Queuing System Elements

**RESOURCES**
- Provide a service
- Permit only exclusive use
- Follow a service discipline
- Passive relative to transactions

**TRANSACTIONS**
- Demand a service
- Display persistent demand (queue)
- Follow a queue discipline
- Active relative to servers

**QUEUE**
- Holding area of transactions awaiting service
- Ordered list based on arrival and/or priority
- Characterized by a queue discipline
- Passive relative to the resource and the transactions
Informal Definitions

**QUEUING SYSTEM** - A system in which a finite service-providing resource is allocated to competing demands.

Queuing systems may be used to model the performance of:
- Transportation systems
- Manufacturing systems
- Communications systems
- Computer systems

**SERVICE** - A process of finite time duration that is provided by a resource

**QUEUE DISCIPLINE** - Rules governing the behavior of transactions who have not received service

**SERVICE DISCIPLINE** - Rules governing the behavior of the resource in allocating service
Queuing Systems Activities

PRESENTATION OF DEMAND
- Check status of resource

ENQUEUE
- Obey queue discipline

SEIZE RESOURCE
- Busy the resource
- Begin service

RELEASE RESOURCE
- End service
- Unbusy the resource
- Obey service discipline
  (Check contents of queue)
Queuing System Characteristics

- **QUEUING Systems** may be configured in a number of ways
  - **Single systems** (single server/ single queue)
  - **Multiple systems** (multi-server/ multi-queue)
  - **Hierarchical systems** (layered arrangements of servers and queues)

- **QUEUING System** can exhibit different behavior
  - **Simple systems** (Exponentially distributed service and interarrival times with infinite transaction populations)
  - **Complex system** (Service and interarrival times not exponentially distributed or finite transaction populations or with feedback loops)

- Expected performance of simple Queuing Systems may be readily determined using formal queuing theory

- Expected performance of more complex Queuing Systems may require computer simulation
Basic Measures Of Queue Performance

SYSTEM PERFORMANCE

\[ \text{SYSTEM PERFORMANCE} = F (\text{Arrival Rate, Service Rate, Discipline, Numbers of Servers}) \]

• TIME IN SYSTEM or QUEUE or RESOURCE
  - MINIMUM
  - MEAN
  - MAXIMUM

• NUMBER OF TRANSACTIONS IN SYSTEM or QUEUE or RESOURCE
  - MINIMUM
  - MEAN
  - MAXIMUM

• SYSTEM or QUEUE or RESOURCE UTILIZATION

• SYSTEM THROUGHPUT
Basic Queuing Systems

SINGLE QUEUING SYSTEMS

SINGLE SERVER / SINGLE QUEUE

MULTIPLE QUEUING SYSTEMS

MULTIPLE SERVER / SINGLE QUEUE

SINGLE SERVER / MULTIPLE QUEUE

MULTIPLE SERVER / MULTIPLE QUEUE
Systems of Multiple Queues

TASK:

SEQUENTIAL

PARALLEL

and

or

SELECTION

CONCURRENT

Queuing System

 TASK 1

 TASK 2

 TASK 1

 TASK 2

 TASK 1

 TASK 2

 TASK 1

 TASK 2

 TASK 1

 TASK 2

 TASK 1

 TASK 2
Introduction To Queuing Theory

**QUEUING THEORY** - Analytic technique for modeling the performance of queuing systems.

**BACKGROUND**

- Era of intuitive understanding
- Markov's definition of memory-less properties
- Erlang's 1917 work on telephone-traffic theory
- Extension to more complex systems

**UTILITY OF QUEUING THEORY**

- reliable method for forecasting the performance of simple queuing systems
- no need for computer support
- limited utility for forecasting the performance of complex queuing systems

**BASIC FACTORS CONSIDERED IN QUEUING THEORY**

- INTERARRIVAL TIME DISTRIBUTION
- SERVICE TIME DISTRIBUTION
- NUMBER OF SERVERS
- (SYSTEM CAPACITY)
- (QUEUE/SERVICE DISCIPLINE)
Resource Scheduling

• Resource scheduling is a key aspect of queue or service discipline
  
  • which transaction is selected for service next

• Resource scheduling also addresses
  
  • how are preempted transactions handled

Some typical resource scheduling algorithms:

• Static scheduling
  
  - FIRST-COME-FIRST-SERVED (FIFO)
  - LAST-COME-FIRST-SERVE (LIFO)
  - PRIORITIZED BASED ON FIXED CHARACTERISTICS

• Dynamic scheduling
  
  - SHORTEST JOB (SERVICE TIME) FIRST
  - TIME-VARIABLE PRIORITY
  - ROUND ROBIN

• System performance with each algorithm depends upon workload characteristics

• Intended service also affects algorithm quality
Queuing System Notation

Example designations:

**TIME DISTRIBUTIONS**

- $M = \text{MARKOVIAN}$
- $G = \text{GENERAL}$
- $D = \text{DETERMINISTIC}$

**DISCIPLINES**

- $FIFO = \text{FIRST IN FIRST OUT}$
- $LIFO = \text{LAST IN FIRST OUT}$
- $SIRO = \text{SERVICE IN RANDOM ORDER}$

The designation of a simple single-server/single-queue system with exponential distributions would be:

\[ M/M/1/\infty/(\text{FIFO}) \]

**ARRIVAL RATE:**  $\lambda = 1/\text{MEAN INTERARRIVAL TIME}$

**SERVICE RATE:**  $\mu = 1/\text{MEAN SERVICE TIME}$

**TRAFFIC INTENSITY** or **SYSTEM UTILIZATION FACTOR**

\[ \rho = \frac{\text{ARRIVAL RATE}}{\text{SERVICE RATE}} \]

**TRAFFIC INTENSITY** is the basic measure of system busyness.
Poisson Process

- Poisson Processes exhibit the Memory-less Markovian Property

- Results in a (Negative) Exponential Distribution or a Poisson Distribution

A Poisson arrival process results in a mean arrival rate of which has a Poisson distribution and a mean inter-arrival time of $1 / \lambda$ which is exponentially distributed.

- **Required assumptions**
  - arrivals drawn from an infinite population
  - the number of arrivals in non-overlapping time intervals is statistically independent
  - the number of arrivals in each time interval is statistically independent of service
  - the number of arrivals in each time interval is statistically independent of queue length or expected waiting time

- **Characteristics of the exponential distribution**
  - single parameter specification (the mean)
  - standard deviation = distribution mean (worst case perspective)
  - most accurate description of most human behavior in resource allocation systems at steady state
  - descriptive of any system whose behavior is independent of previous states
The Poisson Process

- Given a stream of events or states (transaction arrivals) characterized by
  - statistical independence
  
\[ P(\text{Event-time}_n) \text{ and } R(\text{Event-time}_m) = P(\text{Event-time}_n) \times P(\text{Event-time}_m) \]

  - inter-event times which are exponentially distributed

- Within a sufficiently small time interval \( \Delta t \)

\[
P(\text{events} = 0) = 1 - \text{event\_rate} \times \Delta t
\]

\[
P(\text{events} = 1) = \text{event\_rate} \times \Delta t
\]

\[
P(\text{events} > 1) = 0
\]

- Multiple Poisson processes can be combined into a single process with the same Poisson characteristics
  -- queuing systems can extended to include multiple transaction sources
  -- feeding a common queue
  -- independence of sources
  -- aggregate arrival rate = \( \sum \text{Source Arrival Rates} \)
  -- retain exponential distribution
Basic Measures Of M/M/1/∞ /FIFO Queuing System Performance

• Arrivals And Service Are Poisson Processes

• Assume Simple FIFO Queue/service Discipline
  - No Balking or Jockeying or Reneging

• Assume That The System Is In Steady State
  - Avoidance Of Opening And Closing Transients
  - System Is Operating At Statistical Equilibrium (Long-term Performance)

• Notation

Traffic = Traffic Intensity \( \rho \)
Arrival = Arrival Rate \( \lambda \)
Service = Service Rate \( \mu \)
Number = Number of Identical Servers \( C \)

• Expected Number Of Transactions In System

\[
L = \frac{\text{Traffic}}{1-\text{Traffic}} = \frac{\text{Arrival}}{\text{Service} - \text{Arrival}} = \frac{\lambda}{\mu - \lambda}
\]

• Expected Number Of Transactions In Queue

\[
L_q = \frac{\text{Traffic}^2}{1-\text{Traffic}} = \frac{\text{Arrival}^2}{\text{Service}(\text{Service} - \text{Arrival})} = \frac{\lambda^2}{\mu(\mu - \lambda)}
\]
Measures Of Queuing System Performance

• Expected Time In System

"Little's Formula" : \( L = \text{Arrival} \times \text{Wait} \)

\[
\text{Wait} = \frac{L}{\text{Arrival}} = \frac{1}{\text{Service} - \text{Arrival}} = \frac{1}{\mu - \lambda}
\]

• Expected Time In Queue

\[
\text{Wait}_q = \frac{\text{Arrival}}{\text{Service} (\text{Service} - \text{Arrival})} = \frac{\lambda}{\mu(\mu - \lambda)}
\]

"Little's Formula" for queues \( L_q = \text{Arrival} \times \text{Wait}_q \)

Expected System Utilization

\[
\text{UTILIZATION} = \frac{\text{Traffic}}{1} = \frac{\text{Arrival}}{\text{Service}} = \frac{\lambda}{\mu}
\]

• Expected System Throughput (Stable System)

\[
\text{THROUGHPUT} = \frac{\text{Number} \times \text{Utilization}}{1} = \frac{C \times \rho}{\mu}
\]

• Expected System Throughput (Unstable System)

\[
\text{THROUGHPUT} = \frac{\text{Number} \times \frac{1}{\text{Arrival}}}{\text{Utilization}} = \frac{C \times \frac{1}{\lambda}}{\rho}
\]
Throughput Of Queuing Systems

- **Throughput** Measures The Flow Capacity Of A System

- Difference Between Stable And Unstable Systems ("Saturated Vs. Unsaturated" Systems)

  **Throughput In Stable Systems** = Arrival Rate
  
  **Throughput In Unstable Systems** = Service Rate

Parameters Of **Throughput**

- Utilization Of Server: \( U \)
- Expected Service Time: \( E \) [x]
- Number Of Parallel Servers: \( C \)

Throughput With A Stable Single Server System

\[
\text{Throughput} = \frac{U}{E \ [X]}
\]

Throughput With A Stable Multi-server System

\[
\text{Throughput} = C \times \frac{U}{E \ [X]}
\]
Queuing System Performance

Expected Waiting Time

Traffic Intensity

Expected Number of Transactions

Traffic Intensity
Probability of Arrivals as a Function of Time to A M/M/1 System

\[
\frac{1}{\lambda} = 7 \text{ or } 9 \text{ time units}
\]

with discrete Poisson probability distribution

![Bar chart showing the probability of different numbers of arrivals in 1 time unit for mean values of 7 and 9 time units.](chart.png)
Probability of Arrivals as a Function of Time to a M/M/1 System

\[ \frac{1}{\lambda} = 15 \text{ or } 30 \text{ time units} \]

with discrete Poisson probability distribution
Simulated Queue Length as a Function of Time for a M/M/1 System

$\mu = 10$ time units

$\lambda = 7$ or $9$ time units
A Markov Chain is a sequence of events or states generated by a stochastic process each of which is dependant only on the previous state and the generating process.

- Given a sequence of random variables  \( X_0, X_1, X_2, \ldots \) which take on values in the set \( \{0,1,2,\ldots\} \)

  The sequence \( X \) is a Markov chain if for all \( N \) and all possible values of \( X_i \)

\[
P(X_n = x \mid X_0 = i_0, X_1 = i_1, X_2 = i_2, \ldots X_n = i_n) = P(X_n = j \mid X_{n-1} = i_{n-1})
\]

- The Markovian property describes arrivals and service in an idealized queuing system.

- A Markov chain is said to be in steady state if the probability of entering any state is equal to the probability of leaving that state.

- If \( X \) is an random variable that counts the number of transactions in a queuing system when at steady state

\[
P(X = x) = \mathbb{P}_x
\]
A Markov Chain

- **STATE N**: (N transactions in system)
  - **ARRIVAL PROCESS**: (Arrival rate = \( \lambda \) transactions / unit time)
  - **DEPARTURE PROCESS**: (Service rate = \( \mu \) transactions / unit time)
Derivation Of Steady State Equations Using Stochastic Balance

• Derivation Of The Steady State Equations
  Begin With The Expression Of Stochastic Balance

Stochastic Balance Exists Between The Inputs And Output Of A Given State

\[ \text{In}(S_n) = \text{Out}(S_n) \]

Into State \( N \):

\[ \Pr(S_n) = \lambda \Pr(S_{n-1}) + \mu \Pr(S_{n+1}) \]

Out of State \( N \):

\[ \Pr(S_n) = \lambda \Pr(S_n) + \mu \Pr(S_n) \]

In Steady State:

\[ \Pr(S_{n+1}) = \frac{\lambda + \mu}{\mu} \Pr(S_n) - \frac{\lambda}{\mu} \Pr(S_{n-1}) \]

• The Steady State of a System is related to its Traffic Intensity

\[ \Pr(S_1) = \frac{\lambda}{\mu} \Pr(S_0) \]
Derivation Of Steady State Equations Using Stochastic Balance

Steady State Equations prescribe the Probability that a system will be in a Given State

- **Forecasting the Probability that A System will be in a Given State**

\[ P_0'(t) = -\lambda P_0 + \mu P_1(t) \]

\[ P_n'(t) = -\lambda P_{n-1} - (\lambda + \mu)P_n(t) + \mu P_{n+1}(t) \quad \text{for } n \geq 1 \]

- **State Equations are More Readily Solved if the System is in Equilibrium; Time is no longer a Factor**

\[ \frac{dP_n(t)}{dt} = 0 \]

- **Steady State has been reached if the probability of finding N Transactions in the system approaches a Limiting Value**

\[ P_1 = \frac{\lambda}{\mu} P_0 \]

\[ P_{n+1} = \frac{(\lambda + \mu)}{\mu} P_n - \frac{\lambda}{\mu} P_{n-1}(t) \quad \text{for } n \geq 1 \]
Solution Of Steady State Equations

- Solution with Iterative Techniques involve inductive reasoning to solve for $P_n$

\[
P_n = \left( \frac{\lambda}{\mu} \right)^n P_0
\]

Assume: \[\rho = \frac{\lambda}{\mu} < 1\]

\[P_0 = (1 - \rho)\]

which is a necessary condition of Steady State,

Then

\[
P_n = \rho^n (1 - \rho) \text{ for } n \geq 0
\]

- Solution is also possible by obtaining probabilities with generating functions
Multiserver Queuing Systems
(M/M/C)

• Assume simple FIFO queue/service discipline

• Assume that the system is in steady state

• Assume identical servers, service rates, and service disciplines

• In stable systems (ρ < 1) the aggregate mean service rate is

  \[ c \times \text{mean service rate} \]

  if \( c \) or more transactions are in system and

  \[ k \times \text{mean service rate} \]

  where \( k \) (less than \( c \)) transactions are in system.

• The probability that \( k \) transactions are in the system is

  \[
P_k = \frac{\lambda^k}{k!\mu^k} P_0 \quad \text{where } 1 \leq k \leq C
  \]

  \[
P_k = \frac{\lambda^k}{C^{k-C}C!\mu^k} P_0 \quad \text{where } k > C
  \]
Multiserver Queuing Systems
(M/M/C)

- Calculating the Probability that Zero transactions are in system is based on the condition that 0 or more Transactions will certainly be in the system

\[
\sum_{k=0}^{\infty} P_k = 1
\]

- The probability that zero transactions are in the system is based on this certainty

\[
P_0 \left[ \sum_{k=0}^{C-1} \frac{\lambda^k}{k! \mu^k} + \sum_{k=C}^{\infty} \frac{\lambda^k}{C^k \cdot C! \mu^k} \right] = 1
\]

Solving for \( P_0 \)

\[
P_0 = \left[ \sum_{k=0}^{C-1} \frac{1}{k!} \left( \frac{\lambda}{\mu} \right)^k + \frac{1}{C!} \left( \frac{\lambda}{\mu} \right)^C \frac{C\mu}{C\mu-\lambda} \right]^{-1}
\]

- M/M/C Queuing Systems (C>1) are inherently more stable than M/M/1 queuing systems
  - greater capacity to serve
  - stable Traffic Intensities permissible
    (Traffic Intensity, \( \rho = \frac{\lambda}{C\mu} \))
  - queue storage capacity limitations become critical
  - service discipline
Basic Measures Of M/M/C Queuing System Performance

- As restrictions on M/M/1 queuing systems are removed, the analytical forecasting of their performance becomes more difficult

- Expected Number of transactions in System

\[ L = \frac{\lambda}{\mu} + \left[ \frac{(\frac{\lambda}{\mu})^c \lambda \mu}{(C-1)!(C \mu - \lambda)^2} \right] * P_0 \]

- Expected Number of Transactions in Queue

\[ L_q = \left[ \frac{(\frac{\lambda}{\mu})^c \lambda \mu}{(C-1)!(C \mu - \lambda)^2} \right] * P_0 \]

- Expected Time in System

\[ W = \frac{1}{\mu} + \left[ \frac{(\frac{\lambda}{\mu})^c \mu}{(C-1)!(C \mu - \lambda)^2} \right] * P_0 \]

- Expected Time in Queue

\[ W_q = \left[ \frac{(\frac{\lambda}{\mu})^c \mu}{(C-1)!(C \mu - \lambda)^2} \right] * P_0 \]
Limited Capacity Queuing Systems (M/M/1/MAX)

- Ultimately more realistic than queuing systems that permit unlimited occupancy

- Assume simple FIFO queue/service discipline

- Assume that the system is in Steady State

- For less than "MAX" number of transactions, the behavior of the system is the same as an M/M1/∞

- Max + 1 number of transactions is not allowed; further transactions are turned away

- A stable system will be achieved regardless of the traffic intensity, since the system is self-limiting (constrained by queue length)

- Performance measurement will involve consideration of the effective service rate
Solving The Steady State Equations For M/M/1/MAX Systems

Steady State Equations Are:

\[
P_1 = \frac{\lambda}{\mu} P_0 \\
P_{j+1} = \frac{\lambda + \mu}{\mu} P_j - \frac{\lambda}{\mu} P_{j-1}, \quad 1 \leq j \leq k - 1 \\
P_k = \frac{\lambda}{\mu} P_{k-1}
\]

Since the Probability of the System Being in a Given State Depends upon the Initial State and the Traffic Intensity :

\[
P_j = \rho^j P_0, \quad 1 \leq j \leq k
\]

\[
\sum_{j=0}^{k} P_j = 1 \quad \Rightarrow \quad \sum_{j=0}^{k} \rho^j P_0 = 1
\]

and \[
P_0 = \frac{1}{\sum_{j=0}^{k} \rho^j}
\]

The Summation of the System's Utilization at Each State Is :

\[
\sum_{j=0}^{k} \rho^j = 1 - \rho^{k+1}, \quad \rho \neq 1
\]

\[
\sum_{j=0}^{k} \rho^j = k + 1, \quad \rho = 1
\]
Limited Capacity Queuing Systems
(M/M/1/MAX)

• Probability of 0 Transactions In System
\[ P_0 = \begin{cases} \frac{1 - \rho}{1 - \rho^{k+1}}, & \rho \neq 1 \\ \frac{1}{k + 1}, & \rho = 1 \end{cases} \]

• Probability Of N Transactions In System
\[ P_n = \begin{cases} \frac{(1 - \rho)\rho^n}{1 - \rho^{k+1}}, & \rho \neq 1 \\ \frac{1}{k + 1}, & \rho = 1 \quad n = 0, 1, 2, \ldots, k \end{cases} \]
Basic Measures Of M/M/1/MAX Queuing System Performance

- Expected Number Of Transactions In System

\[ L = \frac{\rho [1 - (k + 1) \rho^k + k \rho^{k+1}]}{(1 - \rho^{k+1})(k - \rho)}, \quad \rho \neq 1 \]

\[ L = \frac{k}{2}, \quad \rho = 1 \]

- Expected Number Of Transactions In Queue

\[ L_q = L - (1 - P_0) \]

- Expected Time In System

\[ W = \frac{L}{\lambda'} = \frac{L}{\lambda(1 - P_k)} \]

The Effective Arrival Rate takes account the probability of saturation:

\[ \lambda' = \lambda(1 - P_k) \]

- Expected Time In Queue

\[ W_q = W - \frac{1}{\mu} = \frac{L_q}{\lambda'} = \frac{L_q}{\lambda(1 - P_k)} \]
Priority-based Queue Disciplines

• Non-FIFO Queue Disciplines often rely on a Prioritization scheme to tailor system performance

• Motivations for Non-FIFO Queue Disciples
  – reduce Average Cost Of Providing Service
  – reduce Average Time In Queue
  – reduce Average Number In Queue

• Priority-based Queue Disciplines Fall into two General Classes
  – Static Priority (Transactions Are Assigned
    • an Initial Priority Which Does Not Change)
  – Dynamic Priority (Transaction Priority
    • may Change During Its Time In The System)

• Two Types of Priority-based Queue Disciplines
  – Non-preemptive (Transactions Are Not Removed From Service)
  – Preemptive (Transactions Can Be Removed From Service By One With A Higher Priority)

• Priority-based Queue Disciplines must provide for tie-breaking between Transactions of equal priority