EE5903-CA3-Report
Srikanth Ayalasomayajulu-A0077017

Introduction: This CA3 report deals with the development of real time audio streaming system which serves the incoming audio packets from server, decodes and processes them into audio blocks and in turn send them to audio devices like head phones.

Key specifications:

- Given that arrival of packets are poison distributed whose cumulative distribution function(CDF) can be given as $e^{-\lambda}$ (λt^n/n!)Where λ is the arrival rate and t is the time so that number of packets that arrived in a given time interval can be given as $\rho= \lambda*t$.
- Given that packets arrive with a minimum gap of 2ms between them and average arrival rate is 100 packets/sec.
- The decoding of a block requires approximately 5 ms to complete but may be performed in 5 steps of approximately 1 ms.
- Should be implemented withOUT of the help of multiple threads.

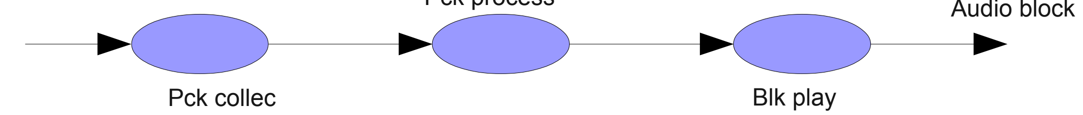
Description of the implemented design and comparison with previous implementation:

- ✔ The implemented design in this exercise(CA3) is NOT much different from the previous implemented exercise(CA2).
- ✔ The three THREADS implemented in CA2 is replaced with three SUBROUTINES in this exercise. They are **`1) pck_collec() 2) pck_process() 3) blk_play().`**
- ✔ All the three subroutines are called sequentially in the **`main().`**
- ✔ Efficiently used the implementation of 2 circular queues. First queue is used for storing of packets and Second queue is used for storing blocks. The reasonable buffer sizes implemented are 200(packets) and 20(blocks) as these values are justified in many executions.
- ✔ Use of *posix* threads and synchronizing elements like *joins* implemented in CA2 are removed in CA3.
- ✔ Blocks start playing after pre-buffering of 10 blocks in its respective buffer. Once packet is played it is made deleted from its buffer.

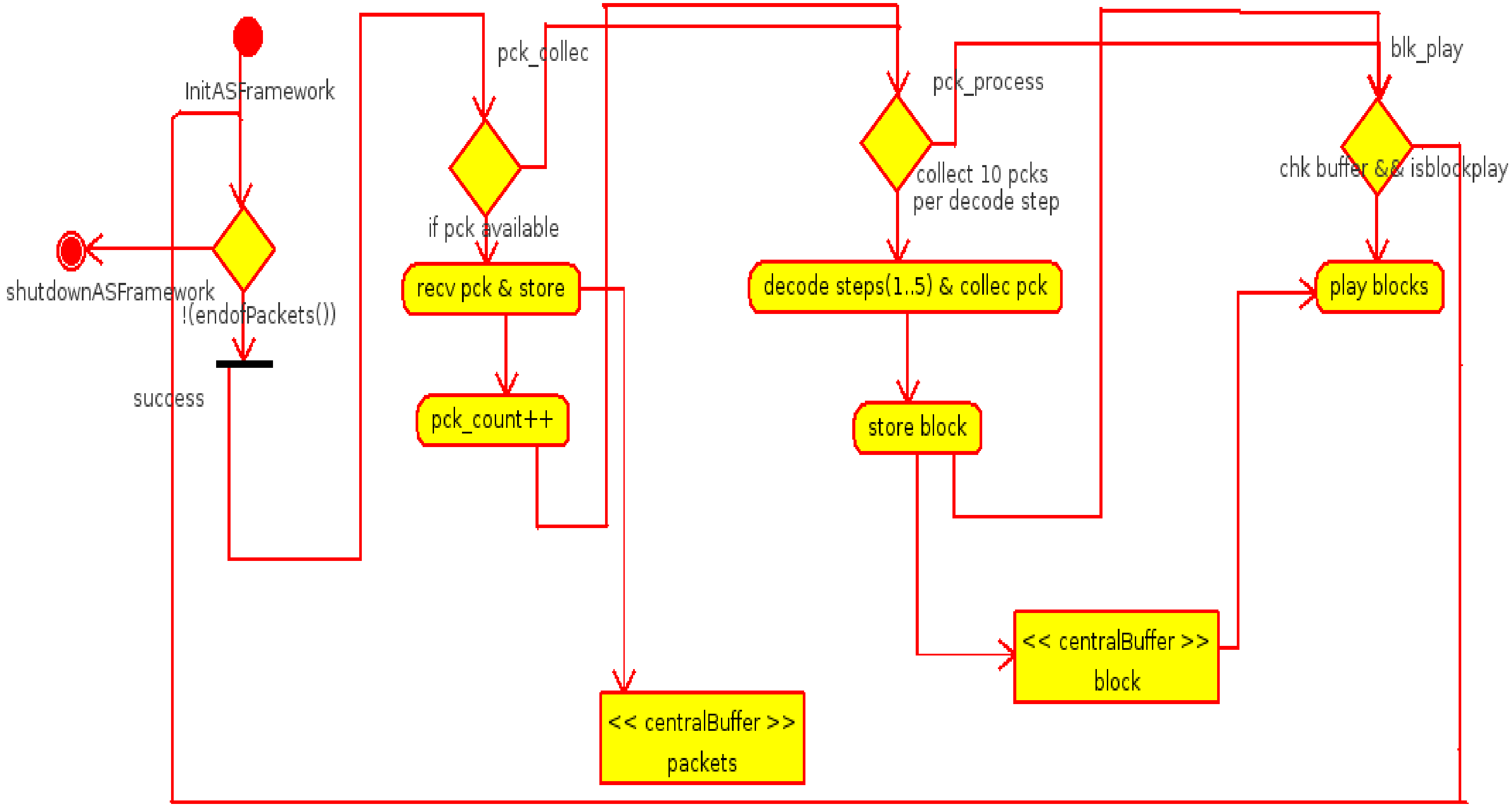
Comparison in Performance:

I am able to achieve similar output as obtained in CA2. I am able to reduce more missing blocks when compared to previous exercise.(i.e 7(previously) to 0(present)) and so can able to produce more quality output in present version. In the rest parameters(i.e. Dropped packets, initial delay, missed blocks) i am able to get same performance as i achieved in previous exercise.

Sequential flow:



Activity Diagram: Activity diagram placed below is in Landscape mode.


InitASFramework
shutdownASFramework
!(endofPackets())
success
pck_collec
if pck available
recv pck & store
pck_count++
<< centralBuffer >>
packets
pck_process
collect 10 pcks
per decode step
decode steps(1..5) & collec pck
store block
<< centralBuffer >>
block
blk_play
chk buffer && isblockplay
play blocks