

4 Pruebas y análisis del software

En este capítulo se presentan una serie de simulaciones donde se analiza el desempeño de ambos sistemas programados en cuanto a exactitud con otros softwares que se encuentran en el mercado y en cuanto al comportamiento entre ellos. En la primera sección se realizan pruebas de efectividad en los sistemas de telefonía para identificar algún posible problema de programación o verificar la funcionalidad del mismo. En la segunda sección se realiza en primera instancia una prueba para verificar que sí esté funcionando el programa de manera efectiva. Finalmente, se realizarán análisis y resultados de ambos sistemas.

Cabe señalar que los softwares que se utilizan para la comparación son: “TrafficFinder!” y “Erlang Calculator”, los cuales son marcas registradas que se encuentran en el mercado para compra. El software “TrafficFinder!” es una marca

registrada por “Staff Software, Inc.”, mientras que el software “Erlang Calculator” es una marca registrada de “Westbay Engineers Limited” desde el año de 1996.

4.1 Pruebas del sistema de telefonía

Para este sistema la primera prueba que se realiza tiene la finalidad de comprobar que se estén leyendo correctamente las tablas de distribución y que se están obteniendo los datos deseados. Así pues, suponiendo dos sistemas diferentes pero con la misma fórmula de probabilidad, se tendrán:

- Caso 1

Variables de entrada: 1200 llamadas, 140 segundos de ACHT, un grado de servicio igual a 0.02% con la distribución de Erlang B, para este caso con respecto a la ecuación (0.31) se obtiene que:

$$Carga_ofrecida = \left(\frac{(1200 \text{ llamadas}) \left(140 \frac{\text{segundos}}{\text{llamada}} \right)}{3600 \text{ seg.}} \right) \approx 4.6667 \text{ Erlangs} \quad (4.1)$$

- Caso 2

Variables de entrada: 50 suscriptores que generan 0.2 Erlangs de tráfico cada uno de ellos, un grado de servicio igual a 0.03% con la distribución de Erlang B, para este caso con respecto a la ecuación (0.32), se tiene que:

$$Carga_ofrecida = (50 \text{ Suscriptores}) \left(0.2 \frac{\text{Erlangs}}{\text{Suscriptor}} \right) = 10 \text{ Erlangs} \quad (4.2)$$

Una vez que se obtuvieron las cargas ofrecidas para ambos casos, se procede a buscar el número de troncales en cada uno de los sistemas, para ello se necesitan los grados de servicios que ofrecen cada uno de ellos y partir de la tabla de distribución de Erlang B se podrán obtener. Hay que tomar en cuenta que como se mencionó en el capítulo anterior, las columnas están inicializadas en 0 al igual que las filas, por lo tanto se tendrá lo siguiente:

	j=0+1=1	j=1+1=2	j=2+1=3 ...			
	0.01%	0.02%	0.03%	0.04%		
erlangb.txt - Notepad						
File Edit Format View Help						
0	0.0001	0.0002	0.0003	0.0004	0.0005	0.0006
1	0.0142	0.0202	0.0248	0.0287	0.0321	0.0353
2	0.0868	0.1102	0.1269	0.1403	0.1517	0.1618
3	0.2347	0.2825	0.3152	0.3409	0.3624	0.381
4	0.452	0.527	0.5773	0.6163	0.6486	0.6764
5	0.7282	0.8316	0.8999	0.9524	0.9957	1.0327
6	1.0541	1.186	1.2723	1.3382	1.3922	1.4384
7	1.4219	1.582	1.686	1.7651	1.8297	1.8848
8	1.8256	2.0133	2.1346	2.2266	2.3016	2.3654
9	2.2601	2.4749	2.6132	2.7177	2.8028	2.875
10	2.7216	2.9629	3.1178	3.2345	3.3294	3.4098
11	3.2069	3.4741	3.645	3.7737	3.8781	3.9666
12	3.7133	4.0058	4.1925	4.3328	4.4465	4.5428
13	4.2387	4.5559	4.7579	4.9095	5.0324	5.1363
14 + 1 = 15 troncales	4.7811	5.1224	5.3395	5.5022	5.6339	5.7453
15	5.3389	5.7039	5.9357	6.1092	6.2496	6.3682
16	5.9109	6.2991	6.5452	6.7294	6.8782	7.004
17	6.4958	6.9067	7.1669	7.3614	7.5186	7.6513
18	7.0927	7.5258	7.7997	8.0045	8.1698	8.3093
19	7.7005	8.1555	8.443	8.6577	8.831	8.9772
20	8.3186	8.795	9.0958	9.3203	9.5014	9.6543
21	8.9462	9.4436	9.7575	9.9916	10.1805	10.3397
22 + 1 = 23 troncales	9.5826	10.1008	10.4275	10.6711	10.8675	11.0331
23	10.2274	10.766	11.1052	11.3582	11.562	11.7339
24	10.88	11.4386	11.7903	12.0524	12.2636	12.4416
25	11.54	12.1183	12.4822	12.7534	12.9717	13.1558
26	12.2069	12.8047	13.1806	13.4606	13.6861	13.8761
27	12.8803	13.4973	13.8851	14.1739	14.4063	14.6022

Figura 4-1 Obtención de troncales con la tabla de Erlang B.

Como puede observarse en la figura 4-1 las troncales necesarias para obtener un sistema eficiente serán de 15 y 23 troncales, respectivamente. De esta manera, utilizando el “Traffic Analyzer” obtendremos el siguiente resultado:

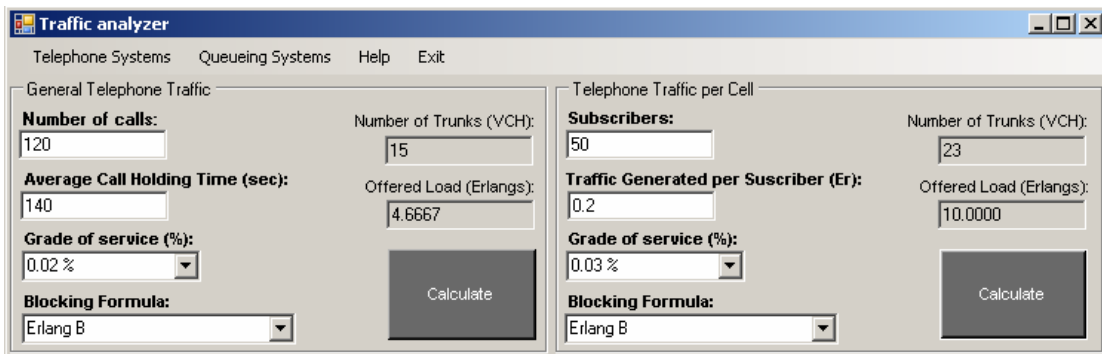


Figura 4-2 Primera simulación en los sistemas telefónicos.

Por lo tanto podemos aprobar la primera prueba que se realizó al sistema, puesto que sí está respetando la lógica con la que se diseñó el programa. Ahora bien, proseguiremos con las demás pruebas, es importante mencionar que hacer comparaciones con otros softwares es un buen recurso para identificar algún problema de programación y poder mejorar el software. De esta manera se realizaron las siguientes pruebas:

- Prueba #2 “Traffic Analyzer” vs. “TrafficFinder!”

Se realizarán pruebas para comprobar la funcionalidad del programa, consecuentemente, tenemos lo siguiente:

- Prueba 2a.- Para empezar, en el “TrafficFinder!” se ingresa un grado de servicio de 0.02, equivalente a 2%, con un tráfico ofrecido de 52 Erlangs y se selecciona la fórmula de bloqueo de Erlang B. Para el caso del “Traffic Analyzer”, se selecciona un GoS igual a 2%, con 40 suscriptores generando un tráfico de 1.3 Erlangs cada uno y seleccionando la fórmula de bloqueo de Erlang B. Como se observa en la figura 4-3 y 4-4 la carga ofrecida y el número de troncales

necesarias para esos sistemas es la misma, por lo tanto esta prueba resulta satisfactoria.

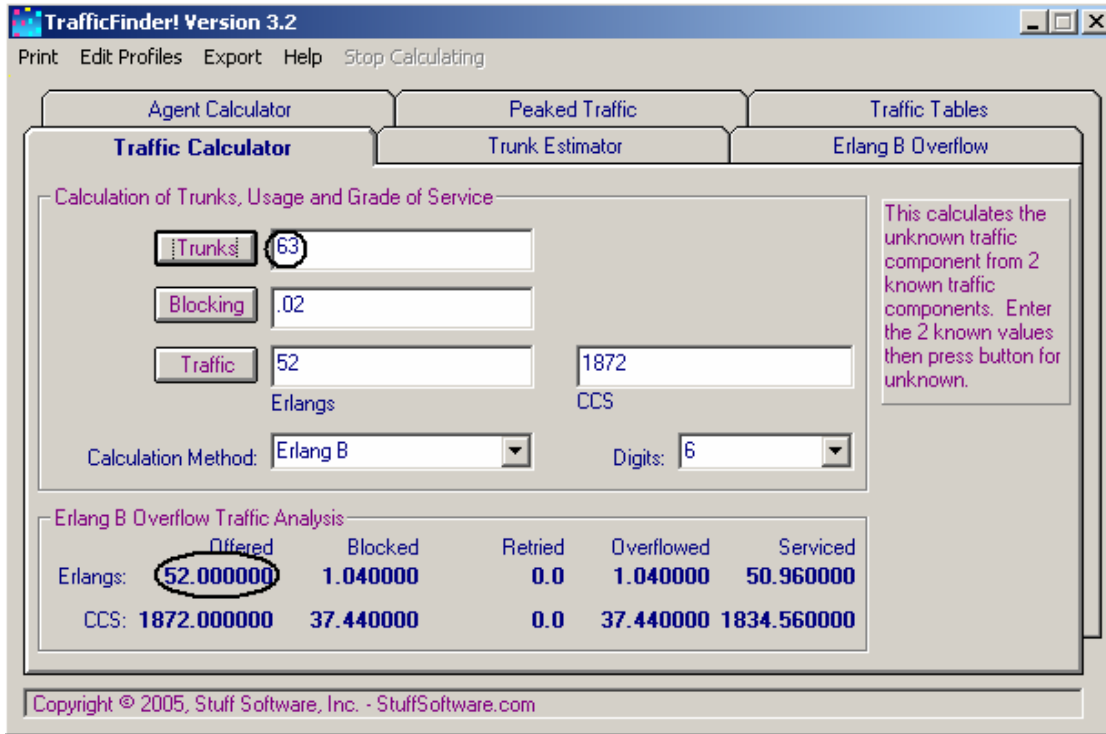


Figura 4-3 Prueba 2a con el “TrafficFinder!”.

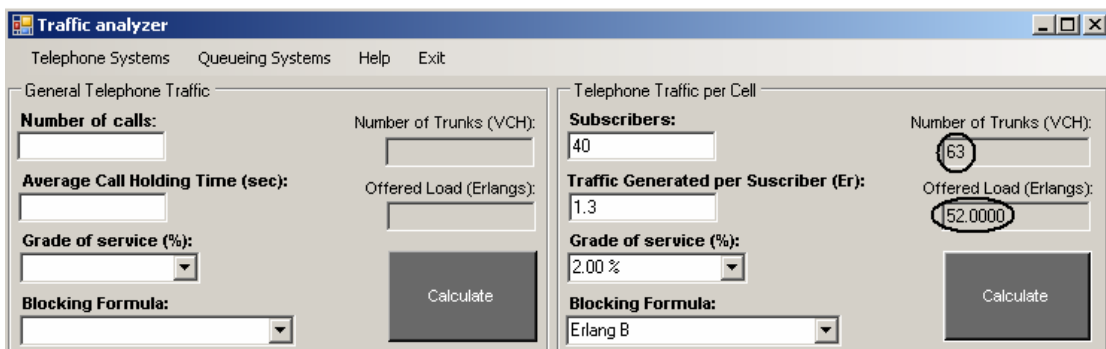


Figura 4-4 Prueba 2a con el “Traffic Analyzer”.

- o Prueba 2b.- De manera análoga a la prueba 2a, en el “TrafficFinder!” se ingresa un GoS equivalente a 1%, con un tráfico de 143 Erlangs, utilizando la fórmula de bloqueo de Erlang B extendido al 50%;

mientras que, en el “Traffic Analyzer” se ingresa un GoS igual al 1%, con 130 suscriptores que están generando 1.1 Erlangs de tráfico cada uno, utilizando la fórmula de bloqueo de Erlang B extendido al 50%. Como se observa en la figura 4-5 y 4-6 la carga ofrecida y el número de troncales necesarias para esos sistemas es la misma, por lo tanto esta prueba resulta satisfactoria.

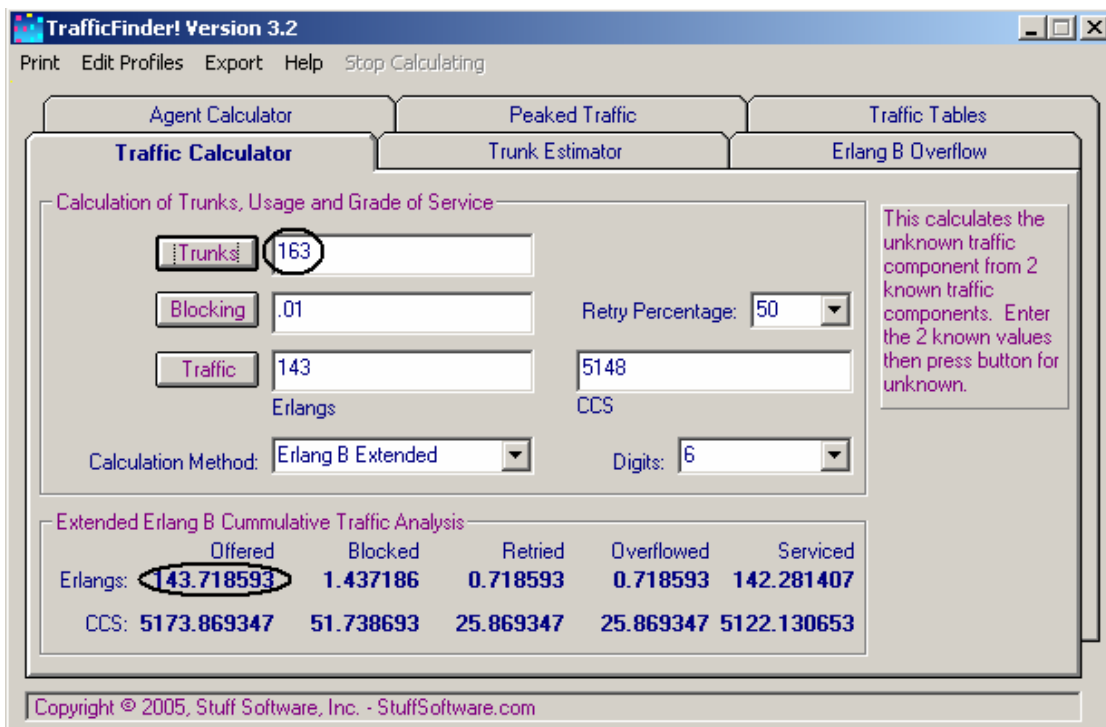


Figura 4-5 Prueba 2b con el “TrafficFinder!”.

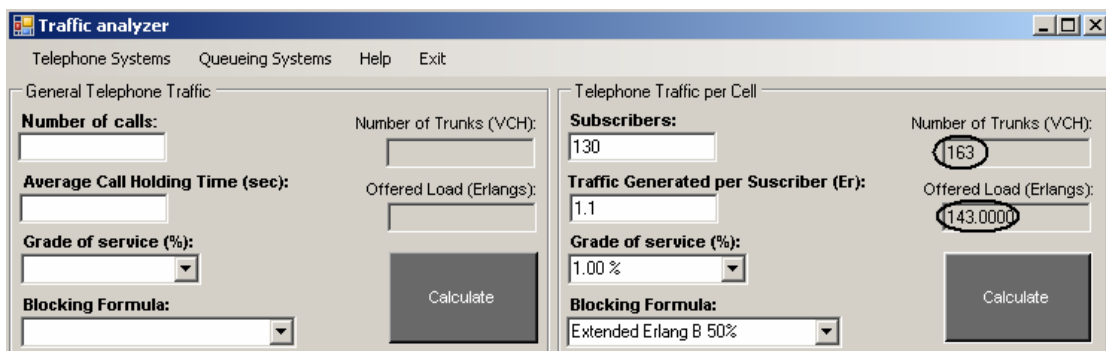


Figura 4-6 Prueba 2b con el “Traffic Analyzer”.

- Prueba 2c.- Para esta prueba se ingresan las mismas variables en ambos softwares, siendo éstas las siguientes: 500 llamadas por hora, un ACHT de 120 segundos, con un GoS aproximado al 4% y utilizando la fórmula de bloqueo de Erlang C. Como se puede observar, el tráfico en el sistema es de 16.6667 Erlangs y se necesitan 25 troncales para obtener un sistema eficiente. Los resultados se pueden verificar en las figuras 4-7 y 4-8, de manera que esta prueba es aprobada.

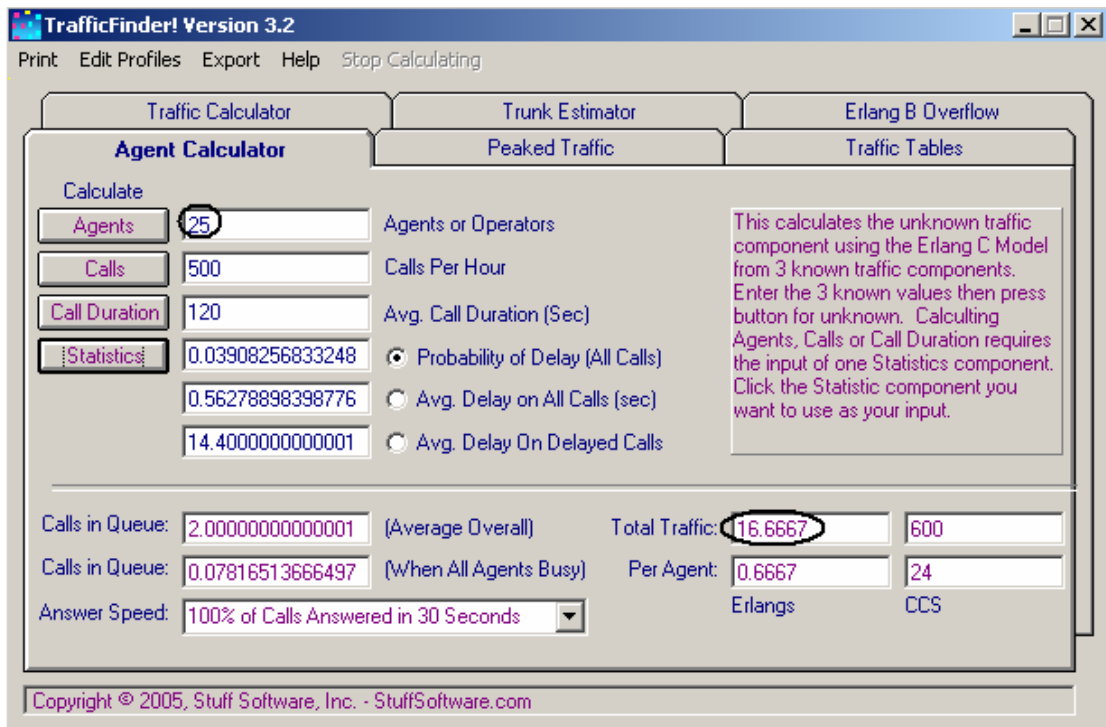


Figura 4-7 Prueba 2c con el “TrafficFinder!”.

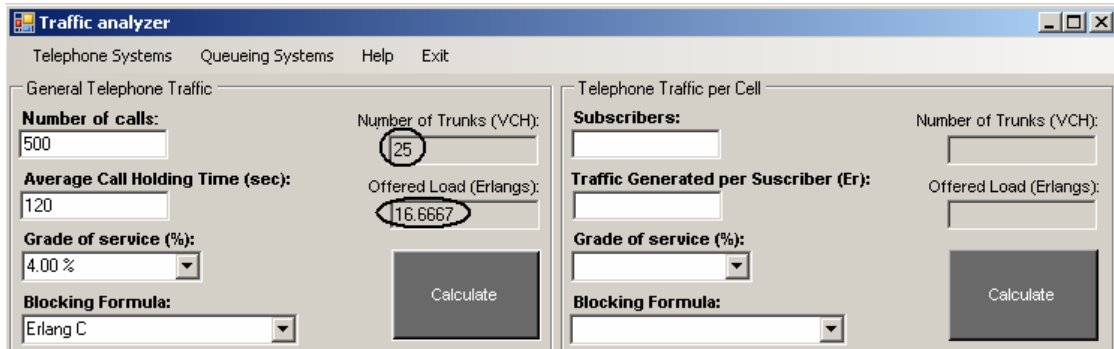


Figura 4-8 Prueba 2c con el “Traffic Analyzer”.

- Prueba 2d.- Finalmente para esta prueba, se utiliza la fórmula de Poisson; para la cual se ingresa un GoS de 0.3%, con un tráfico ofrecido al 100% de 178 Erlangs; siendo que para este sistema se necesitarán 217 troncales, como se puede observar satisfactoriamente en la figura 4-9 y 4-10.

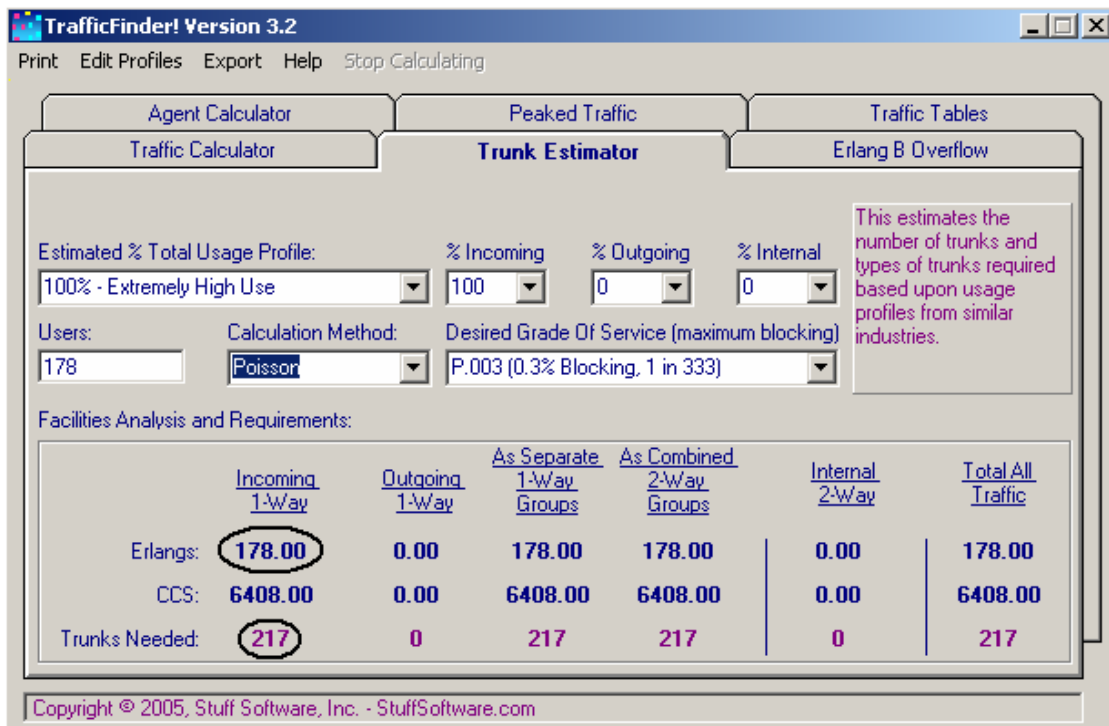


Figura 4-9 Prueba 2d con el “TrafficFinder!”.

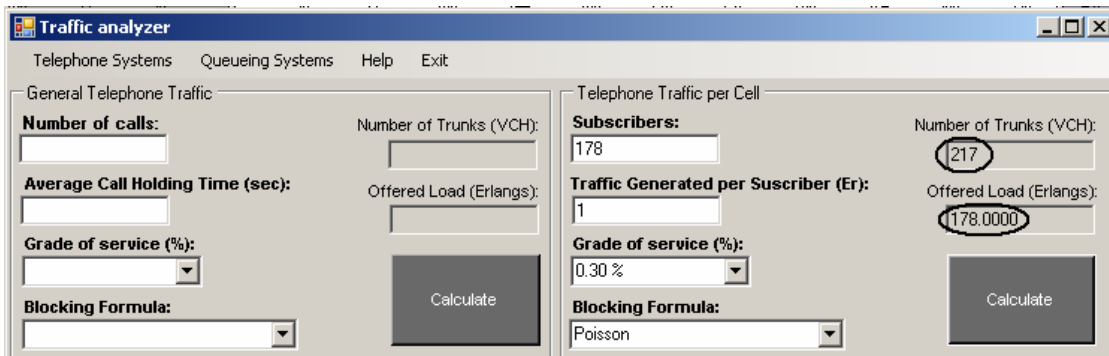


Figura 4-10 Prueba 2d con el “Traffic Analyzer”.

Así pues, se hicieron 4 pruebas con las distintas fórmulas de probabilidad, las 4 coincidieron en resultados. Se cambiaron las variables para todos los casos y así es como aceptamos esta prueba.

- Prueba #3 “Traffic Analyzer” vs. “Erlang Calculator”

Además del software “TrafficFinder!”, otro muy popular es el “Erlang Calculator”, con el cual también se hicieron un par de pruebas para comprobar la funcionalidad del programa, consecuentemente, tenemos lo siguiente:

- Prueba 3a.- Para esta prueba se ingresaron en ambos softwares las siguientes variables: un GoS igual a 2%, un tráfico ofrecido de 100 Erlangs, ocupando la fórmula de bloqueo de Erlang B. Como se puede observar en la figura 4-11, se obtienen 113 troncales como resultado en ambos programas.

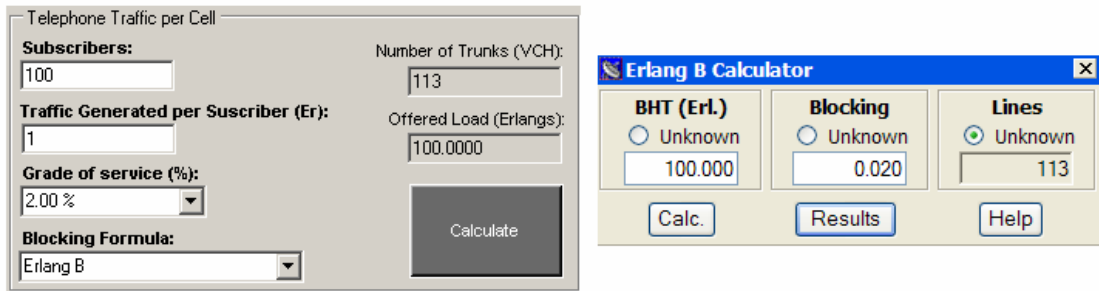


Figura 4-11 Prueba 3a, “Traffic Analyzer” contra “Erlang Calculator” con 100 suscriptores.

- o Prueba 3b.- Para esta prueba se ingresaron los mismos datos que en la prueba 3a; sin embargo, se disminuyó el tráfico ofrecido a 50 Erlangs ó 50 suscriptores generando un tráfico de 1 Erlang cada uno. Se observa exitosamente en la figura 4-12, que para ambos casos se tiene un resultado de 61 troncales.

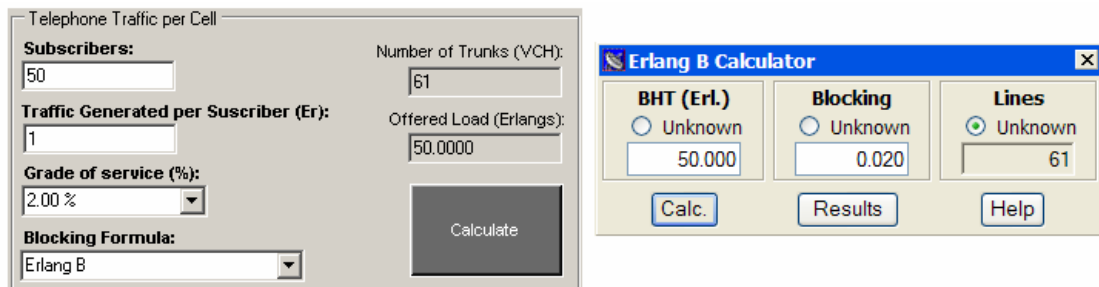


Figura 4-12 Prueba 3b, “Traffic Analyzer” contra “Erlang Calculator” con 50 suscriptores.

4.2 Pruebas de los sistemas de colas

En este tipo de sistema se analizó el comportamiento que tienen las colas en los sistemas, sin embargo lo que se toma en consideración en primer instancia es que el sistema se encuentre en el estado estable o no saturado, es por eso que se irán poniendo los casos que se previnieron para que el programa no se ciclará.

- Prueba 4. En el caso de la cola M/M/1 evitar que la tasa de servicio sea mayor que la de arribo, para poder estar en el estado estable. En la figura 4-13 puede observarse que al no cumplirse la condición el usuario no podrá seguir el proceso, hasta poner una tasa de arribo menor.

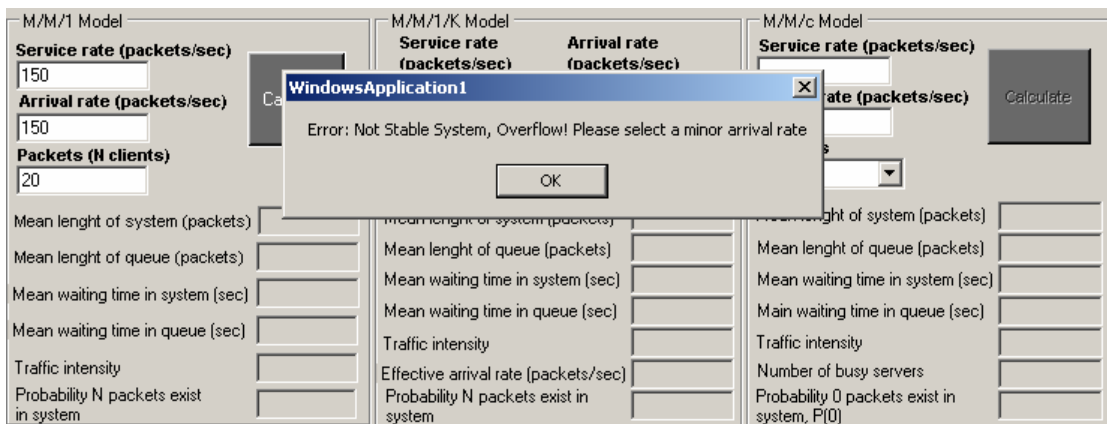


Figura 4-13 Prueba 4, sistema saturado para modelo M/M/1.

- Prueba 5. Para el modelo M/M/1/K el número total de paquetes debe de ser menor a la probabilidad de que N paquetes existan, además de que la tasa de arribo y servicio deben de ser distintas de cero para que no se indefinan; es por ello que a través de la figura 4-14 y 4-15 se puede observar que el programa se ejecutará hasta obtener datos que hagan al sistema estable.

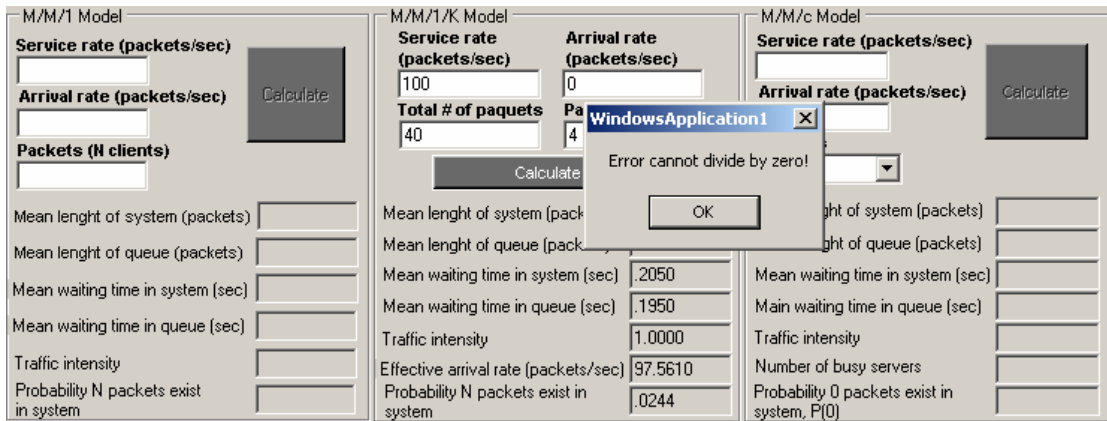


Figura 4-14 Prueba 5, sistema indefinido para el modelo M/M/1/K.

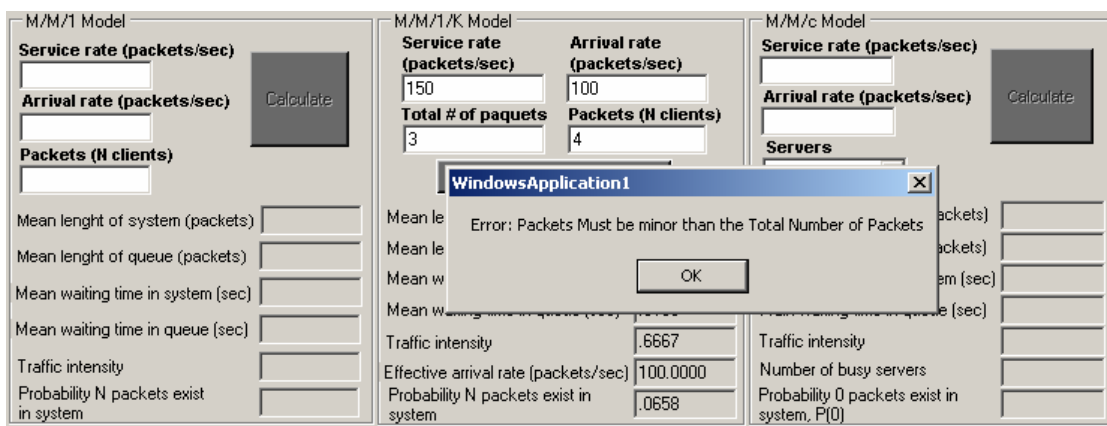


Figura 4-15 Prueba 5, el número total de paquetes K debe ser mayor al número de paquetes para el modelo M/M/1/K.

- Prueba 6. Para el modelo M/M/c, se tiene la consideración de que la tasa de servicio multiplicado por el número de servidores, c, sea mayor al de arriba para que el sistema no se sature, además de que éstos no pueden ser igual a cero porque indefinen el sistema. En las figuras siguientes se muestran las pruebas de que el usuario necesita tener las variables en estado estable para poder continuar con la operación.

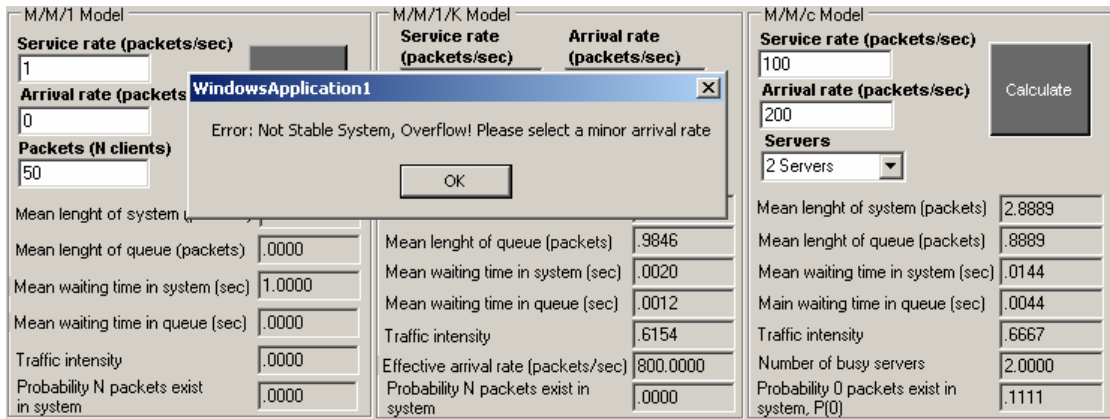


Figura 4-16 Prueba 6, sistema saturado para el modelo M/M/c.

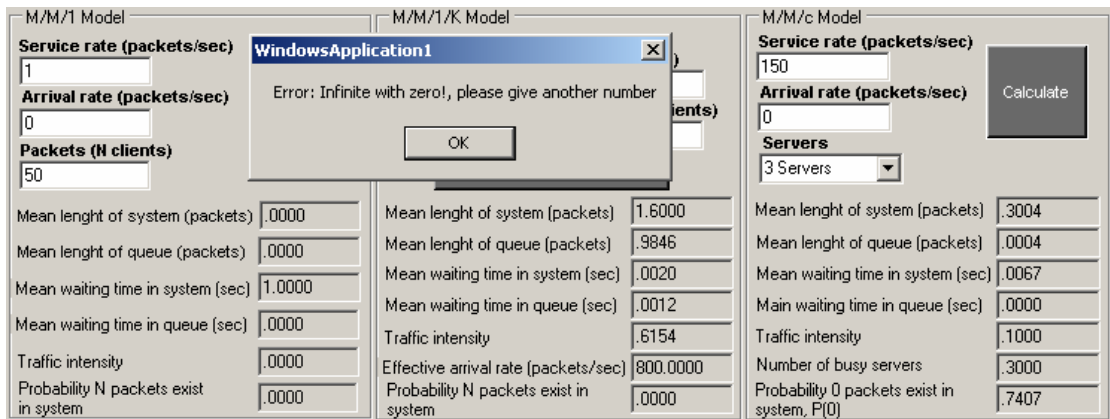


Figura 4-17 Prueba 6, sistema indefinido para el modelo M/M/c.

De esta manera pudimos comprobar que el sistema siempre va a respetar el estado estable o de no saturación y al tener puras variables estables el programa nos arrojará los cálculos de nuestras incógnitas, así como se muestra en la figura 4-18:

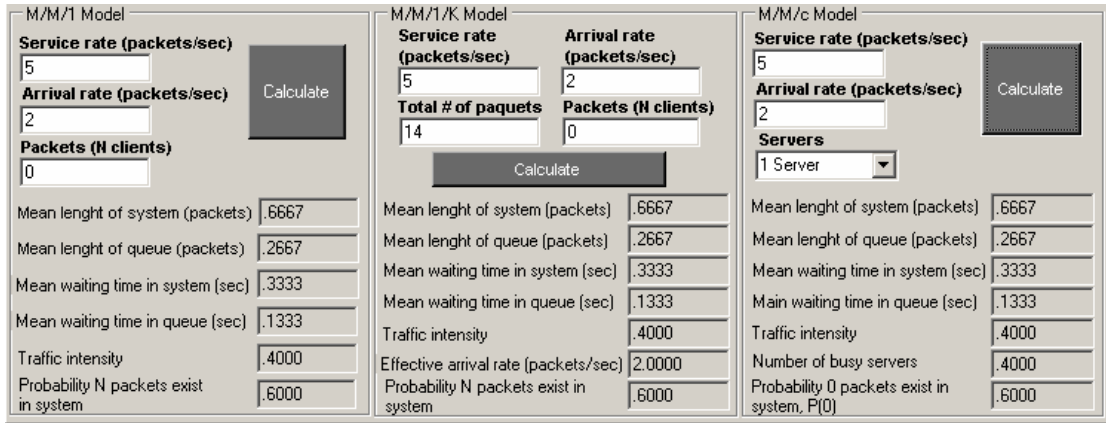


Figura 4-18 Prueba 7, los tres sistemas están en el estado estable.

4.3 Análisis y resultados de los sistemas de telefonía

Para poder hacer un análisis de lo que es tráfico en telefonía, se realizó una serie de tablas y gráficas con ayuda del “Traffic Analyzer” que nos especificarían el tipo de comportamiento que se presentan en cada tipo de distribución que se utiliza. En primer lugar se crearon tablas para cada una de las distribuciones que presenta el software, Erlang B, Erlang B Extendido 50%, Erlang C y Poisson, variando la carga ofrecida de 10 en 10 Erlangs para los diferentes grados de servicio a evaluar, los cuales fueron 0.2, 2 y 20 %. Una vez almacenados los datos y obteniendo el número de troncales necesarios para cada caso, se graficaron y en ellas se puede observar el comportamiento del tráfico para los tres grados de servicio. De esta manera obtuvimos lo siguiente:

En la figura 4-19 y 4-20 se muestra una tabla y una gráfica, respectivamente, donde se realizan comparaciones entre los grados de servicio de 0.2, 2 y 20%, utilizando un rango de 10 hasta 160 Erlangs para la tabla de Erlang B. Se puede

observar que la diferencia entre los 3 resultados es bastante pronunciada, siendo que a mayor grado de servicio se necesitan menos troncales.

Tabla de distribución: Erlang B			
Grado de servicio \ Carga ofrecida	0.20%	2%	20%
10Er	20	17	11
20Er	34	28	19
30Er	46	39	27
40Er	58	50	36
50Er	69	61	44
60Er	81	71	52
70Er	92	82	60
80Er	103	92	68
90Er	114	103	76
100Er	125	113	84
110Er	136	123	92
120Er	147	133	100
130Er	158	144	108
140Er	169	154	116
150Er	180	164	124
160Er	191	174	132

Figura 4-19 Tabla de comparación entre los grados de servicio 0.2, 2 y 20% utilizando la fórmula de Erlang B.

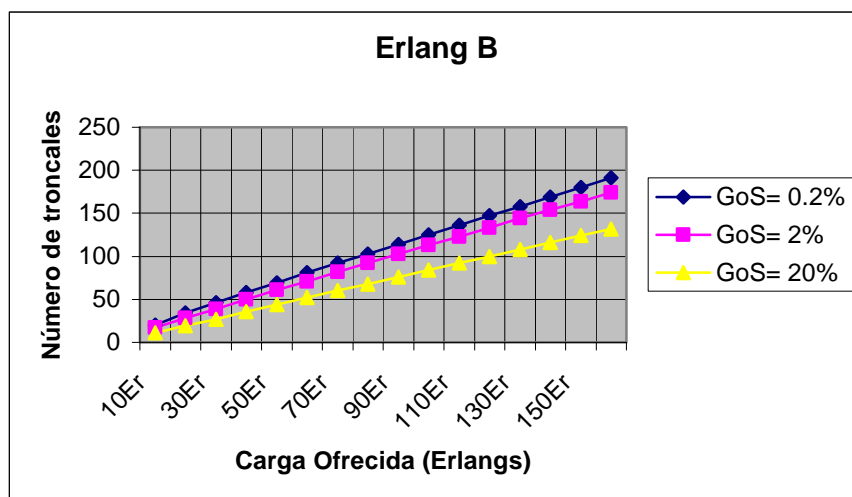


Figura 4-20 Gráfica de comparación entre los grados de servicio (azul) 0.2, (rosado) 2 y (amarillo) 20% utilizando la fórmula de Erlang B.

En la figura 4-21 y 4-22 se muestra una tabla y una gráfica, respectivamente, donde se realizan comparaciones entre los grados de servicio de 0.2, 2 y 20%, utilizando un rango de 10 hasta 160 Erlangs para la tabla de Erlang B extendido un 50%. A continuación se pueden observar los resultados:

Tabla de distribución: Erlang Extendido 50%			
<i>Grado de servicio</i>	0.20%	2%	20%
<i>Carga ofrecida</i>			
10Er	20	17	12
20Er	34	29	21
30Er	46	40	30
40Er	58	51	39
50Er	69	61	48
60Er	81	72	57
70Er	92	83	66
80Er	103	93	75
90Er	115	104	84
100Er	126	114	93
110Er	137	124	102
120Er	148	135	111
130Er	158	145	120
140Er	169	155	129
150Er	180	165	137
160Er	191	176	146

Figura 4-21 Tabla de comparación entre los grados de servicio 0.2, 2 y 20% utilizando la fórmula de Erlang B Extendido 50%.

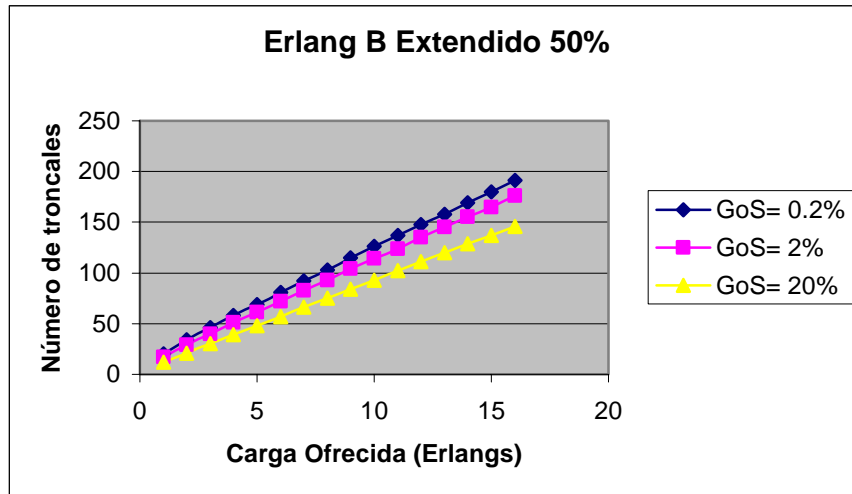


Figura 4-22 Gráfica de comparación entre los grados de servicio (azul) 0.2, (rosado) 2 y (amarillo) 20% utilizando la fórmula de Erlang B Extendido 50%.

En la figura 4-23 y 4-24 se muestra una tabla y una gráfica, respectivamente, donde se realizan comparaciones entre los grados de servicio de 0.2, 2 y 20%, utilizando un rango de 10 hasta 160 Erlangs para la tabla de Erlang C. Los resultados obtenidos fueron los siguientes:

Tabla de distribución: Erlang C			
Grado de servicio \ Carga ofrecida	0.20%	2%	20%
10Er	21	18	14
20Er	35	31	26
30Er	48	43	37
40Er	61	55	48
50Er	73	66	58
60Er	85	78	69
70Er	96	89	80
80Er	108	100	90
90Er	120	112	101
100Er	131	123	111
110Er	143	134	122
120Er	154	145	133
130Er	165	156	143
140Er	177	167	153
150Er	188	177	164
160Er	199	188	174

Figura 4-23 Tabla de comparación entre los grados de servicio 0.2, 2 y 20% utilizando la fórmula de Erlang C.

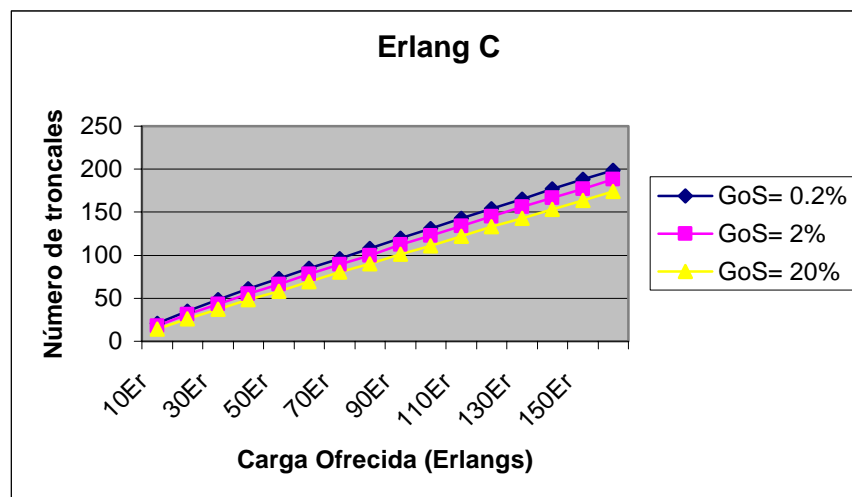


Figura 4-24 Gráfica de comparación entre los grados de servicio (azul) 0.2, (rosado) 2 y (amarillo) 20% utilizando la fórmula de Erlang C.

En la figura 4-25 y 4-26 se muestra una tabla y una gráfica, respectivamente, donde se realizan comparaciones entre los grados de servicio de 0.2, 2 y 20%,

utilizando un rango de 10 hasta 160 Erlangs para la tabla de Poisson. A continuación se muestran los resultados:

Tabla de distribución: Poisson			
<i>Grado de servicio</i>	0.20%	2%	20%
<i>Carga ofrecida</i>			
10Er	21	18	14
20Er	35	31	25
30Er	48	43	36
40Er	60	55	46
50Er	73	66	57
60Er	84	77	67
70Er	96	89	78
80Er	108	100	88
90Er	119	111	99
100Er	131	122	109
110Er	142	133	120
120Er	154	144	130
130Er	165	155	141
140Er	176	166	151
150Er	187	177	161
160Er	199	188	172

Figura 4-25 Tabla de comparación entre los grados de servicio 0.2, 2 y 20% utilizando la fórmula de Poisson.

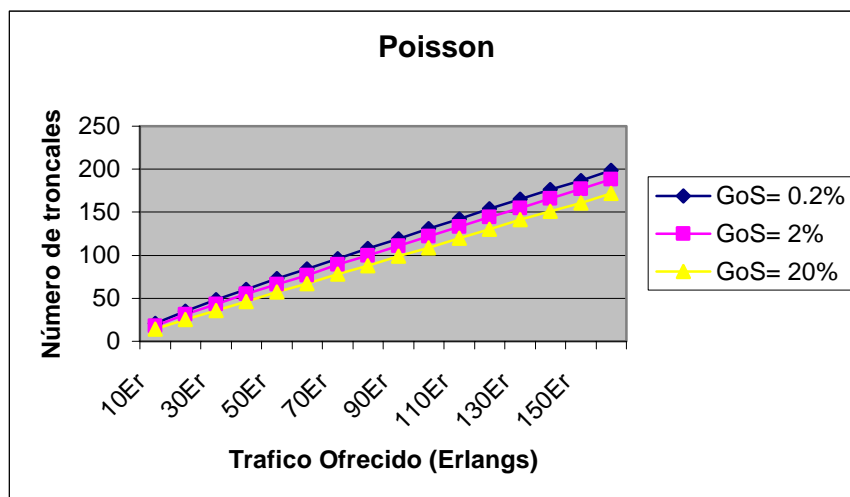


Figura 4-26 Gráfica de comparación entre los grados de servicio (azul) 0.2, (rosado) 2 y (amarillo) 20% utilizando la fórmula de Poisson.

En estas gráficas se pudo observar que el comportamiento de Poisson con respecto a la de Erlang C es muy parecido, el número de troncales necesarios entre ellos es casi igual. Otra cosa importante por señalar es la tabla de distribución de Erlang B, ya que ésta no requiere de tantos canales, o troncales como se ha estado manejando, por lo mismo de que el usuario únicamente hace un intento de llamada, mientras que en Erlang C por ejemplo, se manejan colas que almacenan los trabajos hasta que se establece el servicio, es decir la llamada.

Al obtenerse todos los datos, se procedió a crear una tabla para comparar las tablas de distribución utilizando una constante de grado de servicio del 2%. Para este caso se seguirá utilizando un rango de carga ofrecida de 10 en 10 Erlangs. Por lo tanto, tendremos lo siguiente:

<i>Grado de servicio</i> <i>Carga ofrecida</i>	2% Erlang B	2% Erlang B Extendido 50%	2% Erlang C	2% Poisson
10Er	17	17	18	18
20Er	28	29	31	31
30Er	39	40	43	43
40Er	50	51	55	55
50Er	61	61	66	66
60Er	71	72	78	77
70Er	82	83	89	89
80Er	92	93	100	100
90Er	103	104	112	111
100Er	113	114	123	122
110Er	123	124	134	133
120Er	133	135	145	144
130Er	144	145	156	155
140Er	154	155	167	166
150Er	164	165	177	177
160Er	174	176	188	188

Figura 4-27 Tabla de comparación entre las tablas de distribución de Erlang B, B extendido, C y Poisson utilizando un grado de servicio del 2%.

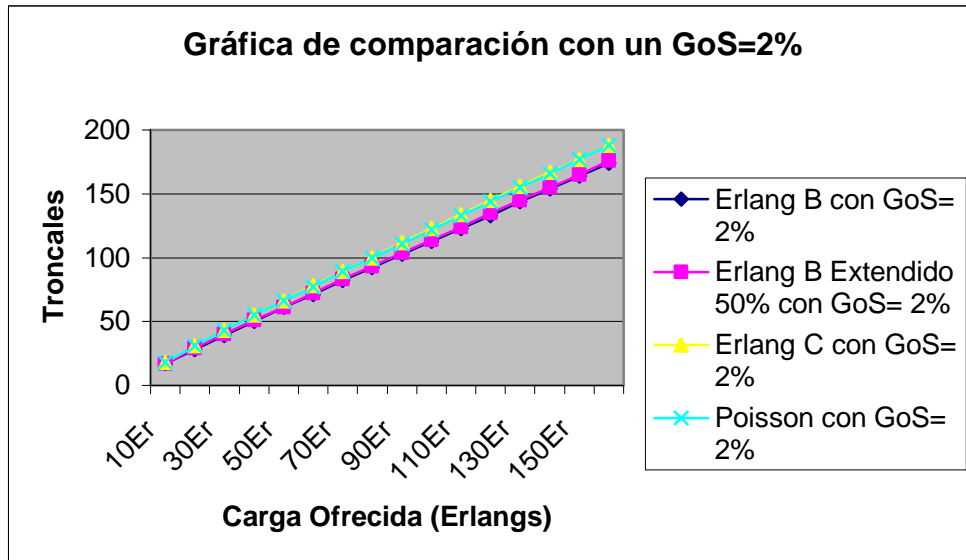


Figura 4-28 Gráfica de comparación entre las tablas de distribución de Erlang B, B extendido, C y Poisson utilizando un grado de servicio del 2%.

Haciendo un “zoom” a la figura 4-28, obtenemos la siguiente gráfica que para efectos de análisis se puede demostrar estadísticamente que el comportamiento de la tabla de Poisson con la de Erlang C es muy parecida, ya que se comportan por medio de colas o almacenamientos de llamadas, es por eso que este tipo de sistemas ocupan más troncales que en las tablas de Erlang B y extendido. Así pues, la tabla de Erlang B es la que ocupa menos troncales para brindar su servicio, esto debido a que el usuario realiza un sólo intento de llamada; sino se realiza la llamada, está es enrutada nuevamente o pérdida. La principal diferencia entre el Erlang B y el B Extendido es que el usuario intenta llamar nuevamente y ese es el porcentaje que se le atribuye a la palabra extendido, es por ello que ocupa un poco más de troncales que el Erlang B.

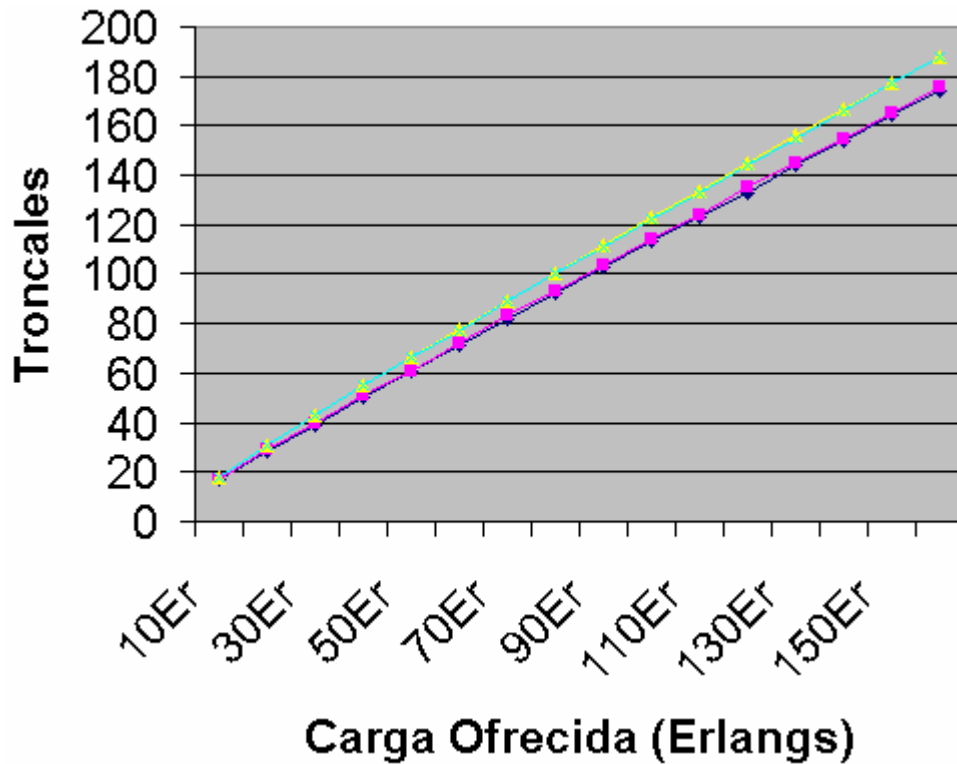


Figura 4-29 Acercamiento de la Figura 4-28.

4.4 Análisis y resultados de los sistemas de colas

Para el análisis de colas se tienen varios supuestos, sin embargo al limitar los sistemas se puede llegar al mismo punto como se muestra en la figura 4-30, donde los 3 sistemas tienen los mismos resultados de salida. Se puede observar que el modelo M/M/1/K tuvo que limitarse a que su cola finita sea de 14 paquetes, mientras que en el modelo M/M/c se tuvo que dejar 1 servidor para efectos de esta prueba.

M/M/1 Model	M/M/1/K Model	M/M/c Model
Service rate (packets/sec) <input type="text" value="5"/>	Service rate (packets/sec) <input type="text" value="5"/>	Service rate (packets/sec) <input type="text" value="5"/>
Arrival rate (packets/sec) <input type="text" value="2"/>	Arrival rate (packets/sec) <input type="text" value="2"/>	Arrival rate (packets/sec) <input type="text" value="2"/>
Packets (N clients) <input type="text" value="0"/>	Total # of paquets <input type="text" value="14"/>	Servers <input type="text" value="1"/>
Calculate	Calculate	Calculate
Mean lenght of system (packets) .6667	Mean lenght of system (packets) .6667	Mean lenght of system (packets) .6667
Mean lenght of queue (packets) .2667	Mean lenght of queue (packets) .2667	Mean lenght of queue (packets) .2667
Mean waiting time in system (sec) .3333	Mean waiting time in system (sec) .3333	Mean waiting time in system (sec) .3333
Mean waiting time in queue (sec) .1333	Mean waiting time in queue (sec) .1333	Main waiting time in queue (sec) .1333
Traffic intensity .4000	Traffic intensity .4000	Traffic intensity .4000
Probability N packets exist in system .6000	Effective arrival rate (packets/sec) 2.0000	Number of busy servers .4000
	Probability N packets exist in system .6000	Probability 0 packets exist in system, P(0) .6000

Figura 4-30 Prueba 1 en programa de colas.

Así pues en la figura 4-31, analizando los sistemas M/M1 y M/M/c, tenemos lo siguiente: el modelo M/M/1 es igual al M/M/c para este caso en específico debido a que se tiene un único servidor por default en el primero y un servidor declarado en el segundo, 1. Sin embargo si se llegará a cambiar el número de servidores la intensidad de tráfico disminuye debido a que los servidores están despachando a los trabajos, paquetes, y así sucesivamente mientras más servidores se estén ocupando como se muestran en las figuras 4-32, 4-33 y 4-34 con 2, 3 y 4 servidores en la cola.

M/M/1 Model	M/M/1/K Model	M/M/c Model
Service rate (packets/sec) <input type="text" value="150"/>	Service rate (packets/sec) <input type="text"/>	Service rate (packets/sec) <input type="text" value="150"/>
Arrival rate (packets/sec) <input type="text" value="100"/>	Arrival rate (packets/sec) <input type="text"/>	Arrival rate (packets/sec) <input type="text" value="100"/>
Packets (N clients) <input type="text" value="0"/>	Total # of paquets <input type="text"/>	Servers <input type="text" value="1"/>
Calculate	Calculate	Calculate
Mean lenght of system (packets) 2.0000	Mean lenght of system (packets)	Mean lenght of system (packets) 2.0000
Mean lenght of queue (packets) 1.3333	Mean lenght of queue (packets)	Mean lenght of queue (packets) 1.3333
Mean waiting time in system (sec) .0200	Mean waiting time in system (sec)	Mean waiting time in system (sec) .0200
Mean waiting time in queue (sec) .0133	Mean waiting time in queue (sec)	Main waiting time in queue (sec) .0133
Traffic intensity .6667	Traffic intensity	Traffic intensity .6667
Probability N packets exist in system .3333	Effective arrival rate (packets/sec)	Number of busy servers .6667
	Probability N packets exist in system	Probability 0 packets exist in system, P(0) .3333

Figura 4-31 Modelos de M/M/1 y M/M/c con número de servidores iguales.

En la figura 4-32 se aprecia como el tiempo medio en los paquetes y en la cola disminuye en comparación al que únicamente tiene 1 servidor. La intensidad de tráfico se decrementa gracias a que existen más agentes brindando el servicio.

The screenshot shows a software interface for an M/M/c Model. It includes input fields for Service rate (150 packets/sec) and Arrival rate (100 packets/sec), a dropdown menu for Servers (set to 2), and a Calculate button. Below these are several output fields with their corresponding values:

Service rate (packets/sec)	150
Arrival rate (packets/sec)	100
Servers	2 Servers
Mean length of system (packets)	.7500
Mean length of queue (packets)	.0833
Mean waiting time in system (sec)	.0075
Main waiting time in queue (sec)	.0008
Traffic intensity	.3333
Number of busy servers	.6667
Probability 0 packets exist in system, P(0)	.5000

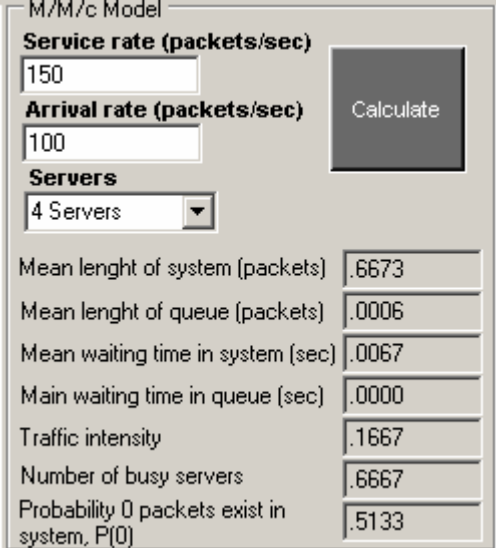
Figura 4-32 Modelo M/M/c con 2 servidores.

De igual forma en la figura 4-33 y 4-34, puede observarse que al aumentar el número de servidores irá disminuyendo la intensidad de tráfico y el tiempo de espera.

The screenshot shows the same M/M/c Model calculator interface but with the Servers dropdown menu set to 3. The output values are as follows:

Service rate (packets/sec)	150
Arrival rate (packets/sec)	100
Servers	3 Servers
Mean length of system (packets)	.6760
Mean length of queue (packets)	.0093
Mean waiting time in system (sec)	.0068
Main waiting time in queue (sec)	.0001
Traffic intensity	.2222
Number of busy servers	.6667
Probability 0 packets exist in system, P(0)	.5122

Figura 4-33 Modelo M/M/c con 3 servidores.



M/M/c Model	
Service rate (packets/sec)	150
Arrival rate (packets/sec)	100
Servers	4 Servers
Calculate	
Mean length of system (packets)	.6673
Mean length of queue (packets)	.0006
Mean waiting time in system (sec)	.0067
Main waiting time in queue (sec)	.0000
Traffic intensity	.1667
Number of busy servers	.6667
Probability 0 packets exist in system, P(0)	.5133

Figura 4-34 Modelo M/M/c con 4 servidores.

Como se observó, mientras más servidores estén a la disposición de atender el sistema será más eficiente, sin embargo en este tipo de colas hay que tener cuidado con no excederse con respecto al número de servidores que deben de atender ni tampoco limitarse. Si un sistema está excedido, el costo será muy alto y se estará pagando por servidores que realmente no son necesarios; sin embargo, si se tiene un sistema muy limitado se tendrán grandes líneas de espera que provocarán costo en el sistema debido al procesamiento en que tardan los paquetes o trabajos en la cola. Es decir, esta parte del software es una herramienta para analizar el comportamiento del tráfico en el sistema y en la cola.

El modelado de colas sirve para poder resolver las redes de colas, y en algunos casos modelos más complejos. Sin embargo, para muchos modelos complejos solamente pueden simularse y no tratarse por modelado matemático como se realizó en este caso. Es importante aclarar que el estado estable se encuentra cuando la tasa

de servicio es mayor que la de arribo, es decir no hay un sobreflujo de trabajos o paquetes, por tal motivo se utilizaron puros modelos estables.