EEP-703 Computer Network Lab

Assignment8-Modeling different scheduling algorithms using Petri Nets

Harshit Kumar Gupta
2013EET2369

Computer Technology
Department Of Electrical Engineering
IIT DELHI

April 16, 2014

Contents

Chapter 1

PROBLEM STATEMENT

roblem Statement C1 (Compulsory; Difficulty level *; 100 points) Simulate the given two scenarios in sharpegui, also analyse and compare the performances:

1. *Scenario 1: Waiter takes order from customer 1; serves customer 1; takes order from customer 2; serves customer 2.

2. * Scenario 2: Waiter takes order from customer 1; takes order from customer 2; serves customer 2; serves customer 1. Assume traffic mean arrival rate and mean service rate accordingly.

3. Scenario 3: customers are served on priority basis. Customer 1 type traffic given higher priority to customer 2 type (ex: voice given priority to data).

4. Scenario 4: Introduce one more type of customer, say type 3 customer and serve them as * Priority scheduling (signaling traffic (type 3) preferred over voice (type 2) which is preferred over data (type 1)) * Round Robin scheduling

Chapter 2

ABSTRACT

1. This work contains an introduction to petrinet model with a customer service scenario in a restaurent is implemented.
2. This focus on the use or stochastic reward petrinet which can include guard function and priority for the timed transaction.
3. Thus different process can be enabled as per requirement.
4. then plot the graph and compare its characteristic.

Chapter 3

INTRODUCTION

A Petri net (also known as a place/transition net or P/T net) is one of several mathematical modeling languages for the description of distributed systems. A Petri net is a directed bipartite graph, in which the nodes represent transitions (i.e. events that may occur, signified by bars) and places (i.e. conditions, signified by circles). The directed arcs describe which places are pre- and/or postconditions for which transitions (signified by arrows).

Like industry standards such as UML activity diagrams, BPMN and EPCs, Petri nets offer a graphical notation for stepwise processes that include choice, iteration, and concurrent execution. Unlike these standards, Petri nets have an exact mathematical definition of their execution semantics, with a well-developed mathematical theory for process analysis.

Chapter 4

SPECIFICATIONS AND ASSUMPTIONS

Specifications

1. This implementation is based on SHARP TOOLS implementation .
2. The implementation is need good knowledge of petri nets.
3. Deals with the simulation of IEEE 802.11 mac protocol with RTS/CTS mechanism.

Assumptions

1. Arrival rate propotional to tha packet size.
2. Virtual load will be propotional to the square of the arrival rate.

Chapter 5

LOGIC USED/METHODOLOGY

The methodology that is used for developing this project work is defined below:

1. First use sharp-gui tools to crete given model .
2. And the architecture is based on petri nets.
3. A Petri net (also known as a place/transition net or P/T net) is one of several mathematical modeling languages for the description of distributed systems.
4. A Petri net is a directed bipartite graph, in which the nodes represent transitions (i.e. events that may occur, signified by bars) and places (i.e. conditions, signified by circles).
5. The directed arcs describe which places are pre- and/or postconditions for which transitions (signified by arrows) occurs.
6. input arcs are directed arcs drawn from places to transitions ,they represent the conditions that need to be satisfied for the event to be activated
7. Output arcs are directed arcs drawn from transitions to places,they represent the conditions resulting from the occurrence of an event
8. Input places of a transition are the set of places that are connected to the transition through input arcs.
9. Output places of a transition are the set of places to which output arcs exist from the transition.

10. Tokens are dots (or integers) associated with places; A place containing tokens indicates that the corresponding condition is active

11. Marking of a Petri net is a vector listing the number of tokens in each place of the net.

12. When input places of a transition has the required number of tokens, the transition is enabled.

13. An enabled transition may fire (event happens) taking a specified number of tokens from each input place and depositing a specified number of tokens in each of its output places

14. first calculates the sum throughput corresponding to the arrival rate .

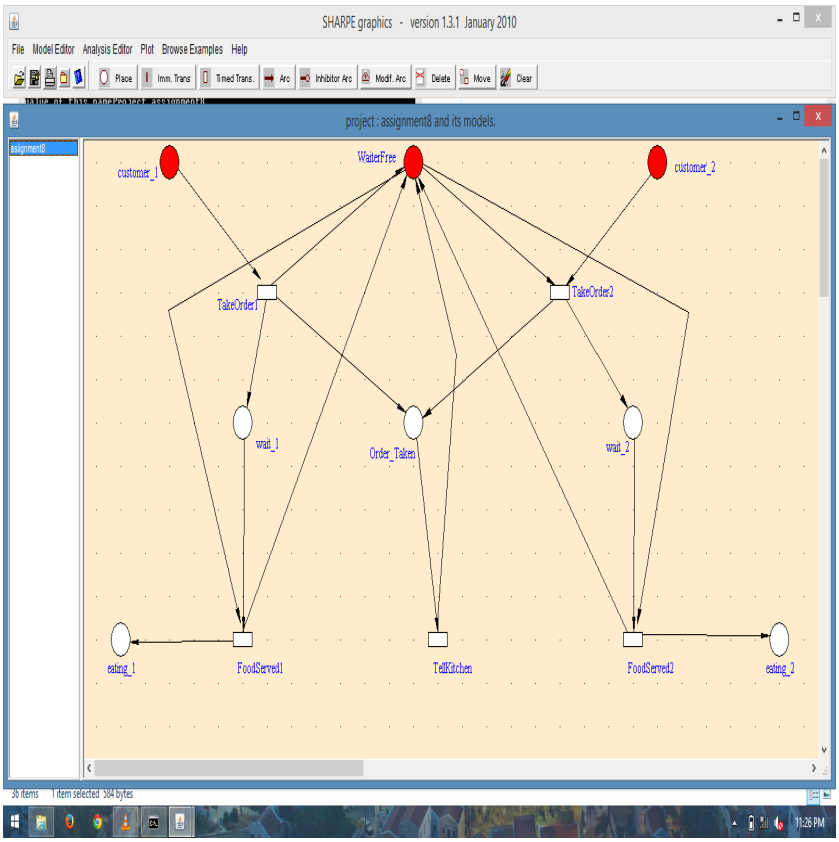
15. then calculate the sum throughput corresponding to the virtual loads.

16. And then compare the characterstic of both the graph.

Chapter 6

EXECUTION DIRECTIVES

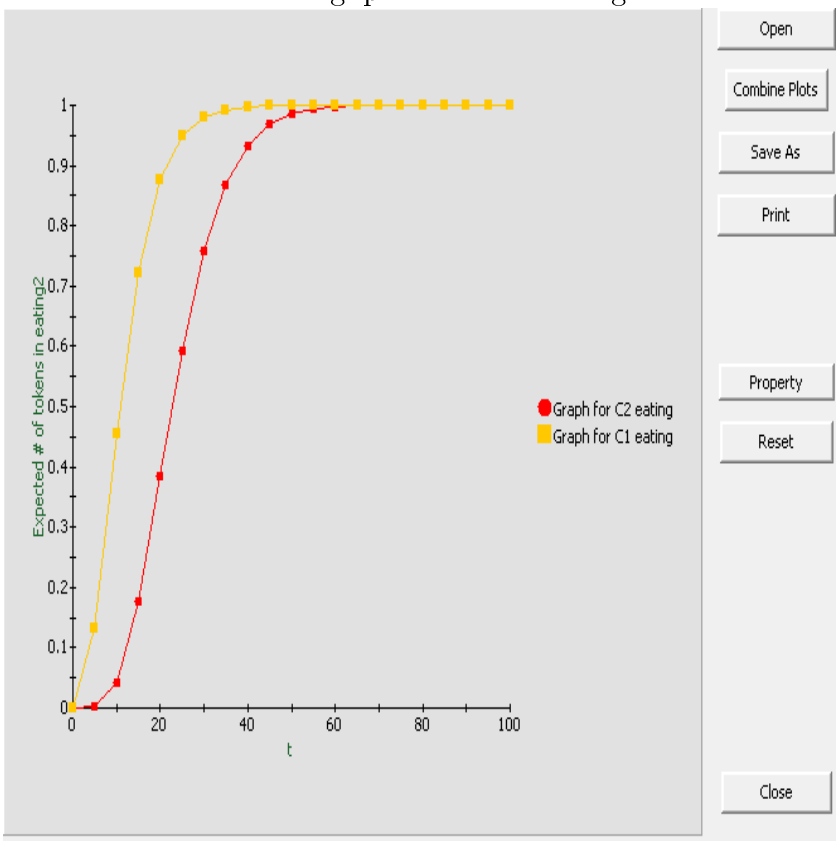
1. Give the project name and experiment name.
2. Then save it after run it create .rgl and .gpn file.
3. it need petri nets architecture to create the model.



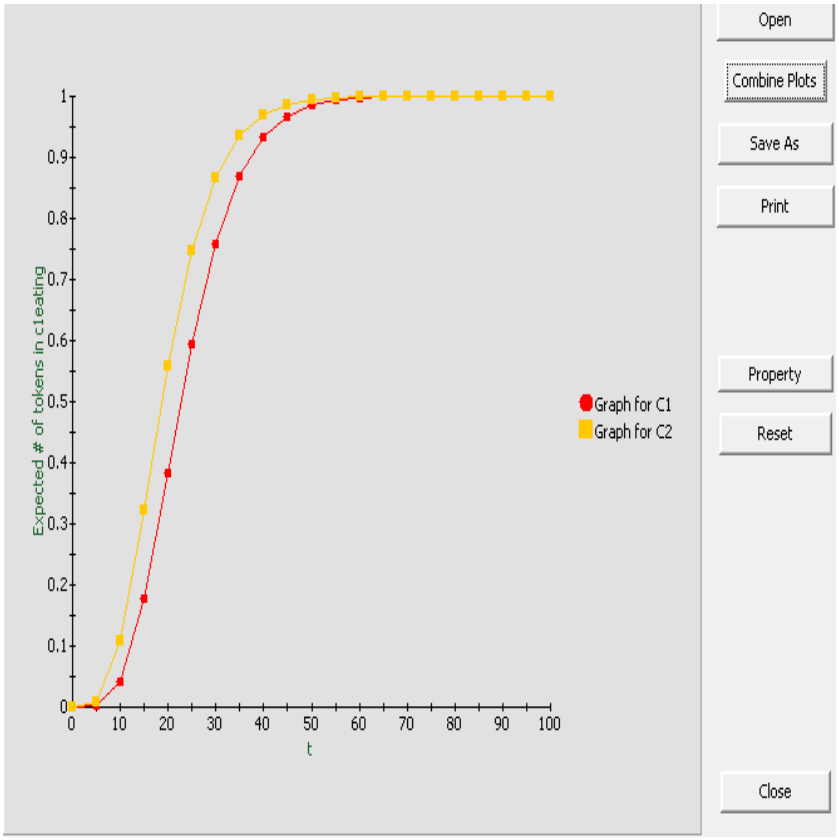
Chapter 7

RESULTS AND CONCLUSIONS

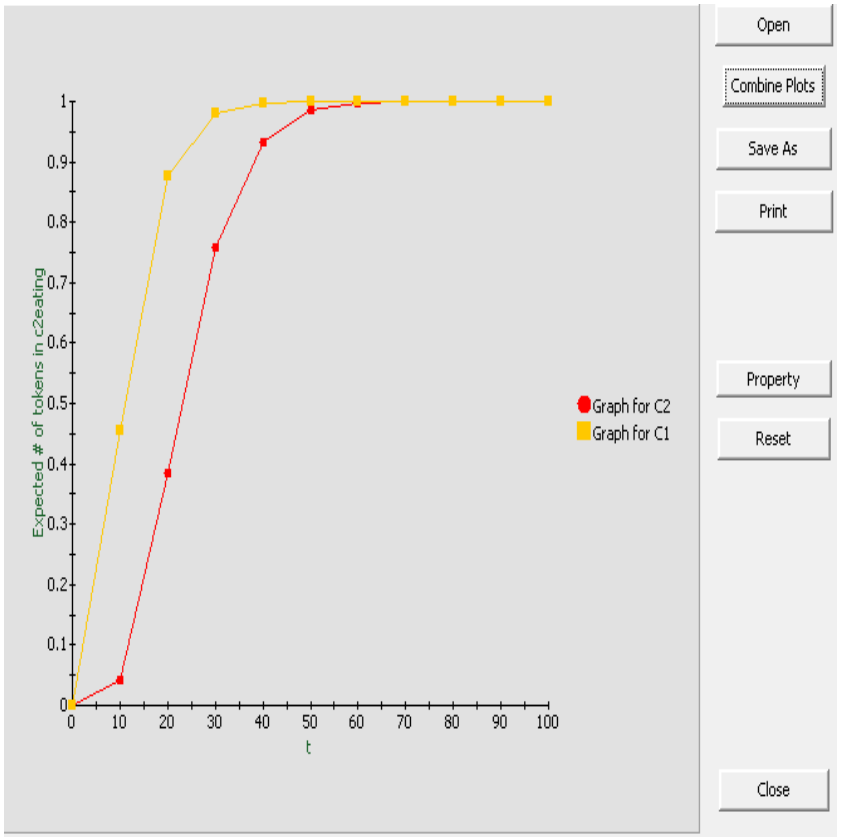
Hence the Simulate of all the given senario in sharp-gui is done and performance measures is verified. The graph between arrivalrate and throughput ,and virtual load and throughput are close enough.



Scenario 1
Waiter takes order from customer 1; serves customer 1; takes order from cus-

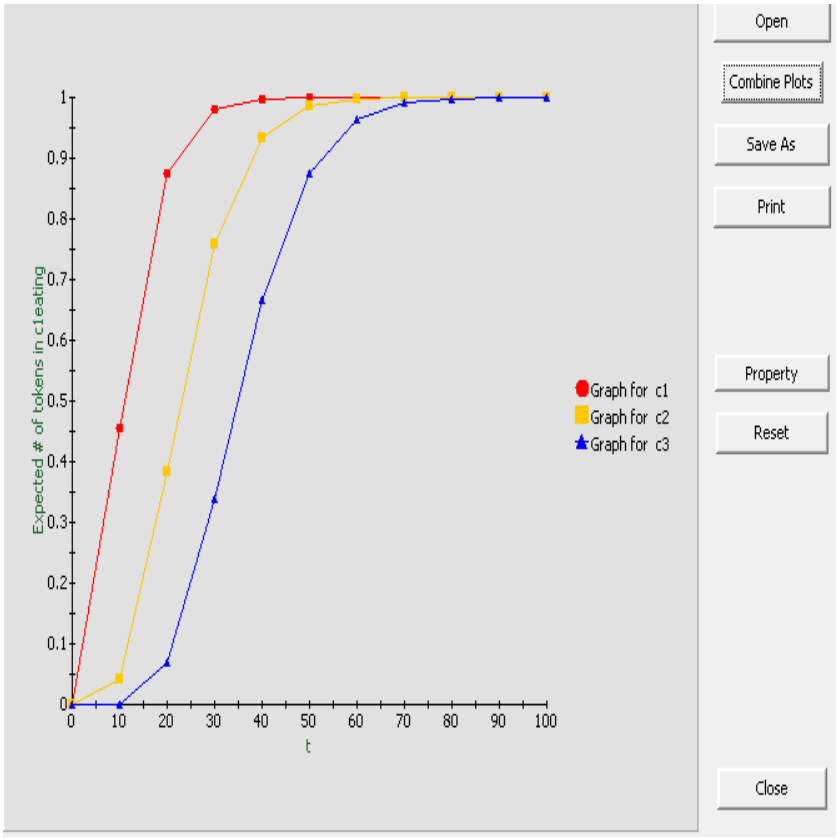


tomer 2; serves customer 2 Scenario 2 Waiter takes order from customer 1; takes order from customer 2; serves customer 2; serves customer 1.



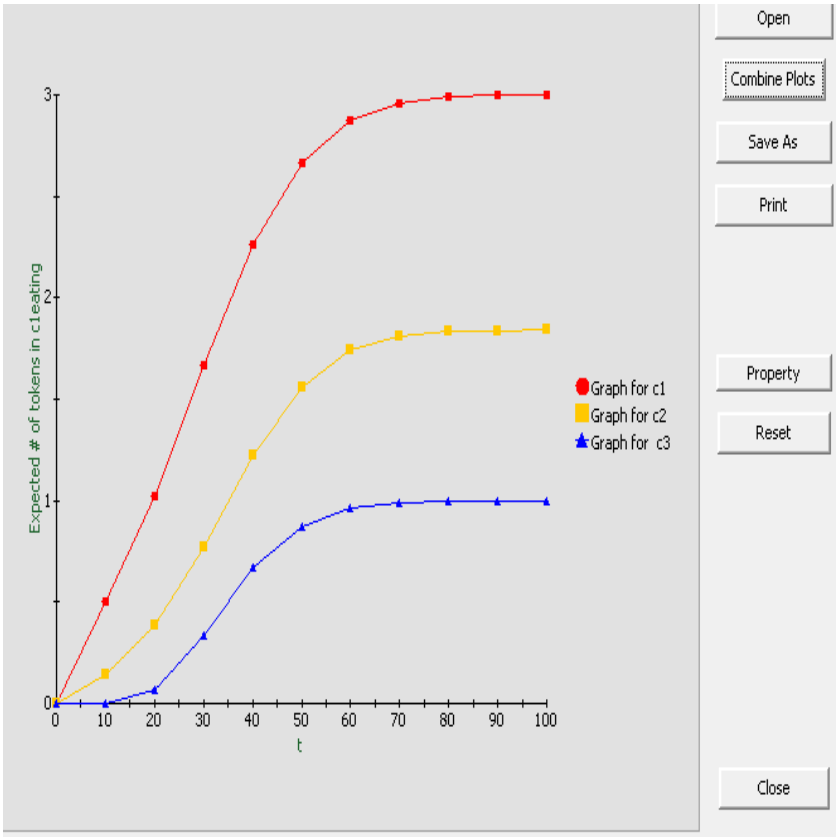
Scenario 3

Customers are served on priority basis. customer 1 type traffic given higher priority to customer 2 type (ex: voice given priority to data)



Scenario 4

Introduce one more type of customer, say type 3 customer and serve them as (i) priority scheduling



Scenario 4

Introduce one more type of customer, say type 3 customer and serve them as (ii) round robin scheduling.