

Laboratory 8 MS

Problem 1:

1. Să se modeleze un sistem ce are două resurse partajate, una cu capacitate patru și alta cu capacitate opt. Un bloc Time-Based Entity Generator generează entități ce vor utiliza aceste resurse, cu intervalul între două entități cu distribuție uniformă cu valorile cuprinse între 0.01s și 0.2s. Entitățile generate vor avea un parametru Type, cu valorile unu și doi cu probabilitățile 0.4 și 0.6. Entitățile cu parametrul Type egal cu unu vor partaja resursa cu capacitate patru. Entitățile cu parametrul Type egal cu doi vor partaja resursa cu capacitate opt. Atributul Type va primi valori într-un bloc Set Attribute ce va avea o intrare Signal port t. Valorile atributului se vor genera cu un bloc Event-Based Random Number cu distribuția Arbitrary discrete cu parametri: Value vector [1 2] și Probability vector [0.4 0.6].

Se va simula sistemul pe 10s. Pentru fiecare resursă se va înregistra numărul de resurse disponibile. Modelul se va simula pentru timpii de servire ai celor două tipuri de entităților din tabelul de mai jos.

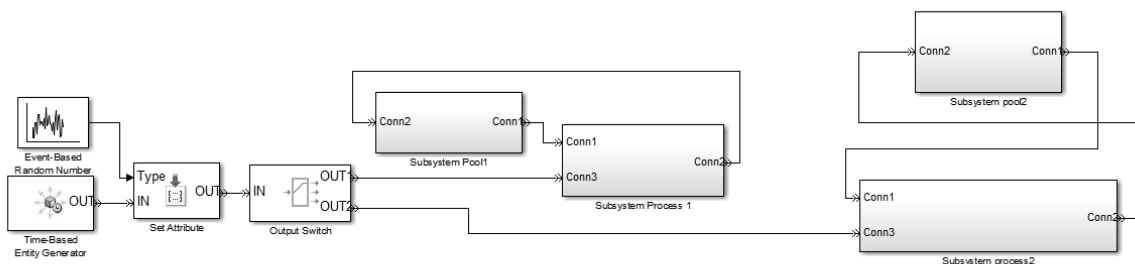
Type = 1	Type = 2
0.5s	0.7s
1s	1.5s
2s	3s

Se vor compara resursele disponibile pentru fiecare caz.

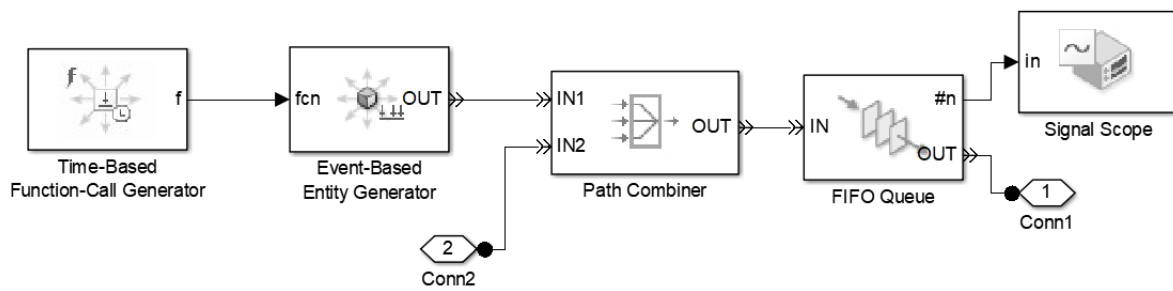
Indicație. Modelul va utiliza pentru fiecare resursă cele două subsisteme din exemplul anterior, Pool și Process, ce conțin:

- Subsistemul Pool - blocurile Time-Based Function-Call Generator, Event-Based Entity Generator, Path Combiner și FIFO Queue, vezi Figura 8.
- Subsistemul Process - blocurile FIFO Queue, Entity Combiner, Infinite Server și Entity Splitter, vezi Figura 7.

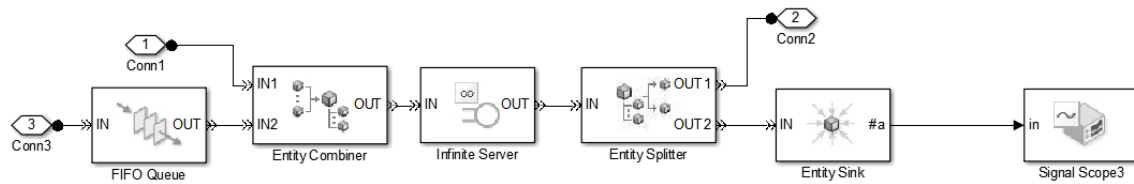
We designed the following schema:



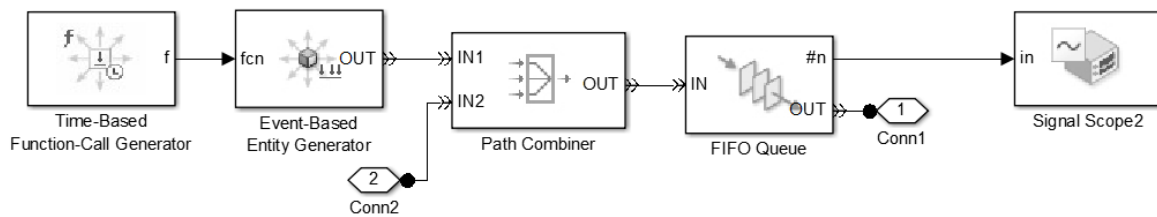
Subsistem pool 1:



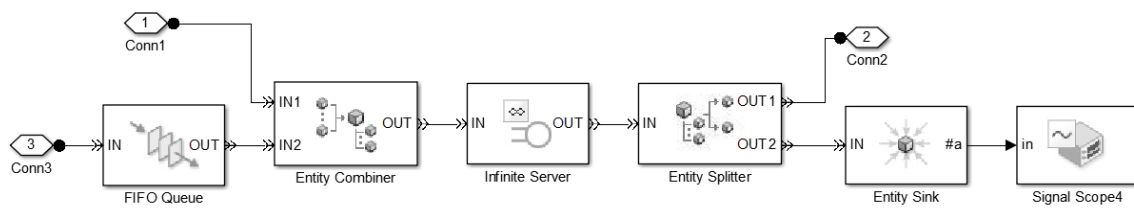
Subsystem process 1:




Subsystem pool 2



Subsystem process 2:



Entity generator:

 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: **Uniform**

Initial seed:
12345


Minimum:
0.01

Maximum:
0.2

Generation event priority:
300

☒ Generate entity at simulation start

Random number generator for type:

 Source Block Parameters: Event-Based Random Number ×

Event-Based Random Number

Generate random numbers from the specified distribution, parameters, and initial seed.

Parameters


Distribution: **Arbitrary discrete**

Value vector:
[1 2]

Probability vector:
[0.4 0.6]

Initial seed:
12346

In output switch we select by type:

 Block Parameters: Output Switch ×

Output Switch

Select an entity output port for departure. The Switching criterion parameter indicates how the block determines which entity output port is selected for departure at any given time.

The block receives entities and outputs them through one of the entity output ports. The port selected for departures can change during the simulation. When the selected entity output port is not blocked, an arriving entity departs through this port.

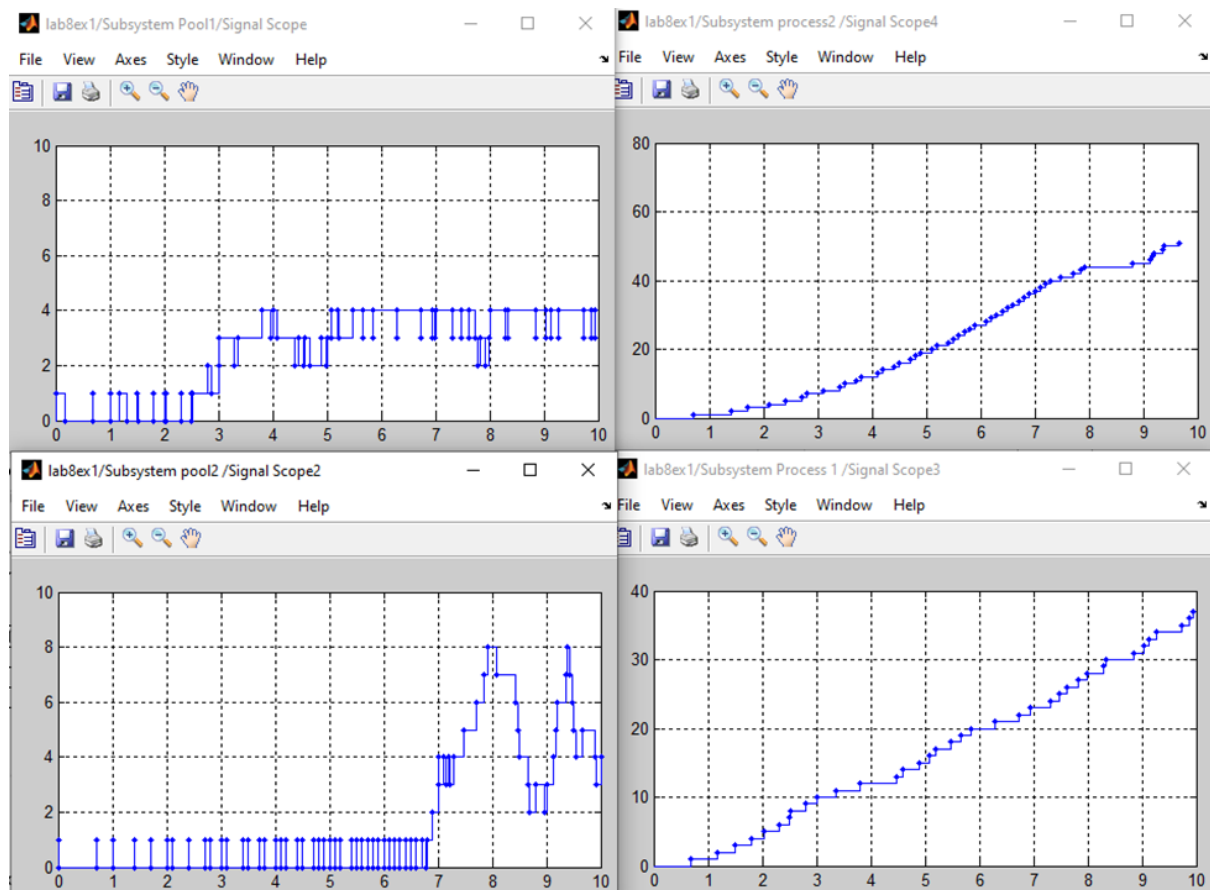
Output Switch Timeout Statistics

Number of entity output ports:
2

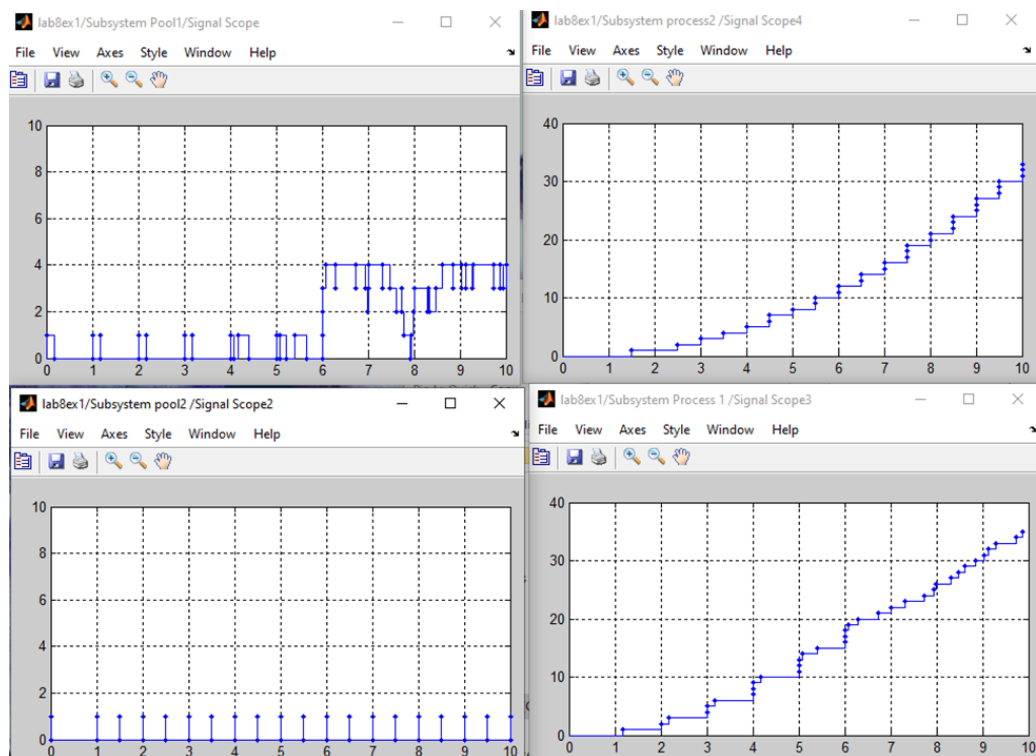
Switching criterion: **From attribute**

Attribute name:
Type

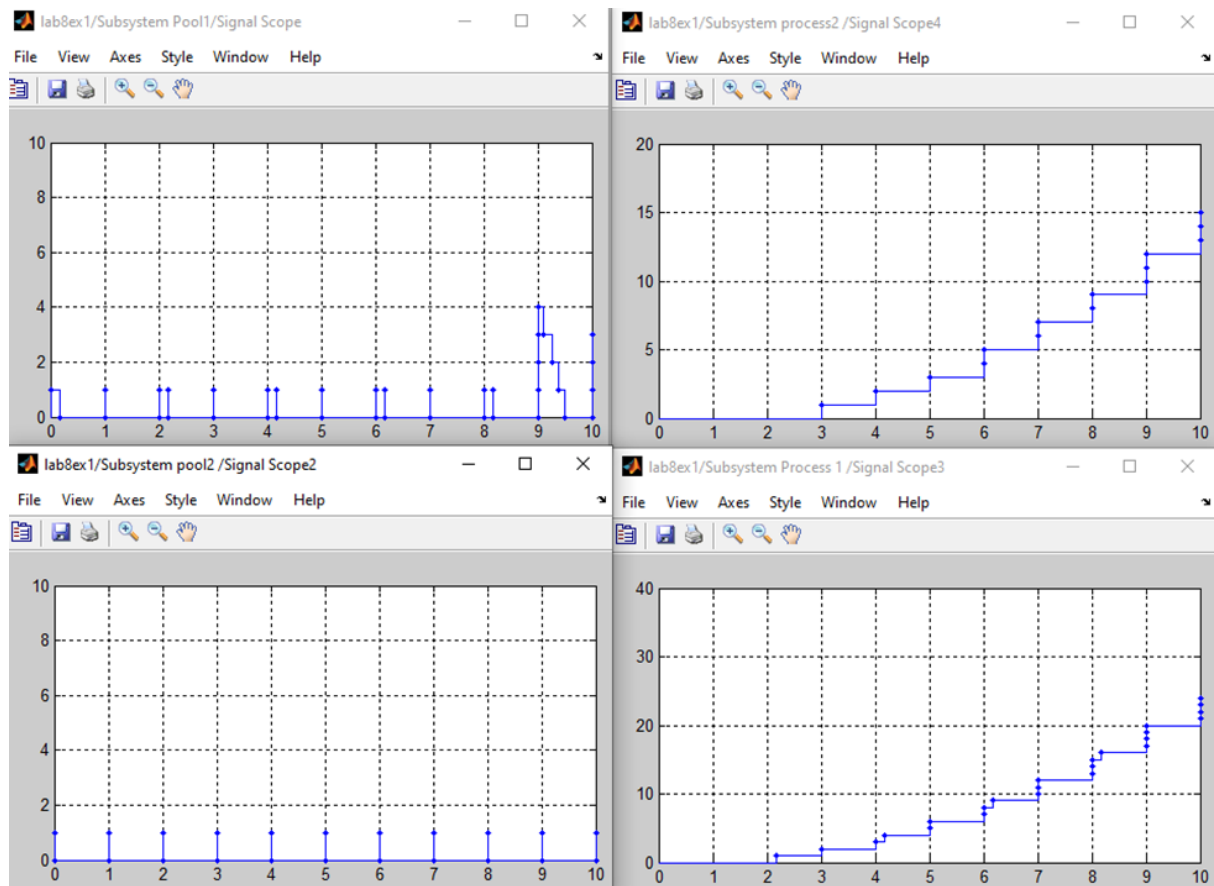
type=1 ,0.5 type=2 0,7



1s & 1.5s:



2s & 3s:



Problem 2:

2. Modelarea unui router. Un router conectează mai multe surse de pachete de date cu mai multe destinații. Router-ul examinează adresa IP destinație a pachetului și îl trimite pe ieșirea corespunzătoare. În problemă router-ul va avea trei surse de date și trei destinații. Fiecare pachet va avea ca atribute adresa IP destinație și lungimea datelor:
- Destination – adresa IP destinație. Pentru simplificare, adresele IP au valorile 1, 2, 3
 - Length - lungimea pachetului, va fi un număr întreg cuprins între șase și zece cu distribuție uniformă.

Indicație. Sursele de date vor fi subsisteme Source din Figura 10.

În cazul blocului Length parametrii sunt: limitele minimă și maximă șase și respectiv zece și numărul de valori cinci.

Unitatea de măsură a timpului va fi milisecunda. Blocul Time-Based Entity Generator generează entități cu distribuție exponențială și valoare medie 8ms.

Routerul va conține un bloc Path Combiner ce combină pachetele într-o singură coadă, un bloc Single Server ce va prelucra pachetele cu timpul de servire 1ms și un bloc Output Switch ce va dirija pachetele către cele trei ieșiri conform atributului Destinație.

Se va simula modelul pe 1000 ms. Se vor afișa adresa sursă a pachetului folosind statistica **last** a blocului Path Combiner și adresa destinație a pachetului folosind statistica **last** a blocului Output Switch. În cazul blocului Path Combiner **last** este numărul portului de intrare pe care a apărut ultima entitate. În cazul blocului Output Switch **last** este numărul portului de ieșire pe care a plecat ultima entitate.

We have the following schema:

Source Block Parameters: Destination

Event-Based Random Number

Generate random numbers from the specified distribution, parameters, and initial seed.

Parameters

Distribution: Discrete uniform

Minimum: 1

Maximum: 3

Number of values: 3

Initial seed: 12346

Path combiner:

Block Parameters: Path Combiner

Path Combiner

Merge paths, that is, accept entities from any input port and output them through a single entity output port.

The Input port precedence parameter indicates how the block determines which entity input port to make available first, whenever the entity output port status changes from blocked to unblocked.

Path Combiner Status Notification Statistics

Number of entity input ports: 3

Entity generator:

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: Exponential

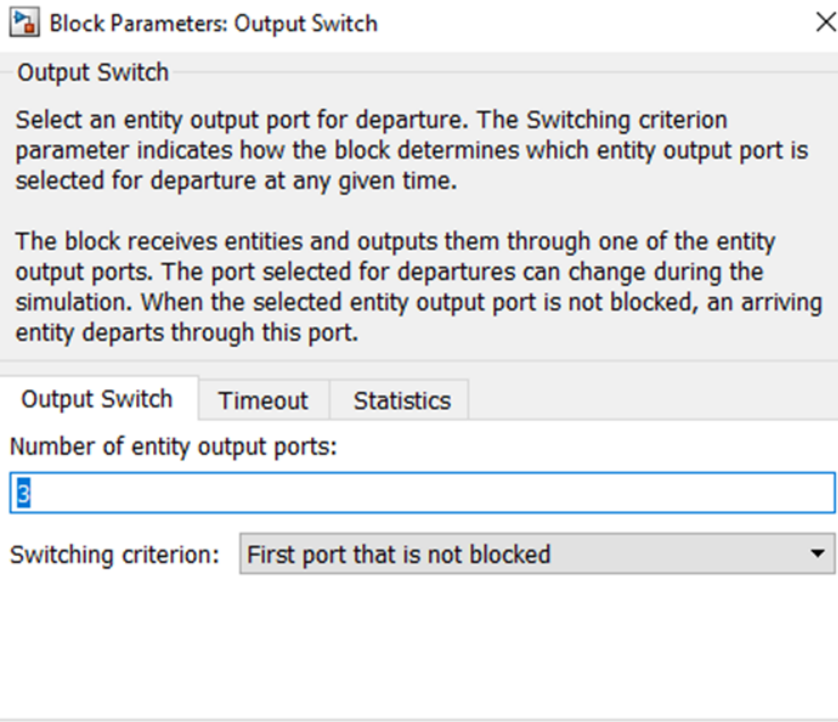
Initial seed: 12333

Mean: 8

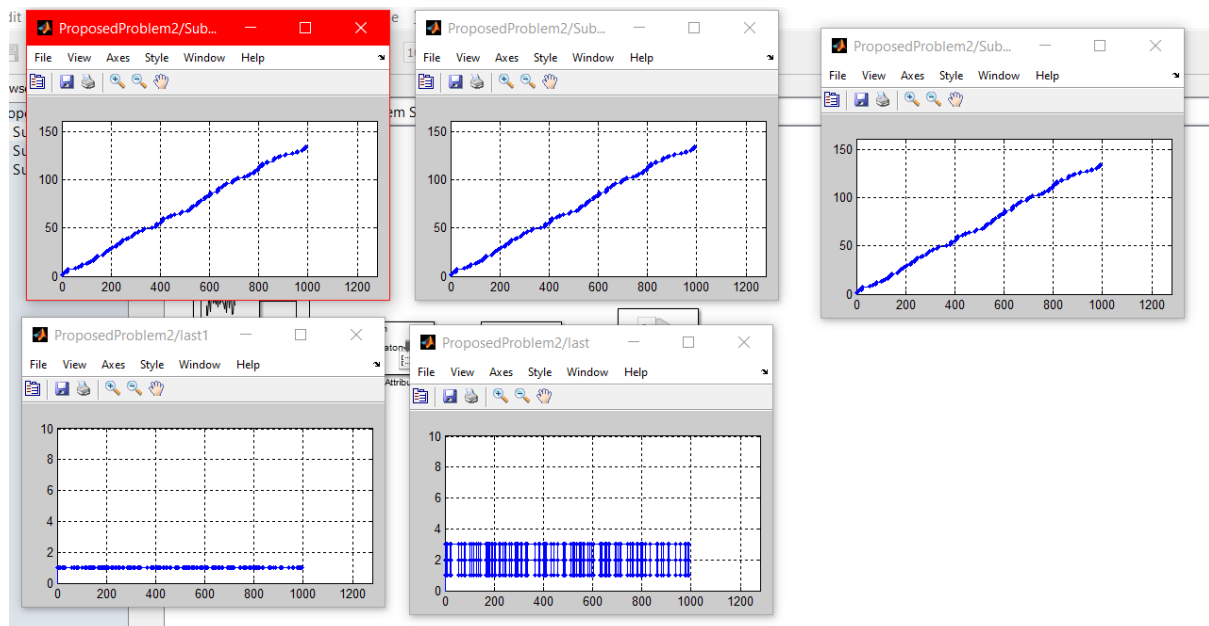
Generation event priority: 300

☒ Generate entity at simulation start

Output switch:



We got the following result:



Problem 3:

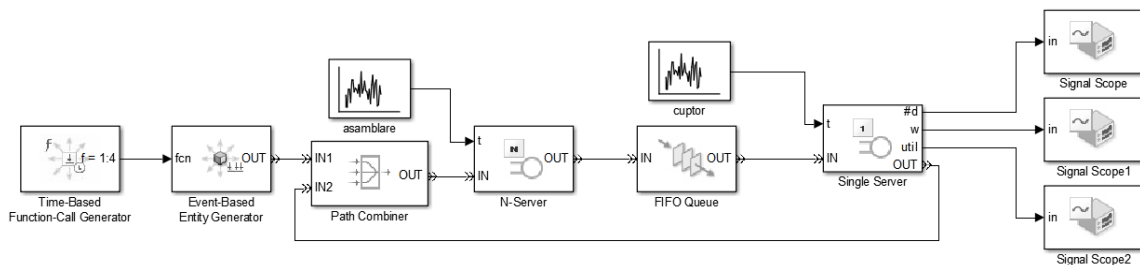
3. Sistem de fabricare a sculelor. In prima etapă un muncitor assemblează piesele ce constituie o sculă. Etapa durează $30 \pm 5'$ cu distribuție uniformă. In a doua etapă același muncitor utilizează un cuptor pentru tratarea termică a sculei. Etapa durează $8 \pm 2'$ cu distribuție uniformă. După aceasta scula este terminată și muncitorul începe procesul de fabricare a unei noi scule. Există patru muncitori și un singur cuptor. Muncitorii utilizează acest cuptor cu ordinea de deservire primul venit, primul servit. Se va simula modelul pe 40 h. Se vor înregistra timpul mediu de utilizare a cuptorului și gradul de utilizare, lungimea cozii și timpul mediu de așteptare în coadă.

Indicație. Etapa de asamblare va fi modelată cu un multiserver cu capacitatea patru. Cuptorul este modelat de un bloc Single Server. Modelul va consta din:


- Un lanț de blocuri Time-Based Function-Call Generator și Event-Based Entity Generator ca în Figura 3, ce vor genera patru entități la începutul simulării
- Un bloc Path Combiner cu două intrări la care sunt conectate: ieșirea blocului Event-Based Entity Generator și ieșirea blocului Single Server
- Un bloc N-Server cu capacitatea patru ce modelează etapa de asamblare. Blocul este conectat la ieșirea blocului Path Combiner
- Un bloc Single Server ce modelează cuptorul.

Pentru înregistrarea statisticilor blocul Single Server vor fi precedate de un bloc FIFO Queue. Modelul este cel din Figura 12.

The schema looks as follows:



Random number for assembly:

 Source Block Parameters: asamblea ×

Event-Based Random Number

Generate random numbers from the specified distribution, parameters, and initial seed.

Parameters


Distribution: Uniform

Minimum:

Maximum:

Initial seed:

Random number for the oven:

 Source Block Parameters: cuptor ×

Event-Based Random Number

Generate random numbers from the specified distribution, parameters, and initial seed.

Parameters

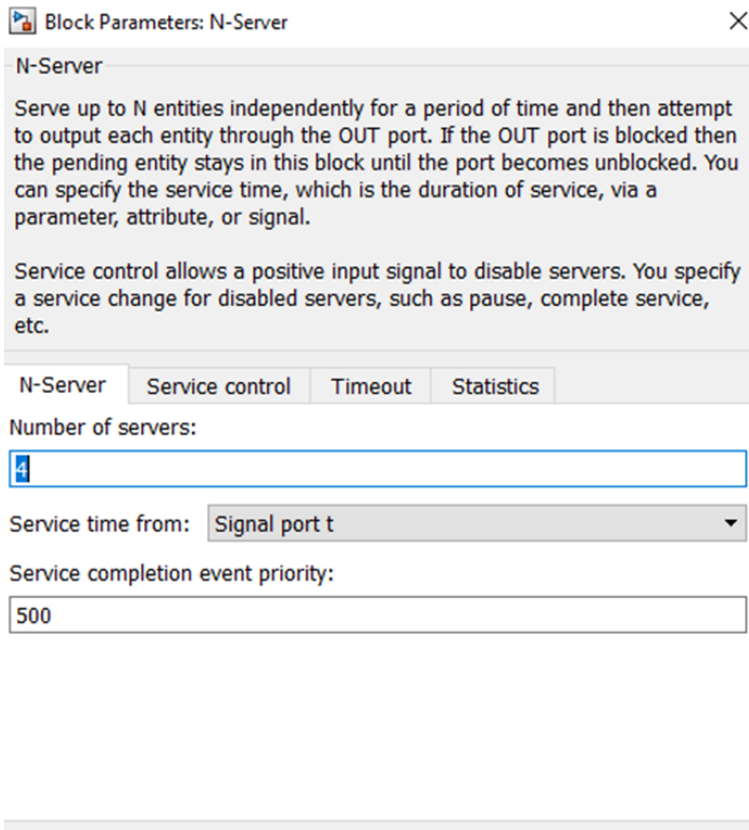
Distribution: Uniform

Minimum:

Maximum:

Initial seed:

The 4 workers that use the oven:



We got the following result:

