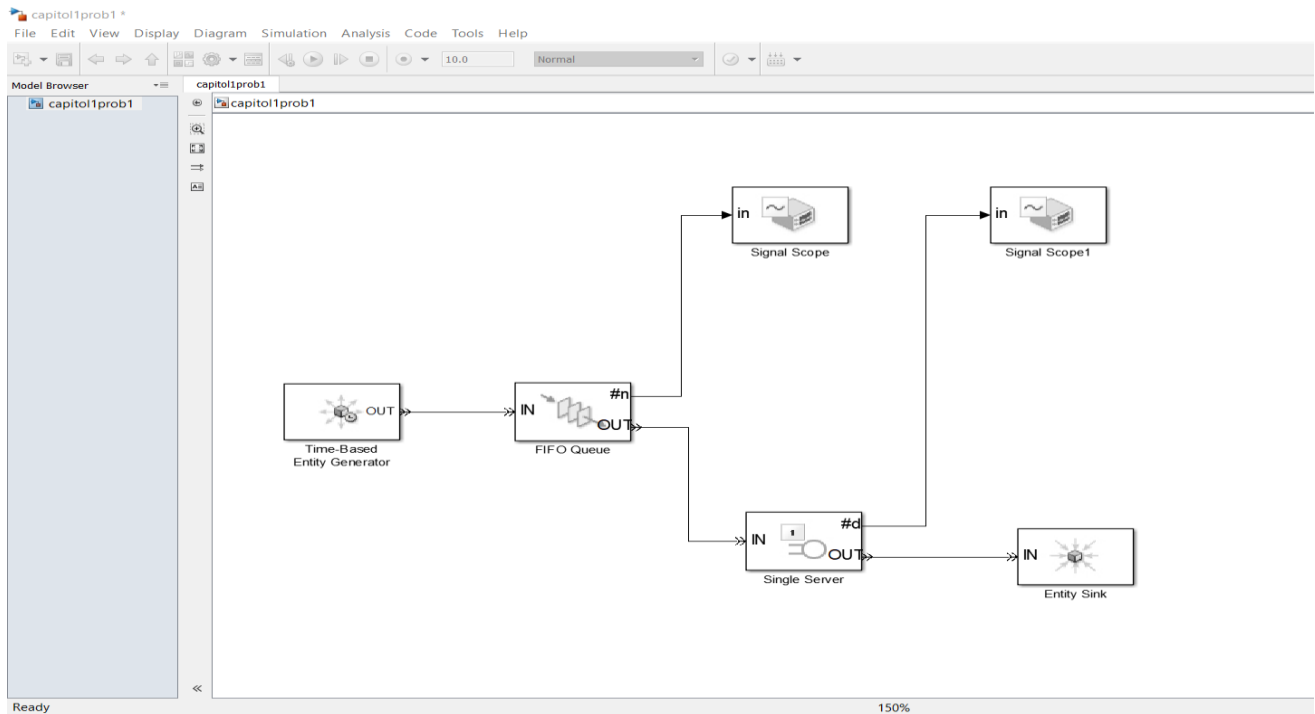


# **Capitol 1 / Lucrarea 1**

## **Crearea unui model SimEvents**

## Problema 1



Initial blocul Single Server are timpul de deservire de o secunda.

a. Modificam timpul de deservire la 1.2 secunde

**Block Parameters: Time-Based Entity Generator**

Time-Based Entity Generator  
Generate entities using intergeneration times from a signal or a statistical distribution.

Entity Generation   Blocking   Entity Type   Statistics

Generate entities upon: **Intergeneration time from dialog**

Distribution: **Constant**

Period: **1**

Generation event priority: **300**

☒ Generate entity at simulation start

OK Cancel Help Apply

**Block Parameters: Single Server**

Single Server  
Serve one entity for a period of time and then attempt to output the entity through the OUT port. If the OUT port is blocked then the entity stays in this block until the port becomes unblocked. You can specify the service time, which is the duration of service, via a parameter, attribute, or signal.

When the block permits preemption, an entity in the server can depart early via the P port. Preemption occurs only if attributes of the current entity and the entity attempting to arrive satisfy specified criteria.

When the block does not permit preemption, the IN port is unavailable whenever this block stores an entity.

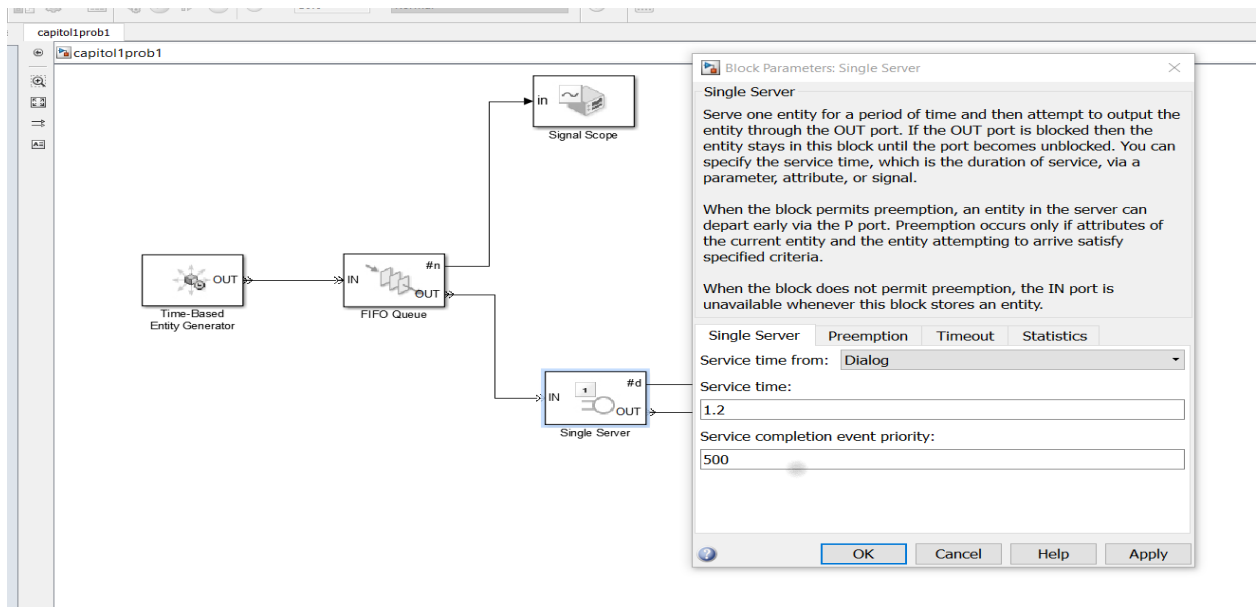
Single Server   Preemption   Timeout   Statistics

Service time from: **Dialog**

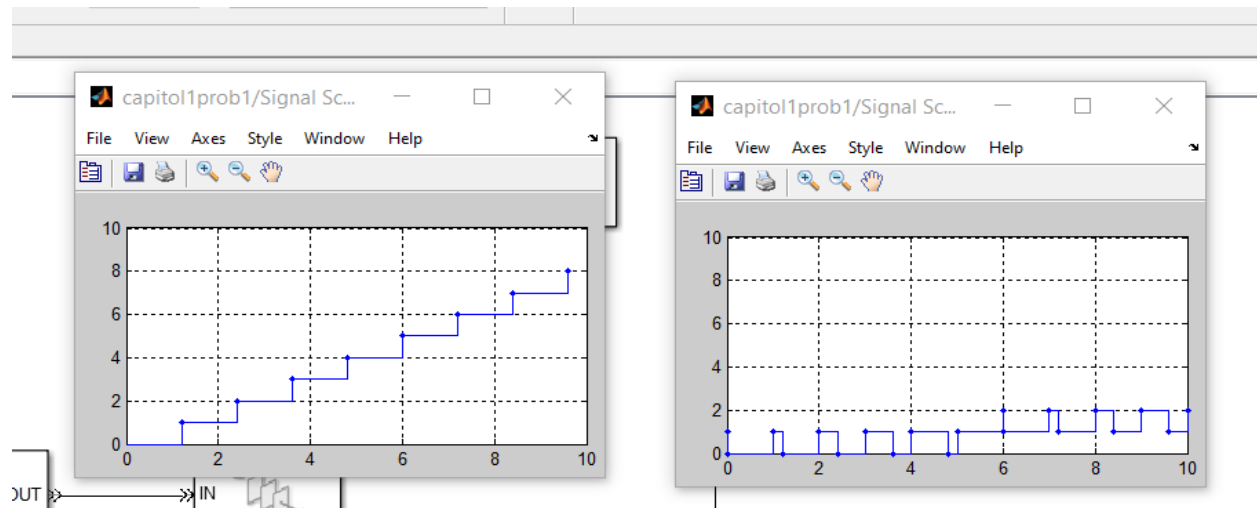
Service time: **1**

Service completion event priority: **500**

OK Cancel Help Apply

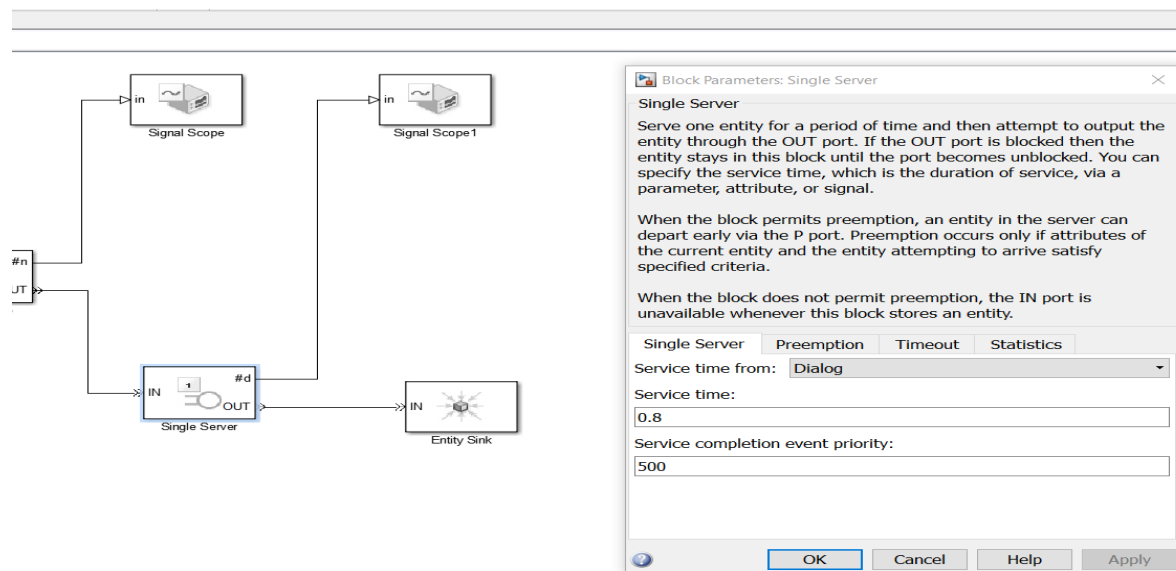


Apoi simulam modelul si obtinem urmatoarele grafice

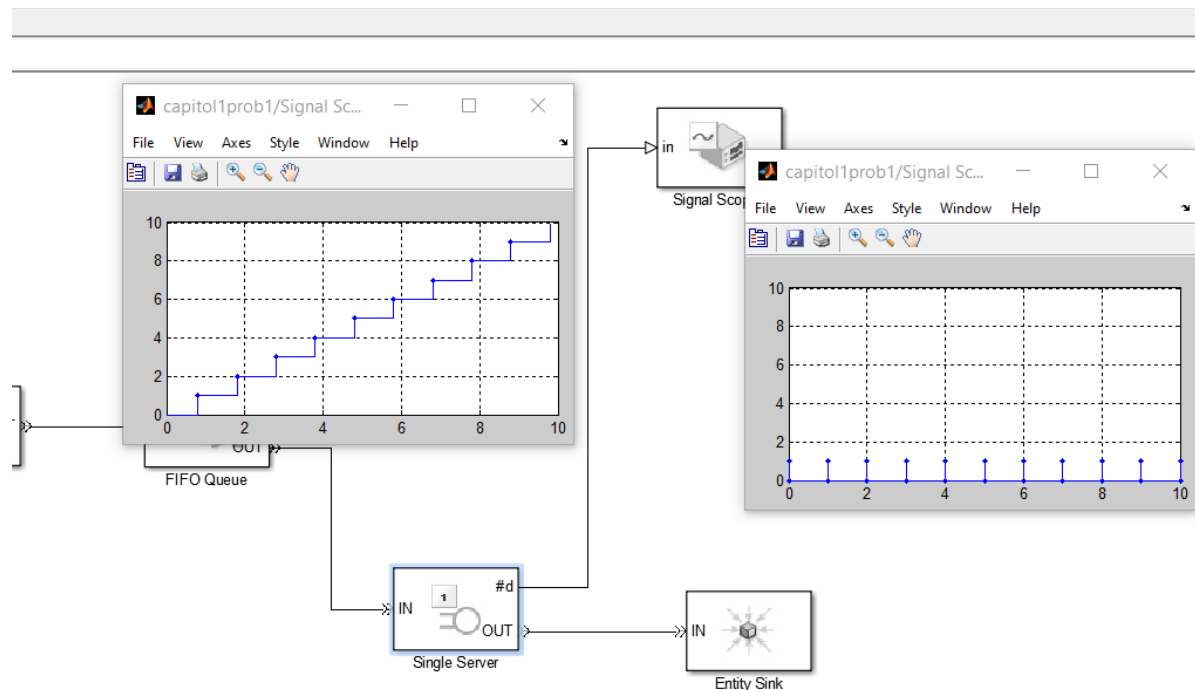


**Concluzie:** In acest caz timpul de deservire este mai mare decat intervalul dintre aparitia a doua entitati ( $1.2s > 1s$ ). Aceasta duce la o intarziere de  $0.2s$  in care FIFO Queue nu poate sa transmita mai departe entitatile, deoarece FIFO Queue stocheaza entitatile generate de Time - Based Entity Generator si le transmite spre Single Server. Astfel, pana cand cele  $1.2s$  de deservire ale lui Single Server nu se termina atunci nu se mai pot transfera entitati.

b. Modificam timpul de deservire la  $0.8$  secunde

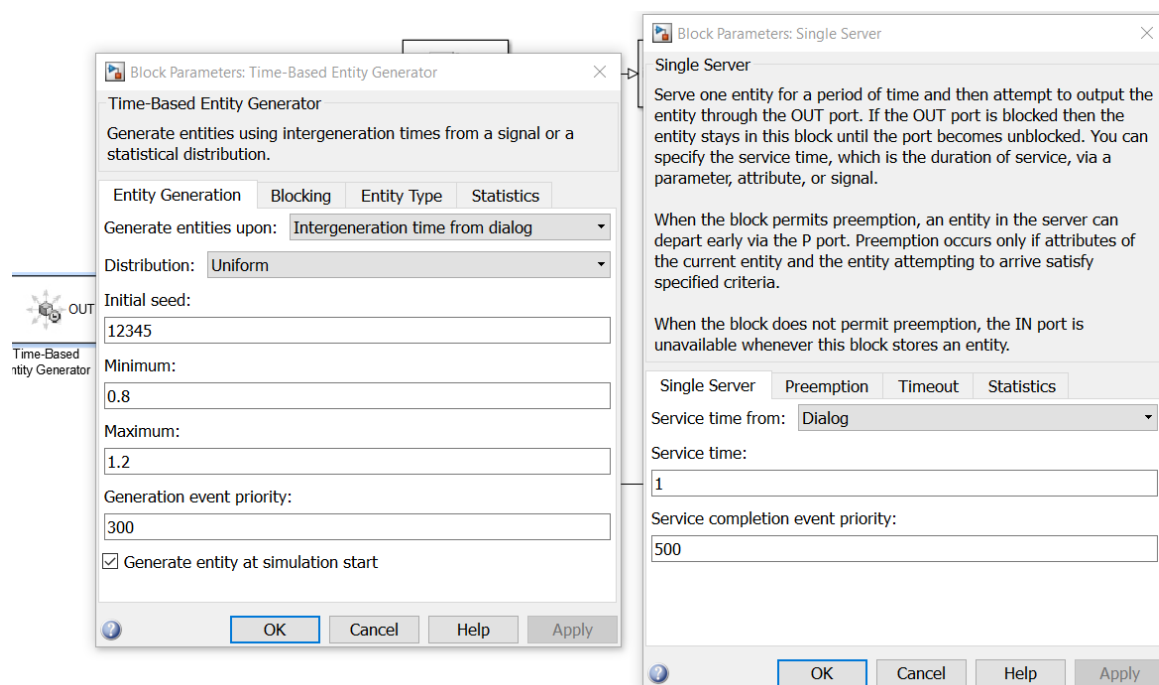


Apoi simulam modelul si obtinem urmatoarele grafice

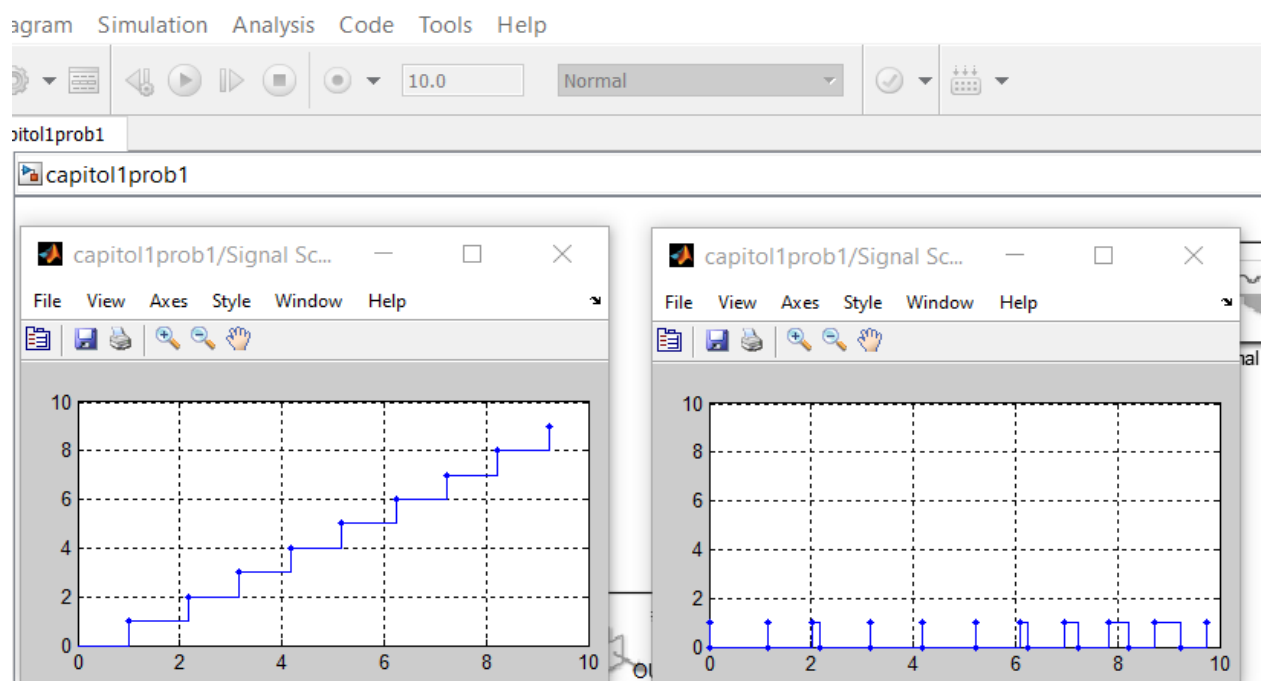


**Concluzie:** Comparand rezultatele cu cele anterioare putem observa ca atunci cand timpul de deservire este mai mic sau egal cu intervalul dintre aparitia a doua entitati, nu mai exista un timp de intarziere sau de pauza in care nu se mai pot genera sau transmite entitati mai departe.

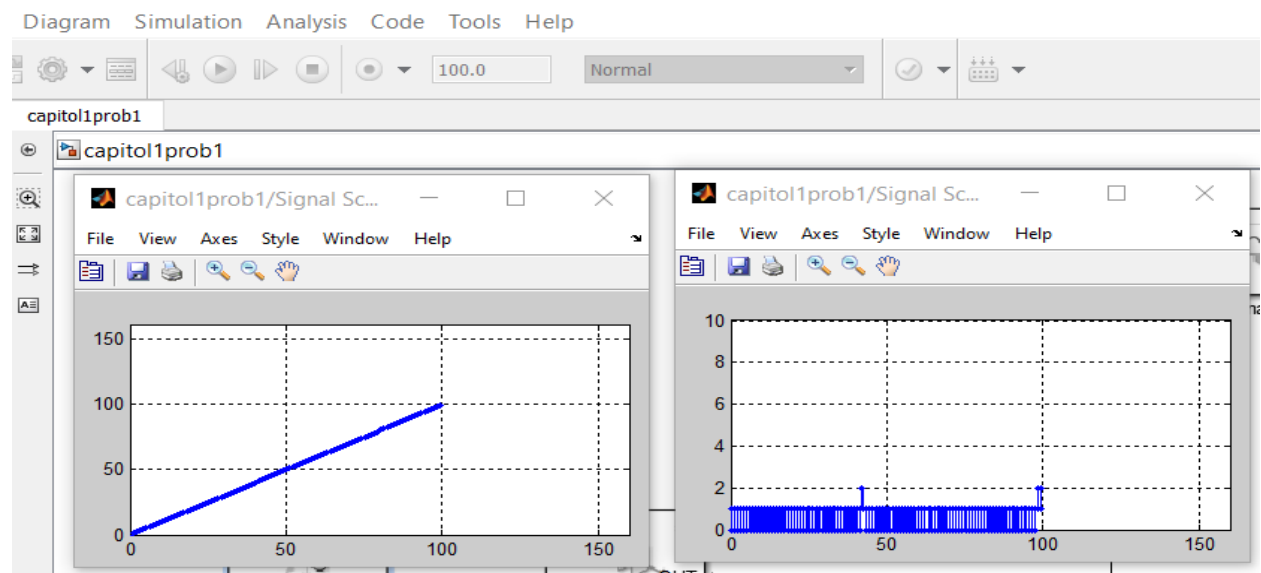
- c. Modificam intervalul dintre aparitia a doua entitati [0.8s, 1.2s] si timpul de deservire 1s



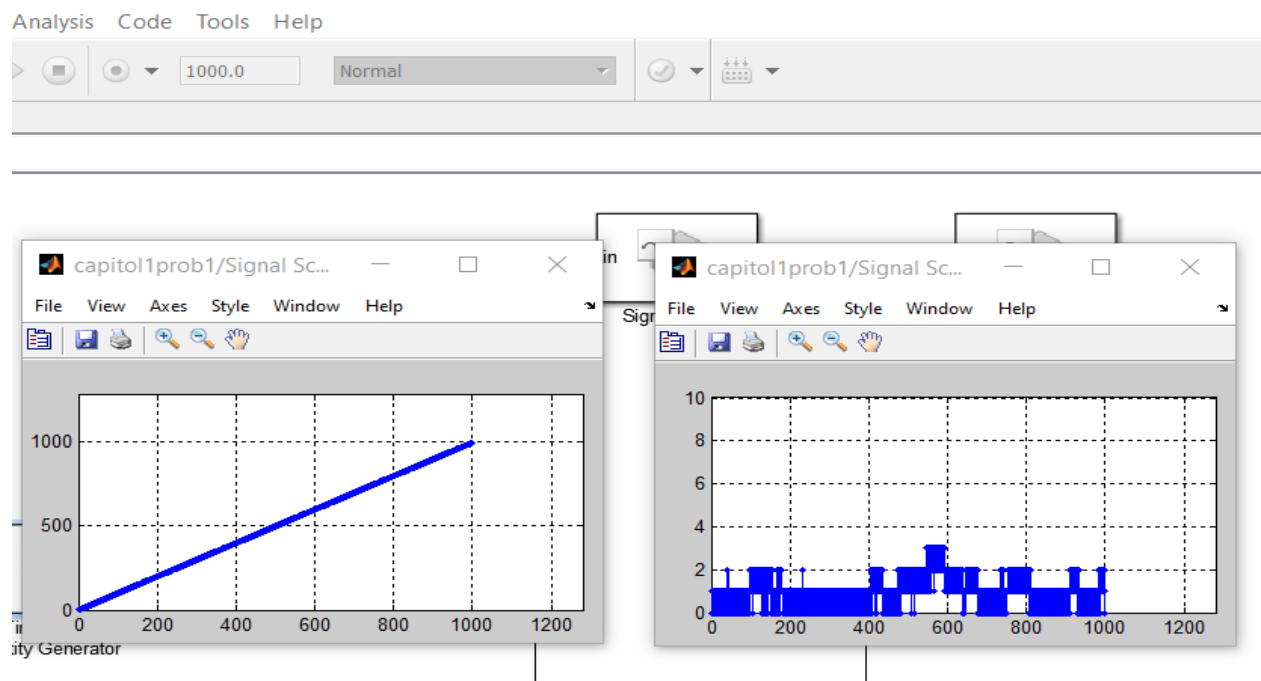
Simulam modelul pe 10s si obtinem



Simulam modelul pe 100s si obtinem

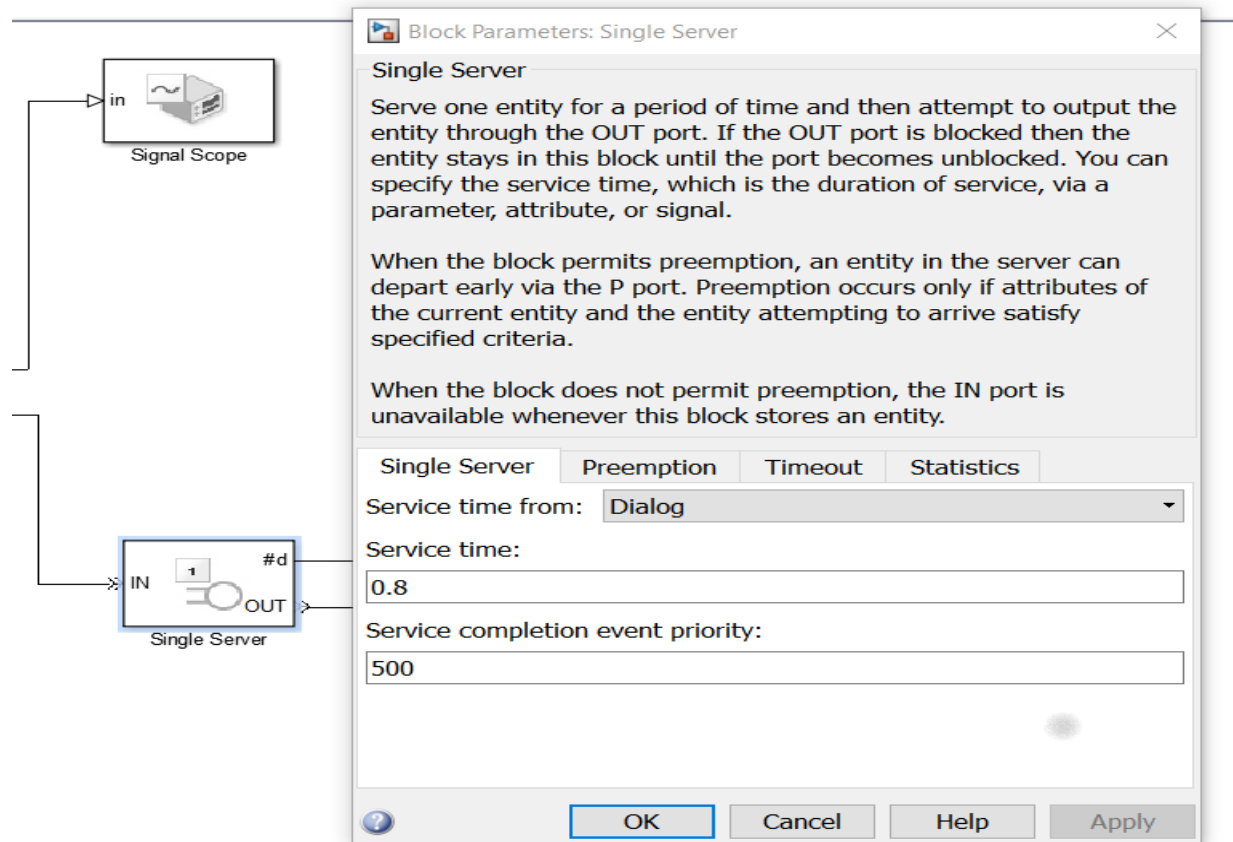


Simulam modelul pe 1000s si obtinem

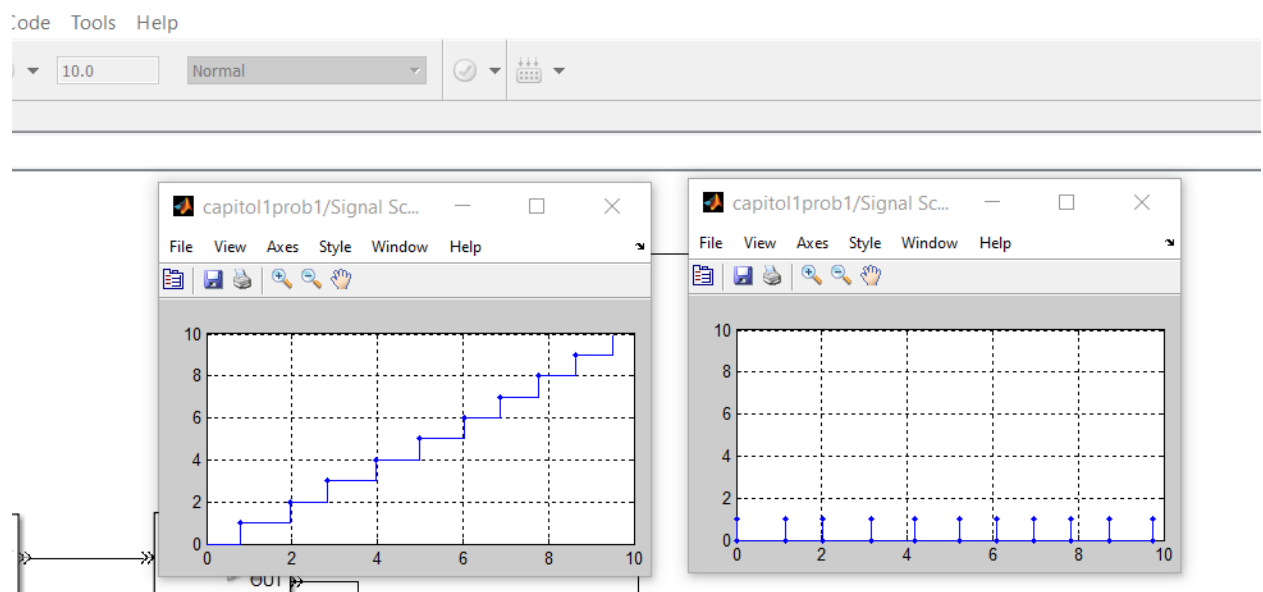


**Concluzie:** Generatorul avand o distributie intre  $[0.8s, 1.2s]$  si timpul de distributie de  $1.2s$  atunci poate aparea sau nu un timp de pauza/intarziere in care nu se mai pot transfera entiatile.

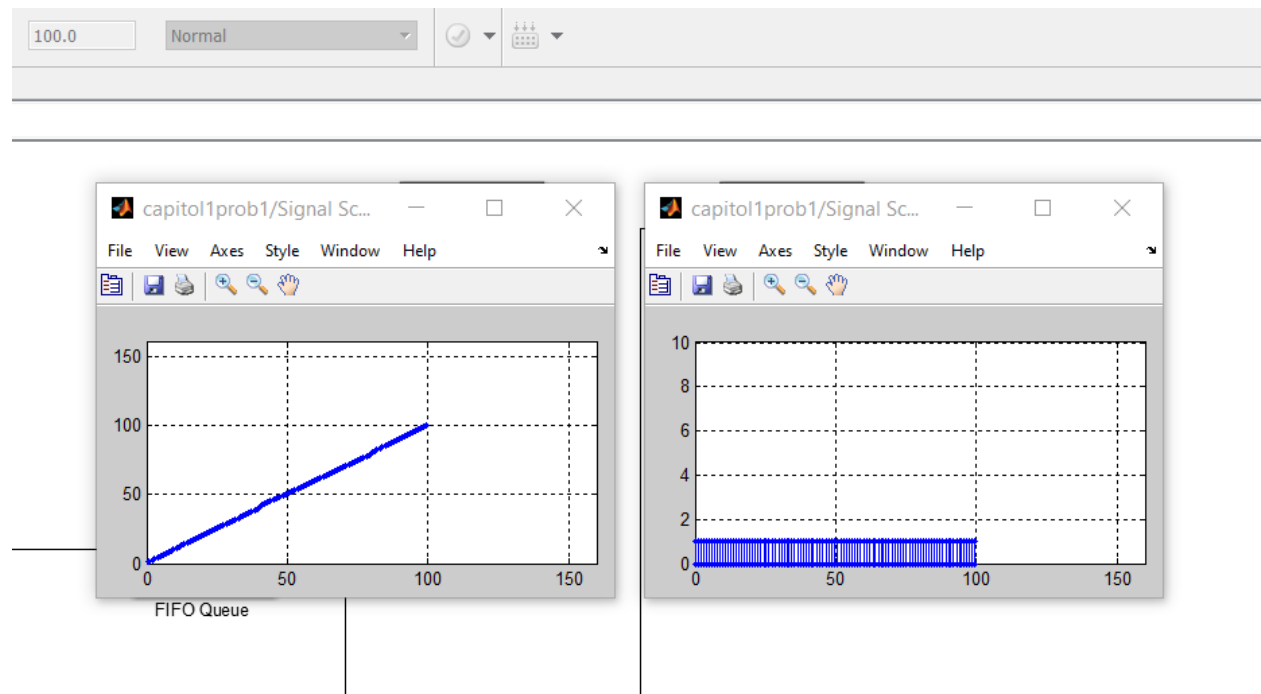
Modificam timpul de deservire la 0.8s, pastrand acelasi interval dintre aparitia a doua entitati [0.8s, 1.2s]



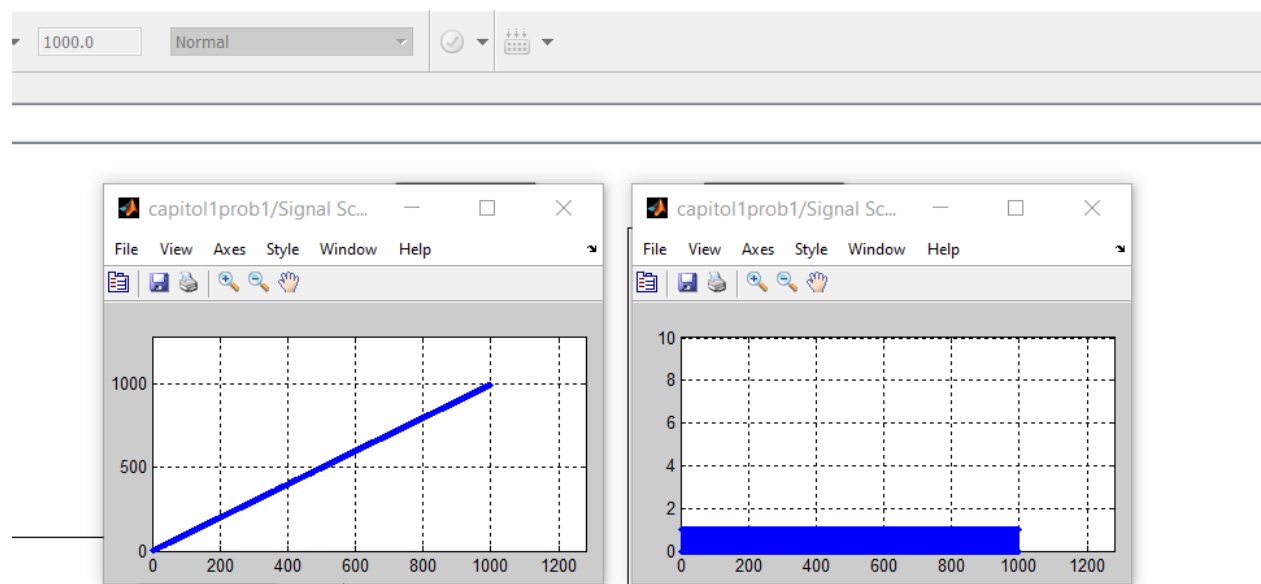
Simulam modelul pe 10s si obtinem



Simulam modelul pe 100s si obtinem

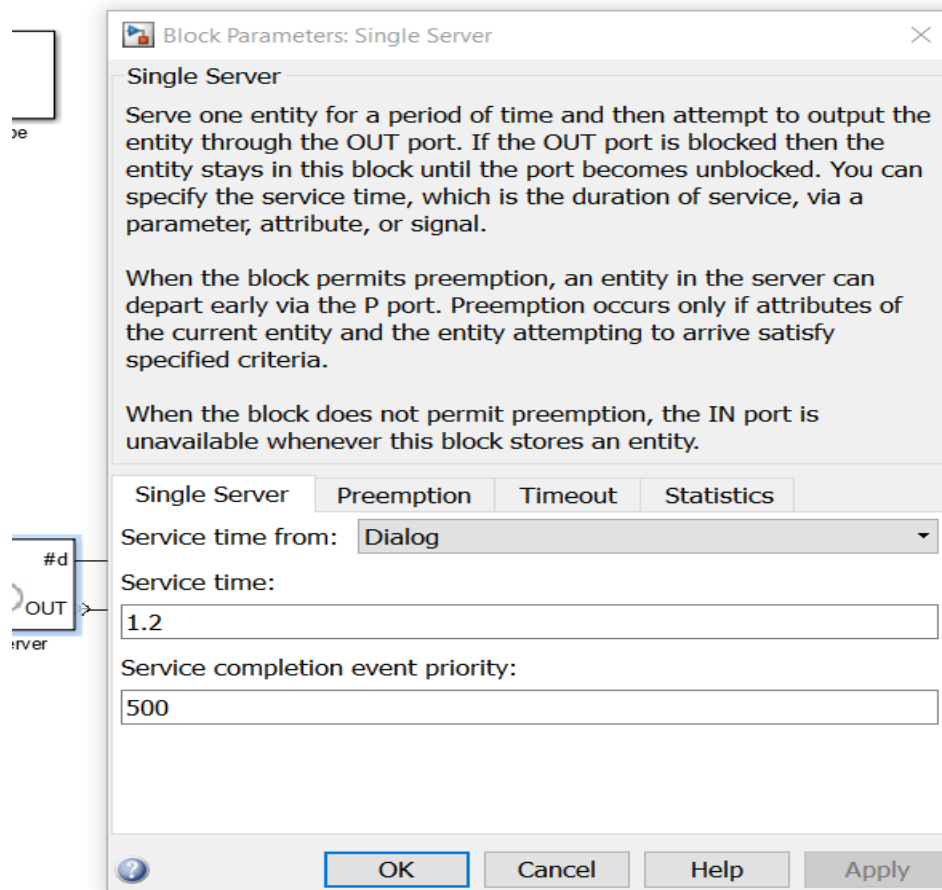


Simulam modelul pe 1000s si obtinem

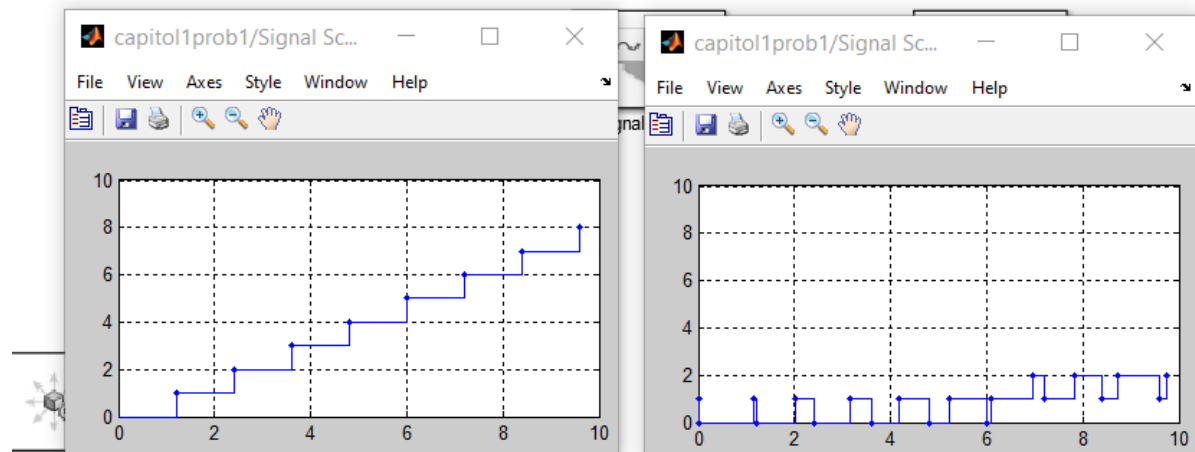




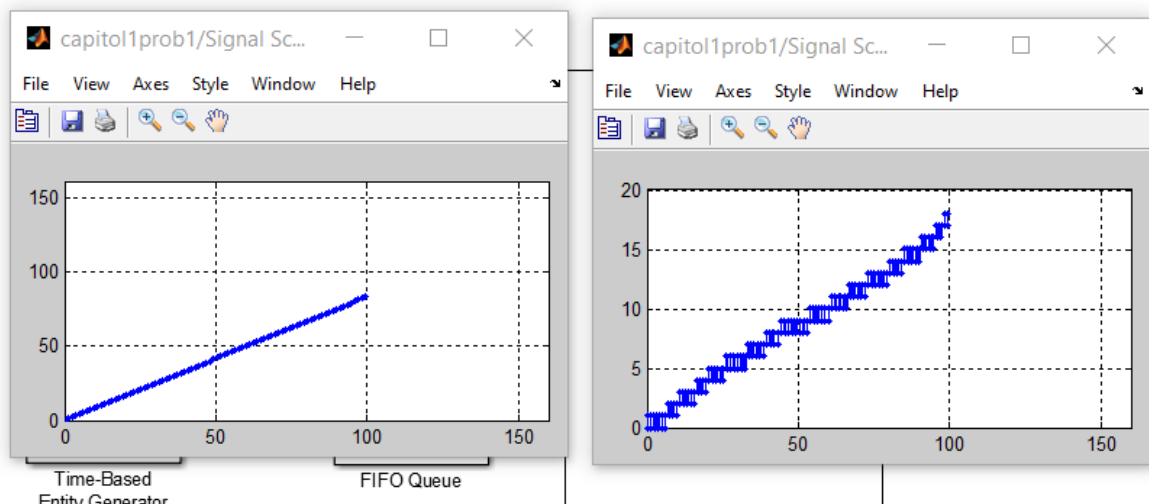
Modificam timpul de deservire la 1.2s, pastrand acelasi interval dintre aparitia a doua entitati [0.8s, 1.2s]



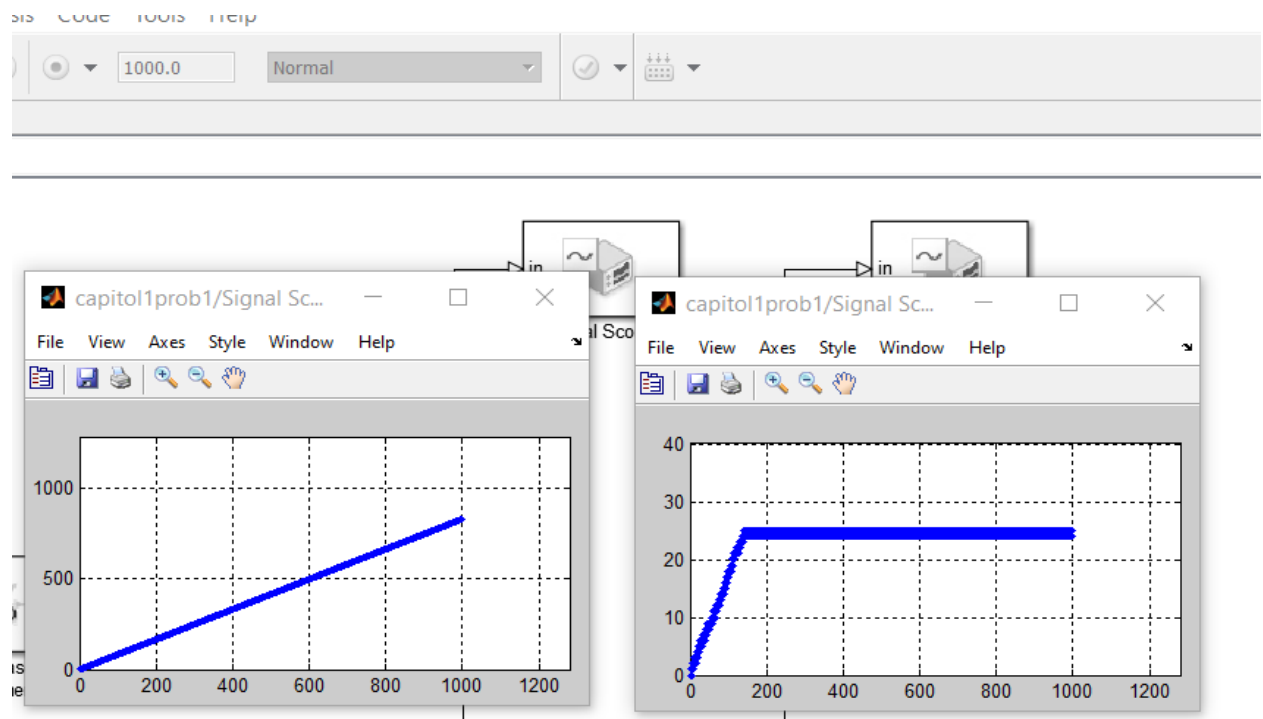
Simulam modelul pe 10s si obtinem



Simulam modelul pe 100s si obtinem

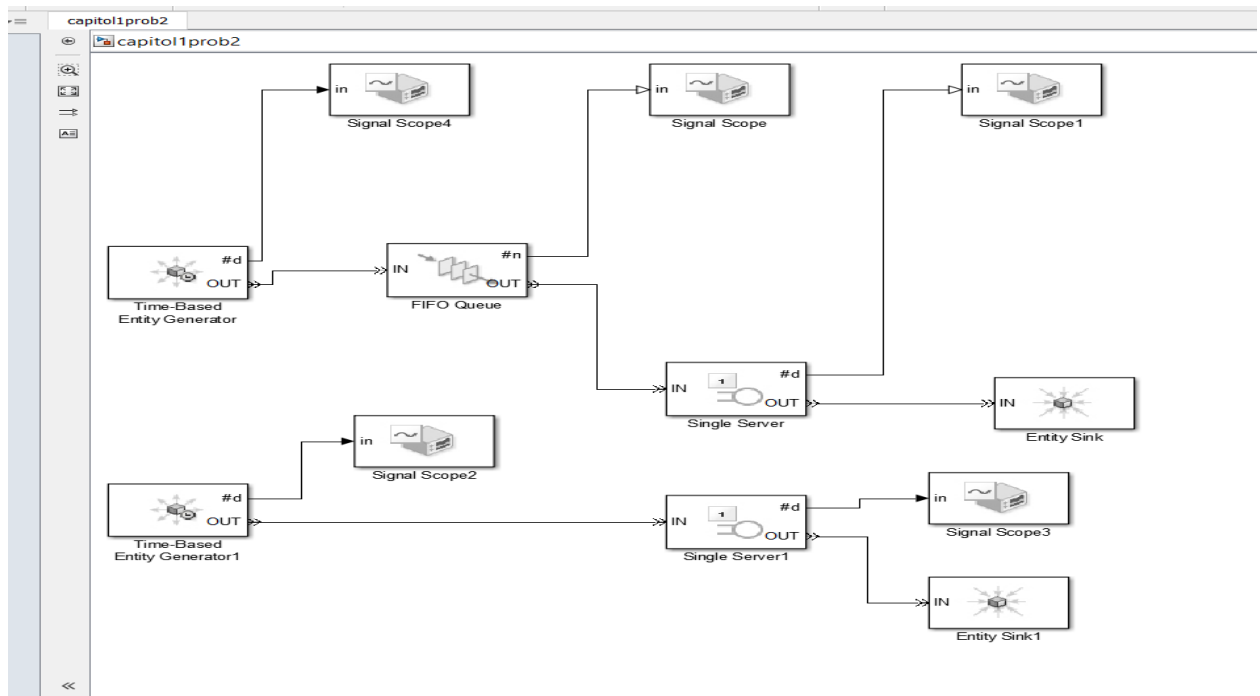


Simulam modelul pe 1000s si obtinem



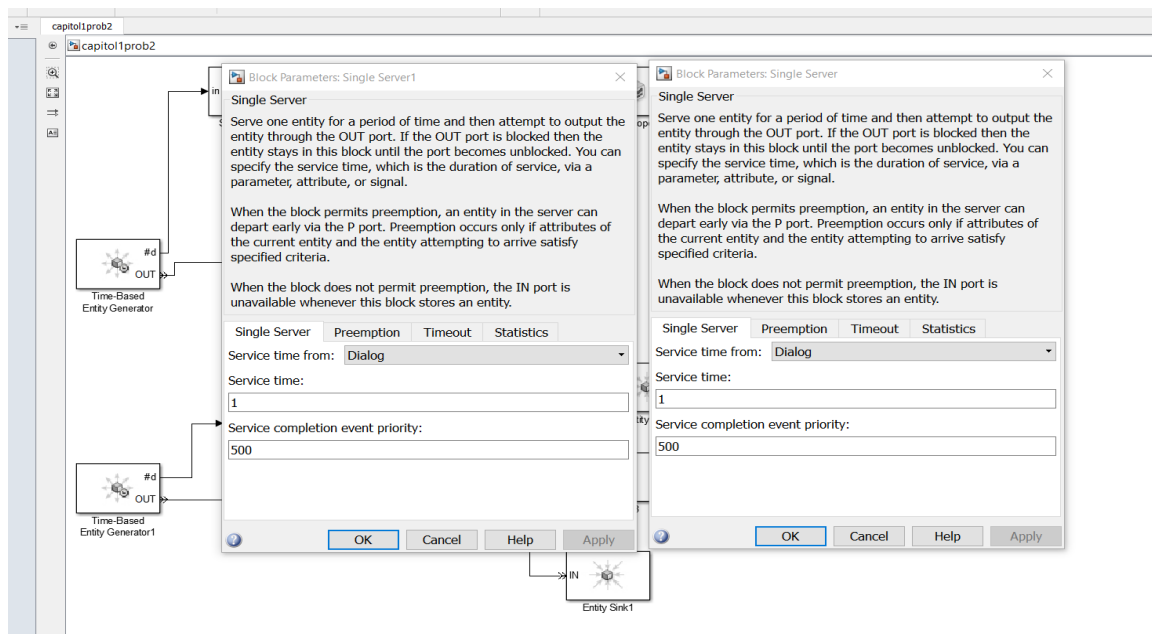
## Problema 2

S-a realizat un nou model respectand indrumarile din ghidul de laborator

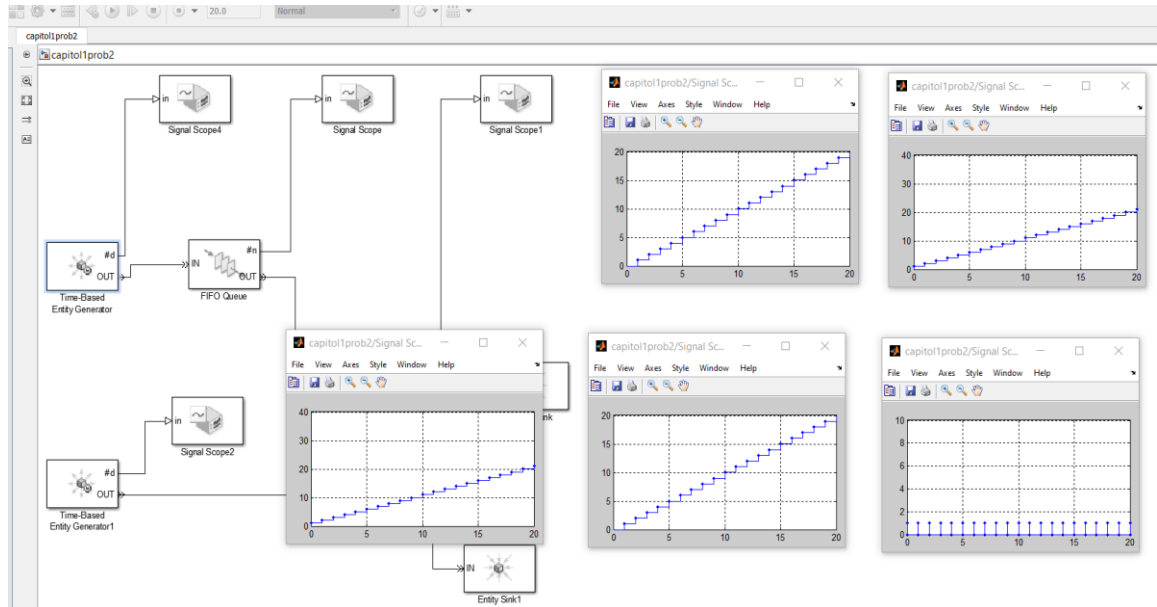


Intervalul intre aparitia a doua entitati este de 1s, iar modelele sunt rulate pe o durata de 20s

a. Pentru timpul de deservire 1s la ambele blocuri server

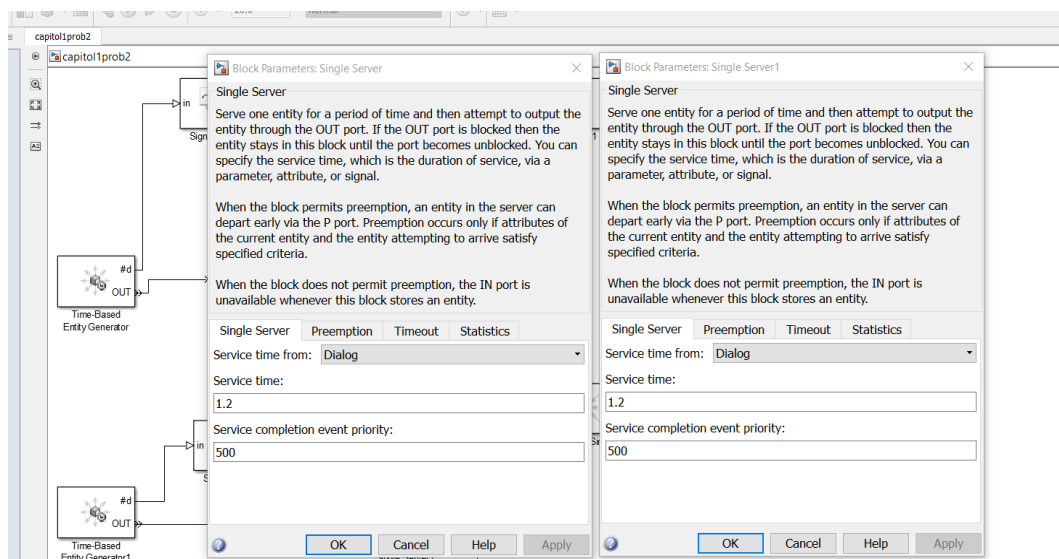


Am obtinut urmatoarele rezultate grafice

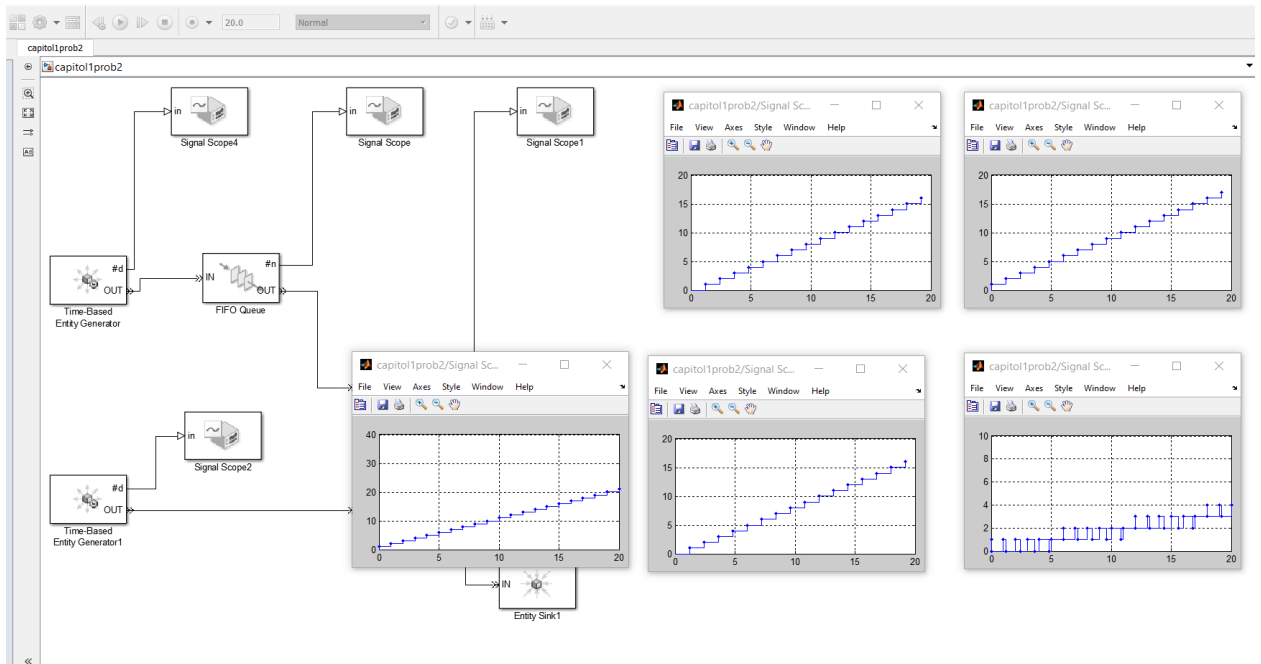


**Concluzie:** Comparand rezultatele dintre cele doua modele observam ca sunt asemanatoare deoarece intervalul dintre aparitia a doua entitati si timpul de deservire este de 1s, iar acestea nu prezinta intarzieri.

b. Pentru timpul de deservire 1.2s la ambele blocuri server



Am obtinut urmatoarele rezultate grafice



### Concluzie:

In acest caz timpul de deservire este mai mare decat intervalul de aparitie a doua entitati, existand un timp de pauza de 0.2s. Numarul de entitati generate de blocul Entity Generator este mai mare decat numarul de entitati generate de Entity Generator 1, deoarece in primul model entitatile sunt stocate in FIFO Queue, iar in al doilea model Entity Generator 1 dupa 1s genereaza o entitate dar trebuie sa astepte 1.2s de deservire pentru a transmite entitatea mai departe si pentru a genera o noua entitate.