3 Diseño del Software “Traffic Analyzer”

En este capítulo se reporta el desarrollo que se llevó a cabo para realizar el software que analiza el tráfico en redes de telefonía y computadoras, denominado “Traffic analyzer”. En el diseño de este software se toman en cuenta varias consideraciones para que la aplicación sea de fácil manejo para el usuario. Cabe señalar que un punto importante en el diseño de cualquier programa radica en que éste sea práctico, útil y sencillo de implementar para los consumidores. Es por ello que para la programación del software se utilizó Visual Studio 2005, específicamente hablando Visual Basic, ya que nos proporciona aplicaciones sencillas y amigables para el usuario, además de que es una herramienta bien cimentada en el área de construcción de aplicaciones visuales orientadas a un servicio, como lo es este caso en específico.

Inicialmente se realizó un diagrama general del “Traffic analyzer” para identificar la secuencia que se iba a seguir. Como se puede ver en la figura 3-1 se tomaron dos sistemas, explicados a detalle en los capítulos 1 y 2, que denotan gran importancia en la ingeniería del tráfico y comportamiento del mismo. Así pues, como se observa en la figura 3-1 se tiene la primera partición del tema, el cual se dividió en sistemas telefónicos y sistemas basados en teorías de colas. A su vez, estos sistemas fueron sub-divididos para que en los sistemas de telefonía se obtuvieran fórmulas para sistemas de tráfico general y por célula, mientras que en los sistemas de colas se obtuvo el análisis matemático para 3 tipos de modelados de colas, M/M/1, M/M/1/K y M/M/c, los cuales se analizaron en el capítulo 2.3. Cabe señalar que en la figura 3-1 y posteriores diagramas de este capítulo, se ocuparán términos, además del español, en inglés ya que el software se programó en este idioma debido a que resulta más práctico para manejar términos utilizados en la ingeniería y modelado de tráfico. Un ejemplo muy particular lo es el ACHT, el cual es el promedio de duración de la llamada.

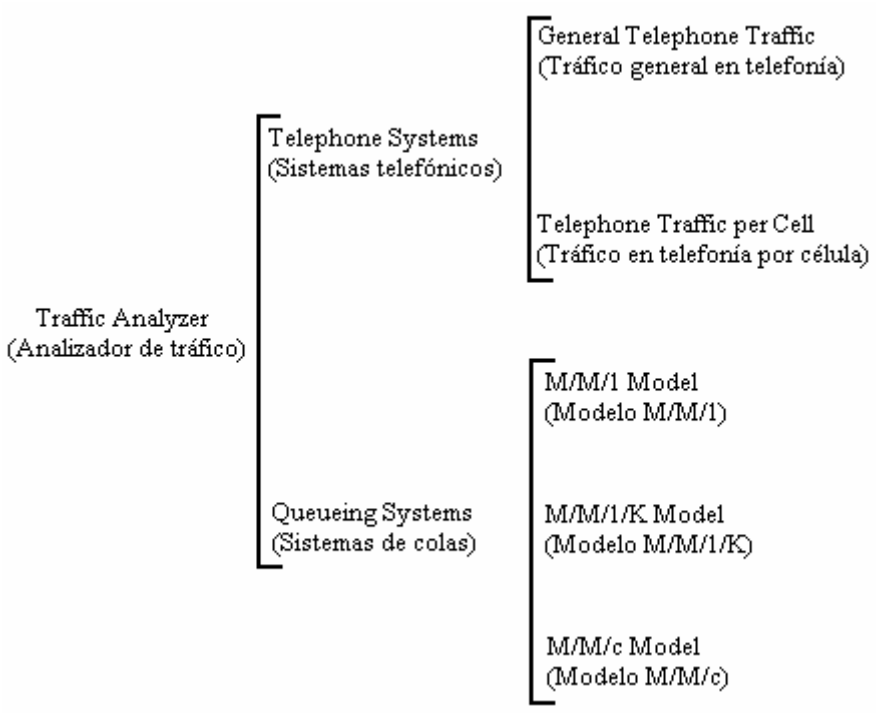


Figura 3-1 Diagrama general del Analizador del tráfico.

En los sub-capítulos siguientes se reportará paso a paso el desarrollo de cada sistema, y sus respectivos sub-sistemas, programados en el software. Además, se mostrarán las variables de entrada y salida requeridas para cada tipo de sistema, así como su interfase gráfica correspondiente.

3.1 Diseño del tráfico en sistemas de telefonía

Para este tipo de sistemas, la ingeniería del tráfico juega un papel muy importante, puesto que en esta parte se calcula el tráfico ofrecido por el sistema y el número de troncales, o canales, necesarios para que el sistema sea costeable y eficiente. Es decir, que el sistema no esté sub o sobre provisto, debido a que si el sistema está sobre provisto, aunque éste asegure que todos los usuarios obtendrán un servicio, se tiene la

gran desventaja de que resulta un medio muy costoso; sí nos vamos al caso extremo del sistema sub provisto tendremos un sistema ineficiente, debido a que muchas llamadas no podrán realizarse por falta de canales disponibles. De esta manera, en la figura 3-2 se muestran las variables de entrada y salida para cada sistema de tráfico en telefonía.

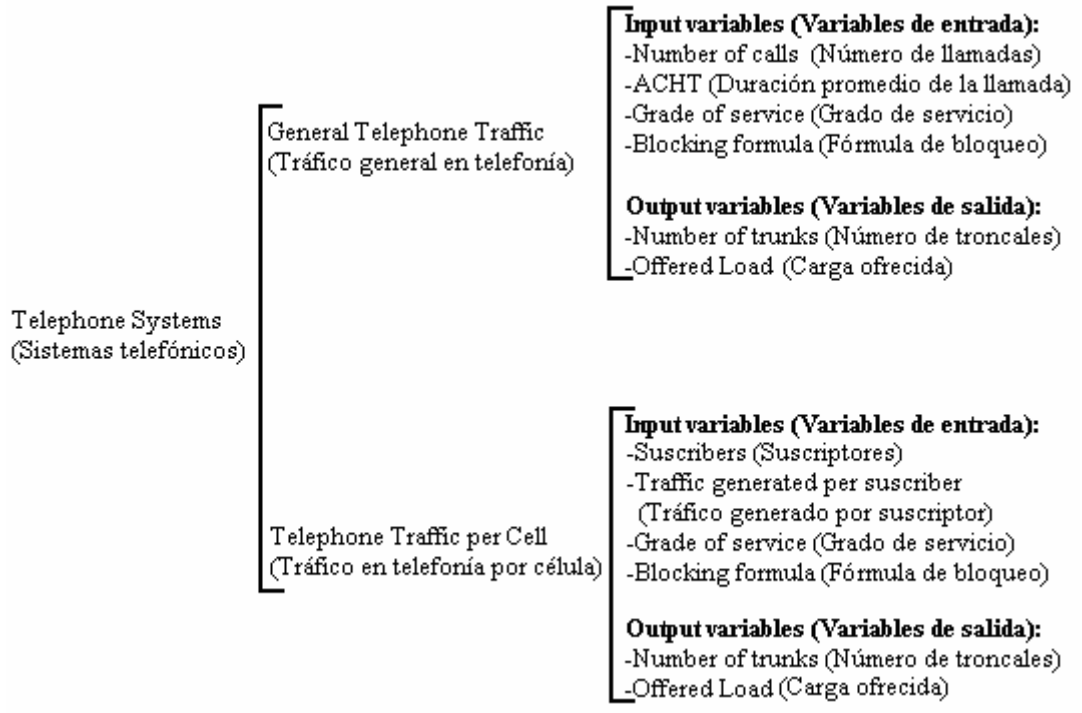


Figura 3-2 Variables consideradas en telefonía para tráfico en general y por célula.

Es decir, se ingresarán las variables de entrada en cualquiera de los dos sistemas de telefonía, y se indicará qué tipo de probabilidad de bloqueo y grado de servicio se desea utilizar, así como otras variables, como lo son el ACHT, suscriptores, etc., las cuales nos dan como resultado la carga ofrecida en Erlangs. Simplificando, a través de este parámetro y el grado de servicio se obtendrá, mediante la tabla de distribución, el número de troncales necesarias para obtener un buen sistema.

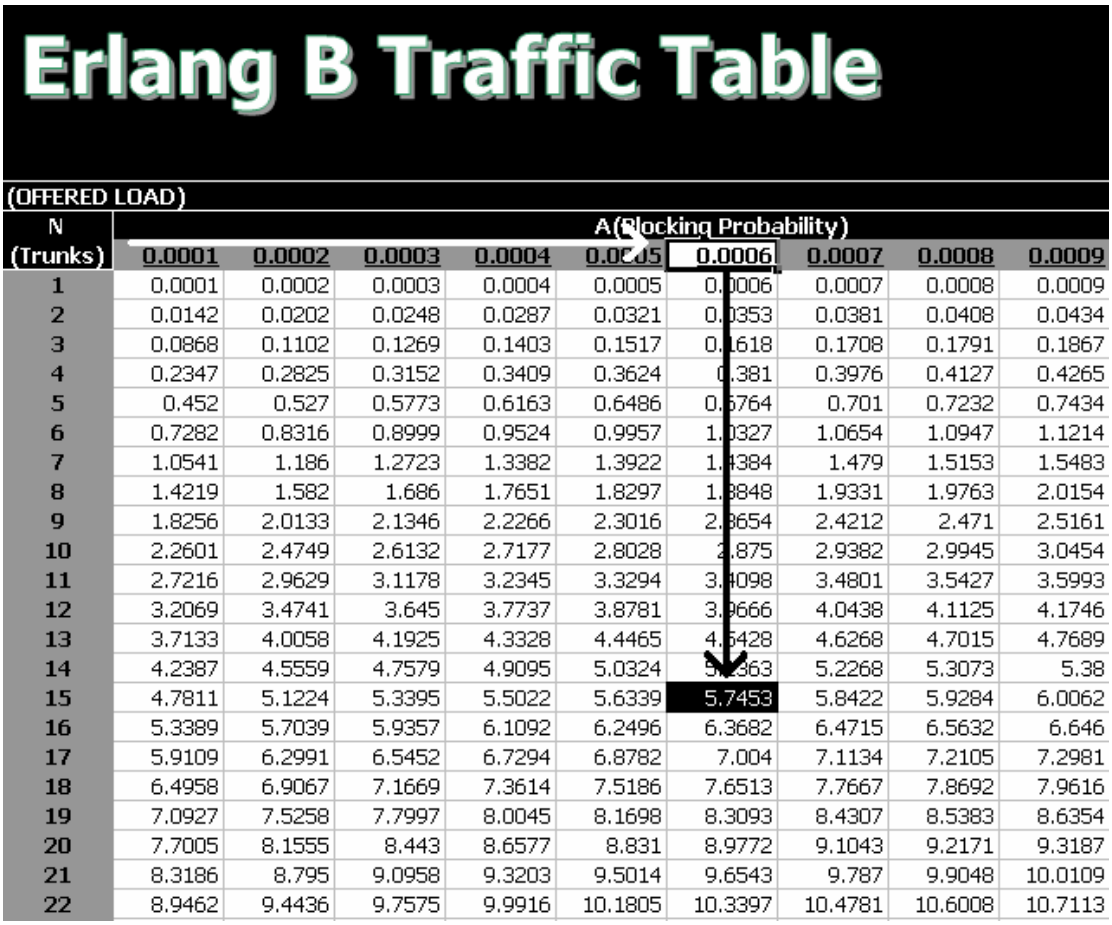
3.1.1 Diseño del tráfico en sistemas de telefonía general

Para este tipo de sistema de tráfico en telefonía, la lógica a seguir es la siguiente:

1) Se proporcionan las 4 variables de entrada, Número de llamadas, ACHT, Grado de servicio y fórmula de bloqueo.
2) El programa a través del Número de llamadas y el ACHT, en segundos, obtiene un parámetro conocido como Carga ofrecida, el cual tiene como unidad de medida el Erlang. Así pues utilizando la ecuación (1.2), comprobamos la unidad para este parámetro:

$$(\cancel{llamadas})\left(ACHT\,\frac{\cancel{segundos}}{\cancel{llamada}}\right)\left(\frac{1Erlang}{3600\,\cancel{segundos}}\right)=Erlangs \qquad (3.1)$$

3) Una vez que se obtiene la carga ofrecida, se utiliza la tabla de distribución correspondiente. Para este programa, se tiene la tabla de Poisson, Erlang C, Erlang B y Erlang B extendido con un porcentaje de 10, 20, 30, 40, 50, 60, 70, 80 y 90%. En la tabla se busca el número que haga intersección entre la probabilidad de bloqueo y este parámetro; para efectos de este programa se tendrá la opción de elegir entre 0.01 – 0.10, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1–1.9, 2–15 y de 20 hasta 95% de 5 en 5 de grado de servicio. A continuación se muestra una tabla de distribución indicando lo que se mencionó anteriormente:



Erlang B Traffic Table

(OFFERED LOAD)

N (Trunks)	A (Blocking Probability) 0.0001	0.0002	0.0003	0.0004	0.0005	0.0006	0.0007	0.0008	0.0009
1	0.0001	0.0002	0.0003	0.0004	0.0005	0.0006	0.0007	0.0008	0.0009
2	0.0142	0.0202	0.0248	0.0287	0.0321	0.0353	0.0381	0.0408	0.0434
3	0.0868	0.1102	0.1269	0.1403	0.1517	0.1618	0.1708	0.1791	0.1867
4	0.2347	0.2825	0.3152	0.3409	0.3624	0.381	0.3976	0.4127	0.4265
5	0.452	0.527	0.5773	0.6163	0.6486	0.6764	0.701	0.7232	0.7434
6	0.7282	0.8316	0.8999	0.9524	0.9957	1.0327	1.0654	1.0947	1.1214
7	1.0541	1.186	1.2723	1.3382	1.3922	1.4384	1.479	1.5153	1.5483
8	1.4219	1.582	1.686	1.7651	1.8297	1.8848	1.9331	1.9763	2.0154
9	1.8256	2.0133	2.1346	2.2266	2.3016	2.3654	2.4212	2.471	2.5161
10	2.2601	2.4749	2.6132	2.7177	2.8028	2.875	2.9382	2.9945	3.0454
11	2.7216	2.9629	3.1178	3.2345	3.3294	3.4098	3.4801	3.5427	3.5993
12	3.2069	3.4741	3.645	3.7737	3.8781	3.9666	4.0438	4.1125	4.1746
13	3.7133	4.0058	4.1925	4.3328	4.4465	4.5428	4.6268	4.7015	4.7689
14	4.2387	4.5559	4.7579	4.9095	5.0324	5.1363	5.2268	5.3073	5.38
15	4.7811	5.1224	5.3395	5.5022	5.6339	5.7453	5.8422	5.9284	6.0062
16	5.3389	5.7039	5.9357	6.1092	6.2496	6.3682	6.4715	6.5632	6.646
17	5.9109	6.2991	6.5452	6.7294	6.8782	7.004	7.1134	7.2105	7.2981
18	6.4958	6.9067	7.1669	7.3614	7.5186	7.6513	7.7667	7.8692	7.9616
19	7.0927	7.5258	7.7997	8.0045	8.1698	8.3093	8.4307	8.5383	8.6354
20	7.7005	8.1555	8.443	8.6577	8.831	8.9772	9.1043	9.2171	9.3187
21	8.3186	8.795	9.0958	9.3203	9.5014	9.6543	9.787	9.9048	10.0109
22	8.9462	9.4436	9.7575	9.9916	10.1805	10.3397	10.4781	10.6008	10.7113

Figura 3-3 Ejemplo para obtener el parámetro de carga ofrecida.

Para poder utilizar las tablas de distribución, las cuales son explicadas detalladamente en el capítulo 1.6, después de ser generadas se pasan a archivos .txt para poder llamarlas desde Visual Basic. Una desventaja de pasar las tablas a archivos .txt es que el software se limita hasta 2500 troncales; si se necesitan más de este número, se indicará que más de 2500 troncales serán asignadas. En la figura 3-4 se muestra la lógica utilizada para programar esta parte del software:

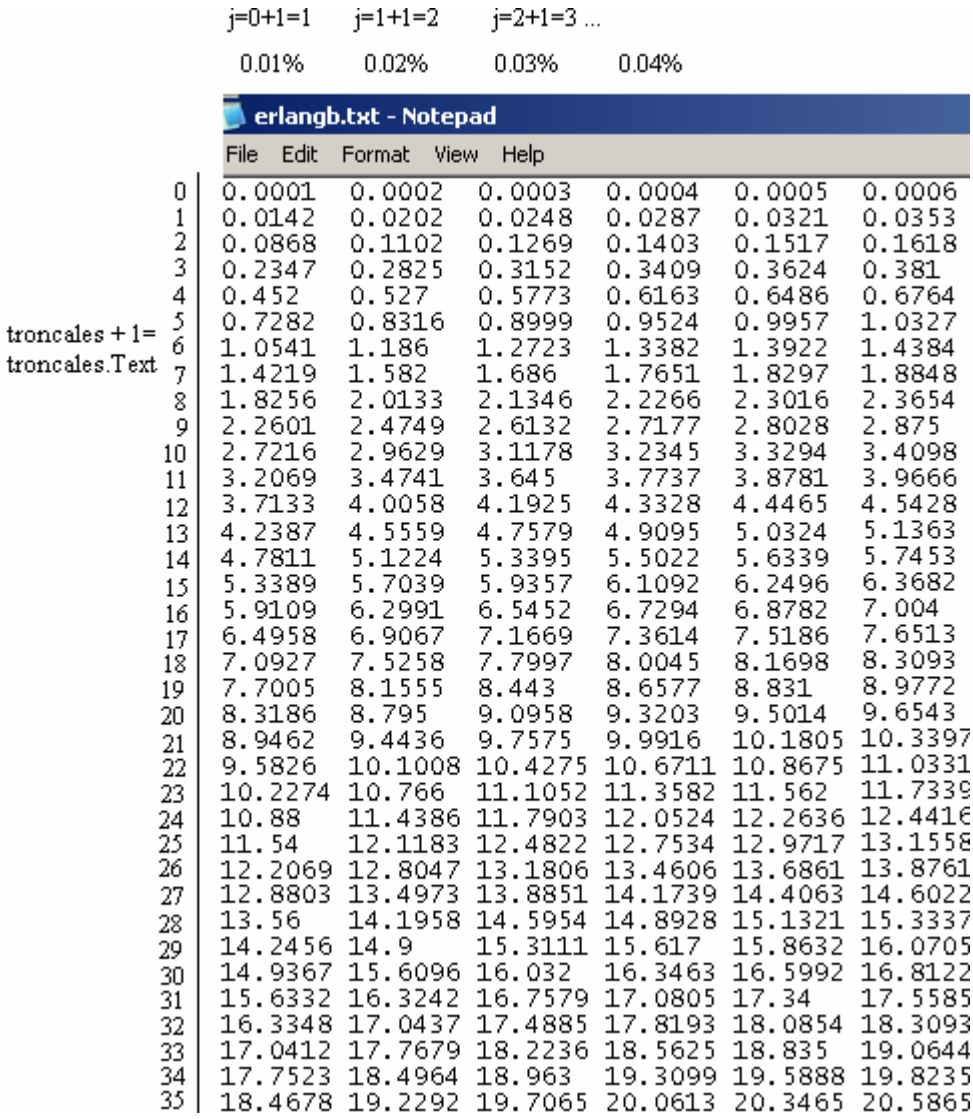


Figura 3-4 Ejemplo de tablas de la lógica en tablas de distribución en archivos txt.

Donde la lógica a utilizar es la siguiente: en la primera fila se encuentran los grados de servicio mencionados anteriormente, al seleccionarse uno se utiliza su posición para identificarlo. Donde la primera posición o índice es cero; es por ello que se utiliza la variable j, para la cual en la figura 3-4 se observa que ésta se calcula al sumar la posición más 1. Por ejemplo, si se tiene un grado de servicio de 0.02%, se observa que éste se localiza en la posición número 1 y al sumarle otro 1, se tendrá que el valor de j es igual a 2. Una vez que se encuentra en la posición correcta, en la columna correspondiente al valor del grado de servicio, se recorre la columna hasta

encontrar el número que sea mayor o igual a la carga ofrecida, una vez encontrado se cuentan las filas que se recorrieron y se le aumenta un 1, así como se muestra en la figura 3-4. A continuación se muestran algunas partes del código para este sistema.

En la figura 3-5 se muestra como se declaran las variables que se van a utilizar, así como el tipo de variable que son. Es importante mencionar que se utiliza la variable FileName para saber que archivo .txt es el que se llamará y así se obtenga el número de troncales necesarias en el sistema.

```
Private Function GetFileName(ByVal BlockType As String) As String
    Dim FileName As String
    FileName = ""
    If BlockType = "Erlang B" Then
        FileName = "erlangb.txt"
    ElseIf BlockType = "Extended Erlang B 10%" Then
        FileName = "exterlangb10.txt"
    ElseIf BlockType = "Extended Erlang B 20%" Then
        FileName = "exterlangb20.txt"
      :
      :
    ElseIf BlockType = "Poisson" Then
        FileName = "poisson.txt"

    End If
    GetFileName = FileName
End Function
```

Figura 3-5 Código para llamar a los archivos txt.

Otra parte importante del código es la lógica que se utiliza para buscar el número de troncales en la tabla de distribución, .txt. Cuando se brindan las variables de entrada, al dar clic en el botón de calcular se obtiene el tráfico ofrecido. De esta manera, al estar ubicado en la columna correcta, se recorre la fila hasta encontrar el grado de servicio calculado (con la condición de que sea mayor o igual, para que cubra todo el servicio). En seguida, se cuenta el número de filas recorridas y se le

suma un 1 debido a que en la primera columna se encuentra la variable de grado de servicio. Los códigos que se utilizaron fueron los siguientes:

```
If cbGOS2.SelectedIndex >= 0 Then
   j = cbGOS2.SelectedIndex
   If Double.TryParse(tbNoC.Text, NoC) Then
       If Double.TryParse(tbACHT.Text, ACHT) Then
          OfferedLoad = (ACHT / 3600) * NoC
          tbOfferedLoad2.Text = FormatNumber(OfferedLoad, 4, vbFalse)
          For Trunks = 0 To MaxRows
              If BlockTable(Trunks, j) >= OfferedLoad Then
                  Exit For
              End If
```

Figura 3-6 Código para asignar el grado de servicio a la variable j.

```
If Trunks >= 0 And Trunks < 2500 Then
    Trunks = Trunks + 1
    tbNoT2.Text = Trunks.ToString
```

Figura 3-7 Código para delimitar el número de troncales posibles (si cumple la condición lo manda a la interfase).

Finalmente, la interfase gráfica de esta parte del software es la siguiente:

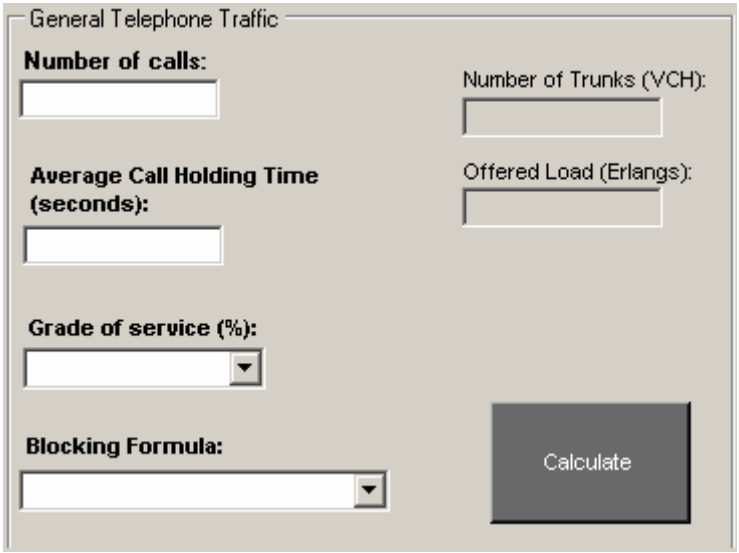


Figura 3-8 Interfase gráfica en el sistema de tráfico por telefonía general.

3.1.2 Diseño del tráfico en sistemas de telefonía por célula

Para este tipo de sistema de tráfico en telefonía, la lógica a seguir es la siguiente:

1) Se proporcionan las cuatro variables de entrada, Número de suscriptores, Tráfico generado por suscriptor, Grado de servicio y fórmula de bloqueo.
2) El programa a través de la multiplicación del número de suscriptores y el tráfico generado por cada uno de ellos obtiene la carga ofrecida, con unidad de Erlang como se menciona en el capítulo 3.1.1.
3) A continuación se sigue la misma lógica que en el diseño de telefonía general; en la tabla de distribución seleccionada se busca a través del grado de servicio y la carga ofrecida el número de troncales. Así pues, la interfase gráfica de esta parte del software es la siguiente:

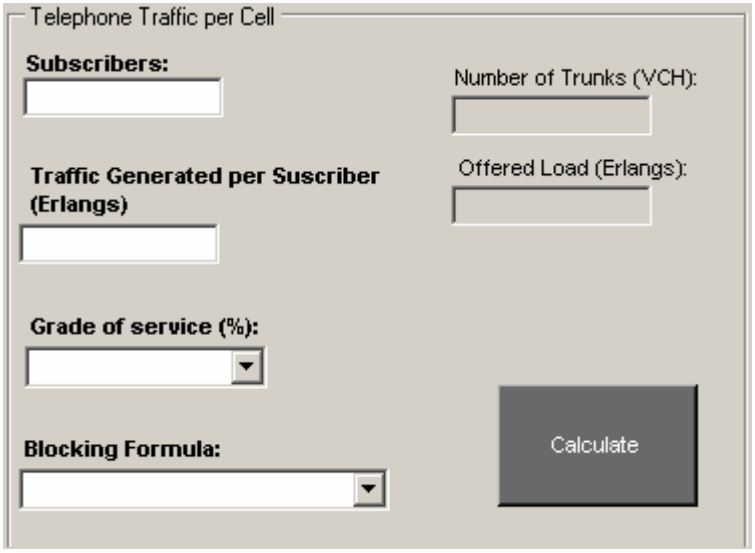
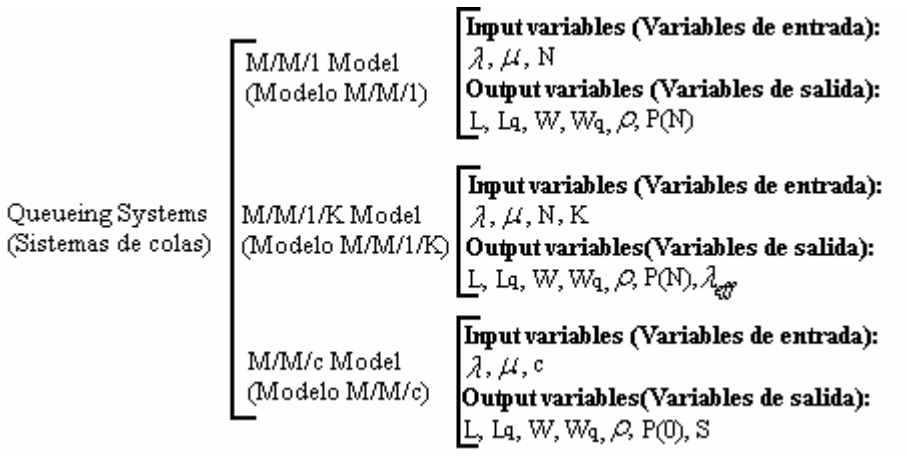


Figura 3-9 Interfase gráfica en el sistema de tráfico de telefonía por célula.

3.2 Diseño de los sistemas de colas

Para el desarrollo de esta parte se utilizaron los modelos matemáticos descritos detalladamente en el capítulo 2. Un factor muy importante es que esta parte del software analiza el comportamiento del tráfico en un sistema. Es decir, obtenemos variables de salida que nos indican como está reaccionando el tráfico dentro del sistema y de la cola, como lo son la longitud, el tiempo de espera, la intensidad de tráfico, probabilidades, etc., dependiendo del tipo de modelo que se esté ocupando. Por lo tanto, en la figura 3-10 se muestran las variables de entrada y de salida para cada tipo de sistema de colas:



Where (donde):
L= Mean lenght of system (Longitud promedio del sistema)
Lq =Mean lenght of queue (Longitud promedio de la cola)
W= Mean waiting time in system (Tiempo de espera promedio en el sistema)
Wq =Mean waiting time in queue (Tiempo de espera promedio en la cola)
ρ = Traffic intensity (Intensidad de tráfico)
P(N)= Probability N packets exist (Probabilidad de que existan N paquetes)
λ_{eff} = Effective arrival rate (Tasa efectiva de llegadas)
P(0)= Probability 0 packets exist (Probabilidad de que no exista ningún paquete)
S= Number of busy servers (Número de servidores ocupados)
N= Packets (Paquetes o clientes)
λ = Arrival rate (Tasa de llegada)
μ= Service rate (Tasa de servicio)
K= Total number of packets (Número total de paquetes)
c= Servers (Servidores)

Figura 3-10 Variables consideradas en los modelos M/M/1, M/M/1/K y M/M/c.

Para esta parte del software un factor muy importante son las condiciones que tiene cada sistema, debido a que para efectos de la aplicación se utiliza únicamente el estado estable o también denominado de no saturación. En seguida se presenta el diseño de cada uno de los modelos:

3.2.1 Diseño del modelo M/M/1

Este modelo es el más sencillo de los tres debido a que se ocupa una sola cola con un único servidor con entradas infinitas; sin embargo, hay que tener cuidado con esta última característica del sistema puesto que si el grado de servicio es menor al grado de arribo, se tendrá un sistema saturado, es decir no estará en el estado estable. Las dos medidas de seguridad que se tomaron para el sistema fueron las siguientes:

1) La intensidad del tráfico debe ser < 1, es decir:

$$\rho = \frac{\lambda}{\mu} < 1 \tag{3.2}$$

donde:

ρ = Intensidad de tráfico en el sistema,

λ = Número medio de llegadas por unidad de tiempo, y

μ = Número medio de paquetes que el servidor es capaz de atender por unidad de tiempo,

2) La tasa de arribos y llegadas debe ser diferente de cero, debido a que al evaluar estos valores en cero se indefinen algunas variables de salida, por lo tanto:

$$\lambda \neq 0 \tag{3.3}$$

$$\mu \neq 0 \tag{3.4}$$

En el programa esta condición se indicó de la siguiente manera:

```
If ArrivalRate >= ServiceRate Then
    MsgBox("Error: Not Stable System, Overflow!")


Else
    If ServiceRate = 0 Then
        MsgBox("Error: Infinite with zero!,give another number")
```

Figura 3-11 Código de no saturación para el modelo M/M/1.

De esta manera hasta que no se cumplan estás condiciones no se pueden obtener los datos de salida. La interfase gráfica realizada para este sistema es la siguiente:

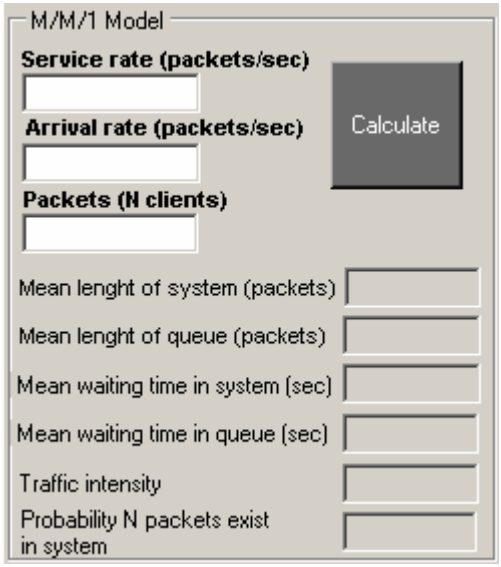


Figura 3-12 Interfase gráfica del modelo M/M/1.

3.2.2 Diseño del modelo M/M/1/K

En el modelo M/M/1/K, a diferencia del modelo M/M/1, se tiene una capacidad finita del sistema denotada por la letra K, y es por ello que este sistema nunca se satura por la intensidad de tráfico; sin embargo, se tendrá que seguir tomando en cuanta que la

tasa de arribo y de servicio no podrán ser igual a cero debido a que algunas variables de salida por medio de las fórmulas se indefinen. En este sistema la condicionante que se tiene es que el número total de paquetes sea menor a los N paquetes, es decir:

$$K < N \tag{3.5}$$

donde:

K = Número total de paquetes que caben en el sistema, y

N = Paquetes en el sistema.

Es por ello que se utiliza el siguiente código para establecer las condiciones de seguridad, para que no se indefina el sistema:

```
If ServiceRate2 = 0 Or ArrivalRate2 = 0 Then
   MsgBox("Error cannot divide by zero!")

Else
   If TrafficIntensity2 = 1 Then
       If Packets2 >= K Then
          MsgBox("Error: Packets Must be minor than Total N of Packets")
```

Figura 3-13 Código de condición para el modelo M/M/1/K.

Para este sistema se tienen dos casos, cuando la tasa de arribo y de llegada eran iguales y cuando eran distintos, es decir cuando la intensidad del tráfico era igual a uno y diferente de uno respectivamente. Por lo tanto:

$$si\{\mu = \lambda\} \rightarrow \rho = 1(caso\_1) \tag{3.6}$$

$$si\{\mu \neq \lambda\} \rightarrow \rho \neq 1(caso\_2) \tag{3.7}$$

Para cada caso se diseñó su sistema correspondiente. Finalmente, la interfase gráfica para este sistema es la siguiente:

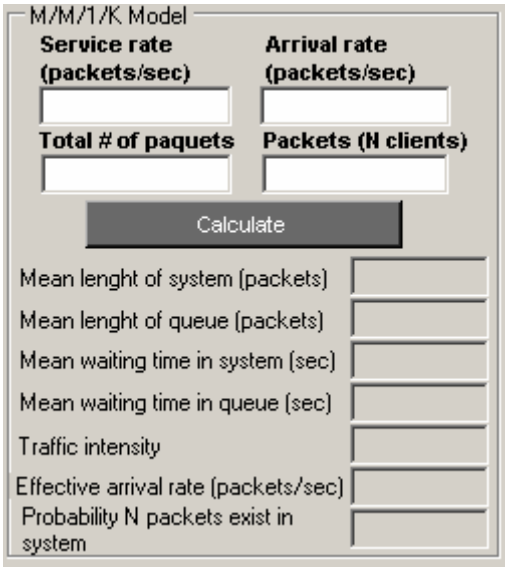


Figura 3-14 Interfase gráfica del modelo M/M/1/K.

3.2.3 Diseño del modelo M/M/c

El sistema M/M/c es más complejo que los sistemas mencionados con anterioridad, M/M/1 y M/M/1/K, debido a que la solución de las ecuaciones, mencionadas en el capítulo 2, son resueltas sumando series finitas. Este sistema se basa en una sola cola pero con la grata diferencia de que se tiene un número de servidores finito. En este caso, el usuario podrá seleccionar si quiere utilizar 1, 2, 3, 4, 5 ó 6 servidores.

Para este sistema se tiene, al igual que en el modelo M/M/1, una cola con capacidad infinita para la cual tenemos que buscar el estado estable o de no saturación. Para ello se tiene que la multiplicación de los servidores y la tasa de arribo deberá ser menor a la tasa de servicio; es decir, la intensidad de tráfico deberá ser menor a uno. Además, en este sistema también se tiene que cuidar que tanto la tasa de servicio como la de arribo sean diferentes de cero, debido a que algunas variables de

salida son indefinidas por las fórmulas. Por lo tanto tenemos las siguientes condiciones:

$$\rho = \frac{\lambda}{c\mu} < 1 \tag{3.8}$$

$$\mu \neq 0 \tag{3.9}$$

$$\lambda \neq 0 \tag{3.10}$$

donde:

ρ = Intensidad de tráfico en el sistema,

λ = Número medio de llegadas por unidad de tiempo,

c = Número de servidores en el sistema, y

μ = Número medio de paquetes que el servidor es capaz de atender por unidad de tiempo.

Como se muestra en la figura 3-15, si el usuario no selecciona nada en la variable de servidores, el programa indicará que se debe de elegir un número de servidores; si lo elige y no está vacío entonces procederá a tomar un valor la variable c, la cual es calculada al sumar el índice del control mas uno, ya que el valor del índice inicialmente es cero. Una vez que se obtiene el valor de c, se ingresa al programa y se verifican las condiciones de saturación. Una vez que pasa las condiciones de saturación prosigue al cálculo de las variables.

```
If cbs3.SelectedItem = Nothing Then
    MsgBox("Please select a Number of Servers")

Else
    C = cbs3.SelectedIndex + 1
    TI3 = ArrivalRate3 / (ServiceRate3 * C)
    If C = 1 Then
        If ArrivalRate3 >= (ServiceRate3 * C) Then
            MsgBox("Error: Not Stable System, Overflow!")

        Else
            If ServiceRate3 = 0 Or ArrivalRate3 = 0 Then
                MsgBox("Error: Infinite with zero!")
```

Figura 3-15 Código de no saturación para el modelo M/M/1.

En la programación del software se utilizaron fórmulas definidas para ρ_0 y L_q, de las cuales se pueden obtener las demás variables de salida con mucha facilidad. Considerando que estas fórmulas sólo pueden ser usadas en el estado estable, que es el caso que estamos diseñando, las series desarrolladas se pueden observar en la siguiente figura:

Núm.de servi-dores	ρ_0	L_q
1	$1-\rho$	$\frac{\rho^2}{1-\rho}$
2	$\frac{1-\rho}{1+\rho}$	$\frac{2\rho^3}{1-\rho^2}$
3	$\frac{2(1-\rho)}{2+4\rho+3\rho^2}$	$\frac{9\rho^4}{2+2\rho-\rho^2-3\rho^3}$
4	$\frac{3(1-\rho)}{3+9\rho+12\rho^2+8\rho^3}$	$\frac{32\rho^5}{3+6\rho+3\rho^2-4\rho^3-8\rho^4}$

5	$\frac{24(1-\rho)}{24+96\rho+180\rho^2+200\rho^3+124\rho^4}$	$\frac{625\rho^6}{24+72\rho+84\rho^2+20\rho^3-75\rho^4-125\rho^5}$
6	$\frac{5(1-\rho)}{5+25\rho+60\rho^2+90\rho^4+54\rho^5}$	$\frac{324\rho^7}{5+20\rho+35\rho^2+30\rho^3-36\rho^5-54\rho^6}$

Figura 3-16 Tabla de resultados para el modelo M/M/c en estado estable [13].

Por último, se tiene que la interfase gráfica para el modelo M/M/c es:

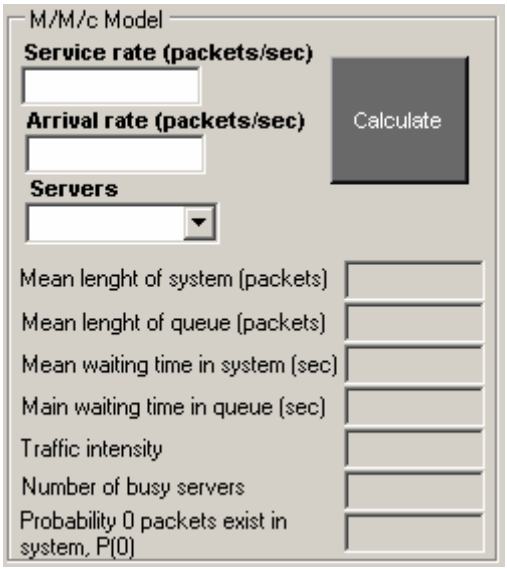


Figura 3-17 Interfase gráfica del modelo M/M/c.

3.3 Otras consideraciones en el diseño

El diseño se pensó de tal manera que fuese simple y fácil de ocupar por los usuarios, es por ello que se agregó un menú de edición para que el usuario pueda cambiar libremente entre el sistema de telefonía y de colas. Adicionalmente, se agregó la herramienta de ayuda para poder tener una guía rápida de los dos sistemas, la cual

sirve para saber las fórmulas y conceptos básicos de ambas. A continuación se muestran las opciones que tienen el menú de ayuda y el menú de edición.

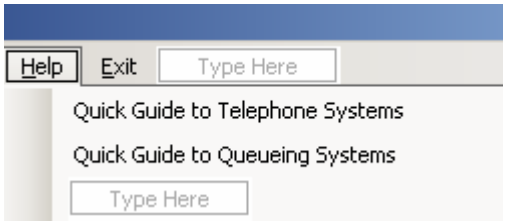


Figura 3-18 Opciones en el menú ayuda.

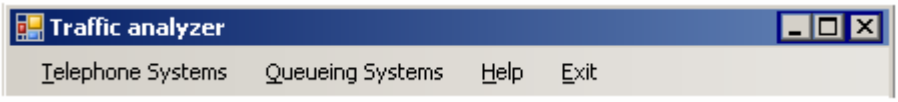


Figura 3-19 Menú de edición del “Traffic Analyzer”.