QNAT

A Graphical *Queuing Network Analysis Tool* for General Open and Closed Queuing Networks

Sanjay K. Bose

QNAT developed at -

Dept. of Elect. Engg., I.I.T., Kanpur, INDIA

by - Sanjay K. Bose *skb@ieee.org*
D. Manjunath *dmanju@ee.iitb.ernet.in*
M.N. Umesh *umesh@sasi.com*
D. M. Bhaskar
Hema Tahilramani *hema@networks.ecse.rpi.edu*

URL: *http://poisson.ecse.rpi.edu/~hema/qnat/*

Beta version is being distributed free over the Internet

System Requirements for QNAT

- •PC - Windows Operating System

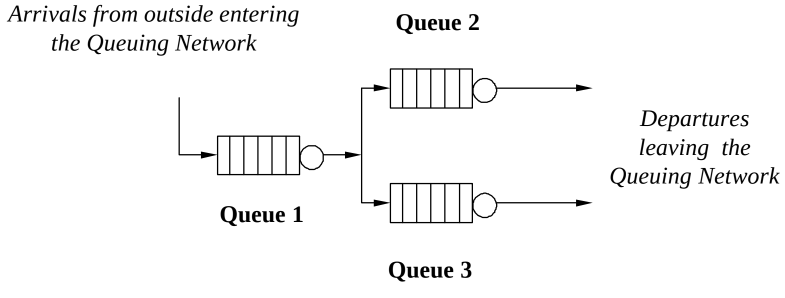
  Win 3.1 or Win95/98/NT

- •Mathematica™ (with Mathlink) as the computing platform

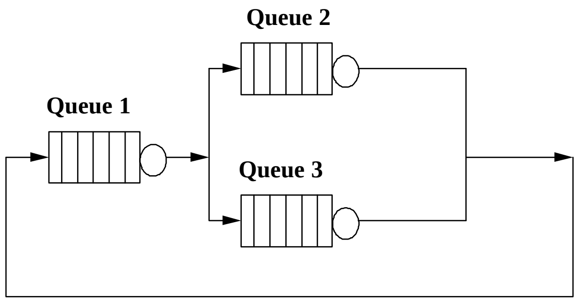
  Ver 2.2 for Win 3.1/Win95

  Ver 3.0 or higher for Win95/98/NT

Open Queuing Network

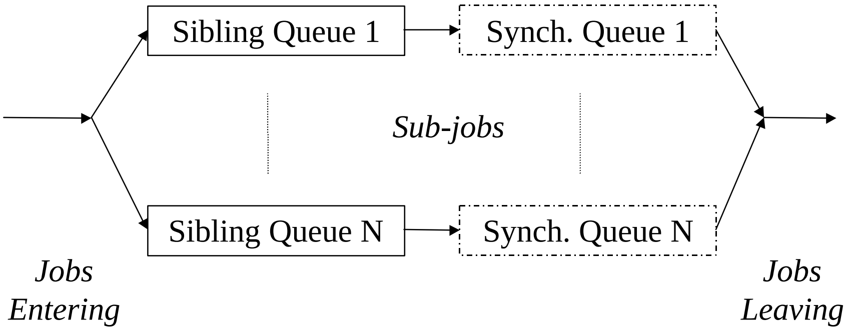


Closed Queuing Network



No External Arrivals or Departures

Fork/Join Queues (Nodes)



Fork/Join Queues (Nodes)

- Single Job *splits* into a number of sub-jobs, one for each sibling queue
- Job exits node only after each sub-job completes its service at its sibling queue
- Optional ***Synchronizing Queue*** to hold sub-jobs; this will free the sibling queue to service later jobs

Features of QNAT

User-friendly ***Graphical User Interface*** (GUI) allows -

- Creating/Saving/Modifying Networks
- Open Previously Saved Networks
- Graphical Input/Output through GUI
- Printing Analysis Results to Files

Features of QNAT

* Analyzes a large variety of general open/closed queuing networks with finite and/or infinite capacity queues

* For finite capacity queues, a wide range of blocking mechanisms can be handled

* For open/closed networks of infinite capacity queues, QNAT can handle multiple customer classes or fork/join nodes (for a single class)

Queuing Networks Analyzed by QNAT

[A] ***Networks of Multi-server Infinite Capacity Queues with Multiple Job Classes***

- Open Networks
- Closed Networks
- Mixed Networks, with some job classes closed and the others open

Queuing Networks Analyzed by QNAT

[B] ***Networks of Multi-server Infinite Capacity Queues with a Single Job Class***

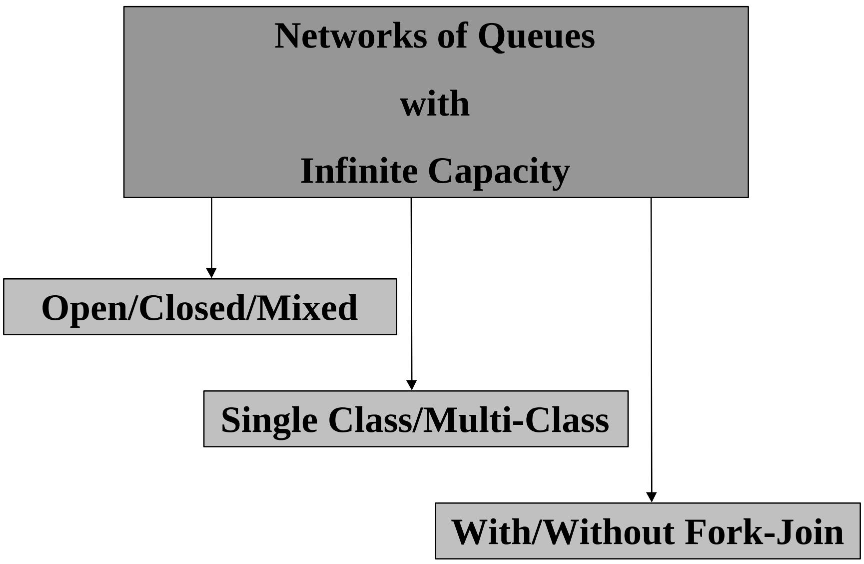
- Open/Closed Networks without Fork/Join Nodes
- Open Networks with "Fork/Join Nodes without Synchronizing Queue"
- Closed Networks with "Fork/Join Nodes with/without Synchronizing Queue"

Queuing Networks Analyzed by QNAT

[C] ***Networks of Multi-server Finite or Infinite Capacity Queues with a Single Job Class***

- Open/Closed Networks with Transfer Blocking
- Open/Closed Networks with Repetitive Service Blocking -
  - Random Destination (*RS-RD*) or
  - Fixed Destination (*RS-FD*)
- Open Networks with Rejection Blocking

Solution Techniques Used in QNAT



Open Networks
of
Infinite Capacity, Multi-server Queues
(Single and Multiple Traffic Classes)

Analytical Approach (Whitt's GI/G/M Approximation)

* Assumes Product-Form Solution and adjusts node and route parameters to remove immediate feedback

* Solve for first & second moments of net arrival process at each node by solving sets of simultaneous equations

* Use this and a GI/G/M model for each queue in isolation to solve for the open network of queues

Open Networks
of
Infinite Capacity, Multi-server Queues
(Single and Multiple Traffic Classes)

Input Parameters

* Mean and SQV (squared coefficient of variance) for each of the external arrival processes (for each class)

* Routing Probabilities (*Static*)

* No. of Servers in each queue

* Mean and SQV of the service times at each queue for each class

Open Networks
of
Infinite Capacity, Multi-server Queues
(Single and Multiple Traffic Classes)

Output Parameters

* Mean and SQV of number of each class in each queue,

* Waiting Times, Effective Arrival Rates etc. and their SQVs for each class of traffic at each queue in the network

* Network Parameters like throughput and sojourn time

Open Networks
of
Infinite Capacity, Multi-server Queues
(Single and Multiple Traffic Classes)

Network of both the following types can be analyzed by QNAT -

1. Networks with Probabilistic Routing between the Queues

    * Most common way of specifying a network
    * Routing specified through a probability matrix

2. Networks with deterministic class-based routing

Open Networks
of
Infinite Capacity, Multi-server Queues, Single Class Traffic
Fork-Join Nodes

QNAT can handle networks where some or all the nodes are of the ***fork-join*** type, subject to the following -

* Only *fork-join without synchronizing queues* are used

* Only *single class* of customers in the system

* All the sibling queues are *single-server* queues of *infinite capacity* and have *exponentially distributed service times* (but with possibly different means)

Open Networks
of
Infinite Capacity, Multi-server Queues, Single Class Traffic
Fork-Join Nodes

Analytical Approach

* *Effective Service Time Distribution* of the fork-join node
 ≡ Distribution of the random variable corresponding to the ***maximum*** of the service times of each sibling

* This is used to find the *mean* and *SQV* of the *effective service time*

* These values are then used in the earlier GI/G/M approach for Open Networks

Closed Networks
of
Infinite Capacity, Multi-Server Queues
(Multiple Traffic Classes)

Analytical Approach

* Assume **exponentially distributed service times** at nodes

* Product-Form Solution

* Use ***Mean Value Analysis*** (MVA) to directly find the mean performance parameters (iterative procedure)

Mean Value Analysis Theorem
for
Queuing Networks
(Single Class, State-Independent Service)

"A customer arriving to a queue in a product-form network sees precisely the same average number in the queue as an outside observer would see, if the network had one less customer"

MVA Theorem can also be stated for queuing networks with Multiple Service Classes and/or with State-Dependent Service Times

QNAT has internally implemented the most general form of the MVA theorem

Closed Networks
of
Infinite Capacity, Multi-server Queues, Single Class Traffic
Fork-Join Nodes without Synchronizing Queue

Analytical Approach

* Get *effective service time distribution* for each node as in open networks
* For each node, fit an exponential distribution (with min. mean square error) to the effective distribution and compute its ***mean***
* Use this mean for the node in the MVA algorithm to compute the mean performance parameters of interest for the network

Closed Networks
of
Infinite Capacity, Multi-server Queues, Single Class Traffic
Fork-Join Nodes with Synchronizing Queue

Analytical Approach (Perros & Liu)

* Approximate computation of a ***Flow Equivalent Server*** (FES) which converts the fork-join node to one with ***state-dependent service times***

User specifies the service times for each of the sibling queues at the fork-join node

Both tabular form and expressions accepted

Closed Networks
of
Infinite Capacity, Multi-server Queues, Single Class Traffic
Fork-Join Nodes with Synchronizing Queue

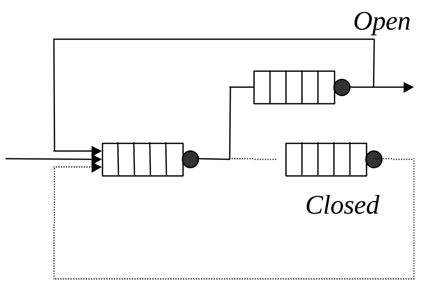
Analytical Approach (Perros & Liu)

Flow Equivalent Server Computation (continued)

Step-by-step procedure first combines two sibling queues for an equivalent FES. Then combines more siblings to this, one at a time, to get the final FES.
FES essentially a node with ***state-dependent service times***

* Replace Fork-Join node by FES and solve network using **State Dependent MVA** algorithm

Mixed Networks
of
Infinite Capacity, Multi-server Queues, Multi-Class Traffic



* *For some of the traffic classes, the network is* **open**
External arrivals/departures occur for these classes

* *For the other traffic classes, the network is* **closed**
No external arrivals/departures for these classes
Fixed number of customers circulating in network

Mixed Networks
of
Infinite Capacity, Multi-server Queues, Multi-Class Traffic

Analytical Approach - (Lazowska, Zahorjan, et al)

* Compute utilizations of the open classes at each node
&
the net utilization of each node due to all the open classes

* Eliminate the open classes and convert the *mixed* network into a *closed* network with ***suitably inflated service demand*** *for the closed classes*

Mixed Networks
of
Infinite Capacity, Multi-server Queues, Multi-Class Traffic

Analytical Approach - (continued)

* Use MVA to solve the modified *closed network* and obtain the corresponding mean performance measures

* Compute the performance of the *open classes* by accounting for the mean queue length of the closed classes at each node in the computational approximations

Networks of Queues with Blocking

Single Class

(some or all queues of finite capacity)

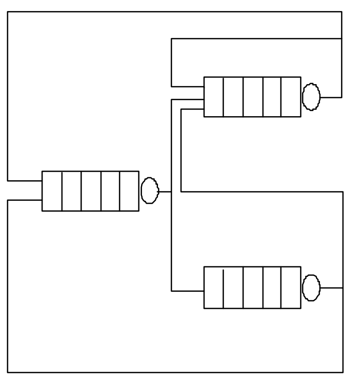
Open/Closed

Blocking Modes

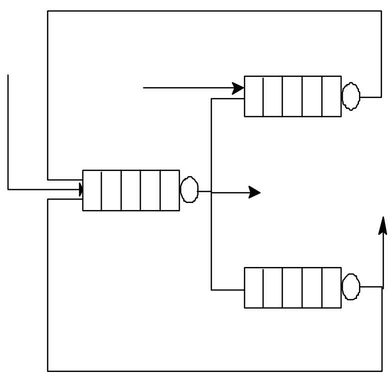
1. Transfer

2. Repetitive Service - ***Random or Fixed Destination***

3. Rejection



A Closed Network



An Open Network

Open Networks with Rejection Blocking

"Customers trying to enter a blocked node are lost"

QNAT can solve open networks with Rejection Blocking for -

* Single Customer Class

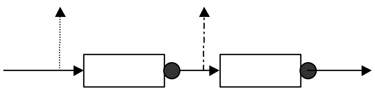
* Exponentially Distributed Service Times

* Poisson Arrival Processes

* Product-Form Solution

Open Networks with Rejection Blocking

Analytical Approach -



* Assume initial values of blocking probabilities

* Solve traffic equations for flows at individual queues

* Use M/M/c/K results for individual queues

* Iterate on convergence of flows for state distributions

* Use "random walk" to find ***sojourn times*** for customers who get *complete service* or get *incomplete service*

Open Networks with Rejection Blocking

Results Provided -

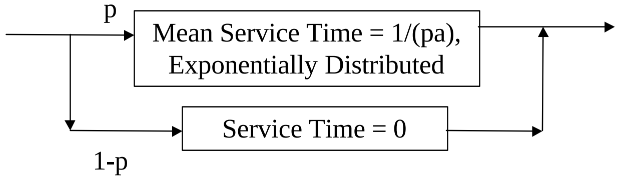
* Mean Number/Waiting Time in each queue

*Rejection Rate (Customers refused entry)
∗Loss Rate (Loss inclusive of Rejection)
∗Departure Rate (Fully served Customers)

∗Average Sojourn Time
∗Average Sojourn Time (Accepted Customer)
∗Average Sojourn Time (Lost Customer)
∗Average Sojourn Time (Departing Customer)

Generalized Exponential (GE) Distributions

GE - Distributions may be used as a two-moment approximation for representing a wide range of general distributions (*specified by their **Mean** and **SQV***)

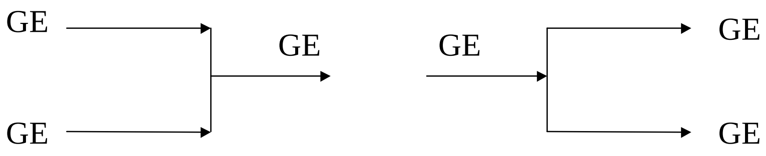


Model for GE-distributed Service Times

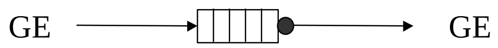
Mean = 1/a

$SQV = Variance/(Mean)^2 = 2/(p) - 1$

Using Generalized Exponential (GE) Distributions in Queuing Networks



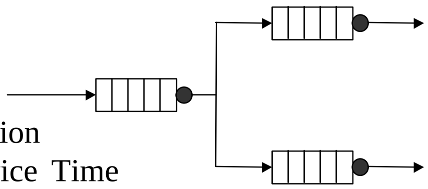
GE/GE/c/N Queue



GE Process through a GE/GE/c/N queue using MEM approach

Open Networks with Repetitive Service Blocking

Fixed or Random Destination



Assumptions -

*Product-Form Approximation
*GE Inter-arrival and Service Time Distributions
* FCFS Service
* Static Routing Matrix

Analytical approach based on a Maximum Entropy based Method (MEM) developed by Kouvastos et al

Open Networks with Repetitive Service Blocking

Fixed or Random Destination

Analytical Approach

(Actual expressions may differ for RS-RD and RS-FD)

* Remove immediate feedback by modifying node and routing parameters

* Assume initial parameters for iteration purposes

* Solve traffic equations for flows

Open Networks with Repetitive Service Blocking

Fixed or Random Destination

Analytical Approach

* Using GE results, find mean and SQV of the net arrival process to each queue

* Modify mean and SQV of service times at each node

* Solve each node using ***MEM*** approximation methods

* Update iteration parameters and iterate until convergence of flows and SQVs is obtained

Open Networks with Repetitive Service Blocking

Fixed or Random Destination

Input Parameters

* External Arrival Processes' Mean and SQVs

* Service Time Means and SQVs at each node

* Number of Servers at each node

* Number of Buffers at each node

Open Networks with Repetitive Service Blocking

Fixed or Random Destination

Output Results

For Each Queue -

Mean Number, Mean Delay, Utilization, Departure Blocking Probability, P{queue empty}, P{queue full}

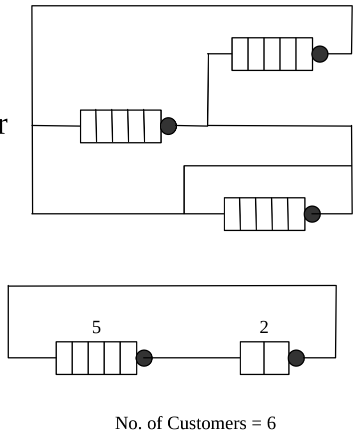
For Network -

Blocking Probability Matrix, Network Throughput and Sojourn Time

Closed Networks with Repetitive Service Blocking

Fixed or Random Destination

* Same assumptions needed as for Open Networks
* Actual expressions may differ for RS-RD and RS-FD but approach is basically similar
* ***Deadlocks*** may happen and are assumed to be immediately resolved
* Queues may get ***censored*** - i.e. the number in the queue may have a non-zero lower limit



Closed Networks with Repetitive Service Blocking

Fixed or Random Destination

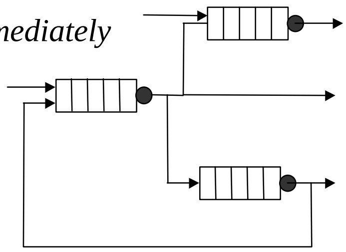
Analytical Approach

* Solve a ***Pseudo-Open Network*** so that the mean number in the pseudo-open network is same as that in the closed network

* Calculate the ***Normalization Constant*** for the network and use this to normalize the flows so that *flow balance* is satisfied

* **State Probabilities** at the individual queues may now be calculated from which performance measures are computed

Open Networks with Transfer Blocking



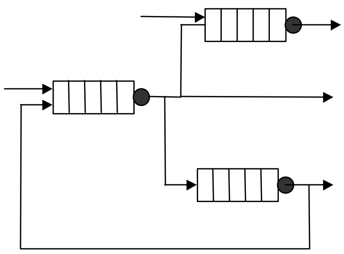
* **Deadlocks** may occur - resolved *immediately*

**Additional Service* under blocking equals the minimum of the residual service times of all the customers getting served at the destination node

* **QNAT** uses a new **MEM** based approximate analysis method proposed by us. This seems applicable to networks where the queues are not *heavily loaded*

Technique has been verified through simulations etc.

Open Networks with Transfer Blocking



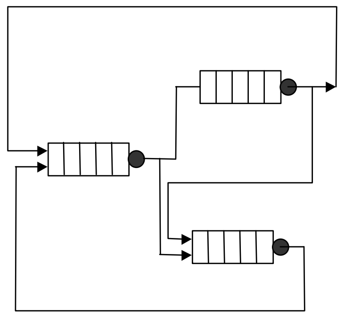
* **Hypothetical Nodes** added to model blocking

* These nodes added to streams which may encounter blocking

* These nodes are infinite server queues with service time modeling the transfer blocking time

*** MEM** used to solve the resultant network appropriately

Closed Networks with Transfer Blocking



Assumptions -

* Exponential inter-arrival and service times
* Network is Deadlock-Free
Number of customers less than the net capacity of any directed loop

Analytical Approach -

Uses a special approximation method (Akyildiz) based on mapping to an equivalent non-blocking network. Method used requires *state enumeration* and is not suitable for large networks and/or large user populations!

Simulation of Queueing Networks in QNAT

- QNAT also supports a Simulation option
- Uses same GUI as for the Analytical option
- Network modelled may either be analyzed or simulated
- Confidence measures not provided by the simulation but random variable seed may be changed for multiple runs
- Deadlock conditions may occur