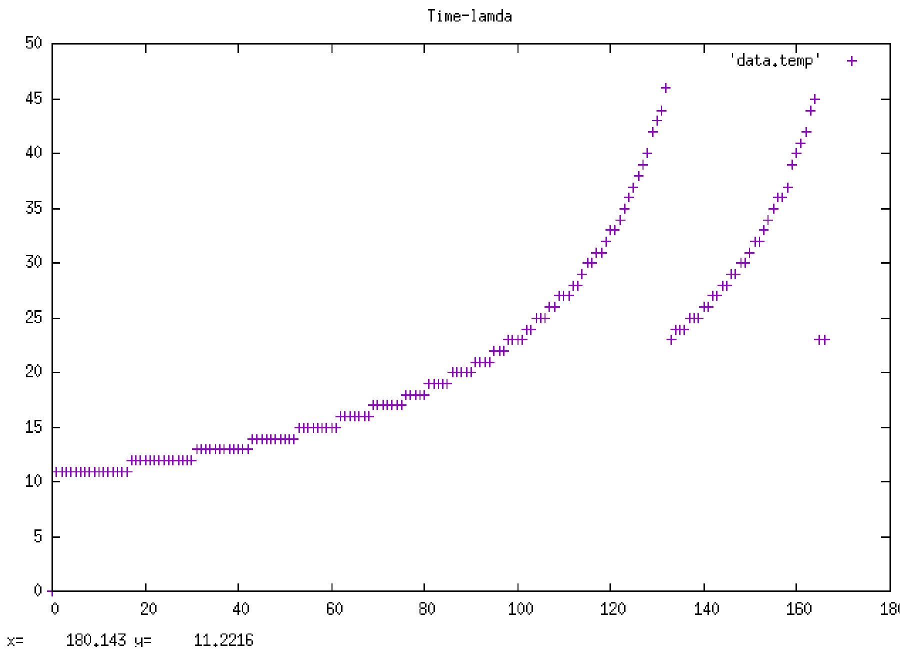
Lamba zig-zag around 40. And is not stable.

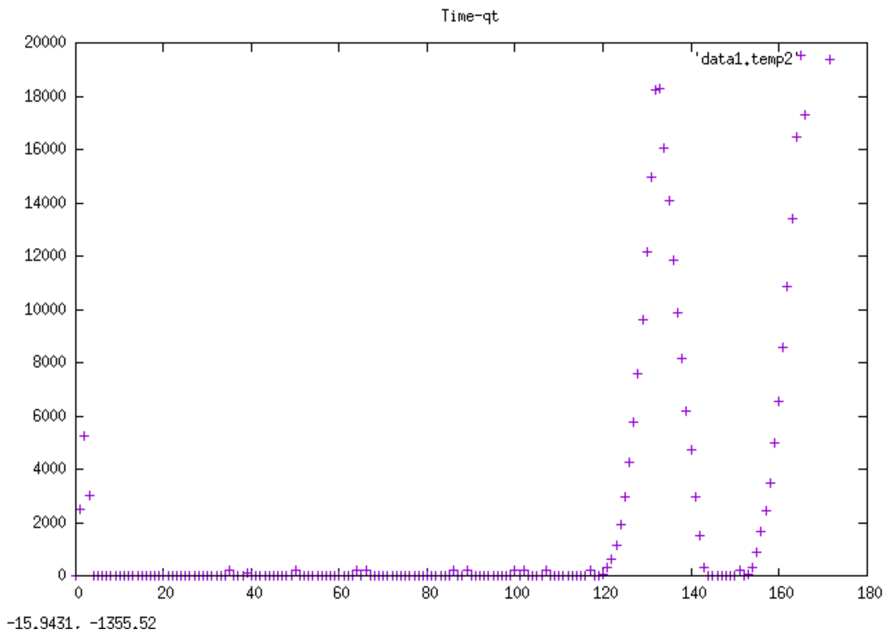
Q(t) range from 0 -40000. Is not stable.





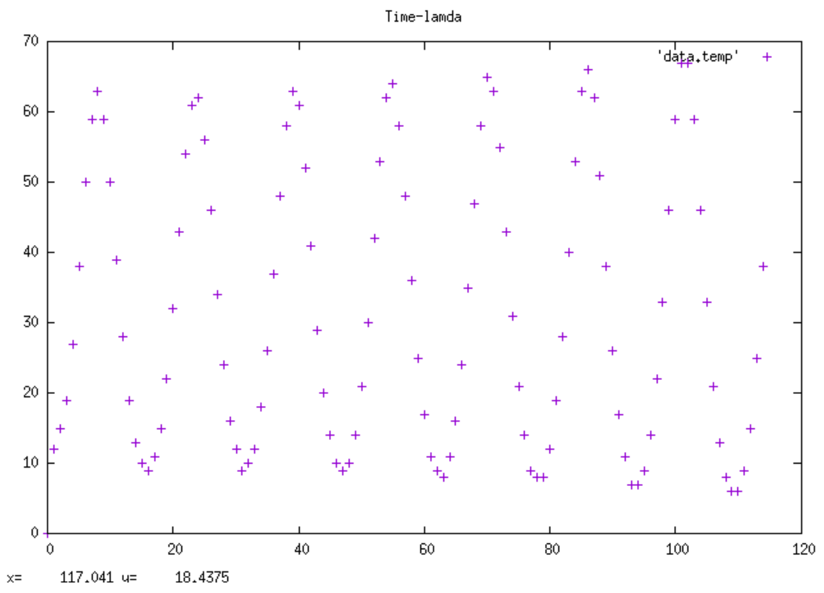
Method 2
Lamba zig-zag around 40. And is not stable.
Q(t) range from 0 to 20000. Is not stable.

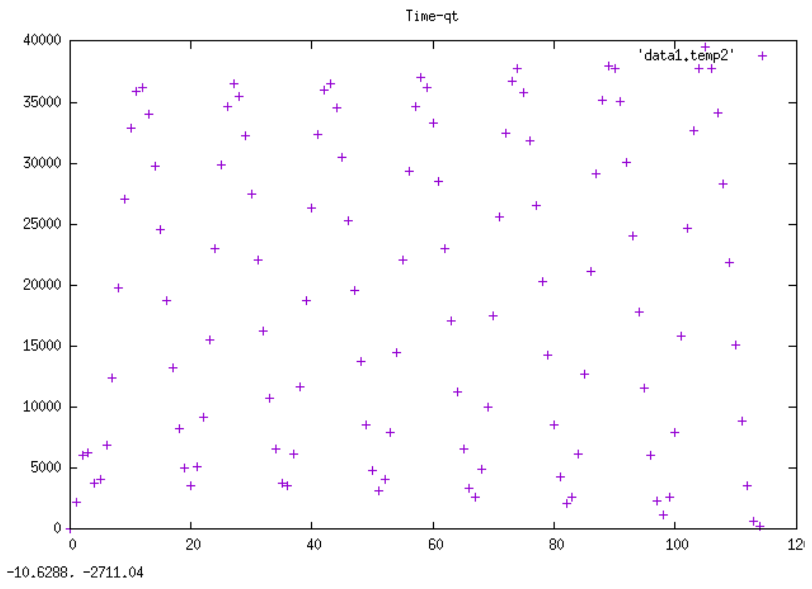




Method 3

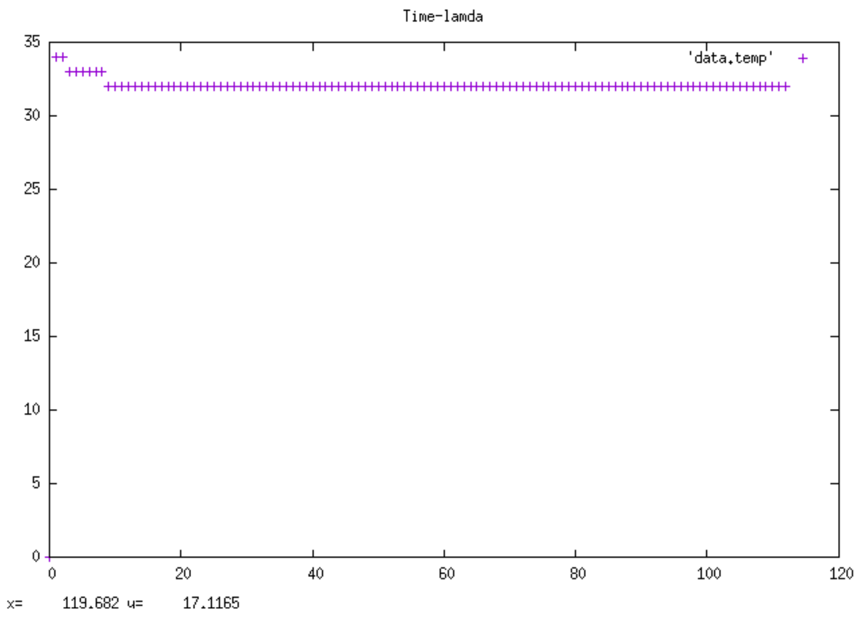
Lamba zig-zag around 40. And is not stable.
Q(t) range from 0 to 40000. Is not stable.

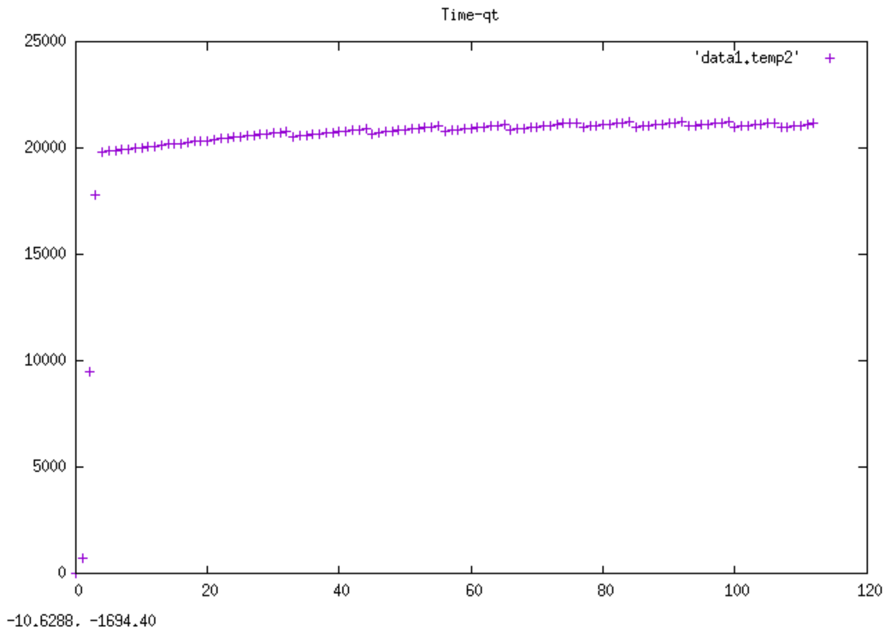




Method 4

Lamba is around 32. And is stable.
Q(t) is stable around 20000.





We can see that the most significant difference between method 4 and other methods is the that lambda on server side will not quickly change.It finally stay on a stable level.The tramission rate is round the same as the playback rate.

And the Q(t) is around the taget Q*, which means method 4 can fully utilized the buffer space.

Two client:

When running two client at the same time. I could not directly tell the difference with only one client. And the total completion time of both the client is the same. I think this might be because the server is efficient and the overhead is low. We could expect the delay will increase as the number of client increase if multiple clients are running at the same time.

In problem 2. We will realized retransmission with special data structures.

On the server side, a circular buffer is maintained that record the last K packet sent out and their ID. if a retransmission request is got, we will check the ID of the packets in the circular buffer. If we have it we will transmit this packet again, otherwise, ignore it.

On the client side, a similar but more complicated circular buffer with list will be maintained. In this structure. One entry store the packet to be played. One entry packet store the sequence number of the packet. The third entry store if the packet is here or not.

We check the lost of a packet by counting the sequence number of the received packet. If one packet's number is >=2 packet number of the previous packet number. We can figure out that one packet is lost, we will leave this position in the list with marking this packet is lost.

Thus, even if the packet is lost, we still have the position to place the packet on our buffer if the retransmission is successful. We will go through the circular buffer to look for the packet number. If we could not find the same packet number discard this packet.