Cellular systems & GSM


Un. of Rome “La Sapienza”

Chiara Petrioli †

† Department of Computer Science – University of Rome “Sapienza” – Italy

Si ringraziano per il materiale fornito, da cui sono state tratte molte di queste slide il Prof. Antonio Capone, Politecnico di Milano (corso di retiradiomobili) e il Prof. Giuseppe Bianchi, Universita’ di Tor Vergata)
Introduction to cellular systems


Un. of Rome “La Sapienza”

Chiara Petrioli†

†Department of Computer Science – University of Rome “Sapienza” – Italy
• How many channels to allocate per cell (planning)
• How to perform frequency reuse
  → with the objective to maximize perceived performance, and decrease cost

• How to use at best available resources
  – voice/data encoding
  – How to deal with multiple access

• How to deal with mobility
Multi-cell systems

- The radio resource is to be divided among base stations.
- The amount of radio resource (bandwidth) is very limited and it is not possible to dedicate it exclusively to a physical channel of a particular cell.
- In the division of the radio resource among cells the resource is reused several times in cells that are sufficiently distant so that the mutual interference becomes strongly attenuated (remember path loss).
- The reuse of frequencies is a critical aspect in the design of cellular systems as it determines on one hand the number of channels to assign to each cell and on the other hand the channel quality.
- We will devote much attention to the problem!
• Regardless of the manner with which the resource is divided the number of channels that we can assign to each cell is limited
• Apart for special cases (which we will see, as those of dynamic allocation) the number of channels is fixed
• The number of simultaneous conversations per cell is limited and it is therefore possible that upon arrival of a new call that requires to establish a circuit (eg. Voice) there are no more available channels in the radio access network (resulting in call blocking)
• To evaluate the performance in terms of call blocking probability we need to characterize the traffic
  – Process of arrivals (voice calls are well modeled by the Poisson process)
  – Rate of arrivals
  – Average call duration
We need to characterize how traffic arriving to a cell is served:

- Service requests arrive to a service system (or queue system) according to a random process.
- Each request is characterized by a non-null time of service which is the time needed to fulfill the request by a server.
- Presence of one or more waiting systems (or queue) where requests await that a serving node is free.
• Commercial service systems (Supermarket/post office cashiers, entrance to a museum, railway ticketing etc.)
• Social service systems (hospital services, outpatient medical service, public offices etc.)
• Transport systems (vehicles waiting at toll booths, or waiting to be loaded/unloaded, planes waiting to take off or land, etc.)
• Production systems (waiting on the part of the production lines of components that must be machined, assembly centers or systems with maintenance-serving workers etc.)
• Communication Systems (waiting for the packets in the queue before being transmitted, etc.)
Serving system is characterized by:

- User population (finite/infinite)
- Number s of servers at the queueing node
- Arrival process
  - what is the distribution of interarrival times?
  - what happens if the arriving user finds the queue full?
- Serving scheme
  - Describes how servers erogate the service:
    ✓ Which is the distribution of serving time?
    ✓ Do servers operate sequentially or in parallel?
- Queuing discipline
  - FIFO/LIFO/priority based/random

We assume that:

a) the arrival times \( t_i^a \) are independent and identically distributed
b) that the service times \( t_i^s \) are independent and identically distributed
Serving system is characterized by:

- User population (finite/infinite)
- Number of servers at the queueing node
- Arrival process
  - what is the distribution of interarrival times?
  - what happens if the arriving user finds the queue full?
- Serving scheme
  - Describes how servers erogate the service:
    ✓ Which is the distribution of serving time?
    ✓ Do servers operate sequentially or in parallel?
- Queuing discipline
  - FIFO/LIFO/priority based/random

Queuing theory:
Allows to answer key questions for system dimensioning:
- Average waiting time?
- Distribution of this time?
- Servers level of utilization?

We assume that a) the arrival times $t_i^a$ are independent and identically distributed
b) that the service times $t_i^s$ are independent and identically distributed
We need to characterize how traffic arriving to a cell is served:

- Service requests arrive to a service system (or queue system) according to a random process.
- Each request is characterized by a non-null time of service which is the time needed to fulfill the request by a server.
- Presence of one or more waiting systems (or queue) where requests await that a serving node is free.

Arrivals = call request
Num. Of servers = Num. Of channels
Queue size = 0
• It is the standard notation used to describe and classify a queueing system
  - A/B/s/c/p/Z
  - A describes the interarrival times probability distribution
  - B represents the probability distribution of the service times
  - s the number of servers at the node
  - c is the capacity of the queue
  - p is the size of the population to be served
  - Z is the queueing discipline

Non negative integers
FIFO if not specified

c and p are infinite if not specified
• In general:
  – Average arrival rate $\lambda = 1/E(t_a)$
    ✓ if there are 5 user arrivals in 30 minutes $\lambda = 1/6 = 0.1666$ users per minute
  – Average service rate $\mu = 1/E(t_s)$
    ✓ if the server is able to serve 4 users per minute (in other words $\mu = 4$) the average service time is $\frac{1}{4}$ of a minute
  – The state of a queueing system $n(t)$ indicates the number of users that are in the system at time $t$
  – The queue size at time $t$ is $n_q(t)$
    ✓ $n_q(t) = 0$ if $n(t) \leq s$
    ✓ $n_q(t) = n(t) - s$ otherwise
• **In general:**
  - Average arrival rate \( \lambda = 1/E(t_a) \)
    - e.g., if there are 5 user arrivals in 30 minutes \( \lambda = 1/6 = 0.1666 \) users per minute
  - Average service rate \( \mu = 1/E(t_s) \)
    - if the server is able to serve 4 users per minute (in other words \( \mu = 4 \)) the average service time is \( \frac{1}{4} \) of a minute
  - The state of a queueing system \( n(t) \) indicates the number of users that are in the system at time \( t \)
  - The queue size at time \( t \) is \( n_q(t) \)
    - \( n_q(t) = 0 \) if \( n(t) \leq s \)
    - otherwise \( n_q(t) = n(t) - s \)
• In general:
  – Average arrival rate $\lambda = 1/E(t_a)$
    ✓ if there are 5 user arrivals in 30 minutes $\lambda = 1/6=0,1666$ users per minute
  – Average service rate $\mu = 1/ E(t_s)$
    ✓ if the server is able to serve 4 users per minute (in other words $\mu = 4$) the average service time is’ $\frac{1}{4}$ of a minute
  – The state of a queueing system $n(t)$ indicates the number of users that are in the system at time $t$
  – The queue size at time $t$ is $n_q(t)$
    ✓ $n_q(t)=0$ if $n(t)<=s$
    ✓ $n_q(t)=n(t)-s$ otherwise
• Distribuzione esponenziale

\[
F_X(t) = P(X \leq t) = \begin{cases} 
1 - e^{-\alpha t} & \text{per } t \geq 0 \\
0 & \text{per } t < 0 
\end{cases}
\]

\[
E(X) = \frac{1}{\alpha}, \quad Var(X) = \frac{1}{\alpha^2}.
\]

Proprietà E1: La densità di probabilità \( f_X(t) \) è una funzione strettamente decrescente di \( t \) (\( t \geq 0 \))

Proprietà E2: Assenza di memoria. Per ogni \( t > 0 \) e \( s > 0 \), vale la seguente uguaglianza

\[
P \left( X > s + t \mid X > s \right) = P(X > t). \quad (1.3.2)
\]
Definizione 1.3.1 Un processo stocastico \( \{X(t), t \geq 0\} \) è detto processo di conteggio (counting process) se \( X(t) \) rappresenta il numero totale di eventi che accadono fino all’istante \( t \).

Definizione 1.3.2 Un processo di conteggio ha incrementi indipendenti se il numero degli eventi che accadono in intervalli di tempo disgiunti sono indipendenti.

Definizione 1.3.4 Un processo di conteggio \( \{X(t), t \geq 0\} \) è un processo di Poisson di tasso \( \lambda > 0 \) se valgono

i) \( X(0) = 0 \)

ii) il processo ha incrementi indipendenti

iii) il numero di eventi che accadono in ogni intervallo di tempo di ampiezza \( t \) (dato da \( X(s+t) - X(s) \)) ha distribuzione di Poisson di parametro \( \lambda t \), ovvero per ogni \( s, t \geq 0 \) risulta

\[
P(X(s+t) - X(s) = n) = e^{-\lambda t} \frac{(\lambda t)^n}{n!}, \quad n = 0, 1, \ldots \quad (1.3.6)
\]
• T1 is the time to wait before the first event occurs and Tk the waiting time between the (k-1) and the k-th event:
• \( \{T_k, \ k \geq 1\} \) the sequence of random variables expressing the time between two successive events;
• Sn is the time when the n-th event happens
  \[ S_n = T_1 + T_2 + \cdots + T_n, \]
• The process \( \{X(t), \ t \geq 0\} \) indicates the number of events that occur in \([0,t]\)
  \[ X(t) = \max\{n \ | \ S_n \leq t\}, \]
• \( \{T_k, \ k \geq 1\} \) the sequence of random variables expressing the time between two successive events;

• The process \( \{X(t), \ t \geq 0\} \) indicates the number of events that occur in \([0,t]\)

\[
X(t) = \max\{n \mid S_n \leq t\},
\]

The following statements are equivalent:

1) \( \{X(t), \ t \geq 0\} \) is a Poisson process with rate \( \lambda \)

2) Variables \( T_i \) are independent identically distributed random variables, which follow an exponential distribution of parameter \( \lambda \)

, i.e.

\[
P(T_i \leq t) = 1 - e^{-\lambda t}, \quad i = 1, 2, \ldots
\]
It is the standard notation used to describe and classify a queueing system:

- \( A/B/s/c/p/Z \)
- \( A \) describes the interarrival times probability distribution
- \( B \) represents the probability distribution of the service times
- \( s \) the number of servers at the node
- \( c \) is the capacity of the queue
- \( p \) is the size of the population to be served
- \( Z \) is the queueing discipline

Arrivals follow a Poisson process.

Examples:

- \( M/M/1 \)
- \( M/M/k \)
- \( M/G/1 \)

M denotes the exponential distribution

D is the constant distribution

\( E_k \) denotes the Erlang distribution of order \( k \)

G indicates a generic distribution
The Poisson Process

- \( \{T_k, \ k \geq 1\} \) is the sequence of random variables expressing the time between two successive events;
- The process \( \{X(t), \ t \geq 0\} \) indicates the number of events that occur in \([0, t]\)

\[
X(t) = \max\{n \mid S_n \leq t\},
\]

The following statements are equivalent:
1) \( \{X(t), \ t \geq 0\} \) is a Poisson process with rate \( \lambda \)
2) Variables \( T_i \) are independent identically distributed random variables, which follow an exponential distribution of parameter \( \lambda \)

, i.e.

\[
P(T_i \leq t) = 1 - e^{-\lambda t}, \quad i = 1, 2, \ldots
\]

Poisson process well expresses the process of call arrivals to a cellular system
- In the case of a queue system that represents the management of calls in a cell
- The probability that the number of arrivals \( N(t, t + \tau) \) in a time interval between \( t \) and \( t + \tau \) is equal to \( k \) is given by:

\[
P[N(t, t + \tau) = k] = \frac{(\lambda \tau)^k}{k!} e^{-\lambda \tau}
\]

Poisson arrival process
In steady state $E[A(T)] = A$

$$A = \lambda X$$

- **Average traffic (active calls)** in an interval of size $T$
- **$A$** is adimensional
- **Traffic is measured in Erlang**
- **$\lambda$** Arrival rate of calls (call/s)
- **$X$** average duration of calls (s)
We need to characterize how traffic arriving to a cell is served:

- Service requests arrive to a service system (or queue system) according to a random process.
- Each request is characterized by a non-null time of service which is the time needed to fulfill the request by a server.
- Presence of one or more waiting systems (or queue) where requests await that a serving node is free.

Arrivals = call request
Num. Of servers = Num. Of channels
Queue size = 0
Evolution of a queueing system
Per modellare l’arrivo delle chiamate in una cella con un numero di canali disponibili pari a \( n \) basta usare un sistema a pura perdita (senza posti in coda) con \( n \) serventi.

Si mostra che, nell’ipotesi di arrivi di Poisson, la probabilità di rifiuto di una chiamata è data dalla formula di Erlang:

\[
B(n, A) = \frac{A^n}{n!} \sum_{k=0}^{n} \frac{A^k}{k!}
\]

dove \( A = \lambda T \) (in Erlang), \( \lambda \) frequenza media degli arrivi (call/s), \( T \) durata media delle chiamate.

NOTA: vale per qualunque distrib. della durate delle chiamate.
• The blocked traffic which is not served is:

\[ A_p = A \cdot B(n, A) \]

• Carried traffic:

\[ A_s = A \cdot (1 - B(n, A)) = A - A_p \]

• Channel utilization coefficient is given by:

\[ \rho = \frac{A_s}{n} = \frac{A \cdot (1 - B(n, A))}{n}, \quad 0 \leq \rho \leq 1 \]
• Fundamental formula for telephone networks planning
  - $A_o =$ offered traffic in Erlangs

$\Pi_{block} = \frac{A_o^C}{C!} \sum_{j=0}^{C} \frac{A_o^j}{j!} = E_{1,C}(A_o)$

$E_{1,C}(A_o) = \frac{A_o E_{1,C-1}(A_o)}{C + A_o E_{1,C-1}(A_o)}$

$→$ Efficient recursive computation available

Blocking probability:
Erlang-B
Target: support users with a given Grade Of Service (GOS)

- GOS expressed in terms of upper-bound for the blocking probability
  - GOS example: subscribers should find a line available in the 99% of the cases, i.e. they should be blocked in no more than 1% of the attempts

Given:
- C channels
- Offered load $A_o$
- Target GOS $B_{\text{target}}$

C obtained from numerical inversion of

\[
B_{\text{target}} = E_{1,C}(A_o)
\]
Channel usage efficiency

Offered load (erl) \( A_o \) \[ \xrightarrow{\text{C channels}} \]
Carried load (erl) \( A_c = A_o (1 - B) \)

Blocked traffic \( A_o B \)

Efficiency: \[ \eta = \frac{A_c}{C} = \frac{A_o (1 - E_{1,c}(A_o))}{C} \approx \frac{A_o}{C} \text{ if small blocking} \]

Fundamental property: for same GOS, efficiency increases as \( C \) grows!! (trunking gain)
GOS = 1% maximum blocking.

Resulting system dimensioning and efficiency:

<table>
<thead>
<tr>
<th>Traffic Load (erl)</th>
<th>Capacity (C)</th>
<th>Efficiency (η)</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 erl</td>
<td>&gt;= 53</td>
<td>74.9%</td>
</tr>
<tr>
<td>60 erl</td>
<td>&gt;= 75</td>
<td>79.3%</td>
</tr>
<tr>
<td>80 erl</td>
<td>&gt;= 96</td>
<td>82.6%</td>
</tr>
<tr>
<td>100 erl</td>
<td>&gt;= 111</td>
<td>84.6%</td>
</tr>
</tbody>
</table>
**Example:** How many channels are required to support 100 users with a GOS of 2% if the average traffic per user is 30 mE?

100x30mE = 3 Erlangs
3 Erlangs @ 2% GOS =

8 channels
The higher the offered load (Erlang) the higher the blocking probability

Teoria del traffico: Probabilità di blocco
Given a desired blocking probability, the higher the number of channels, the higher the offered traffic that can be supported.
Given a desired max. blocking probability and a given expected traffic (target audience) which is the minimum number of channels?