

Capitol 3 / Lucrarea 3

Cozi de asteptare si statii de deservire

Problema 1

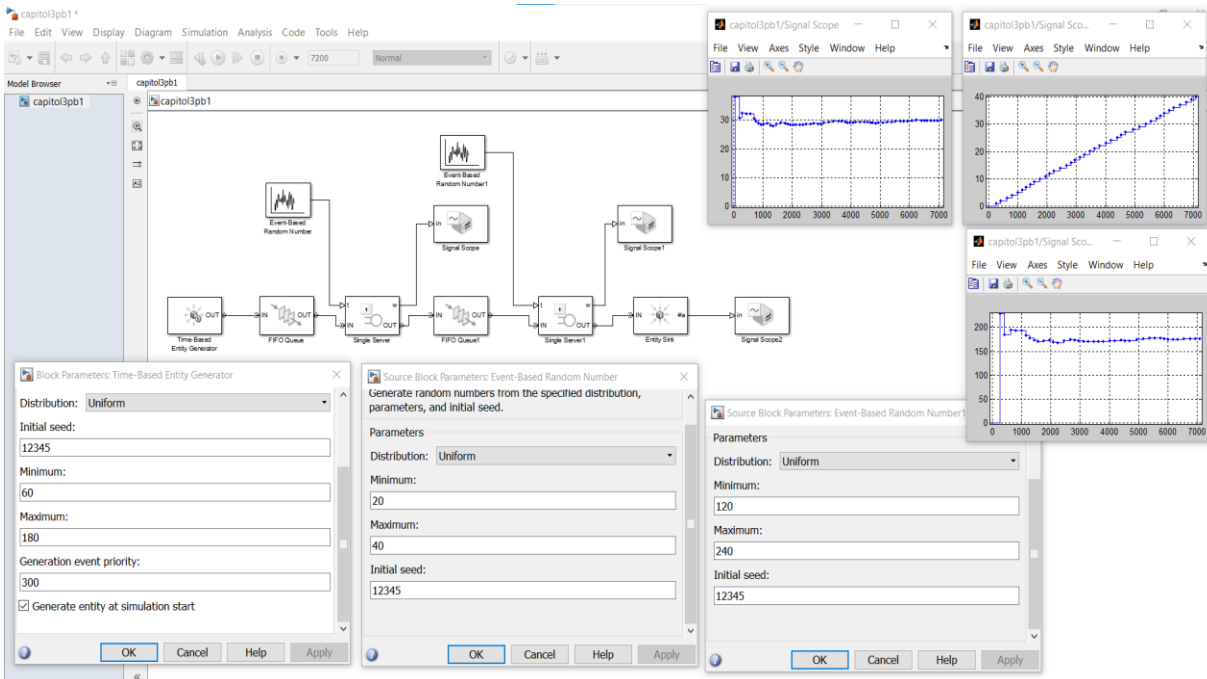
La sistemul de calcul sosesc programe la interval de $2+(-)1$ minute, rezulta ca intervalul de aparitie dintre doua entitati va avea o valoare minima de un minut adica 60 secunde, si o valoare maxima de 3 minute adica 180 secunde, avand setata o distributie uniforma. Timpul de distributie este primit ca input prin portul t, pentru aceasta folosim un Event-Based Random Number prin care setam distributia ca fiind uniforma, minimul pentru timpul de citire este de $30-10=20$ secunde, iar maximul este de 40 secunde.

Pentru Single Server 1 timpul de distributie este tot uniform, de aceea folosim din nou Event-Based Random Number unde timpul de executie este de minim 2 minute adica, 120 secunde, si maxim de 4 minute, adica 240 secunde. In concluzie, primul Single Server reprezinta citirea datelor de pe disc, iar cel de-al doilea procesarea acestora, fiecare statie de deservire avand cate o coada.

Scope, w=timpul de citire de pe disc.

Scope 1, w=timpul de deservire.

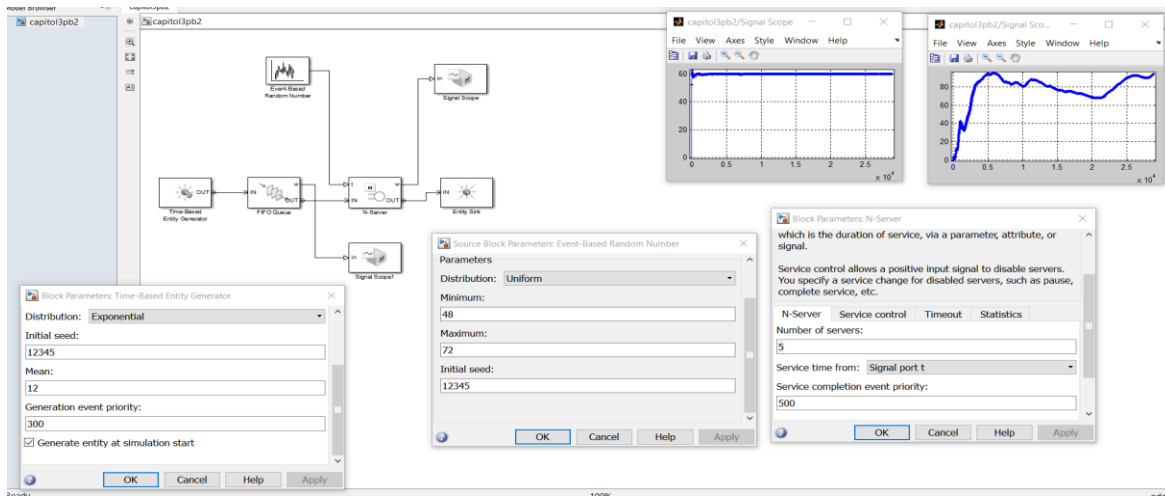
Simulare pe o durata de 2 ore, 7200 secunde.



Problema 2

Cunoastem ca cererile apar la intervale cu distributie exponentiala cu valoare medie de 0.2 minute, deci intervalul de aparitie dintre doua entitati este egal cu 0.2 minute adica 12 secunde, iar distributia este exponentiala. Folosim un N-Server deoarece se cere un multiserver cu capacitatea de cinci. Cum distributia este uniforma vom folosi un Event-Based Random Number pentru a seta timpul, durata convorbirii fiind de $1+(-)0.2$, minimul este de 0.8 minute, adica 48 secunde, iar maximul este de 1.2 minute, adica 72 secunde. **In concluzie**, timpul de prelucrare de asociata ca si semnal de la blocul Event-Based Random Number pentru a avea distributie uniforma.

Simularea pe o durata de 8 ore, adica 28.800 secunde.

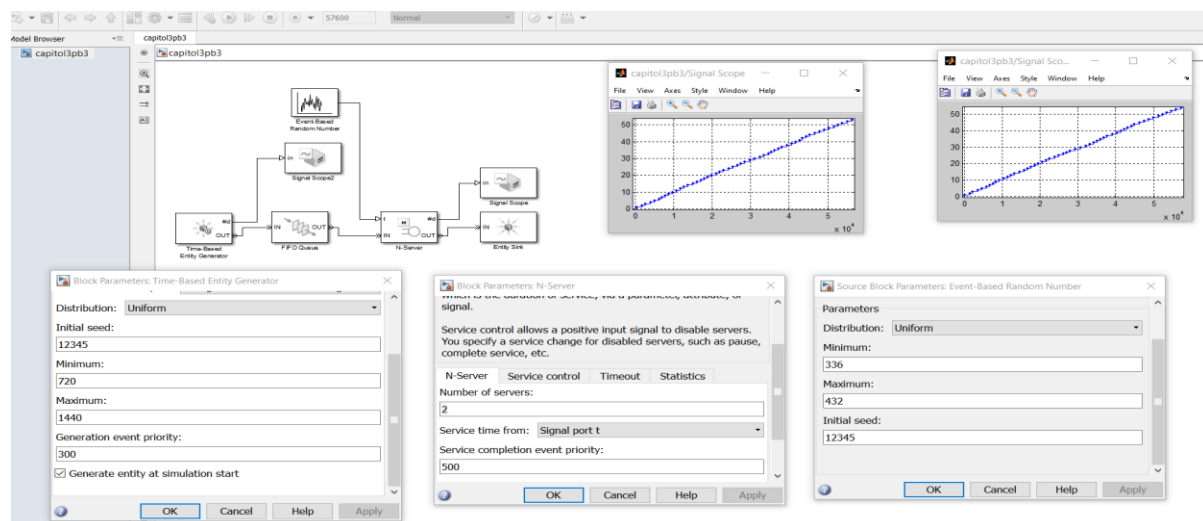


Problema 3

Cum loturile sosesc la interval de $18+(-)6$ minute cu distributie uniforma, rezulta ca intervalul de aparitie dintre doua entitati pentru blocul Time-Based Entity Generator va avea valoarea minima de 12 minute, adica 720 secunde, iar maximul de 24 minute, adica 1440 secunde. Cum loturile trebuie prelucrate la o statie cu capacitatea doi, vom utiliza un server N-Server cu o distributie uniforma care va fi introdusa prin portul t folosind un bloc Event-Based Random Number, unde prelucrarea dureaza $32+(-)4$ minute, valoarea minima fiind de 28 de minute, adica 336 secunde, iar valoarea maxima fiind de 36 minute, adica 432 secunde.

Pentru numarul de entitati generate utilizam un Signal Scope, in cazul de fata Signal Scope2, #d.
Pentru numarul de entitati deservite utilizam un Signal Scope, in cazul de fata Signal Scope, #d.

Simularea pe o durata de 960 minute, adica 57600s



Problema 4

Cum loturile de primul tip sosesc la interval de 35 ± 10 minute cu distributie uniforma, rezulta ca intervalul de aparitie dintre doua entitati pentru primul lot va avea valoarea minima de 25 minute adica 1500 secunde, si valoarea maxima de 45 minute, adica 2700 secunde in Time-Based Entity Generator.

Cum loturile de al doilea tip sosesc la interval de 45 ± 7 minute cu distributie uniforma, rezulta ca intervalul de aparitie dintre doua entitati pentru cel de al doilea lot va avea valoarea minima de 38 minute, adica 2280 secunde, iar valoarea maxima de 52 minute, adica 3120 secunde in Time-Based Entity Generator 1.

Ambele tipuri de loturi vor intra in aceeasi coada printr-un bloc Path Combiner cu 2 intrari pentru a fi transmise in FIFO Queue unde entitatile sunt introduce in coada in ordinea in care apar, si mai departe intr-un N-Server cu capacitatea doi pentru a fi prelucrate. Prelucrarea loturilor dureaza 18 ± 4 minute, si vom utiliza un bloc Event-Based Random Number unde vom seta valoarea minima pe 14 minute, adica 840 secunde si valoarea maxima pe 22 minute, adica 1320 secunde.

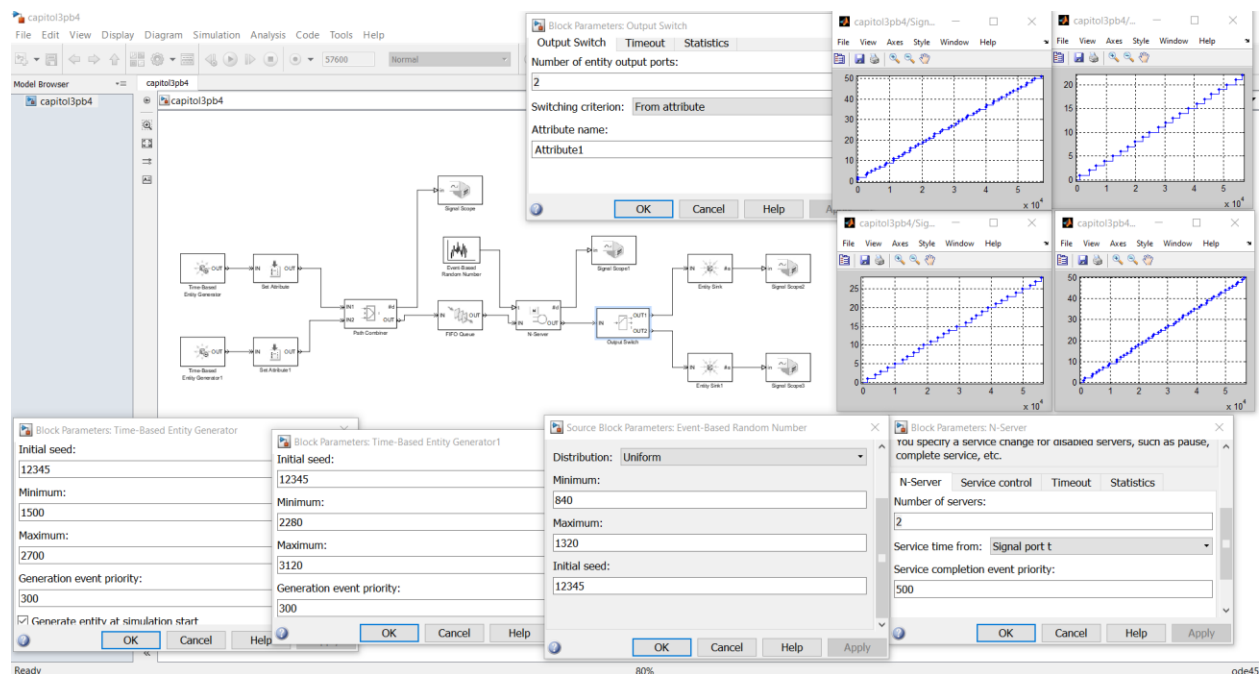
Pentru fiecare lot stabilim cate un atribut cu valori diferite, deoarece cum prelucrarea necesita acelasi timp pentru ambele loturi, statia de deservire va prelucra loturile pe baza unui atribut.

In final loturile prelucrate sunt preluate de un bloc Output Switch pentru a distribui loturile spre entitati diferite, unde putem sa vedem cate entitati au prelucrate din fiecare lot.

Signal Scope2 ne arata numarul de entitati deservite de tipul 1.

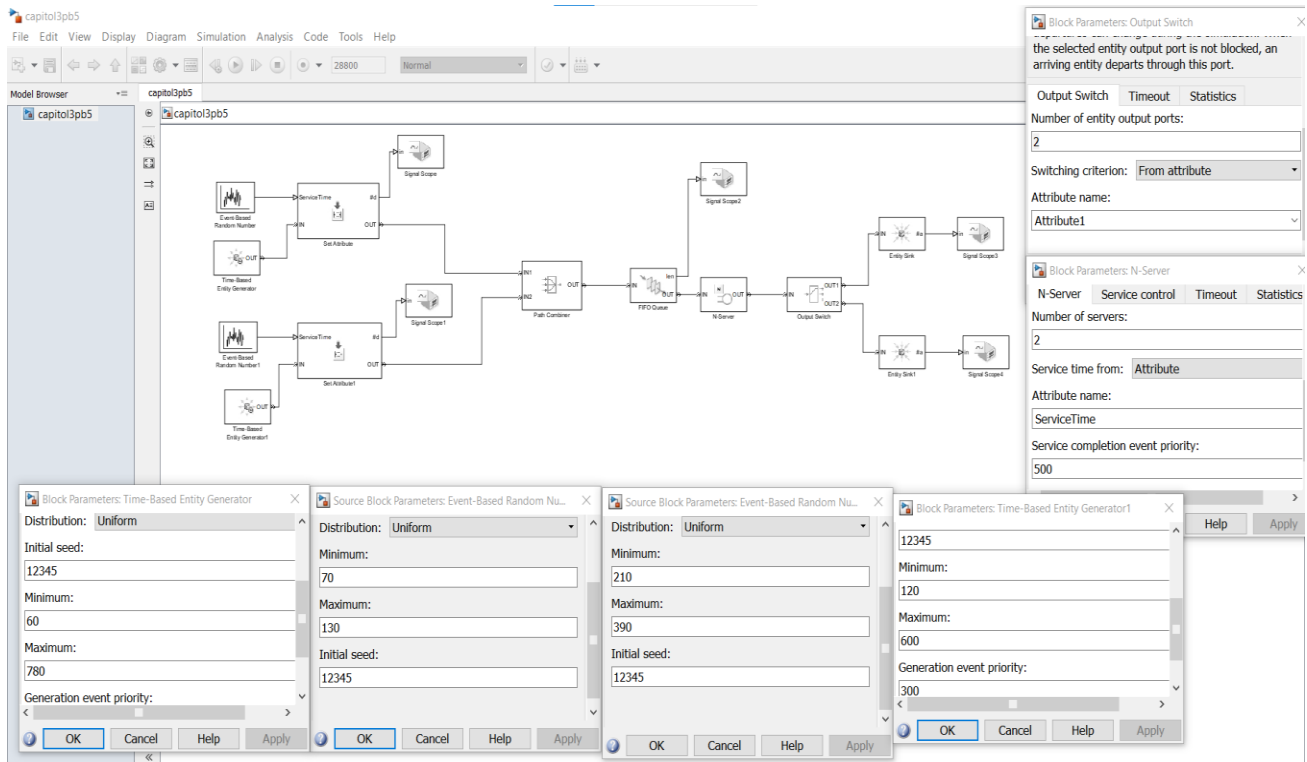
Signal Scope3 ne arata numarul de entitati deservite de tipul 2.

Simulare pe o durata de 960 minute, adica 57600s



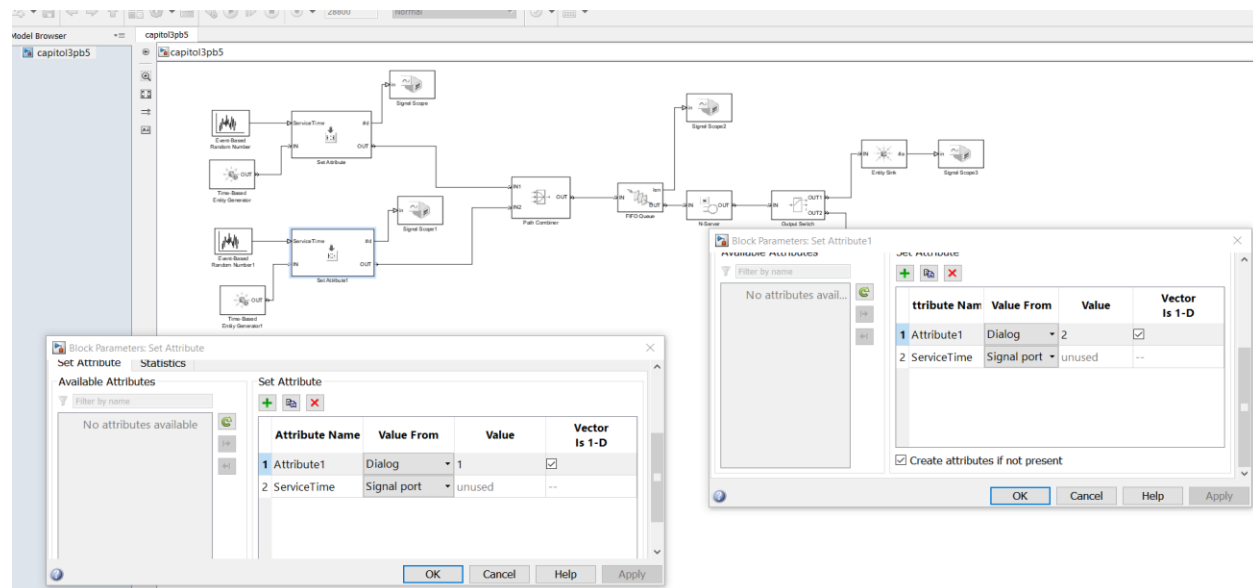
Problema 5

Cunoastem ca muncitorii de primul tip vin la interval de $420+(-)360$ secunde cu o distributie uniforma, adica rezulta ca intervalul de aparitie dintre doua entitati pentru primul tip va avea valoarea minima de 60 secunde si valoarea maxima de 780 secunde. Timpul de deservire pentru acest tip de muncitori va fi cum cuprinsa intre $300+(-)90$ secunde, adica o valoare minima de 210 secunde si o valoare maxima de 390 secunde. Muncitorii de tipul doi vin la interval de $360+(-)240$, adica intervalul dintre aparitia a doua entitati va avea valoarea minima de 120 secunde si valoarea maxima de 600 secunde. Timpul de deservire pentru al doilea tip de muncitori este cuprins in intervalul $100+(-)30$ secunde, adica o valoare minima de 70 secunde si o valoare maxima de 130 secunde.

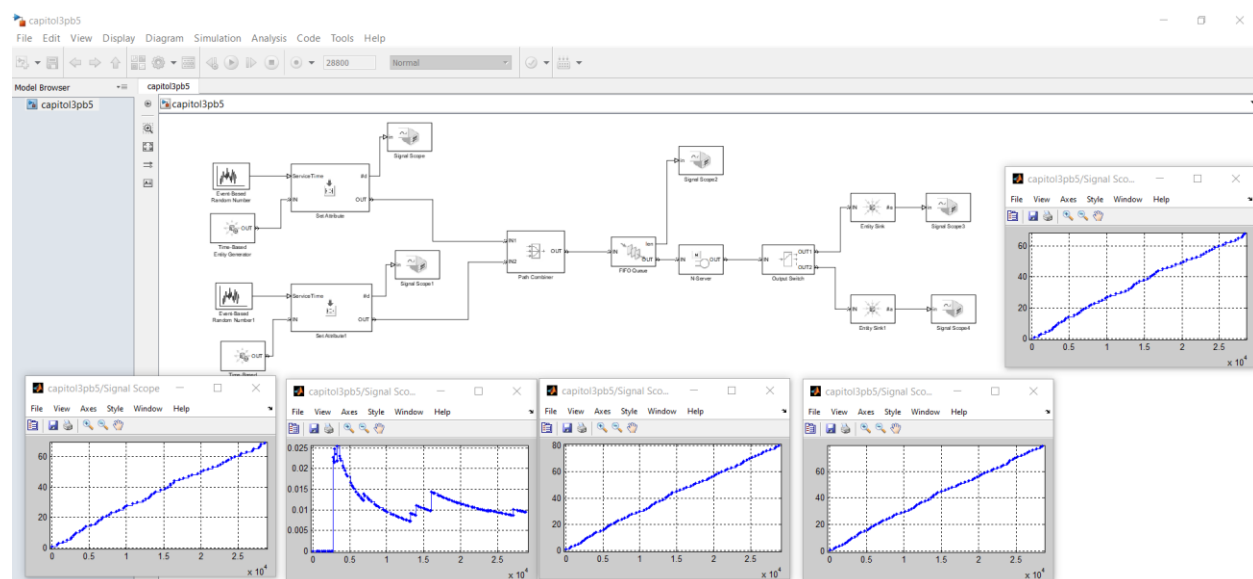


Pentru fiecare tip de muncitori exista un timp de deservire diferit, pentru asta setam cate un Service Time pentru fiecare tip folosind cate un bloc Event-Based Random Number cu distributie uniforma, acesta va fi transmis ca input in blocul Set Attribute. In acelasi timp setam si atributul cu cate o valoare pentru fiecare tip care mai tarziu va fi folosit pentru blocul Output Switch. Acest bloc dirijeaza entitatile dupa prioritate catre blocurile Entity Sink, separandu-le, criteriul de selectare a iesirii este valoarea atributului Attribute1.

Folosim un bloc N-Server cu capacitatea doi pentru a putea deservii ambele tipuri de muncitori, iar pentru blocul Output Switch setam criteriul pentru atribut, pentru a separa entitatile dupa atribut catre blocurile Entity Sink.



Simularea pe o durata de 28800 secunde



Problema 6

Diferenta intre problema 5 si 6 este data de reprezentarea blocului cozii, in cazul de fata pentru problema 6 am utilizat un bloc Priority Queue care sorteaza entitatile in ordine crescatoare sau descrescatoare dupa valorile unui atribut, prioritatea fiind un numar natural. Am setat timpii de deservire cu ajutorul a doua blocuri Event-Based Random Number ca si attribute. Entitatile sunt stocate in aceeasi coada si deservite cu prioritaet pentru muncitorii de tipul 1, iar prelucrarea se face pe baza atributului ServiceTime, iar mai apoi entiatile sunt separate dupa atributul de prioritate PriorityAttributeName pentru a vedea usor cate entitati din fiecare tip au ajuns la capatul simularii.

Draghici Andreea-Maria

CR 3.1B

Simularea pe o durata de 28800 secunde

