2.3 Birth and Death Processes

In the previous section we considered the Poisson process and saw that it can be used to describe the arrivals of service requests in many cases of great practical interest. In a practical queuing system, the request arrivals result in resource allocation and eventually the users get served and leave the queue. It is customary to view this process as a member of a wider class of stochastic processes that are commonly referred to as the *birth and death*. Within this framework, every incoming request is regarded as a birth and every user that, after being served, leaves the system is regarded as a death. For the Poisson process the average birth rate is specified by the distribution parameter $\lambda$. The birth rate can change as a function of the state of the queuing system. However, we can still say that in a short time interval $h$, the probability of a single birth is equal to $\lambda_n h + o(h)$, where subscript $n$ indicates one of the system states. Likewise, it is reasonable to assume that in a short time interval $h$, the number of users leaving the system is equal to $\mu_n h + o(h)$, where $\mu_n$ indicates the average death rate, and index $n$ references the state of the queuing system.

The birth and death process is frequently used as a mathematical model of a queuing system and in this section we provide its description. The framework of the birth and death process will allow us to derive some results that describe the behavior of the queuing systems in general.

The formal definition of the birth and death process is given as [1]:

**Definition 2.2.** Consider a stochastic process $N(t)$ that is continuous in time but has a discrete state space $\Omega = \{0, 1, 2, \ldots\}$. Suppose that this process describes a physical system that is in state $E_n$, $n = 0, 1, 2, \ldots$ at time $t$, if and only if $N(t) = n$. Then the system is described by the *birth-and-death* process if there exist nonnegative birth rates $\lambda_n, n = 0, 1, 2, \ldots$, and nonnegative death rates $\mu_n, n = 0, 1, 2, \ldots$, such that the following postulates (sometimes called nearest neighbor assumptions) are true:

1. State changes are only allowed between state $E_n$ to state $E_{n+1}$ or from state $E_n$ to $E_{n-1}$ if $n \geq 1$, but from state $E_0$ to state $E_1$ only.
2. If at time $t$ the system is in state $E_n$, the probability that between time $t$ and time $t+h$ a transition from state $E_n$ to state $E_{n+1}$ occurs equals $\lambda_n h + o(h)$, and the probability of transition from $E_n$ to $E_{n-1}$ is $\mu_n h + o(h)$ (if $n \geq 1$).
3. The probability that in time interval from $t$ to $t+h$ more than one transition occurs is $o(h)$.

Before we proceed, let us provide some examples of processes that can be classified as birth and death. As a first example, let us consider a stochastic process modeling the population in a closed system. The assumption that the system is closed is necessary to assure that the only mechanisms that the population can change are death and birth. Referring back to Definition 1 we can identify the following analogies:
1. State of the system $E_n$ represents the total population count at a given time $t$. One should note that although we can determine the population at an arbitrary time (that is the process is continuous in time), the actual values of $E_n$ can be only nonnegative integers, i.e. 0,1,2, ... etc. Therefore, the process is continuous in time, but desecrate in state space.

2. At any given state $E_n$, the population increases at the rate of $\lambda_n$ and decreases at the rate of $\mu_n$. Obviously, the population experiences growth if $\lambda_n > \mu_n$ and it is subject to decline if $\mu_n > \lambda_n$. In actual physical systems, no deaths or births can occur if the system is in state $E_0$. However, definition of the birth-and-death process allows for a nonzero birth rate even when the system is in state $E_0$.

3. Let’s assume that we count the population at times $t$ and $t+h$. If the time increment $h$ is kept small we expect the probability of birth to be given as the product of birth rate and the given time increment, i.e., $\lambda_n h$. Similarly, the probability of death is given as $\mu_n h$. Given that $h$ is infinitesimally small, the probability of both death and birth occurring within such a small time increment can be neglected, that is assumed as essentially zero.

4. As a final note, we point out that in general, birth and death rates are a function of the current population count. In other words, if the population grows, both the rate of birth and the rate of death can be expected to grow. Likewise if the population plummets, the rate of birth and the rate of death decrease as well. However, if the population is very large, the impact of the actual population count on the birth and death rates becomes smaller. In a boundary case for infinite population we would expect the rates of death and birth to remain constant.

As a second example, let us examine the modeling of traffic served in a cell of a cellular communication system.

1. The state of system $E_n$ represents the total number of users that are being served by a given cell. Unlike the previous example in which the set of possible states encompasses all positive integers, the possible states in this case are limited by the number of available resources at the cell site. In other words, $E_n \in \{0,1,2,\ldots,C\}$, where $C$ is the number of trunks (that is, voice channels) that are available at the site.

2. The process of birth is analogous to a new user trying to set up a call. Therefore, the birth rate $\lambda_n$ gives the rate at which the users request the service. In a similar way the death corresponds to a user that has completed the call and released the voice channel.

2.3.1 State Diagram Representation of Birth and Death Process

A useful visualization of the birth and death process is provided through the state transition rate diagram. An example of such a diagram is given in Fig. 2.8.
Figure 2.8. An example of the state diagram for birth and death process

The number inside the circle indicates the state of the system. For example, in a cellular system this would be the number of users serviced by a given site. Values \( \lambda_i \) indicate the birth rates at each of the system states. Similarly, values \( \mu_i \) represent the death rates. The state diagram allows only "the nearest neighbor" transitions and only the birth transition is allowed from state zero.

State diagram representation of the birth and death process will be frequently used for analyses presented in subsequent sections. For that reason, we derive differential-difference equations for \( P_n(t) = P_r\{N(t) = n\} \), that is, the probability that the system is in state \( E_n \) at time \( t \). Note that the derivation presented here is generalized, and as such, it is valid for any system that can be described using the birth and death processes.

If \( n \geq 1 \), the probability \( P_n(t+h) \) that at the time \( t+h \) the system will be in the state \( E_n \) has four components listed as follows:

1. **The system was in state \( E_n \) at time \( t \) and no births or deaths have occurred.** Knowing that the probability of birth is \( \lambda_n h + o(h) \) and the probability of death is \( \mu_n h + o(h) \), this component can be expressed as:

   \[
P_n^{(1)}(t+h) = P_n(t)[1 - \lambda_n h + o(h)][1 - \mu_n h + o(h)] = P_n(t)(1 - \lambda_n h - \mu_n h) + o(h)
   \]

   (2.9)

2. **The system was in state \( E_{n-1} \) at time \( t \) and a birth has occurred.** The probability of this event is given as:

   \[
P_n^{(2)}(t+h) = P_{n-1}(t)\lambda_{n-1} h + o(h)
   \]

   (2.10)

3. **The system was in state \( E_{n+1} \) and a death has occurred.** The probability of this event is given as:

   \[
P_n^{(3)}(t+h) = P_{n+1}(t)\mu_{n+1} h + o(h)
   \]

   (2.11)

4. **Two or more transitions have occurred.** By the properties of the birth and death process stated in Definition 2, this probability is:
\[ P_n^{(t)}(t+h) = o(h) \]  \hspace{1cm} (2.12)

From (2.9) through (2.12) we have:

\[ P_n(t+h) = \sum_{i=1}^{A} P_n^{(i)} = \left[ 1 - \lambda_n h - \mu_n h \right] P_n(t) + \lambda_{n-1} h P_{n-1}(t) + \mu_{n+1} h P_{n+1}(t) + o(h) \]  \hspace{1cm} (2.13)

or

\[ \frac{P_n(t+1) - P_n(t)}{h} = -(\lambda_n + \mu_n) P_n(t) + \lambda_{n-1} P_{n-1}(t) + \mu_{n+1} P_{n+1}(t) + \frac{o(h)}{h} \]  \hspace{1cm} (2.14)

By letting \( h \to 0 \), (2.14) reduces to:

\[ \frac{dP_n(t)}{dt} = -(\lambda_n + \mu_n) P_n(t) + \lambda_{n-1} P_{n-1}(t) + \mu_{n+1} P_{n+1} \]  \hspace{1cm} (2.15)

Equation (2.15) is valid for \( n \geq 1 \). For \( n = 0 \), following the same procedure one obtains:

\[ \frac{dP_0(t)}{dt} = -\lambda_0 P_0(t) + \mu_i P_i(t) \]  \hspace{1cm} (2.16)

If the initial state of the system is \( E_i \), then initial conditions are given as:

\[ P_i(0) = 1 \text{, and } P_j(0) = 0 \text{, for } j \neq i \]  \hspace{1cm} (2.17)

From (2.15) and (2.16), we see that the birth and death process can be described using an infinite set of differential equations, with initial conditions given in (2.17). Although it can be proven that the solution of these equations exists under very general circumstances [1], it can be rarely obtained in an analytical form.

The steady state solution of (2.16) and (2.17) are of a special practical interest. The steady state solution assumes that a sufficient time has elapsed and that the system has reached statistical equilibrium. In a steady state, all system state probabilities (\( P_n(t) \) values), become constant and hence the derivatives on the left-hand sides of (2.15) and (2.16) are equal to zero. Therefore, under the steady state assumptions

\[ 0 = \lambda_{n-1} P_{n-1} + \mu_{n+1} P_{n+1} - (\lambda_n + \mu_n) P_n \text{, for } n \geq 1 \]  \hspace{1cm} (2.18)

and

\[ 0 = \mu_i p_i - \lambda_0 p_0 \text{, for } n = 0 \]  \hspace{1cm} (2.19)

Equation (2.19) can be rewritten as
\[ p_i = \frac{\lambda_0}{\mu_1} p_0 \]  

(2.20)

Also, (2.18) can be rearranged in the form

\[ \mu_{n+1} p_{n+1} - \lambda_n p_n = \mu_n p_n - \lambda_{n-1} p_{n-1} \]  

(2.21)

Since (2.21) is valid for every \( n \), using (2.20) we can conclude that

\[ p_{n+1} = \frac{\lambda_n}{\mu_{n+1}} p_n \]  

for \( n = 0, 1, 2, \ldots \)  

(2.22)

Using (2.22) we can compute

\[ p_1 = C_1 p_0 = \frac{\lambda_0}{\mu_1} p_0 \]  

(2.23)

\[ p_2 = C_2 p_0 = \frac{\lambda_1}{\mu_2} p_1 = \frac{\lambda_1 \lambda_0}{\mu_1 \mu_1} p_0 \]  

(2.24)

\[ p_3 = C_3 p_0 = \frac{\lambda_2}{\mu_3} p_2 = \frac{\lambda_2 \lambda_1 \lambda_0}{\mu_3 \mu_2 \mu_1} p_0 \]  

(2.25)

In general, we have

\[ p_n = C_n p_0 = \frac{\lambda_{n-1} \lambda_n \cdots \lambda_0}{\mu_n \mu_{n-1} \cdots \mu_1} p_0 \]  

(2.26)

Since the sum of all state probabilities has to be equal to 1,

\[ p_0 \left( 1 + \frac{\lambda_0}{\mu_1} + \frac{\lambda_1 \lambda_0}{\mu_2 \mu_1} + \cdots + \frac{\lambda_{n-1} \lambda_{n-1} \cdots \lambda_0}{\mu_n \mu_{n-1} \cdots \mu_1} + \cdots \right) = p_0 S = 1 \]  

(2.27)

Finally, as a summary, we have

\[ p_0 = P_e \{ N(t) = 0 \} = \frac{1}{S}, \]  

(2.28)

and

\[ p_n = P_e \{ N(t) = n \} = \frac{C_n}{S} \]  

(2.29)

where

\[ C_n = \frac{\lambda_0 \lambda_1 \cdots \lambda_{n-1}}{\mu_1 \mu_2 \cdots \mu_n}, \text{ and } S = 1 + C_1 + C_2 + \cdots + C_n + \cdots \]  

(2.30)
From (2.28) through (2.30) we see that the birth and death process has a steady state solution if the sum $S$ converges. In such a case, there is a finite probability of a system occupying state zero. This would mean that from time to time the system “catches up” and manages to serve all users. On the other hand, if $S$ diverges, this indicates of an unstable system in which births are occurring at faster rates than deaths. For practical applications of the birth and death processes, we will assume that the system is not unstable, that a steady state exists, and that the state probabilities are constant and given by (2.29).

### 2.3.2 Little's Formula

Little's formula is a simple but very important equation that applies to any system in equilibrium in which customers arrive, spend some time and then depart. The formula is given by

$$ L = \lambda W $$

where $L$ is the average number of customers in the system, $\lambda$ is the average rate of customer arrivals, and $W$ is the average time that customers spend in the system. The proof of (2.31) is relatively complex and is beyond the scope of this document. To get an intuitive understanding of Little's formula, consider a system with a single server and an infinite queue. If the average service time is $W$, the number of users that arrive while one user is being served is $\lambda W$. Since the resource is occupied, these users are placed in queue and the state of the system is described by (2.31). The most important aspect of (2.31) is its universal applicability, therefore it is used frequently throughout this document.

**Example 2.4.** As an illustration of a birth and death process, consider a queuing system having only one server. Assume that that the service request arrivals can be accurately modeled as a Poisson process with an average rate of $\lambda = 1\text{min}^{-1}$, and that the average time required to service one request is given by $W_s = 0.5\text{ min}$. Also assume an infinite queue capacity with a FIFO queuing discipline. This kind of queuing system can be used to model many practical "real life" scenarios. For example, it can be used to model the queue formed at the printer server, or the queue formed in a supermarket with only one cash register. Estimate the probability that exactly $n$ users are in the queue, an average number of users in the queuing system, and the average time that users spend in this queuing system.

First, we estimate the average death rate, that is, the average rate at which the users would be leaving the system providing that the server has no idle time. This rate is estimated as:

$$ \mu = \frac{1}{W_s} = \frac{1}{0.5} = 2\text{ min}^{-1} $$

(2.32)

Using (2.29) and (2.30) we have

$$ C_n = \left(\frac{\lambda}{\mu}\right)^n = \left(\frac{1}{2}\right)^n = \frac{1}{2^n} $$

(2.33)

and
\[ S = 1 + C_1 + C_2 + \cdots = 1 + \frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \cdots = \frac{1}{1 - 1/2} = 2 \]  
(2.34)

Therefore, the probability of having exactly \( n \) users within the queue is given by:

\[ p_n = \frac{C_n}{S} = \frac{1/2^n}{2} = \frac{1}{2^{n+1}} \]  
(2.35)

The average number of users in the system can be calculated as

\[ L = \sum_{n=0}^{\infty} n \cdot p_n = 0 \cdot \frac{1}{2} + 1 \cdot \frac{1}{4} + 2 \cdot \frac{1}{8} + \cdots + n \cdot \frac{1}{2^{n+1}} + \cdots \]  
(2.36)

Multiplying both sides in (2.36) with \( 1/2 \) we obtain

\[ \frac{1}{2} L = 0 \cdot \frac{1}{4} + 1 \cdot \frac{1}{8} + 2 \cdot \frac{1}{16} + \cdots + (n-1) \cdot \frac{1}{2^{n+1}} + \cdots \]  
(2.37)

Subtracting (2.37) from (2.38)

\[ \frac{1}{2} L = \frac{1}{4} + \frac{1}{8} + \cdots + \frac{1}{2^{n+1}} + \cdots = \frac{1}{4} \left( 1 + \frac{1}{2} + \frac{1}{2^2} + \cdots \right) = \frac{1}{4} \cdot \frac{1}{1 - 1/2} = \frac{1}{2} \]  
(2.38)

Therefore the average number of users in the queuing system is given by

\[ L = 1 \]

The average time that users spend in the system can be calculated using Little's formula as

\[ W = \frac{L}{\lambda} = \frac{1}{1} = 1\text{ min} \]  
(2.39)

### 2.4 Kendall's Notation

Kendall's notation is frequently used for describing queuing systems of various properties. This is a shorthand notation in the following form

\[ A/B/C/K/m/Z \]

where the interpretation of individual terms is as follows:

- **A** - distribution of the interarrival times
- **B** - distribution of the service times
- **C** - number of servers within the service facility
- **K** - maximum number of users within the queuing system
Within Kendall's notation for the description of the arrival process and service times, the following symbols are used:

- GI - general independent arrival/service times
- G - general (not necessarily independent) arrival/service times
- $H_k$ - k-stage hyperexponential distribution
- $E_k$ - Erlang-k distribution
- M - exponential distribution (Poisson process)
- D - constant interarrival/service times
- U - uniform distribution

As an illustration, consider the queuing system described in Example 2.4. In Kendall's notation, this queue can be described as follows. Since the arrivals are modeled using the Poisson process $A = M$. Due to exponentially distributed service times $B = M$. Since there is only one server, $C = 1$. Both the queue and the population are of an infinite size and therefore $K = \infty$ and $m = \infty$. As the queuing discipline is First-In-First-Out, $Z = FIFO$. Therefore, Kendall's notation for the queuing system in Example 2.4 is $M/M/1/\infty/\infty/FIFO$. Very often, if the queue and population are infinite and the queuing service discipline is FIFO, the last three designators of the notation are omitted. In this example, the notation would reduce to $M/M/1$.

### 2.5 Examples

In this section we illustrate the application of the queuing theory in the analysis of some commonly encountered queuing systems. Two examples will be presented. The first example analyzes the problem of connecting two workstations to a central server. The second example shows the applicability of the queuing theory in the design of reliable microwave communication links.

**Example 2.5.** Consider a problem illustrated in Fig. 2.9.

Two work stations need to be connected to a single server and we examine two possible configurations that can be used to accomplish the task. In the first configuration, the connection is achieved by using two separate lines. The second configuration uses one line with a bandwidth that is twice as large. Let us assume that each workstation generates $\lambda$ messages per second and that the average for the message delivery is given as $1/\mu$ for the individual lines and $1/(2\mu)$ for the line with the larger bandwidth. Both configurations in Fig. 2.9 can be modeled using the theory developed in previous sections. We will examine some performance matrix as they are observed from individual workstations.
Figure 2.9. Two different configurations examined in Example 2.5

**Configuration 1.** In configuration 1, we essentially have two separate M/M/1 queuing systems with the same performance. Using the results of the birth and death process analysis (c.f. Section 2.3), the probability of having exactly $n$ messages in a transmission line (or associated buffer), is given by

$$p_n = \frac{C_n}{S}, \quad \text{(2.40)}$$

where

$$C_n = \left(\frac{\lambda}{\mu}\right)^n = \rho^n, \quad \rho = \frac{\lambda}{\mu} \quad \text{(2.41)}$$

and

$$S = 1 + \frac{\lambda}{\mu} + \left(\frac{\lambda}{\mu}\right)^2 + \left(\frac{\lambda}{\mu}\right)^3 + \cdots = \frac{1}{1 - \lambda/\mu} = \frac{1}{1 - \rho} \quad \text{(2.42)}$$

Therefore,

$$p_n = (1 - \rho)\rho^n \quad \text{(2.43)}$$

The average number of messages within each of the transmission lines is given by

$$\bar{n}_i = \sum_{n=0}^{\infty} np_n = 0 \cdot (1 - \rho)\rho^0 + 1 \cdot (1 - \rho)\rho + 2 \cdot (1 - \rho)\rho^2 + \cdots = \frac{\rho}{1 - \rho} \quad \text{(2.44)}$$

Using Little's formula, the average time required for the message delivery is given by
\[ W_1 = \frac{\bar{n}_1}{\lambda} = \frac{\rho}{(1 - \rho)\lambda} = \frac{1}{\mu - \lambda} \]  

(2.45)

Therefore, in the first configuration each of the workstations experiences an average throughput

\[ R_1 = \frac{1}{W_1} = \mu - \lambda \]  

(2.46)

**Configuration 2.** Configuration 2 can be seen as one $M/M/1$ queuing system with a birth rate of $2\lambda$ and a death rate of $2\mu$. Following the same approach as in the case of configuration 1, we obtain the following results

\[ p_n = (1 - \rho)^n = \left(1 - \frac{2\lambda}{2\mu}\right)^n = \left(1 - \frac{\lambda}{\mu}\right)^n \]  

(2.47)

\[ \bar{n}_2 = \frac{\rho}{1 - \rho} = \frac{\lambda/\mu}{1 - \lambda/\mu} \]  

(2.48)

\[ W_2 = \frac{\bar{n}_2}{2\lambda} = \frac{\lambda/\mu}{2\lambda(1 - \lambda/\mu)} = \frac{1}{2(\mu - \lambda)} \]  

(2.49)

and

\[ R_2 = \frac{1}{W_2} = 2(\mu - \lambda) = 2R_1 \]  

(2.50)

Therefore, the second configuration is twice as efficient as the first one.

**Example 2.6.** In this example we illustrate the impact of the link diversity on the reliability of a microwave connection. Consider a microwave link with a hot standby [4]. Let us assume that a mean time between a single link failure is given as $T_f$. When a link fails (either the main one or the hot standby), the mean time to repair is given by $T_r$. If we assume the same reliability of the main link and the hot standby, let us estimate the reliability improvement over a system without the link diversity.

The microwave link in this example can be modeled as a birth and death process with just three states and the state diagram shown in Fig. 2.10.

![State diagram for the microwave system in Example 2.6](image)

**Figure 2.10.** State diagram for the microwave system in Example 2.6
The state of the system corresponds to the number of non-working links. In other words, state 0 corresponds to the case when both the main link and its hot standby are operational; state 1 corresponds to the case when one of the links fails; and state 2 corresponds to failure of both the main link and the hot standby. The birth and death rates are indicated in Fig. 2.10, where

\[ \lambda = \frac{1}{T_f}, \quad \mu = \frac{1}{T_r} \]  

(2.51)

and

(2.52)

To calculate the mean time between the failure for the system with the link diversity we use the diagram in Fig. 2.10 to estimate the steady state rate at which the system reaches state 2. From Fig. 2.10, this rate can be calculated as

\[ \lambda_{2f} = \lambda \cdot p_1 \]  

(2.53)

where

\[ p_1 = \frac{C_1}{S} = \frac{2 \lambda/\mu}{1 + 2 \lambda/\mu} \]  

(2.54)

Therefore,

\[ \lambda_{2f} = \lambda \frac{2 \lambda/\mu}{1 + 2 \lambda/\mu} \]  

(2.55)

and the time between the failures becomes

\[ T_{2f} = \frac{1}{\lambda_{2f}} = \frac{1 + 2 \lambda/\mu}{2 \lambda^2/\mu} = \frac{1 + 2T_r/T_f}{2T_r/T_f^2} \]  

(2.56)

To illustrate the resulting improvement, let us consider the following numerical data. The average time between link failure is \( T_f = 4000 \) hours and the average repair time is \( T_r = 24 \) hours. When the link diversity is used, the average time between failures becomes

\[ T_{2f} = \frac{1 + 2 \cdot \frac{24}{4000}}{2 \cdot \frac{24}{4000^2}} = 337.333 \text{ [hours]} \]  

(2.57)

which is a significant improvement.
3 Traffic Planning for Circuit Switched Voice Services

A dominant type of traffic in cellular communication systems of the first and the second generation is voice traffic. In all cellular standards, voice service is implemented as a circuit switched service. As explained in Section 1, the circuit switched nature of the voice service implies that a network allocates a cell site communication resource for the entire duration of the call. Strictly speaking, in cellular communication this assumption does not hold. Due to handoff, the occupancy of the cell site resource is generally shorter than the call duration. However, in practical traffic dimensioning, the effect of the handoff is usually neglected. This simplification can be qualitatively justified as follows:

1. In the state of traffic equilibrium there is approximately the same number of calls leaving and entering the cell site coverage area. Therefore, the average traffic load is not modified by the handoff.
2. The overhead traffic that results from the handoff processing is usually handled by a separate group of communication resources referred to as the control channels. Therefore, the actual handoff processing does not increase the traffic volume on the voice channels.
3. The service time for the voice call is usually assumed to be distributed exponentially. Due to so-called "memoryless" property of the exponential distribution, the incoming calls can be thought of as the continuation of the outgoing calls. Although this is truly not the case, we may assume that all calls terminate in the same cell where they originate.

3.1 Common Descriptors of the Circuit Switched Voice Service

There are several common terms used in everyday traffic engineering of cellular systems. In this section we define and illustrate their typical use.

3.1.1 Busy Hour

The volume of voice traffic carried by a cellular network is continuously changing with respect to time. Most systems will experience a low level of usage during the afternoon and early evening hours with peak levels of usage during the morning rush hour, lunch hour, and evening rush hour (see Fig. 3.1). The uninterrupted period of sixty minutes during which the traffic is at a maximum is known as the busy hour. The amount of traffic experienced during the busy hour is generally used as the basis for traffic calculations and resource dimensioning. Shorter peaks (less than sixty minutes) of traffic usage can be experienced due to special events such as traffic jams, automobile accidents or poor weather. However, system dimensioning which would cater to these special events would not be economically viable.

The busy hour can be defined as either fixed or bouncing. The fixed busy hour is a set period of 60 minutes that does not change from day to day. When the fixed busy hour is used, the data collection process is limited to one hour per day and the requirements for storage memory space are reduced. The bouncing busy hour is determined for each day independently. To determine
the bouncing busy hour, the switch has to collect traffic data over the entire 24 hours. Although twenty four hour data collection results in relatively large volumes of measured data, most contemporary switches are capable of providing reports based on the bouncing busy hour.

![Graph of daily traffic usage](image)

**Figure 3.1.** Typical daily traffic usage in a cellular system

### 3.1.2 QoS parameters

In general, a definition of the quality of service depends largely on the type of communication service provided by the network. For the circuit switched voice, the most common indicator of the service quality is *probability of blocking*. The probability of blocking is defined as probability that a service request is denied due to all cell site trunks being occupied. In common engineering practice, the probability of blocking is commonly referred to as the *Grade Of Service* (GOS), and it can be formally defined as

\[
GOS = 1 - \frac{\text{Served Traffic}}{\text{Offered Traffic}}
\]

(3.1)

The GOS definition in (3.1) assumes no queuing. In other words, if the call attempt is made and all resources are occupied, the call is immediately rejected. This is analogous to the situation encountered in landline telephone networks when a user picks up the phone and gets a busy signal even before dialing the number. While in contemporary landline telephone networks this is a fairly rare event, the mobile communication networks are designed with a GOS target between 1 and 2%. In other words, we may expect that in a typical cellular network, 1 out of 50 calls may be blocked.
Some of the services provided in the circuit switched mode allow queuing of individual calls. For example, dispatch voice services as well as myriad of circuit switched data services may delay transmission until the channel becomes available. In such scenarios, GOS does not represent a valid measure of system quality and alternative indicators are used. The most common ones are probability of delay, probability of delay exceeding given limit, an average number of calls in the queue, maximum number of calls in the queue. Some of these parameters will be more thoroughly addressed in Section 3.2.2.

### 3.2 Trunking Models for Circuit Switched Services

In this section we will analyze two trunking models that are commonly used in traffic dimensioning of wireless cellular networks. The main difference between the two models is in the treatment of the service requests that occur while all servers are occupied. In the case of the Erlang B trunking model, requests that cannot find an available server are cleared from the system. This regime of operation is sometimes referred to as the "lost calls cleared" regime. The Erlang C model represents the other extreme. In the case of Erlang C, service requests made during time intervals when all servers are busy are placed in a queue of an infinite capacity. Therefore, no calls are lost, and this regime of operation is referred to as the "lost calls held" regime. There are numerous systems that can be closely modeled using either Erlang B or Erlang C formulas. However, in analyzing a particular traffic problem, an engineer needs to make sure that the assumptions of either Erlang B or Erlang C models are still valid.

#### 3.2.1 Erlang B Formula

The Erlang B formula is the most commonly used formula for the GOS calculation in a cellular system. The formula is applied on a per cell basis assuming that each cell site can be modeled as a queuing system of a type $M/M/C/C$. In other words, the assumptions used in derivation of the Erlang B formula are as follows:

1. Call arrival process is a Poisson process.
2. Service time (i.e., call holding time) is exponentially distributed.
3. There are $C$ identical servers within the service facility. For voice service this translates into $C$ voice channels available at the site.
4. There is no queuing in the system. Any call attempt that is made while all trunks are occupied is cleared from the system and is considered lost.

A commonly adopted QoS parameter in $M/M/C/C$ systems is the probability of call blocking or GOS. This is the probability that an incoming call will find all channels at the site occupied. The Erlang B loss formula is used for calculation of the GOS in $M/M/C/C$ systems.

A state transition diagram for a $M/M/C/C$ system is presented in Fig. 3.2.
Figure 3.2. State diagram for $M/M/C/C$ queuing system

From the state diagram we have

$$C_n = \frac{\lambda^n}{n! \mu^n} = \frac{a^n}{n!}$$ (3.2)

and

$$S = 1 + \frac{\lambda}{\mu} + \frac{\lambda^2}{2 \mu^2} + \frac{\lambda^3}{3! \mu^3} + \cdots + \frac{\lambda^n}{C! \mu^n} = \sum_{k=0}^{C} \frac{(\lambda/\mu)^k}{k!} = \sum_{k=0}^{C} \frac{a^k}{k!}$$ (3.3)

where $a = \lambda/\mu$ is the offered traffic. Therefore, the probability of finding exactly $n$ occupied trunks is given by

$$p_n = \frac{C_n}{S} = \frac{a^n/n!}{\sum_{k=0}^{C} \frac{a^k}{k!}}$$ (3.4)

An incoming call will be blocked (i.e., a call request will be denied), if all of the trunks are occupied. The probability of such an event is given by

$$B[a, C] = \frac{a^C/C!}{\sum_{k=0}^{C} \frac{a^k}{k!}}$$ (3.5)

Equation (3.5) is the Erlang B loss formula.

To illustrate the use of (3.5), consider the following example.

**Example 3.1.** A cell site has 5 AMPS radios. The average rate of call origination within the cell site coverage area is 60 calls per hour. If the call holding times are distributed exponentially with an average of 90 seconds, calculate the blocking probability.

The average birth rate is given by: $\lambda = 60$ calls/hour

The average death rate is given by: $\mu = \frac{1}{H} = \frac{1}{90/3600} = 40$ calls/hour
The offered traffic is: \[ a = \frac{\lambda}{\mu} = \frac{60}{40} = 1.5 \ E \]

Using (3.5), the blocking probability can be calculated as:

\[
B[a = 1.5, C = 5] = \frac{a^C/C!}{\sum_{k=0}^C \frac{a^C}{C!} + \frac{1.5^k}{k!}} \approx 0.0142
\]

To facilitate the use of the Erlang B formula in everyday engineering practice, (3.5) is frequently presented either in the form of parametric curves or in the form of a table. Erlang B curves are shown in Fig. 3.3, while Appendix A provides the Erlang B table.

**Figure 3.3.** Representation of the Erlang B blocking formula through a family of curves

To illustrate the use of the Erlang B table, consider the following examples.

**Example 3.2.** Determine the traffic capacity in erlangs for a 30-channel cell such that the GOS will not exceed a) 2% and b) 1% using the Erlang B table provided in Appendix A.

**Part a:** In Appendix A, find the row for \( N = 30 \). Find the intersection of this row with the column for 2% GOS. Read the corresponding traffic capacity to be 21.9 erlangs.
Part b): In Appendix A, find the row for N = 30. Find the intersection of this row with the column for 1% GOS. Read the corresponding traffic capacity to be 20.3 erlangs.

Example 3.3. Determine the number of voice channels required to support 20 erlangs (720 CCS) at a GOS of a) 2% and b) 1% using the Erlang B table provided.

Part a): In Appendix A, find the column for a GOS of 2% and follow this column down to a traffic capacity value of 20.2 erlangs. Follow this row to the left and read the number of voice channels to be 28.

Part b): In Appendix A, find the column for a GOS of 1% and follow this column down to a traffic capacity value of 20.3 erlangs. Follow this row to the left and read the number of voice channels to be 30.

Example 3.4. Determine the GOS that a 30 voice channel cell will provide with a) 28 erlangs of traffic and b) 20 erlangs of traffic using the Erlang B table provided.

Part a): In Appendix A, find the row for N = 30 and read across until a traffic capacity of 28.1 erlangs is found. Follow this column up and read the GOS to be 10%.

Part b): In Appendix A, find the row for N = 30 and read across until a traffic capacity of 20.3 erlangs is found. Follow this column up and read the GOS to be 1%.

Several performance indicators can be derived for the M/M/C/C queuing systems. The values for some of them, including the ones derived above, are provided in Table 3.1.

Table 3.1. Summary of the performance parameters for the M/M/C/C queuing system

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Symbol</th>
<th>Value (Formula)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offered traffic</td>
<td>$a$</td>
<td>$a = \lambda / \mu$</td>
</tr>
<tr>
<td>Probability of having no calls at the cell site</td>
<td>$p_0$</td>
<td>$p_0 = \frac{1}{1 + a + \frac{a^2}{2!} + \cdots + \frac{a^C}{C!}}$</td>
</tr>
<tr>
<td>Probability of having exactly $n$ calls at the cell site</td>
<td>$p_n$</td>
<td>$p_n = \frac{a^n}{n!} p_0$</td>
</tr>
<tr>
<td>Blocking probability</td>
<td>$B[a,C]$</td>
<td>$B[a,C] = \frac{a^C}{C!} p_0$</td>
</tr>
<tr>
<td>Average arrival rate of served calls</td>
<td>$\lambda_a$</td>
<td>$\lambda_a = \lambda (1 - B[a,C])$</td>
</tr>
<tr>
<td>Average number of active calls</td>
<td>$\bar{n}$</td>
<td>$\bar{n} = \frac{\lambda_a}{\mu}$</td>
</tr>
<tr>
<td>Average channel utilization</td>
<td>$\rho$</td>
<td>$\rho = \frac{\lambda_a}{\mu} \frac{1}{C}$</td>
</tr>
</tbody>
</table>
Example 3.5. Consider a GSM cell that implements GPRS service with 2 radios. Let us assume that on the voice side the average rate of the call origination is 300 calls/hour. Assuming that the average call holding time is 120s, calculate the average number of time slots available for the GPRS service.

The average birth rate: \( \lambda = 300 \) calls/hour

The average death rate (per time slot): \( \mu = \frac{1}{120/3600} = 30 \) calls/hour

The offered traffic is; \( a = \frac{\lambda}{\mu} = \frac{300}{30} = 10 \) erlangs

Assuming that one of the time slots is used as a control channel the number of time slots (i.e. trunks) available for the voice service is given as

\[ C = 2 \cdot 8 - 1 = 15 \]

The probability of blocking at this cell site is given by

\[ B[10,15] = \frac{10^{15}/15!}{\sum_{k=0}^{15}10^k/k!} = 0.0365 \]

Using Table 3.1, we find the average served traffic as

\[ a_s = \frac{\lambda}{\mu} = \frac{\lambda(1 - B[a,C])}{\mu} = \frac{300(1 - 0.0365)}{30} = 9.635 \text{ erlangs} \]

The average utilization of time slots can be calculated as

\[ \rho = \frac{a_s}{C} = \frac{\lambda}{\mu C} = \frac{9.635}{15} = 0.64 \]

Therefore for about \( 1 - \rho = 36\% \) of time, each time slot is available for the GPRS data traffic. Alternatively, if we assume negligible setup overhead for the data traffic, the average number of time slots available to service GPRS can be calculated as

\[ C_{GPRS} = (1 - \rho)C = 0.36 \cdot 15 = 5.4 \]

As a final remark we note that the Erlang B formula is derived under the assumption of exponentially distributed service times. However, it has been proven [1,4] that this formula remains valid for an arbitrary distribution of the service times as well.
3.2.2 Erlang C Formula

The queuing system used for the derivation of the Erlang C formula is of type $M/M/C$. In other words, we assume the following:

1. Service request arrivals can be modeled as a Poisson process
2. Call holding times follow exponential distribution
3. The system has $C$ identical servers
4. The queue is of an unlimited capacity

Unlike $M/M/C/C$ system which was used in the derivation of the Erlang B formula, $M/M/C$ does not reject any call request. Provided that the average birth rate is smaller than the average death rate, the system is stable and ultimately all requests will be served. For that reason, in $M/M/C$ systems, GOS provided by the Erlang B formula is inadequate and different performance indicators need to be examined. The type of the most relevant metric depends on the application. Some the most commonly used ones include

- Probability of service request delay
- Average delay for all requests
- Average delay for the requests that are placed in the queue
- 90% delay percentile
- The average number of requests in the queue, and so on.

![State transition model](image)

**Figure 3.4.** State transition model for a queuing system of type $M/M/C$

A system satisfying the assumptions of the Erlang C formula can be modeled as a birth and death process with the state transition diagram shown in Fig. 3.4. Using the results of the queuing theory (c.f. Section 2.3), we can write:

$$p_n = \frac{C_n}{S}$$  \hspace{1cm} (3.6)

where

$$C_n = \begin{cases} \frac{(\lambda/\mu)^n}{n!} = \frac{a^n}{n!} & , n \leq C \\
\frac{(\lambda/\mu)^n}{C!C^{n-C}} = \frac{a^n}{C!C^{n-C}} & , n > C \end{cases}$$ \hspace{1cm} (3.7)

and
RF Engineering Continuing Education
Introduction to Traffic Planning

\[ S = 1 + a + \frac{a^2}{2} + \cdots + \frac{a^c}{C!} + \frac{a^{c+1}}{C!(1-C)} + \cdots \]  \hspace{1cm} (3.8)

If we define a new variable \( \rho = a/c \), (3.8) can be rewritten as

\[ S = 1 + a + \frac{a^2}{2} + \cdots + \frac{a^{c-1}}{(C-1)!} + \frac{a^C}{C!(1-\rho)} + \sum_{k=0}^{C-1} \frac{a^k}{k!} \]  \hspace{1cm} (3.9)

Therefore,

\[ p_0 = \left[ \sum_{k=0}^{C-1} \frac{a^k}{k!} + \frac{a^C}{C!(1-\rho)} \right]^{-1} \]  \hspace{1cm} (3.10)

and

\[ p_n = \begin{cases} 
\frac{a^n}{n!} p_0 & , n \leq C \\
\frac{a^C}{C!} \rho^{n-C} p_0 & , n > C 
\end{cases} \]  \hspace{1cm} (3.11)

Using (3.11) we can derive some important performance metrics for \( M/M/C \) systems. The first one is the probability that an arriving service request is placed in queue, that is, the probability of a request being delayed. The delay probability can be calculated as

\[ \Pr(>0) = \Pr\{N \geq C\} = \sum_{n=C}^{\infty} p_n = 1 - \sum_{n=0}^{C-1} p_n \]  \hspace{1cm} (3.12)

\[ \Pr(>0) = 1 - \sum_{n=0}^{C-1} \frac{a^n}{n!} p_0 = 1 - \frac{\sum_{n=0}^{C-1} \frac{a^n}{n!} + \frac{a^C}{C!(1-\rho)}}{\sum_{n=0}^{\infty} \frac{a^n}{n!} + \frac{a^C}{C!(1-\rho)}} \]  \hspace{1cm} (3.13)

or

\[ \Pr(>0) = \frac{\frac{a^C}{C!(1-\rho)}}{\sum_{n=0}^{\infty} \frac{a^n}{n!} + \frac{a^C}{C!(1-\rho)}} \]  \hspace{1cm} (3.14)

Formula (3.14) is commonly referred to as the **Erlang C** delay formula.

Now, we compute the average number of call requests that are in the system queue.

\[ L_q = \sum_{n=C}^{\infty} (n-c) p_n = \sum_{k=0}^{\infty} k p_{C+k} \]  \hspace{1cm} (3.15)

\[ L_q = \sum_{k=0}^{\infty} k \frac{a^C}{C!(1-\rho)^2} p_0 = p_0 \frac{a^C}{C!} \sum_{k=0}^{\infty} k p^k = p_0 \frac{a^C}{C!} \frac{\rho}{(1-\rho)^2} \]  \hspace{1cm} (3.16)
or

\[ L_q = \Pr(>0) \frac{\rho}{1-\rho} = \Pr(>0) \frac{a/C}{1-a/C} = \Pr(>0) \frac{a}{C-a} \]  

(3.17)

The average time that a call request spends in the queue is given by Little's formula

\[ D_1 = W_q = \frac{L_q}{\lambda} = \Pr(>0) \frac{a}{C-a} \cdot \frac{1}{\lambda} = \Pr(>0) \frac{H}{C-a} \]  

(3.18)

where \( H = 1/\mu \) is the average call holding time

The average delay experiences by the calls that are placed in the queue is given by

\[ D_2 = \frac{D_1}{\Pr(>0)} = \frac{H}{C-a} \]  

(3.19)

Finally, the probability of a delay exceeding some given time \( t \) can be calculated as

\[ \Pr(>t) = \Pr(>0) \exp \left( -\frac{C-a}{H} t \right) \]  

(3.20)

Proof for (3.20) can be found in [1].

Table 3.2 summarizes some of the relations that are valid for \( M/M/C \) queuing systems.

**Example 3.6.** Consider a dispatch system in which 100 users transmitting back to a central system using the *push to talk* approach. The users can begin transmitting at an arbitrary instant in time and the lengths of their messages are distributed exponentially with a mean value of 6 sec. The system is operating using 5 channels. Assuming that each user generates 10 messages per hour, calculate the following:

- Probability of a message being delayed
- Average delay for all messages
- Average delay for delayed messages
- Probability that a message delay exceeds 4 seconds

The average offered traffic \( a = 100 \cdot \frac{10 \cdot 6}{3600} = 1.667 \text{ erlangs} \)

Average channel utilization is given as \( \rho = \frac{a}{C} = \frac{1.667}{5} = 0.333 \)

Probability of a delayed message is given by

\[
\Pr(>0) = \frac{a^C}{C! (1 - \rho)} = \frac{1.667^5}{5! (1 - 0.333)} = \frac{0.1608}{5.1503 + 0.1608} = 0.03
\]

\[
\sum_{n=0}^{\infty} \frac{a^n}{n! C! (1 - \rho)} = \sum_{n=0}^{4} \frac{1.667^n}{n!} + \frac{1.667^5}{5! (1 - 0.333)}
\]
The average delay for all messages is given by

\[ D_1 = \Pr(>0) \frac{H}{C - a} = 0.03 \frac{6}{5 - 1.667} = 0.054 \text{ seconds} \]

The average delay for delayed messages

\[ D_2 = \frac{H}{C - a} = \frac{6}{5 - 1.667} = 1.8 \text{ seconds} \]

Probability of a delay exceeding 4 seconds can be calculated as

\[ \Pr(>4) = \Pr(>0) \exp \left( - \frac{C - a}{H} t \right) = 0.03 \exp \left( - \frac{5 - 1.667}{6} \times 4 \right) = 0.0033 \]

**Table 3.2.** Summary of the performance parameters for \( M/M/C \) queuing system

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Symbol</th>
<th>Value (Formula)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offered traffic</td>
<td>( a )</td>
<td>( a = \lambda/\mu )</td>
</tr>
<tr>
<td>Channel utilization</td>
<td>( \rho )</td>
<td>( \rho = \frac{a}{C} )</td>
</tr>
<tr>
<td>Probability of having no calls at the cell site</td>
<td>( p_0 )</td>
<td>( \left{ \begin{array}{l} \frac{\sum_{k=0}^{C-1} a^k}{k!} \frac{a^c}{C!(1-\rho)} \ \sum_{n=0}^{C} \frac{a^n}{n!} p_0 \quad , n \leq C \ \frac{a^c}{C!} \rho^{n-C} p_0 \quad , n &gt; C \end{array} \right. )</td>
</tr>
<tr>
<td>Probability of having exactly ( n ) calls at the cell site</td>
<td>( p_n )</td>
<td>( \left{ \begin{array}{l} \frac{a^n}{n!} p_0 \quad , n \leq C \ \frac{a^c}{C!} \rho^{n-c} p_0 \quad , n &gt; C \end{array} \right. )</td>
</tr>
<tr>
<td>Probability of service request delay</td>
<td>( \Pr(&gt;0) ) or ( E_C [C,a] )</td>
<td>( \frac{a^c}{C!(1-\rho)} ) ( \sum_{n=0}^{C} \frac{a^n}{n!} + \frac{a^c}{C!(1-\rho)} )</td>
</tr>
<tr>
<td>Average number of requests in the queue</td>
<td>( L_q )</td>
<td>( \Pr(&gt;0) \frac{a}{C - a} )</td>
</tr>
<tr>
<td>Average time that all call requests spend in the queue</td>
<td>( D_1 ) or ( T_q )</td>
<td>( \Pr(&gt;0) \frac{H}{C - a} )</td>
</tr>
<tr>
<td>Average time that delay call request spend in the queue</td>
<td>( D_2 )</td>
<td>( \frac{H}{C - a} )</td>
</tr>
</tbody>
</table>
To facilitate the use of Erlang C in everyday engineering practice, (3.14) is frequently provided either in the form of curves or in the form of a table. Erlang C curves are given in Fig. 3.4 while in Appendix B we provide the Erlang C table. To illustrate the use of the Erlang C table, consider the following examples.

![Erlang C delay formula curves](image)

**Figure 3.4.** Representation of the Erlang C delay formula through a family of curves

**Example 3.7.** Determine the delay loss probabilities (Pr(> 0) and Pr(> t)) and delays (D₁ and D₂) for a cell with:

(a) 24 voice channels if the busy hour traffic is 20 erlangs, the acceptable delay time is 4 seconds, and the average call holding time is 4 seconds

(b) 20 voice channels if the busy hour traffic is 16 erlangs, the acceptable delay time is 3 seconds, and the average call holding time is 5 seconds.

**Part a:** In Appendix B, find the table for N = 24 and then find the row for a = 20 erlangs. Read across to find \( P(>0) = 0.29807 \) or 29.8% and \( P(>t) = 0.00546 \) or 0.55% for \( t/H = 1.0 \). Calculate \( D_1 \) and \( D_2 \) using (3.18) and (3.19) as follows:

\[
D_1 = \text{Pr}(>0) \frac{H}{C - a} = 0.298 \frac{4}{24 - 20} = 0.298 \text{ seconds}
\]

\[
D_2 = \frac{H}{C - a} = \frac{4}{24 - 20} = 1 \text{ second}
\]
**Part b):** In Attachment B, find the table for \( N = 20 \) and then find the row for \( a = 16 \) erlangs. Read across to find \( \Pr(>0) = 0.256 \) or 25.6\% and \( \Pr(>t) = 0.0232 \) or 2.32\% for \( t/H = 0.6 \). Calculate \( D_1 \) and \( D_2 \) using (3.18) and (3.19) as follows:

\[
D_1 = \Pr(>0)\frac{H}{C-a} = 0.256\frac{5}{20-16} = 0.32 \text{ seconds}
\]

\[
D_2 = \frac{H}{C-a} = \frac{5}{20-16} = 1.25 \text{ seconds}
\]

**Example 3.8.** Determine the offered traffic that

a) 24 voice channels can serve and still provide a delay probability of 2\% or better assuming that the average call holding time is 4 seconds and that the acceptable delay is 4 seconds

b) 20 voice channels can serve and still provide a delay probability of 1\% or better assuming that the average call holding time is 5 seconds and the acceptable delay is 3 seconds.

**Part a):** In Attachment B, find the table for \( C = 24 \) and read down the \( P(>t) \) column for \( t/H = 1.0 \) to find a probability of delay greater than 4 seconds of 0.01851. Read across this row to find \( a = 20.9 \) erlangs.

**Part b):** In Attachment B, find the table for \( C = 20 \) and read down the \( P(>t) \) column for \( t/H = 0.6 \) to find a probability of delay greater than 3 seconds of 0.00994. Read across this row to find \( a = 15.2 \) erlangs.

**Example 3.9.** Determine the number of voice channels

a) required to handle a busy hour traffic of 11 erlangs at a grade of service of 3\% or better, given an acceptable delay time of 8 seconds and an average call holding time of 10 seconds

b) required to handle a busy hour traffic of 15 erlangs at a grade of service of 1\% or better, given an average call holding time of 5 seconds and an acceptable delay time of 6 seconds.

**Part a):** Using a trial-and-error method, find that the table for \( C = 14 \) in Attachment B shows a probability of delay greater than 8 seconds to be 0.02748 for \( t/H = 0.8 \). Therefore, fourteen voice channels are required.

**Part b):** Using a trial-and-error method find that the table for \( C = 18 \) in Appendix B shows a probability of delay greater than 6 seconds to be 0.00987 for \( t/H = 1.2 \). Therefore, eighteen voice channels are required.