Queueing Networks

Simonetta Balsamo, Andrea Marin
Università Ca' Foscari di Venezia
Dipartimento di Informatica, Venezia, Italy

School on Formal Methods 2007: Performance Evaluation
Bertinoro, 28/5/2007

Queueing Networks

Stochastic models of resource sharing systems
computer, communication, traffic, manufacturing systems

Queueing Network a system model
set of service centers representing the system resources that provide service to a collection of customers that represent the users

Customers compete for the resource service => **queue**

QN are powerful and versatile tool for system performance evaluation and prediction

Stochastic models based on queueing theory
* **queuing system models** *(single service center)*
  represent the system as a *unique* resource
* **queueing networks**
  represent the system as a *set of* interacting resources
  => model system structure
  => represent traffic flow among resources

System performance analysis
* derive performance indices
  (e.g., resource utilization, system throughput, customer response time)
* analytical methods exact, approximate
* simulation

Outline

I) **Queueing systems**
various hypotheses
analysis to evaluate performance indices
underlying stochastic Markov process

II) **Queueing networks (QN)**
model definition
analysis to evaluate performance indices
types of customers: multi-chain, multi-class models
types of QN

Markovian QN
underlying stochastic Markov process

III) **Product-form QN**
have a simple closed form expression of the stationary state distribution
BCMP theorem
=> efficient algorithms to evaluate average performance measures

Solution algorithms for product-form QN
Convolution, MVA, RECAL, …

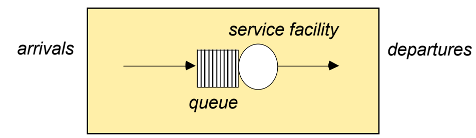
IV) **Properties of QN**
arrival theorem - exact aggregation - insensitivity

Extensions and application examples
special system features (e.g., state-dependent routing, negative customers, customers batch arrivals and departures and finite capacity queues)

Introduction: the queue

- basic QN: Queueing Systems

- Customers
  arrive to the service center
  ask for resource service
  possibly wait to be served => queueing discipline
  leave the service center



- under exponential and independence assumptions
  one can define an associated stochastic *continuous-time Markov process*
  to represent system behaviour

- performance indices are derived from the solution of the *Markov process*

Stochastic processes

Stochastic process: set of random variables

$$\{X(t) \mid t \in T\}$$

defined over the same probability space indexed by the parameter t, called **time**

each X(t) random variable
takes values in the set Γ called **state space** of the process

Both T (time) and Γ (space) can be either *discrete* or *continuous*

Continuous-time process if the time parameter t is *continuous*

Discrete-time process if the time parameter t is *discrete*

$$\{X_n \mid n \in T\}$$

Joint probability distribution function of the random variables $X(t_i)$

$$Pr\{X(t_1) \le x_1; X(t_2) \le x_2; \ldots; X(t_n) \le x_n\}$$

for any set of times $t_i \in T$, $x_i \in \Gamma$, $1 \le i \le n$, $n \ge 1$

Markov processes

Discrete-time Markov process

$$\{X_n \mid n=1,2,\ldots\}$$

if the state at time **n + 1** only depends on the state probability at time **n** and is independent of the previous history

$$\boxed{Prob\{X_{n+1}=j \mid X_0=i_0; X_1=i_1; \ldots; X_n=i_n\} = Prob\{X_{n+1}=j \mid X_n=i_n\}}$$

$\forall n>0, \forall j, i_0, i_1, \ldots, i_n \in \Gamma$

Continuos-time Markov process

$$\{X(t) \mid t \in T\}$$

$$\boxed{Prob\{X(t)=j \mid X(t_0)=i_0; X(t_1)=i_1; \ldots; X(t_n)=i_n\} = Prob\{X(t)=j \mid X(t_n)=i_n\}}$$

$\forall t_0, t_1, \ldots, t_n, t : t_0<t_1<\ldots<t_n<t$, $\forall n>0, \forall j, i_0, i_1, \ldots, i_n \in \Gamma$

Markov property

The *residence time* of the process in each state is distributed according to

geometric — for *discrete-time* Markov processes
negative exponential distribution — for *continuous*-time Markov processes

Discrete-space Γ Markov processes are called **Markov chain**

Analysis of Markov processes

Discrete-time Markov chain

$\{X_n \mid n=1,2,...\}$

homogeneous if the one-step conditional probability is independent on time n

$p_{ij} = Prob\{X_{n+1}=j|X_n=i\}$ $\forall n>0, \forall i,j \in \Gamma$

$\mathbf{P}=[p_{ij}]$ state transition **probability** matrix

If the stability conditions holds, we can compute the **stationary state probability**

$\boldsymbol{\pi} = [\pi_0, \pi_1, \pi_2, \ldots]$

$\pi_j = Pr\{X=j\}$ $\forall j \in \Gamma$

For ergodic Markov chain (irreducible and with positively recurrent aperiodic states) $\boldsymbol{\pi}$ can be computed as

$\boxed{\boldsymbol{\pi} = \boldsymbol{\pi}\,\mathbf{P}}$ with $\sum_j \pi_j = 1$

system of global balance equations

Analysis of Markov processes

Continuous-time Markov chain

$\{X(t) \mid t \in T\}$

homogeneous if the one-step conditional probability only depens on the interval width

$p_{ij}(s) = Prob\{X(t+s) = j \mid X(t) = i\}$ $\forall t>0, \forall i,j \in \Gamma$

$\mathbf{Q} = \lim_{s \to 0} (\mathbf{P}(s) - \mathbf{I})/s$

$\mathbf{Q}=[q_{ij}]$ matrix of state transition **rates** *(infinitesimal generator)*

If the stability conditions holds, we compute the **stationary state probability**

$\boldsymbol{\pi} = [\pi_0, \pi_1, \pi_2, \ldots]$

For ergodic Markov chain (irreducible and with positively recurrent aperiodic states) as

$\boxed{\boldsymbol{\pi}\,\mathbf{Q} = 0}$ with $\sum_j \pi_j = 1$

system of global balance equations

Birth-death Markov processes

State space $\Gamma = \mathbb{N}$ $\boldsymbol{\pi} = [\pi_0, \pi_1, \pi_2, \ldots]$

The only non-zero state transitions are those

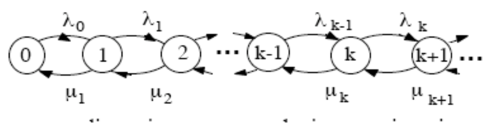
from any state i to states $i-1, i, i+1$, $\forall i \in \Gamma$

Matrix **P** *(discrete-time)* or **Q** *(continuous-time)* is tri-diagonal

λ_i birth transition rate , $i\geq 0$
μ_i death transition rate, $i\geq 1$

continuos-time Markov chain

$q_{i\,i+1} = \lambda_i$ $i\geq 0$
$q_{i\,i-1} = \mu_i$ $i\geq 1$
$q_{i\,i} = -(\lambda_i + \mu_i)$ $i\geq 1$
$q_{00} = -\lambda_0$
$q_{i\,j} = 0$ $|i-j|>1$



$$\boxed{\pi_i = \pi_0 \prod_{j=0}^{i-1} \frac{\lambda_j}{\mu_{j+1}}}$$

$$\boxed{\pi_0 = \left\{ \sum_{i=0}^{\infty} \prod_{j=0}^{i-1} \frac{\lambda_j}{\mu_{j+1}} \right\}^{-1}}$$

Closed-form expression Normalizing condition

Birth-death Markov processes

Sufficient condition for stationary distribution

$$\boxed{\exists\, k_0 : \forall k > k_0 \quad \lambda_k < \mu_k}$$

Special case: constant birth and death rates

$\lambda_i = \lambda$ birth transition rate , $i\geq 0$
$\mu_i = \mu$ death transition rate, $i\geq 1$

Let $\rho = \lambda / \mu$
If $\rho<1$

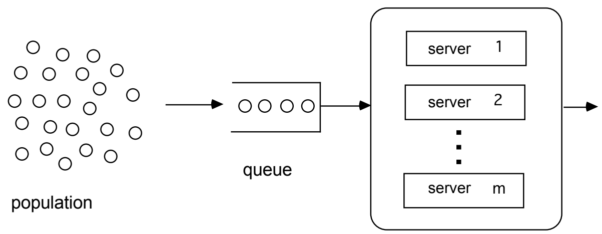
$\pi_0 = [\sum_k \rho^k]^{-1} = 1-\rho$
$\pi_k = \pi_0 \ (\lambda / \mu)^k$

$$\boxed{\pi_k = (1-\rho)\, \rho^k \qquad k\geq 0}$$

Geometric distribution

Basic queueing systems

Single service center

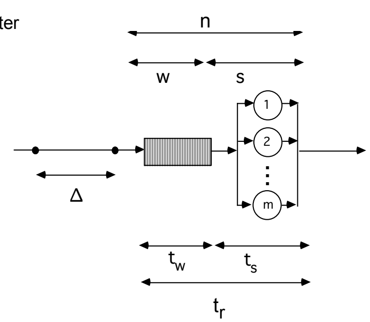


Customers resources offering a service

=> resource contention

Basic queueing systems

Single service center



Δ : interarrival time

w : number of customers in the queue

s : number of customers in service

n : number of customers in the system

t_w : queue waiting time

t_s : service time

t_r : response time

Definition of a queueing systems

- The queueing system is described by
  - * the arrival process
  - * the service process
  - * the number of servers and their service rate
  - * the queueing discipline process
  - * the system or queue capacity
  - * the population constraints

- Kendall's notation **A/B/X/Y/Z**

A	interarrival time distribution (Δ)
B	service time distribution (t_s)
X	number of servers (m)
Y	system capacity (in the queue and in service)
Z	queueing discipline

A/B/X if Y = ∞ and Z = FCFS *(default)*

Examples: A,B : D deterministic (constant)
M exponential (Markov)
E_k Erlang-k
G general

Examples of queueing systems: D/D/1, M/M/1, M/M/m (m>0), M/G/1, G/G/1

Analysis of a queueing systems

- system analysis
  - **Transient** for a time interval, given the initial conditions
  - **Stationary** in steady-state conditions, for stable systems
- Analysis of the associated stochastic process that represents system behavior
  - Markov stochastic process
  - birth and death processes

- Evaluation of a set of **performance indices** of the queueing system

* number of customers in the system	n
* number of customers in the queue	w
* response time	t_r
* waiting time	t_w
* utilization	U
* throughput	X

random variables: evaluate probability distribution and/or the moments

average performance indices

* average number of customers in the system	N=E[n]
* mean response time	$R=E[t_r]$

Some basic relations in queueing systems

Relations on random variables

$$n = w + s$$

$$t_r = t_w + t_s$$

=>

$$N = E[w] + E[s]$$

$$R = E[t_w] + E[t_s]$$

Little's theorem

$$N = X\,R$$

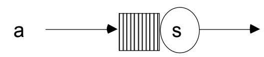
$$E[w] = X\,E[t_w]$$

> The average number of customers in the system is equal to the throughput times the average response time

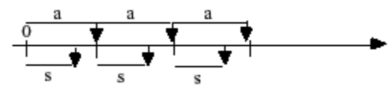
Very general assumptions
Can be applied at various abstraction levels (queue, system, subsystem)

Basic relation used in several algorithms for Queueing Network models and solution algorithms for product-form QN

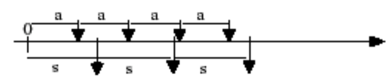
A simple example: D/D/1



- deterministic arrivals: constant interarrival time (a)
- deterministic service: each customers have the same service demand (s)
- transient analysis
  from time t=0
- if s<a



then U= …
- if s=a
- if s>a



then U= …

A simple example: D/D/1

- transient analysis

n(t) number of customer in the system at time t

if n(0)=0 empty system at time 0

then

$n(t) = 0$ if $s<a$, $i\,a + s < t < (i+1)\,a$, $i\geq 0$

$n(t) = 1$ if $s<a$, $i\,a \leq t \leq i\,a + s$, $i\geq 0$

$n(t) = 1$ if $s=a$

$n(t) = \lfloor t/a \rfloor - \lfloor t/s \rfloor$ if $s>a$ for $t\geq 0$

- stationary analysis

stability condition: $s\leq a$

If arrival rate $(1/a) \leq$ service rate $(1/s)$

$\Rightarrow$The system reaches the steady-state

$\Rightarrow n \in \{0,1\}$

Prob{n=0} = (a-s)/a

Prob{n=1} = s/a

- $w = 0$ $t_W = 0$ $t_r = s$ (deterministic r.v.)
- $X=1/a$ throughput
- $U = s/a$ utilization

Basic queueing systems: M/M/1

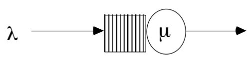
Arrival Poisson process, with rate λ

(exponential interarrival time)

Exponential service time with rate μ

$E[t_s] = 1/\mu$

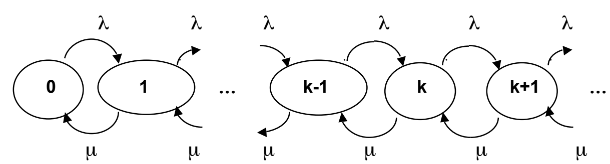
Single server



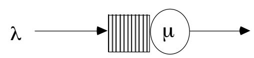
System state: n

Associated stochastic process:

birth-death *continuous-time* Markov chain with constant rates λ and μ



Basic queueing systems: M/M/1



- stationary analysis
  stability condition: $\lambda < \mu$

  traffic intensity $\boxed{\rho = \lambda / \mu}$

stationary state probability $\pi_k = \text{Prob}\{n = k\} \quad k \in \mathbb{N}$

$$\boxed{\pi_k = \rho^k (1 - \rho) \qquad k \geq 0}$$

$N = \rho / (1 - \rho)$

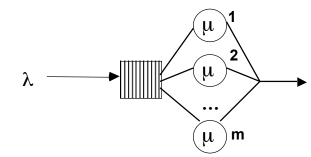
$R = 1 / (\mu - \lambda)$ *(Little's theorem)*

$X = \lambda$
$U = \rho$

$E[w] = \rho^2 / (1 - \rho)$
$E[t_w] = (1 / \mu)\, \rho / (1 - \rho)$ *(Little's theorem)*

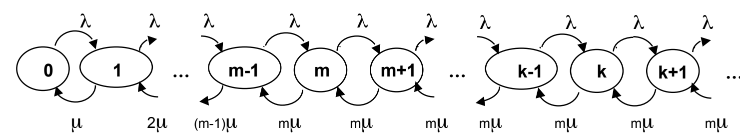
Basic queueing systems: M/M/m

Arrival Poisson process, with rate λ
(exponential interarrival time)
Exponential service time with rate μ
$E[t_s] = 1/\mu$
m servers

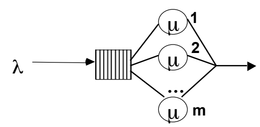


System state: n
Associated stochastic process:

birth-death *continuous-time* Markov chain with rates
$\lambda_k = \lambda$
$\mu_k = \min\{k,m\}\, \mu$



Basic queueing systems: M/M/m



- stationary analysis

stability condition: $\boxed{\lambda < m\,\mu}$

traffic intensity $\rho = \lambda / m\,\mu$

stationary state probability

$\pi_k = \pi_0\,(m\,\rho)^k / k!$	$1 \le k \le m$
$\pi_k = \pi_0\,m^m \rho^k / m!$	$k > m$

$$\pi_0 = \left[\sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} + \frac{(m\rho)^m}{m!} \frac{1}{1-\rho} \right]^{-1}$$ *(Erlang-C formula)*

$N = m\,\rho + \pi_m\,\rho / (1-\rho)^2$

$R = \pi_m / (m\,\mu\,(1-\rho)^2) + 1/\mu$ *(Little's theorem)*

$X = \lambda$ $U = \rho$ Prob{queue}$= \sum_{k \ge m} \pi_k = \pi_0\,(m\,\rho)^m / m!\,(1-\rho)$

$E[w] = \pi_m\,\rho / (1 - r\rho)^2$

$E[t_w] = \pi_m / ((1-\rho)^2\,\mu)$ *(Little's theorem)*

Basic queueing systems: M/M/∞

Arrival Poisson process, with rate λ

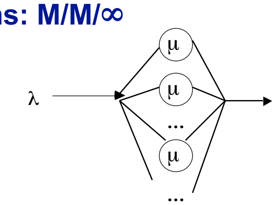
(exponential interarrival time)

Exponential service time with rate μ

$E[t_s] = 1/\mu$

infinite identical servers

No queue



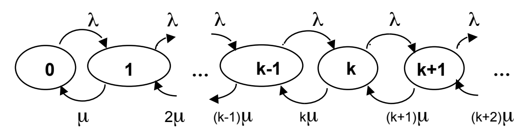
System state: n

Associated stochastic process:

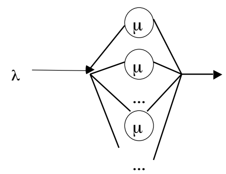
birth-death *continuous-time* Markov chain with rates

$\lambda_k = \lambda$

$\mu_k = k\,\mu$



Basic queueing systems: M/M/∞



• stationary analysis

the system is always stable

traffic intensity $\rho = \lambda / \mu$

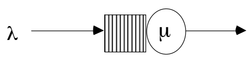
stationary state probability $\boxed{\pi_k = e^{-\rho} \rho^k / k! \quad k \geq 0}$

Poisson distribution

$N = \rho$
$R = 1 / \mu$
$X = \lambda$
$U = \rho$ $E[w] = E[t_w] = 0$

Delay queue

Basic queueing systems: M/G/1



Arrival Poisson process, with rate λ
(exponential interarrival time)
General service time with rate μ
$E[t_s] = 1/\mu$
$C_B = (Var[t_s])^{1/2} / E[t_s]$ coefficient of variation of t_s

Single server

The state defined as *n* (number of customers in the system) does not lead to a Markov process

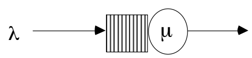
State description *n* for system M/M/1 gives a (birth-death) continuous-time Markov chain because of the exponential distribution (*memoryless* property)

We can use a different (more detailed) state definition to define a Markov process (e.g., the number of customers and the amount of service already provided to the customer currently in service)

The associated Markov process is not birth-death
Analysis of an *embedded* Markov process
Z-transform technique

Basic queueing systems: M/G/1

$E[t_s] = 1/\mu$
$C_B = (Var[t_s])^{1/2} / E[t_s]$



Khinchine Pollaczeck theorem
for any queueing discipline independent of service time without pre-emption

$$E[w] = \rho + \frac{\rho^2 (1 + C_B^2)}{2(1-\rho)}$$

$N = E[w] + l\,E[t_s]$
$R = N / X$

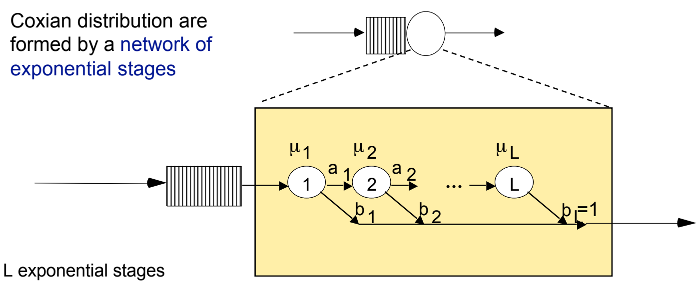
Stability condition: $\lambda < \mu$

PASTA theorem: ***Poisson Arrivals See Time Average***

The state distribution and moments seen by a customer at arrival time is the same as those observed by a customer at arbitrary times in steady-state conditions

Coxian distribution

Coxian distribution are formed by a network of exponential stages



L exponential stages
Stage ℓ service rate μ_ℓ
probabilities a_ℓ, b_ℓ: $a_\ell + b_\ell = 1$

Coxian distributions have rational Laplace transform

Can be used
- to represent general distribution with rational Laplace transform
- to approximate any general distribution with known bounds

PH-distributions *(phase-type)* have similar representation and property

Queueing disciplines

• scheduling algorithms

FCFS	first come first served
LCFS	last come first served
LCFSPr *	idem with pre-emption
Random	
Round Robin	each customer is served for a fixed quantum δ
PS *	Processor Sharing for $\delta \rightarrow 0$ all the customers are served at the same time for service rate μ and n customers, each receives service with rate μ /n
IS *	Infinite Serves no queue (*delay* queue)
SPTF	Shortest Processing Time First
SRPTF	Shortest Remaining Processing Time First

- with/without priority
- abstract priority/dependent on service time
- with/without pre-emption

**** Immediate service***

Queueing Networks

• A queueing *system* describes the system as a **unique** resource
• A queueing *network* describes the system as **a set** of interacting resources

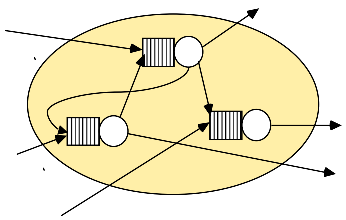
Queueing Network

a **collection** of service centers that provide service to a set of **customers**
- open external arrivals and departures
- closed constant number of customers (finite population)
- mixed if it is open for some types of customers, closed for other types

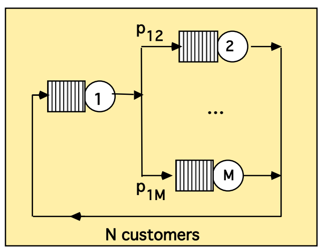
- Customers
  arrive to a service center (node) *(possibly external arrival for open QN)*
  ask for resource service
  possibly wait to be served (queueing discipline)
  at completion time exit the node and
  - immediately move to another node
  - or exit the QN
  in closed QN customers are always in queue or in service

Queueing Networks Examples

open QN



closed QN



Queueing Networks Definition

Informally, a QN is defined by
- the set of **service centers** $\Omega = \{1, \ldots, M\}$
- the set of **customers**
- the network **topology**

Each **service center** is defined by
- the number of servers
  - usually independent and identical servers
- the service rate
  - either constant or dependent on the station state
- the queueing discipline

Customers are described by
- their total number (closed QN)
- the arrival process to each service center (open QN)
- the service demand to each service center
  - service demand: expressed in units of service
  - service rate of each server: units of service / units of time
  - => service time = service demand/service rate
  - non-negative random variable with mean denoted by $1/\mu$

Queueing Networks Definition

The **network topology**

models the customer behavior among the interconnected service centers

- assume a non-deterministic behavior represented by a probabilistic model

- p_{ij} probability that a customer completing its service in station i immediately moves to station j, $1 \le i,j \le M$

- p_{i0} for open QN probability that a customer completing its service in station i immediately exits the network from station i

- **P** = [p_{ij}], **routing probability** matrix $1 \le i,j \le M$
  where $0 \le p_{ij} \le 1$, $\sum_j p_{ij} = 1$ for each station i

A QN is **well-formed** if it has a well-defined long-term customer behavior:

- for a closed QN if every station is reachable from any other with a non-zero probability
- for an open QN add a virtual station 0 that represents the external behavior, that generates external arrivals and absorbs all departing customers, so obtaining a closed QN. Definition as for closed QN.

Types of customers: classes, chains

In simple QN we often assume that all the customers are statistically identical

Modeling real systems can require to identify different types of customers -

- service time
- routing probabilities

Multiple types of customers: concepts of **class** and **chain**.

A **chain** forms a permanent categorization of customers

a customer belongs to the same chain during its whole activity in the network

A **class** is a temporary classification of customers

a customer can switch from a class to another during its activity in the network (usually with a probabilistic behavior)

The customer **service time** in each station and the **routing probabilities** usually **depend on the class** it belongs to

Multiple-class single-chain QN

Multiple-class and multiple-chain QN

$\mathcal{R}$	set of classes of the QN	R	number of classes
$\mathcal{C}$	set of chains	C	number of chains

Types of customers: classes, chains

R classes
C chains

Classes can be partitioned into chains, such that there cannot be a customer switch from classes belonging to different chains

$\mathbf{P}^{(c)}$ **routing probability matrix** of customers for each **chain** $c \in C$

- $p_{ir,js}^{(c)}$ probability that a customer completing its service in station i class r immediately moves to station j, class s, $1 \le i,j \le M$, r,s in $\mathcal{R}$, classes of chain *c*
- $p_{ir,0}^{(c)}$ probability that a customer completing its service in station i class r immediately exits the network

- $K^{(c)}$ population of a closed chain $c \in C$
- $p_{0,ir}^{(d)}$ probability of an external arrival to station i class r for an open chain $d \in C$

A QN is said to be

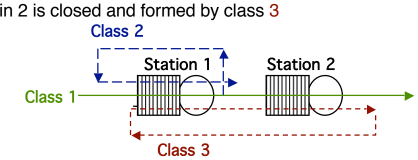
open	if all its chains are open
closed	if they are all closed
mixed	otherwise

Types of customers: classes, chains

Example of **multiple-class and multiple-chain QN**

M=2 stations
R=3 classes *C*=2 chains $\mathcal{R}$ ={1,2,3}

Chain 1 is open and formed by classes 1 and 2
Chain 2 is closed and formed by class 3



$\mathcal{R}_i$ set of classes served by station i
$E_c = \{(i,r): r \in \mathcal{R}_i \text{ set}, 1 \le i \le M, \text{class } r \in \text{chain } c\}$
$\mathcal{R}_i^{(c)}$ set of classes served by station i and belonging to chain c

$\mathcal{R}_1 = \{1,2,3\}$ $\mathcal{R}_2 = \{1,3\}$
$E_1 = \{(1,1), (1,2), (2,1)\}$ $E_2 = \{(1,3), (2,3)\}$
$\mathcal{R}_1^{(1)} = \{1,2\}$ $\mathcal{R}_2^{(1)} = \{1\}$ $\mathcal{R}_1^{(2)} = \mathcal{R}_2^{(2)} = \{3\}$

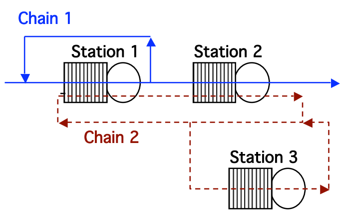
Types of customers: classes, chains

Example of **single-class and multiple-chain QN** *R=C*

(only one class in each chain)

M=3 stations
R=2 classes *C*=2 chains $\mathcal{R}$ ={1,2}
no class switching

Chain 1 is open and formed by one class
Chain 2 is closed and formed by one class



QN performance indices

Local performance indices
Related to a single resource i (a service center)
average indices

U_i	utilization
X_i	throughput
N_i	mean queue length
R_i	mean response time

random variables

- n_i number of customers **in station i**
- n_{ir} number of customers **in station i and class r**
- $n_i^{(c)}$ number of customers **in station i and chain c**

t_i customer passage time through the resource

distribution of n_i

$\pi_i(n_i)$ at arbitrary times

Global performance indices
Related to the overall network
average indices

U	utilization
X	throughput
N	mean population (for open networks)
R	mean response time

passage time

Notation - QN

Network model parameters (single class, single chain)

- **M** number of nodes
- **K** number of customers (closed network)
- **P**=$[p_{ij}]$ routing matrix
- $\mathbf{e}_i$ **visit ratio** at node i, solution of **traffic equations**
- λ total arrival rate
- μ_i service rate of node i
- p_{0i} arrival probability at node i

$$e_i = \lambda\, p_{0i} + \Sigma_j\, e_j\, p_{ji} \qquad 1 \le i \le M$$

S = $(S_1,\ldots,S_M)$ system state

S_i node i state which includes n_i, $1 \le i \le M$

Example: M nodes, *R* classes and *C* chains , single-class multi-chain (*R*=*C*)

$\mathbf{n} = (\mathbf{n}_1,\ldots, \mathbf{n}_M)$ network state

$\mathbf{n}_i = (n_{i1},\ldots, n_{iR})$ station i state, $1 \le i \le M$

We can describe QN behavior by an **associated stochastic process**

Under exponential and independence assumption we can define an homogeneous continuous-time Markov process

Markovian QN

S network state

E set of all feasible states of the QN

Markovian network

the network behavior can be represented by a homogeneous continuous time Markov process *M*

E discrete state space of the process

Q infinitesimal generator

if P (network routing matrix) irreducible

then ∃ ! stationary state distribution $\boldsymbol{\pi} = \{\pi(\mathbf{S}),\ \mathbf{S} \in \mathbf{E}\}$

solution of the **global balance equations**

$$\boldsymbol{\pi}\,\mathbf{Q} = \mathbf{0}\ , \quad \Sigma_{\mathbf{S}\in\mathbf{E}}\ \pi(\mathbf{S}) = 1$$

- the **definition of S**, **E** and **Q** depends on
  - the network characteristics
  - the network parameters

Exact analysis of Markovian QN

Solution algorithm for the evaluation of average performance indices and joint **queue length distribution** at arbitrary times (π) in Markovian QN

1. Definition of **system state** and state space E
2. Definition of **transition rate matrix** Q
3. **Solution** of global balance equations to **derive π**
4. **Computation** from π of the average **performance indices**

This method becomes unfeasible as |E| grows,
i.e., proportionally to the dimension of the model
(number of customers, nodes and chains)
Example: single class closed QN with M nodes and K customers $|E|=\binom{M+K-1}{K}$

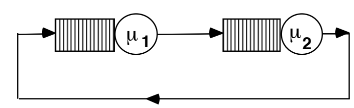
⇒ **exact product-form solution** under special constraints

⇒ **approximate solution** methods

Example of Queueing Network: two-node cyclic

closed network

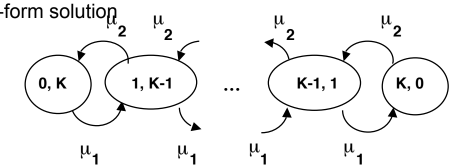
FCFS service discipline
exponential service time
Independent service time



$S = (S_1,S_2)$ system state
$S_i = n_i$

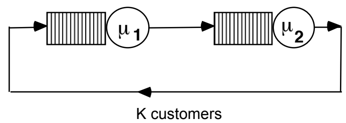
K customers

birth-death Markov process
closed-form solution



Example of Queueing Network: two-node cyclic

closed network



Let $\rho = (\mu_1/\mu_2)$

$\boxed{\pi(n_1, n_2)=(1/G)\, \rho^{n2}}$ $0 \le n_1 \le K$, $n_2=K-n_1$

closed-form solution

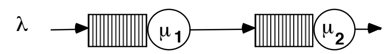
$G = \Sigma_{0 \le k \le K}\ \rho^{k} = (1-\rho^{K+1}) / (1-\rho)$

$\pi(K-k,k)= \pi(K,0)\, \rho^{k}$ $0 \le k \le K$

$\pi(K,0)= (1-\rho) / (1-\rho^{K+1})$

Example of Queueing Network: tandem

open network



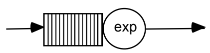
Arrival Poisson process
Exponential service times
Independence assumption
FCFS discipline
$S = (S_1,S_2)$ system state
$S_i = n_i$
E $= \{ (n_1, n_2) \mid n_i \ge 0\}$ state space
$\pi(n_1, n_2)$ stationary state probability

NON birth-death Markov process - complex structure - global balance equations

BUT node 1 can be analyzed independently
=> it is an M/M/1 system with parameters λ and μ_1
=> $\pi_1(k)= \rho_1^{k} (1-\rho_1)$ $k \ge 0$ if $\rho_1 = \lambda / \mu_1 < 1$

node 2 ?
arrival process at node 2?

Burke's Theorem

Poisson(λ)  Poisson(λ)

> The departure process of a stable M/M/1 is a Poisson process with the same parameters as the arrival process

Burke theorem's also holds for M/M/m and M/G/∞

For the tandem two node network:

⇒Node 2 has a Poisson arrival process (λ)

⇒Isolated node 2 is an M/M/1 system with parameters λ and μ_2

⇒$\pi_2(k) = \rho_2^k (1-\rho_2)$ $k \geq 0$ if $\rho_2 = \lambda / \mu_2 < 1$

Moreover for the independence assumption

$$\boxed{\pi(n_1, n_2) = \pi_1(n_1)\, \pi_2(n_2) = \rho_1^{n_1} \rho_2^{n_2} (1-\rho_1)(1-\rho_2)}$$

closed-form solution

That satisfies the process global balance equation **$\pi Q=0$**

Some extensions

An immediate application of

Burke theorem's together with

the property of **composition and decomposition** of **Poisson** processes

leads to a closed form solution of the state probability for a class of QN with

- exponential service time distribution
- FCFS discipline
- exponential interarrival time (Poisson arrivals, parameter γ_i)
- independence assumption
- acyclic probabilistic routing topology (triangular routing matrix P)

$$\pi(n_1, n_2, \ldots, n_M) = \prod_{i=1}^{M} \mathrm{Prob}_i\{n_i\}$$

where $\mathrm{Prob}_i(k) = \rho_i^k (1-\rho_i)$ $k \geq 0$ if $\rho_i = \lambda_i / \mu_i < 1$

and $\lambda_i = \gamma_i + \Sigma_j \lambda_j p_{ji}$ $0 \leq i \leq M$

(traffic equations)

Note: Burke's theorem does not hold when feedback is introduced, but…

Product-form Queueing Networks

product-form solution of π (under certain constraints)

$$\pi(\mathbf{S}) = \frac{1}{G} d(n) \prod_{i=1}^{M} g_i(n_i)$$

The stationary state probability **π** can be computed as the product of a set of functions each dependent only on the state of a station

Other average performance indices can be derived by state probability **π**

Jackson theorem open exponential-FCFS networks

Gordon-Newell theorem closed exponential-FCFS networks

BCMP theorem open, closed, mixed QN with various types of nodes

The solution is obtained **as if**

the QN is formed by independent M/M/1 (or M/M/m) nodes

Computationally efficient exact solution algorithms
- Convolution Algorithm
- Mean Value Analysis

BCMP Queueing Networks

Types of node

1) FCFS and exponential chain independent service time
2) PS
3) IS
4) LCFCPr

(2–4) and Coxian service time

For types 2-4 the service rate may also depend on the customer chain.
Let $\mu_i^{(c)}$ denote the service rate for node i and chain c.
=> $\mu_i^{(c)} = \mu_i$ for each chain c, for type-1 nodes.

Consider single-class multiple-chain QN

Consider open, closed, mixed QN with M nodes of types 1-4,
Poisson arrivals with parameter $\lambda(n)$ dependent on the overall QN population n,
R classes and *C* chains, population $K^{(c)}$ for each closed chain $c \in C$,
external arrival probabilities $p_{0,i}^{(c)}$ for each open chain $c \in C$,
routing probability matrices $\mathbf{P}^{(c)}$ for each chain $c \in C$,
that define the **traffic equation system** derive the **visit ratio** of (relative) throughputs $e_i^{(c)}$

$$e_i^{(c)} = p_{0,i}^{(c)} + \Sigma_j e_j^{(c)} p_{ji} \qquad 1 \le i \le M,\ 1 \le c \le C$$

BCMP Queueing Networks

BCMP theorem [Baskett, Chandy, Muntz, Palacios 1975]

For open, closed, mixed QN with M nodes of types 1-4 and the assumptions above, let $\rho_i^{(c)} = e_i^{(c)} / \mu_i^{(c)}$ for each node i and each chain c.

If the system is stable, i.e., if $\rho_i^{(c)} < 1\ \forall i, \forall c$,

then the steady state probability can be computed as the product-form:

$$\boxed{\pi(\mathbf{S}) = \frac{1}{G}\, d(n) \prod_{i=1}^{M} g_i(n_i)}$$

where G is a normalizing constant,

function $d(n)=1$ for closed network, and for open and mixed network depends on the arrival functions as follows:

$$d(\mathbf{n}) = \prod_{k=0}^{n-1} \lambda(k) \qquad d(\mathbf{n}) = \prod_{c=1}^{C} \prod_{k=0}^{n^{(c)}-1} \lambda_c(k)$$

for arrival rates dependent on the number of customers in the network n, or in the network and chain c,

functions $g_i(n_i)$ depend on node type as follows:

BCMP Queueing Networks

functions $g_i(n_i)$ depend only on

node i parameters $e_i^{(c)}$ and $\mu_i^{(c)}$ and

node i state $n_i^{(c)}$

$$g_i(n_i) = n_i! \prod_{c=1}^{C} \frac{(\rho_i^{(c)})^{n_i^{(c)}}}{n_i^{(c)}!}$$ for nodes of type 1, 2 and 4 (FCFS, PS and LCFSPr)

$$g_i(n_i) = \prod_{c=1}^{C} \frac{(\rho_i^{(c)})^{n_i^{(c)}}}{n_i^{(c)}!}$$ for nodes of type 3 (IS)

Where $n_i^{(c)}$ number of customers in node i and chain c

n_i number of customers in node i

The proof is based on the a detailed definition of the network state and by substitution of the product-form expression into the global balance equations of the associated continuous-time Markov process.

BCMP Queueing Networks - extensions

The service rate of node i may depend on the

A) the state of the node n_i
let $x_i(n_i)$ a positive functions (*capacity function*)
that gives the relative service rate ($x_i(1)=1$)
and the actual service rate for class r customers at node i is
$x_i(n_i)\,\mu_{ir}$

=> function $g_i(n_i)$ in the product-form is multiplied by factor

$$\prod_{a=1}^{n_i} (1/\, x_i(a))$$

B) the state of the node in chain c $n_i^{(c)}$
let $y_i^{(c)}(n_i^{(c)})$ a positive functions defined similarly to x_i above
=> function $g_i(n_i)$ in the product-form is multiplied by factor

$$\prod_{c \in \mathcal{R}_i} \prod_{a=1}^{n_i} (1/\, y_i^{(c)}(a))$$

(not for type 1 nodes)

BCMP Queueing Networks - extensions

The service rate of node i may depend on the

C) the state of a subnetwork H $n_H = \Sigma_{h \in H}\, n_h$
where H is a subset of stations
let $z_H(n_H)$ a positive functions defined similarly to x_i above
relative service rate when $n_H=1$
=> the product of functions $\prod_{h \in H} g_h(n_h)$ in the product-form
is multiplied by factor

$$\prod_{a=1}^{n_H} (1/\, z_H(a))$$

Note that multiservers can be modeled by type-A and type-B functions

Example: PS or LCFSPr node with class dependent service rates and *m* servers can be modelled by

$$x_i(n_i) = \min\{m, n_i\}/\, n_i$$
$$y_i^{(c)}(n_i^{(c)}) = n_i^{(c)}$$

Further BCMP extensions include
- other serving disciplines
- special form of state-dependent routing
- special cases of blocking and finite capacity queues

BCMP QN - multi-class multi-chain

Consider **multi-class multiple-chain QN**

Customers can move within a chain with class switching
routing probability matrices $\mathbf{P}^{(c)} = [p_{ir,js}{}^{(c)}]$ for each chain $c \in \mathcal{C}$,
that define the **traffic equation system**
from which we derive the **visit ratio** of (relative) throughputs $e_{ir}{}^{(c)}$

$$e_{ir}{}^{(c)} = p_{0,ir}{}^{(c)} + \Sigma_j \Sigma_{s\in \mathcal{R}_i{}^{(c)}} e_{js}{}^{(c)} p_{ir,js}{}^{(c)} \qquad 1\le i\le M,\ s \in \mathcal{R}_i{}^{(c)}\ \ 1\le c\le C$$

Let $\rho_{ir}{}^{(c)} = e_{ir}{}^{(c)} / \mu_{ir}{}^{(c)}$
M nodes, *R* classes *C* chains, multi-class multi-chain (*R≠C*)

$\mathbf{n} = (\mathbf{n}_1,\dots, \mathbf{n}_M)$ network state
$\mathbf{n}_i = (\mathbf{n}_i{}^{(1)},\dots, \mathbf{n}_i{}^{(C)})$ station i state, $1\le i\le M$
$\mathbf{n}_i{}^{(c)}$ has components $n_{ir}{}^{(c)}$ for each class $r \in \mathcal{R}_i{}^{(c)}$

n_i number of customers **in station i**
$n_i{}^{(c)}$ number of customers **in station i and chain c**
$n_{ir}{}^{(c)}$ number of customers **in station i and class r of chain c**

BCMP QN - multi-class multi-chain

functions $g_i(n_i)$ depend only on
node i and class r parameters $e_{ir}{}^{(c)}$ and $\mu_{ir}{}^{(c)}$ and
node i and class r state $n_{ir}{}^{(c)}$

$$g_i(n_i) = n_i! \prod_{c=1}^{C} \prod_{r\in\mathcal{R}_i{}^{(c)}} \frac{(\rho_{ir}{}^{(c)})^{n_{ir}{}^{(c)}}}{n_{ir}{}^{(c)}!}$$ for nodes of type 1, 2 and 4 (FCFS, PS and LCFSPr)

$$g_i(n_i) = \prod_{c=1}^{C} \prod_{r\in\mathcal{R}_i{}^{(c)}} \frac{(\rho_{ir}{}^{(c)})^{n_{ir}{}^{(c)}}}{n_{ir}{}^{(c)}!}$$ for nodes of type 3 (IS)

Estensions:
the state dependent functions can be defined for each class r and node i
capacity function: $y_{ir}{}^{(c)}(n_{ii}{}^{(c)})$ for each class $r \in \mathcal{R}_i{}^{(c)}$ $1\le c\le C$
for node types 2,3 and 4

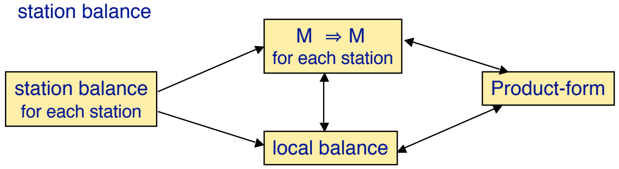
Product-form QN

Under some assumptions
(e.g., non-priority scheduling, infinite queue capacity, non-blocking factors, state-independent routing)
it is possible to give conditions on
- service time distributions
- queueing disciplines

to determine whether a well-formed QN yield a BCMP-like product-form solution

Properties strictly related to product-form

local balance
M $\Rightarrow$ M
quasi-reversibility
station balance



For non priority service centers

Product-form QN and properties

local balance

the effective rate at which the system i leaves state ξ due to a service completion of a chain r customer at station i	=	the effective rate at which the system enters state ξ due to an arrival of a chain r customer to station i

If **π** satisfies the local balance equations (LBEs) => it satisfies also the global balance equations (GBEs)
$\not\Leftarrow$

LBEs are a sufficient condition for network solution
Solving LBEs is computationally easier than solving GBEs
but it still requires to handle the set of reachable states
(can be a problem for open chains or networks)

LBE is a property of a station embedded in a QN, since the considered states are still network states

Note: a service center with work-conserving discipline and independent on service time and exponentlial service time holds LBE

Product-form QN and properties

M => M property

For a single queueing system:
an open queueing system holds M => M property if
under independent Poisson arrivals per class of customers,
then the departure processes are also independent Poisson processes

M ⇒ M property applies to the station in isolation

It can be used to decide whether a station (with given queueing discipline and service time distribution) can be embedded in a product-form QN

A station with M ⇒ M => the station has a product-form solution

An open QN where each station has the M ⇒ M => the QN has M ⇒ M

For a QN with stations with non-priority scheduling disciplines property

M ⇒ M for every station <=> local balance holds

Product-form QN and properties

quasi-reversibility property

if the queue length at a given time t is independent
of the arrival times of customers after t and
of the departure times of customer before t

then a queueing systems holds quasi-reversibility

A QN with quasi-reversible stations => QN has product-form solution

Quasi-reversibility property is defined for isolated stations

One can prove that
all the arrival streams to a quasi-reversible system should be independent and Poisson, and
all departure streams should be independent and Poisson

A system is quasi-reversible <=> it has M ⇒ M

Product-form QN and properties

Station balance

A scheduling discipline holds station balance property if
the service rates at which the customers in a position of the queue are served are proportional to
the probability that a customer enters this position

symmetric scheduling disciplines
p position in the queue $1 \le p \le n$
$\delta(p,n+1)$ probability that an arrival enters position p
$\mu(n)$ service rate
$\varphi_i(p,n)$ proportion of the service to position p
A symmetric discipline is such that: $\delta(p,n+1) = \varphi_i(p,n+1) \quad \forall p, \forall n$

examples:
IS, PS, LCFSPr are symmetric
but FCFS does not yields station balance
$\delta(p,n+1) = 1$ if $p=n+1$, 0 otherwise, $\varphi_i(p,n) = 1$, if $p=1$, 0 otherwise

station balance is defined for an isolated station

It is a sufficient condition for product-form

Product-form QN and properties

Insensitivity

For symmetric disciplines the QN steady state probabilities only depend on
the **average** of the service time distribution and
the (relative) visit ratio

State probabilities and average performance indices are independent of
- higher moments of the service time distribution
- possibly different routing matrices that yield the same (relative) visit ratios

Note:

only symmetric scheduling disciplines allow product-form solution for non-exponential service distribution

symmetric disciplines immediately start serving a customer at arrival time => they are always pre-emptive discipline

Product-form QN: further extensions

Special forms of **state-dependent routing**
depending on state
of the entire network or
of subnetworks and/or
single service centers

Special forms of QN with finite capacity queues and various **blocking** mechanisms
various types of blocking
constraints depending on blocking type, topology and types of stations

Batch arrivals and batch departures

Special disciplines
example: Multiple Servers with Concurrent Classes of Customers

G-networks: QN with positive and negative customers that can be used to represent special system behaviors
Negative customer arriving to a station reduces the total queue length by one if the queue length is positive and it has no effect otherwise. They do not receive service.
A customer moving can become either negative or remain positive
Exponential and independence assumptions, solution based on a set of non linear traffic eq.
Various extentions: e.g.,multi-class, reset-customers, triggered batch signal movement

Product-form QN: algorithms - single chain

Algorithms for **closed** QN with M stations and K customers (single chain)

Polynomial time computational complexity

Convolution
evaluation of the normalizing constant G and average performance indices

MVA
direct computation of average performance indices
(mean response time, throughput, mean queue length)
PASTA theorem *(arrival theorem)*

Convolution

based on a set of recursive equations, derivation of

- marginal queue length distribution $\pi_i(n_i)$
- mean queue length N_i
- mean response time R_i
- throughput X_i
- utilization U_i

time computational **complexity:** ***O(M K)***

Product-form QN: Convolution Algorithm

Direct and efficient computation of the normalizing constant G in product-form formula

Assume that stations

1,…, D have constant service rate IS discipline (type-3) (delay stations)

D + 1,…,D+I have load independet service rates (load-independent) (simple stations)

D + I + 1,…, D + I + L = M have load-dependent service rates

$G_j(k)$ normalizing constant for the QN considering a population of k customers and the first j nodes

$\mathbf{G}_j = (G_j(0)\ G_j(1) \ldots G_j(K))$

$$G = \sum_{\mathbf{n} \in E} \prod_{i=1}^{M} g_i(n_i)$$ Then $G = G_M(K)$

$$G_j(k) = \sum_{n=0}^{k} g_j(n)\, G_{j-1}(k-n)$$ **convolution** of vectors $\mathbf{G}_j$ and $(g_j(0) \ldots g_j(K)_j)$

Product-form QN: Convolution Algorithm

$G_j(0) = 1$ $1 \le j \le M$
$G_0(0) = 1$
$G_0(k) = 0$ $0 < k \le K$
$G_1(k) = g_1(k)$ $0 \le k \le K$

For the first D stations with IS disciplines we immediately obtain, for $0 \le k \le K$

$$G_D(k) = \left[\sum_{j=1}^{D} \rho_j \right]^k \frac{1}{k!}$$

For the I stations with load-independent service rate we can write

$$G_j(k) = G_{j-1}(k) + \rho_j\, G_j(k-1) \qquad 0 \le k \le K,\ D+1 \le j \le D+I$$

For the remaining stations with load-dependent service rate we apply convolution

Product-form QN: Convolution Algorithm

Performance indices

throughput

$$X_j(K) = e_j \frac{G_M(K-1)}{G_M(K)}$$

utilization

* $U_j(K) = 0$ 1≤j≤D, IS node
* $U_j(K) = X_j(K) / \mu_j m_j$ D+1≤j≤D+I, simple station, m_j servers
* $U_j(K) = \sum_{k=1}^{K} \min\{k, m_j\} \pi_j(k) / m_j$ D+I+1≤j≤M, load-dependent station

mean queue length

* $N_j(K) = X_j(K) / \mu_j$ 1≤j≤D, IS node
* $N_j(K) = \sum_{k=1}^{K} \rho_j^k \frac{G_M(K-k)}{G_M(K)}$ D+1≤j≤D+I, simple station
* $N_j(K) = \sum_{k=1}^{K} k\, \pi_j(k)$ D+I+1≤j≤M, load-dependent station

Product-form QN: Convolution Algorithm

Performance indices

queue length distribution

$$\pi_i(k) = g_i(n_i) \frac{G_{M-\{j\}}(K-k)}{G_M(K)}$$

$G_{M-\{j\}}(k)$ normalizing constant of the network obtained by the original network with station j removed

that simplifies as

$$\pi_j(k) = \rho_j^k \left\{ G_M(K-k) - \rho_j G_M(K-k-1) \right\} / G_M(K)$$

for D+1≤j≤D+I, simple station, m_j servers
with $G_M(n)=0$ if $n<0$

Potential numerical instability
- scaling techniques

Computational complexity

without load-dependent service rates *O(MK)*

with load-dependent service rates *$O(K+IK+L^2K^2)$*

Convolution Algorithm - multi-chain QN

Performance indices for each node i and chain c

$K^{(c)}$ population of a closed chain $c \in C$

$G_j(\mathbf{k})$ normalizing constant for the QN with a population of $\mathbf{k} = (k_1,\dots k_R)$ customers and the first j nodes

$\mathbf{G}_j = (G_j(\mathbf{0}) \dots G_j(\mathbf{K}))$

$$G_j(\mathbf{k}) = \sum_{\mathbf{n} \in E(j,\mathbf{k})} \prod_{i=1}^{j} g_i(\mathbf{n}_i)$$

$E(j,\mathbf{k})$ state space of the QN with j stations and $\mathbf{k}$ customers

Then $G = G_M(\mathbf{K})$

$$G_j(\mathbf{k}) = \sum_{\mathbf{n}=\mathbf{0}}^{\mathbf{k}} g_j(\mathbf{n})\, G_{j-1}(\mathbf{k}-\mathbf{n})$$ **convolution**

For the I stations with load-independent service rate we can write

$$G_j(\mathbf{k}) = G_{j-1}(\mathbf{k}) + \sum_{c=1}^{C} \rho_j^{(c)} G_{j-1}(\mathbf{K}-\mathbf{1}_c) \qquad \mathbf{0} \le \mathbf{k} \le \mathbf{K} \quad D+1 \le j \le D+I$$

Convolution Algorithm - multi-chain QN

For the first D stations with IS disciplines we immediately obtain, for $0 \le k \le K$

$$G_D(\mathbf{K}) = \left[\sum_{j=1}^{D} \rho_j^{(1)}\right]^{K^{(1)}} \dots \left[\sum_{j=1}^{D} \rho_j^{(c)}\right]^{K^{(c)}} \frac{1}{K^{(1)}! \dots K^{(c)}!}$$

Throughput $X_j^{(c)} = e_j^{(c)}\, G_M(\mathbf{K}-\mathbf{1}_c) / G_M(\mathbf{K})$

Utilization $U_j^{(c)} = \rho_j^{(c)}\, G_M(\mathbf{K}-\mathbf{1}_c) / G_M(\mathbf{K})$

Mean queue length $N_j^{(c)} = \sum_{1 \le a \le K^{(c)}} \sum_{\mathbf{n}:\, n^{(c)}=a} \mathrm{Prob}\{\mathbf{n}_i = \mathbf{n}\}$

Computational complexity

$H = \prod_{1 \le c \le C} (K^{(c)} + 1)$

an iteration step of Convolution

for a simple station requires $O(C\,H)$

for a load- dependent station requires $O(H^2)$

Special case: QN where all the chains have $K^{(c)}=K=K/C$, with load-independent stations, then $O(M\,C\,K^C)$

Product-form QN: MVA

MVA

- directly calculates the QN performance indices
- avoids the explicitly computation of the normalizing constant
- based on the arrival theorem and on Little's theorem

Arrival theorem

[Sevcik - Mitrani 1981; Reiser - Lavenberg 1980]

In a closed product-form QN
the steady state distribution of the number of customers at station i
at customer arrival times at i
is identical to
the steady state distribution of the number of customers at the same station
at an arbitrary time with that user removed from the QN

This leads to a recursive scheme

Assume:

1,…, D	constant service rate and IS discipline (type-3) (delay stations)
D + 1,…,D+I	load independent service rates (simple stations)
D + I + 1,…, D + I + L = M	load-dependent service rates

Product-form QN: MVA

1) Mean response time

$R_j(K) = 1 / \mu_j$ $\quad$ $1 \le j \le D$, IS node (delay node)

$R_j(K) = \frac{1}{\mu_j}(1 + N_j(K-1))$ $\quad$ $D+1 \le j \le D+I$, simple node (load-independent)

(Arrival theorem)

$R_j(K) = \sum_{n=1}^{K} \frac{n}{\mu_j(n)} \pi_j(n-1 \mid K-1)$ $\quad$ $K>0$

$D+I+1 \le j \le M$, load-dependent

2) Throughput

$X_j(K) = \frac{K}{\sum_{i=1}^{M} \frac{e_i}{e_j} R_i(K)}$ $\quad$ for each node j $\quad$ *(Little's theorem)*

3) Mean queue length

$N_j(K) = X_j(K)\, R_j(K)$ $\quad$ for each node j $\quad$ *(Little's theorem)*

Product-form QN: MVA

3) For load dependent stations
queue length distribution

$$\pi_j(n \mid K) = \frac{X_j(K)}{\mu_j(n)} \pi_j(n-1 \mid K-1) \qquad 1 \le n \le K,\ K > 1$$

$$\pi_j(0 \mid K) = 1 - \sum_{n=1}^{K} \pi_j(n \mid K)$$

Initial conditions:
$\pi_j(0|0) = 1$ for each node j
$N_j(0) = 0$

Potential numerical instability
MMVA modified MVA

$$\pi_j(0 \mid K) = \pi_j(0 \mid K-1) \frac{X_i(K)}{X_i^{M-\{j\}}(K)}$$

$X_i^{M-\{j\}}(K)$ throughput of any node i computed for the QN with node j removed

Product-form QN: MVA

Computational complexity

- For single-chain QN without load-dependent stations with K customers and M nodes, as Convolution *O(KM)*
- For single-chain QN with has only load-dependent stations *O(MK²)* (better than Convolution *O(M² K²)*)
- To overcome numerical instability MMVA has the same complexity as Convolution
- For multi-chain QN
  $H = \prod_{1 \le c \le C} (K^{(c)} + 1)$
  an iteration step of MVA
  for a simple station requires *O(C H)*
  for a load- dependent station requires *O(K C H)*
  (better than Convolution *O(H²)*)
- Special case: QN where all the chains have $K^{(c)}$=K=K/C , with load-independent stations, then as Convolution $O(M\, C\, K^{C})$
- MVA considers only type A capacity function, Convolution types A and B
- MVA is generalized to compute higher moments of performance measures

Product-form QN: algorithms for multi-chain

- Convolution
- MVA (Mean Value Analysis)
- Recal (Recursion by Chain Algorithm)
- DAC
- Tree convolution
- Tree-MVA

RECAL

- for networks with many customers classes but few stations

- main idea:
  recursive scheme is based on the formulation of the normalizing constant G for C chains as function of the normalizing constant for C − 1 chain

- if $K^{(c)} = \mathcal{K}$ for all the chains c, M and $\mathcal{K}$ constant, then for C → ∞ the time requirement is $O(C^{M+1})$

Product-form QN: algorithms for multi-chain

MVAC and DAC

- extends MVA with a recursive scheme on the chains
- direct computation of some performance parameters
- numerically robust, even for load-dependent stations
- possible numerical problems

Tree-MVA and Tree-Convolution

- for sparse network is sparse,
  (most of the chains visit just a small number of the QN stations)
- main idea: build a tree data structure where QN stations are leaves that are combined into subnetworks in order to obtain the full QN (the root of the tree)
- locality and network decomposition principle

For networks with class switching:
Note: it is possible to reduce a closed QN with C ergodic chains and class switching to an equivalent closed network with C chains without class switching

Approximate analysis of QN

Many approximation methods
Most of them do **not provide any bound** on the introduced error
Validation by comparison with exact solution or simulation

Basic principles

- **decomposition** applied to the Markov process
- **decomposition** applied to the network (aggregation theorem)
- **forced product-form** solution
- for multiple-chain models: approximate algorithms for product-form QN based on MVA
- exploit structural properties for special cases
- other approaches

Various accuracy and time computational complexity

Markov Process Decomposition

Markov process with state space **E** and transition matrix **Q**
- **Identify** a **partition** of **E** into **K subsets**

$$\mathbf{E} = \bigcup_{1 \le k \le K} \mathbf{E}_k$$

⇒ **decomposition of Q**
- **decomposition-aggregation procedure**

$$\boxed{\pi(\mathbf{S}) = Prob(\mathbf{S} \mid \mathbf{E}_k)\, \pi^a(\mathbf{E}_k)}$$

$Prob(\mathbf{S} \mid \mathbf{E}_k)$	conditional distribution
π^a	aggregated probabilities

- **computation of π(S) reduces to**
  the computation of $Prob(\mathbf{S} \mid \mathbf{E}_k)\ \forall\, \mathbf{S},\ \forall\, \mathbf{E}_k$
  the computation of π^a
- **exact computation soon becomes computationally intractable EXCEPT FOR special cases** (symmetrical networks)
- **approximation** of $Prob(\mathbf{S} \mid \mathbf{E}_k)$ and $Prob(\mathbf{E}_k)$

Process and Network Decomposition

Heuristics take into account the network model characteristics

NOTE: the identification of an appropriate state space partition affects
the algorithm **accuracy**
the **time** computational complexity

If the partition of E corresponds to a **NETWORK partition** into **subnetworks**
⇒ network decomposition subsystems are (possibly modified) subnetworks

The decomposition principle applied to QN
is based on the **aggregation theorem** for QN

1. network decomposition into a set of subnetworks
2. analysis of each subnetwork in isolation to define an aggregate component
3. definition and analysis of the new aggregated network

Exact aggregation (Norton's theorem) holds for product-form BCMP QN

Approximate analysis of QN

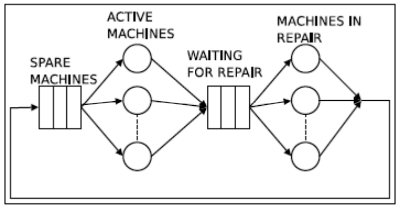
For multiple-chain QN
approximate algorithms for product-form QN based on MVA
- Bard and Schweitzer Approximation
- (SCAT) Self-Correcting Approximation Technique
generalized as the **Linearizer Algorithm**

Main idea:
approximate the MVA recursive scheme and
apply an approximate iterative scheme

Mean queue length
For population **K**, the MVA recursive equations require $N_j^{(c)}(\mathbf{K} - \mathbf{1}_d)$
for each chain d
Approximation: $N_j^{(c)}(\mathbf{K} - \mathbf{1}_d) = (|\mathbf{K} - \mathbf{1}_d|_c \ / \ K^{(c)}) \ N_j^{(c)}(\mathbf{K})$

where $|\mathbf{K} - \mathbf{1}_d|_c = K^{(c)}$ if $c \neq d$
$= K^{(c)} - 1$ if $c = d$

Example: machine repair model



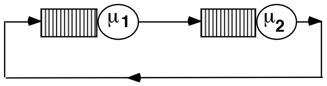
α identical machines which can achieve the same task with identical speed

They operate independently, in parallel and are subject to breakdown

At most $\beta \le \alpha$ of them can be operating simultaneously (active)
An active machine operates until failure
The active-time is a random time exponentially distributed with mean $1/\mu_1$

After a failure a machine waits for being repaired
At most γ machines can be in repair, and
the repair-time is a random time exponentially distributed with mean $1/\mu_2$

Example: machine repair model

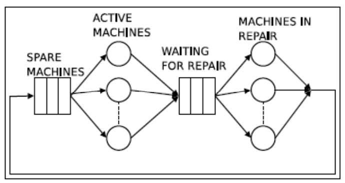


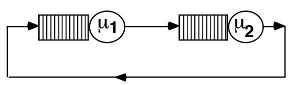
Model: single-chain single-class closed BCMP QN
With M = 2 stations, where
- station 1 represents the state of operative machines
- station 2 the machines in repair

K= α customers.
Visit ratios $e_1 = e_2$
Service rates μ_1 and μ_2
Multiple servers:
(β servers for station 1, γ servers for station 2)
BCMP representation with a single server with load-dependent service rate with capacity functions
$x_1(k) = \min\{k, \beta\}$, $x_2(k) = \min\{k, \gamma\}$

We can choose any BCMP-type discipline to compute product-form solution

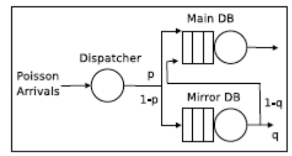
Example: machine repair model





Examples of performance measures
- steady state probability distribution $\pi(n_1, n_2)$
  (n_1 active machines and n_1 machines in waiting to be repaired)
- mean number of working machines, and mean number of machines in repair
  (N_1 and N_2)
- the utilization at station 1
  (U_1 ratio between the effective average work and the maximum work)
- mean time that a machine is broken (R_2)

Example: database mirror



A database (DB) repository system with two servers: master and a mirror

The arrival is dispatched to the primary or to the mirror
When a query is sent to the mirror:if the data are not found, then the mirror redirects the query to master

Design of the optimal dispatcher routing strategy
min system response time, given the DB average service times,
the cache hit probability for the slave database, and the arrival rate

Under some independence and exponential assumptions: BCMP QN

Example: database mirror

Model: open BCMP QN where a request is a customer

M = 3 stations
- station 1 represents the dispatcher
- station 2 the master
- station 3 the mirror

Assume Poisson arrivals of requests with rate λ

If we assume request independent routing
=> single chain QN, otherwise we should use a multi-chain model

A request can be fulfilled by the mirror with probability q and
it generates a new request to the master with probability 1−q

dispatcher -> delay station
DB stations -> with PS queueing discipline Coxian service time distribution
Visit ratios: $e_1=1$, $e_2=p+(1-p)(1-q)$, $e_3= 1-p$

Then by setting $\rho_i= \lambda e_i/\mu_i$ where μ_i is the mean service rate of station i, i=1,2,3
BCMP formulas if $\rho_i=<1$
Examples: evaluate average response time for each node i R_i
and average overall response time
$$R = R_1+e_2R_2+e_3R_3$$

Possible parametric analysis of response time R as function of probability p
to identify the optimal routing strategy

Open problems

Extension of QN

- special features of real systems
- models of classes of systems
  e.g.: software architectures, mobile systems,real time systems, …
  LQN, G-nets, …

Solution algorithms and product-form QN

- identify efficient algorithms to evaluate average performance measures
  consider the new classes of product-form QN

- explore/extend product-form class of QN

- identify efficient approximate and bound algorithms
  for non product-form QN to evaluate average performance measures

Properties of QN

- explore the relations with other classes of stochastic models
  - product-form classes
- hybrid modeling
  - explore the possible integration of various types of performance models