Computer Networks Lab Report

111701019 Md Talha Yaseen Khan

Lab – 3 (MAC Protocols)

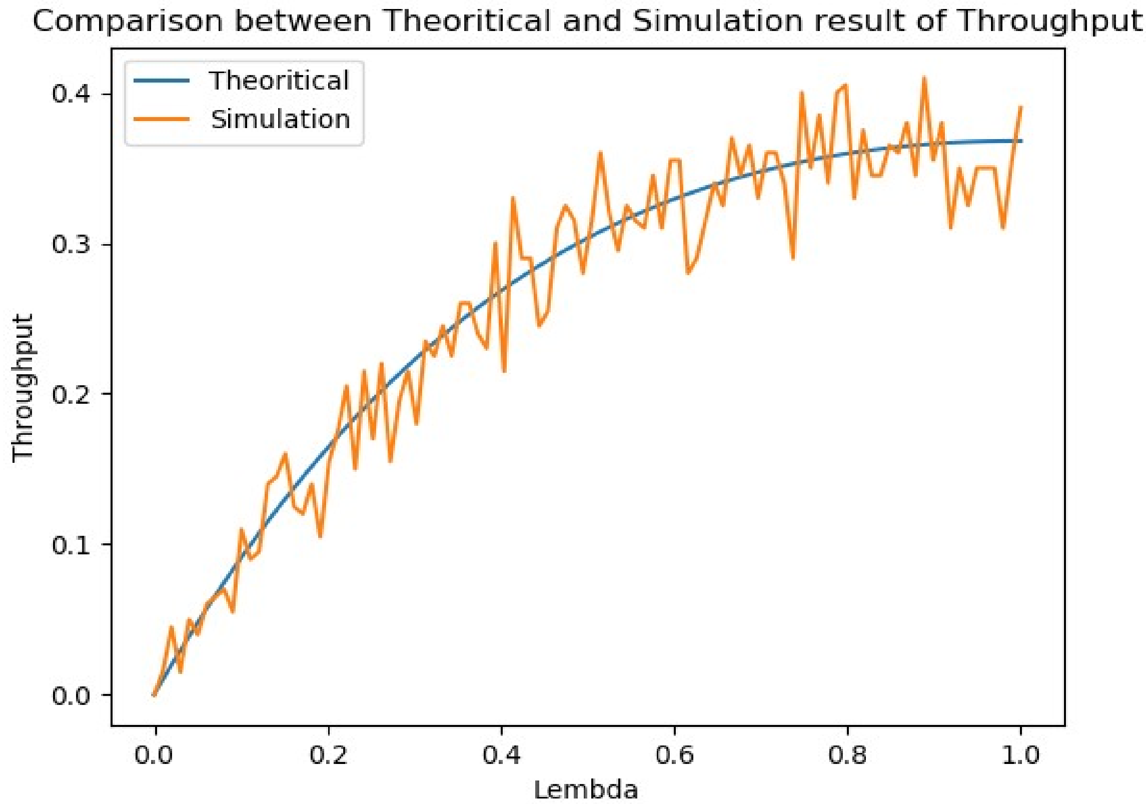
1.

a) Slotted ALOHA

- This code required 3 inputs
  - n (Number of users per slot, which is 100 here)
  - nL (Number of λ you want to generate)
  - nS (Number of slots)
- Users are generating frame with probablity λ/n.
  - X_i = { 0 ; if not generated | 1 ; if generated }
  - G = Expected Transmisson = Σ 0(1 – λ/n) + 1(λ/n) = λ
  - Throughput for Slotted ALOHA = S = G e^{-G} = λ $e^{-\lambda}$ (For theoritical prediction)
- For simulation result, I am generating “nL” random λ where λ € [0, 1] and simulating the throughput for each lembda.
- Then by fixing a λ, for each “nS” slots and “n” users in each slots generating frame with probablity λ/n and checking for collision. If “frames” value comes 1 means only one frame transmitted i.e. no collision and if there are only one frame generated the increasing successfully frame generated count (cnt) by 1.
- Then calculating throughput (= number of successfully generated user frame / total slots) and appending this on throughput list.
- Repeat above steps for every λ.

- Then plotted the both theoritical and simulation result graph in one frame for comparison.

Below is the comparison between theoritical and simulated result of throughput for (n = 100, nL = 100 and nS = 200).



b) p-persistent CSMA

- This code required 3 inputs
  - n (Number of users per slot, which is 100 here)
  - nL (Number of λ you want to generate)
  - nS (Number of slots)
- First I stored value of ***p*** in a list named pS (pS[0] = 0.5 and pS[1] = 0.01). And then initialized a 2D list for simulation result throughput for both ***p*.**
- Then generating "nL" random λ, where λ € [0, 1] and simulating the throughput for each lembda for both ***p***.

- Then by fixing a λ, for each "nS" slots and "n" users in each slots generating frame with probablity λ/n and storing value in "frameGen" and adding into queued list for that user.
- If channel is found idle and frames are queued, then sending frame with probablity p = pS[k] and waiting for to send in start of next slot with prbablity 1-p.
- If frame is sent, increase the frameCnt (counting frame which are sent) by 1 and append that user into the sentUser and decrease the queued value by 1.
- Set timer when slot became again idle is 3.
- If more than 1 or no frame sent, increase queued value by 1 for each sent users and if only one frame sent, increase the successful frame count by 1.
- Then calculating throughput (= number of successfully generated user frame / total slots) and appending this on throughput list.
- Repeat above steps for every λ.
- Then plotted the simulation result for both ***p = 0.5 and p = 0.01*** graph in one frame for comparison.

Below is the comparison between simulated result for both *p = 0.5 and p = 0.01* of throughput for (n = 100, nL = 100 and nS = 200).