**AddisAbaba Institute of Technology**

**Dept. Electrical and Computer Engineering**

***Subject: Computer Simulation and Modeling***

**Project Title: Three-Tier Architecture E-banking**

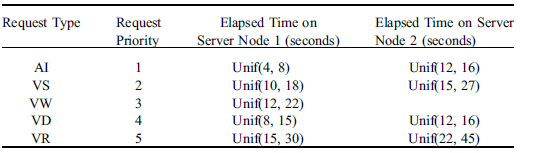
**Name: Addisalem .T**

**Id : GSR/2216/07**

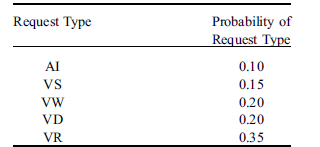
**Submitted to Dr. Derje**

**Date 2/18/2015**

**Problem description**

2. three-tier architecture in e-banking. The First New Brunswick Savings (FNBS) bank has introduced e-banking to its customers. The e-banking infrastructure consists of a cluster of two server nodes linked to each other via a TP monitor, which dynamically balances the workload of each server in the cluster. The processing time at the TP monitor is 0.05 seconds. The e-banking program offers a number of services to its customers, as follows: account identification (AI), view account summary (VS), view last 15 withdrawals (VW), view last 15 deposits (VD), and view interim account report (VR). The elapsed times of services are server-node dependent. These are given in following table along with request priorities.

Interarrival times of requests are iid exponentially distributed with a mean of 10 seconds. The probabilities of service request types arriving from any node are given in the next table.



Server processes implement priority queuing disciplines with smaller priority values of requests given higher priority.

a. Develop an Arena model for the FNBS bank’s e-banking system and simulate it for an 8-hour period.

b. Estimate the following statistics:

1. Server process utilizations
2. Transmission network utilization
3. Average pooled network delay
4. Average response time for each service request type
5. Average delays at server process queues

**General Description**

**Design of Three-tier architecture in E-banking**

Entity and their attributes

Customer

1. get the customer's ID
2. get the customer's arrival time
3. get the customer's service Time
4. get the customer's Request Type
5. get the customer's Priority

Server (Teller)

1. get the server's ID
2. start serving a customer
3. get information on the customer being served
4. check to see when the server will be free
5. finish serving the customer

Service Center(main method)

* 1. get the current time
  2. add a customer
  3. do business (serve customer if available)
  4. check to see if any customers remain
  5. calculate all performance measure

### Events

Each thing that happens is called an *event*. An event stores a time, its type, and its target object. The time is the point in time at which the event happens. The type is the kind of event. Our simulation only has a CUSTOMER\_ARRIVAL event and a SERVICE\_FINISHED event. The target is the object that is affected by the event (either a customer or a Server).

1. Events
   * 1. New Customer arrival
     2. customer departure
2. States
   * 1. Number of busy servers
     2. number of customers waiting service
3. State transitions
   * 1. Arrive event – schedules another new event
     2. Depart event

Generate the initial events and insert them in the event queue while the event queue is not empty:

1. remove the next event (with the smallest time) from the event queue
2. update current time to event's time
3. process the event (possibly generating new events and inserting them in the event queue)

**Arrival event**

Enque customer

Is Server free?

Yes no

Remove server from idle queue customer

Remove customer from customer queue customer

Create service end event customer

Schedule server and service end event

**Depart Event**

Schedule server and service end event

Is customer queue empty?

No yes

Remove first customer from customer queue

Put server idle sate

Create service end event

**Object Flow Diagram**

Sp11 server Node1

Sp11 queue Node1

Sp12 server Node1

Sp12 queue Node1

General queue

SP22 server Node2

Sp22 queue Node2

Sp21 server Node2

Sp21 queue Node2