



Escuela
Técnica **Ingeniería de**
Superior **Telecomunicación**

TRABAJO ALOHA RANURADO

Diego Ismael Antolinos García
Andrés Ruz Nieto

ÍNDICE

1. INTRODUCCIÓN	2
a. ¿Qué es el protocolo Aloha Puro?.....	2
b. ¿Qué es el protocolo Aloha Ranurado?	2
2. PROGRAMACIÓN	3
a. Emisor	3
b. Canal.....	6
i. Emitir Colisión	7
ii. Emitir TICs	7
iii. Leer intención.....	8
c. Usuario.....	8
d. Parámetros y variables	9
3. SIMULACIÓN, VALIDACIÓN Y OBTENCIÓN DE DATOS	9
a. Simulación.....	9
b. Validación	10
c. Obtención de datos	11
4. ANÁLISIS DE DATOS	14

1. INTRODUCCIÓN

a. ¿Qué es el protocolo Aloha Puro?

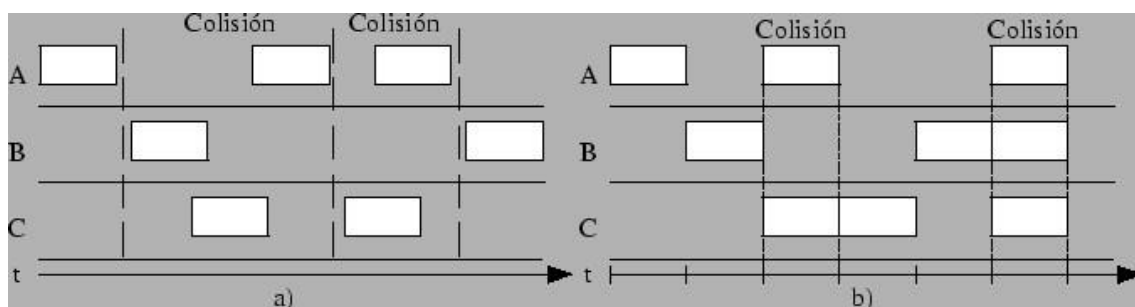
El protocolo Aloha Puro es un mecanismo de envío de mensajes, en el que cuando queremos transmitir un mensaje este será enviado sin comprobar si ya hay alguien más transmitiendo en el mismo canal. Si nadie más se encuentra transmitiendo nuestro mensaje llegará con éxito. Por otro lado, cuando ya hay otro usuario transmitiendo se produce una colisión entre los mensajes, lo que provoca que ninguno de los dos llegue a su destinatario. Es decir, un mensaje enviado, en un tiempo establecido T_0 , será enviado correctamente si ningún otro mensaje es enviado entre $t = T_0 - T$ y $t = T_0 + T$, donde T es la longitud del mensaje.

En este protocolo la probabilidad de éxito es la probabilidad de que recibamos correctamente un mensaje en $2T$ segundos.

b. ¿Qué es el protocolo Aloha Ranurado?

El protocolo Aloha Ranurado es una variante del protocolo Aloha en el que se pretende disminuir el número de colisiones. En esta versión, los mensajes solo pueden ser enviados en determinados slots que poseen la misma longitud y los cuales denominaremos como ranuras temporales. Solo en el caso de que se quieran enviar dos mensajes simultáneamente, por el mismo canal, se producirá una colisión. Si esta se llega a producir se utiliza el algoritmo de Back-Off que se encarga de realizar la retransmisión del mensaje.

En este protocolo, en el caso de colisión, el reenvío del mensaje se produce tras esperar $[0, K-1]$ ranuras temporales. Todo este proceso se irá repitiendo hasta que se transmite la trama con éxito, el valor de K va aumentando conforme suceden intentos de transmisión. La probabilidad de éxito en Aloha Ranurado es la probabilidad de que se envíe un mensaje en T segundos.



(a) Protocolo Aloha Puro

(b) Protocolo Aloha Ranurado

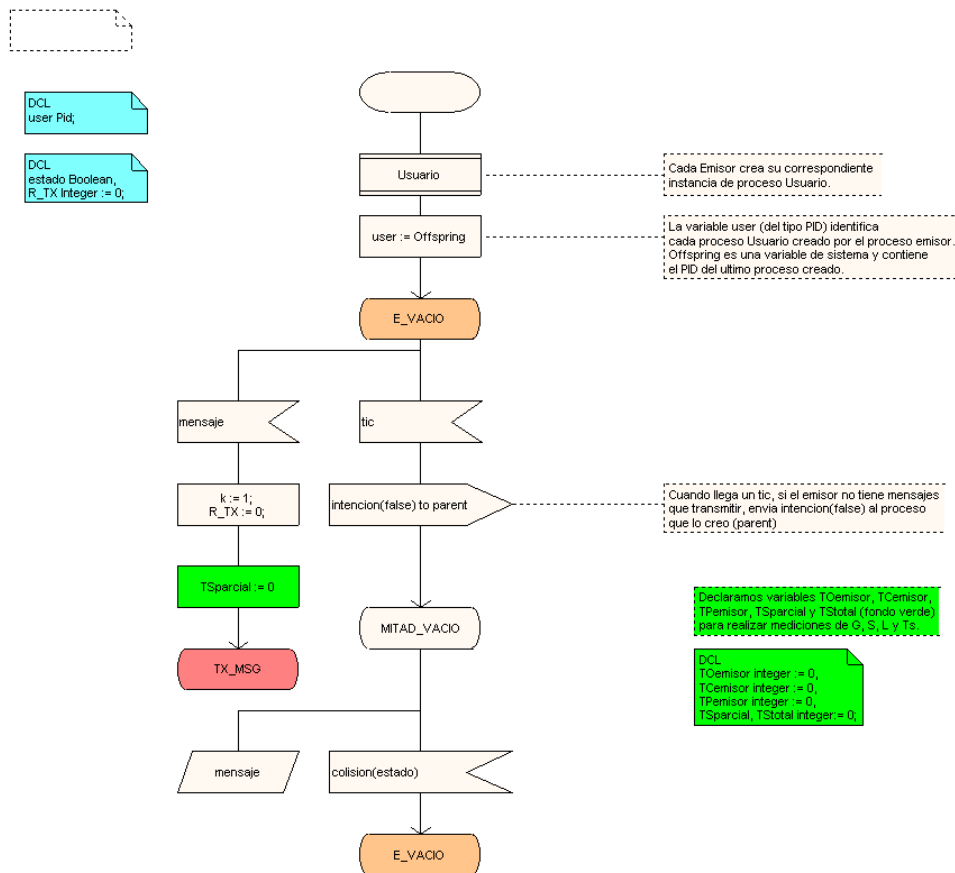
2. PROGRAMACIÓN

En este punto procederemos a comentar como se ha realizado la programación de las diferentes partes de nuestro protocolo Aloha Ranurado. Diferenciamos los códigos de Emisor, Canal y Usuario.

a. Emisor

process Emisor

1(3)



Este bloque cuenta con 3 procesos, en el primero crearemos los usuarios con una intención. El emisor creará un proceso “n” el cual estará vacío. En este momento pueden ocurrir dos cosas:

- Rama derecha.

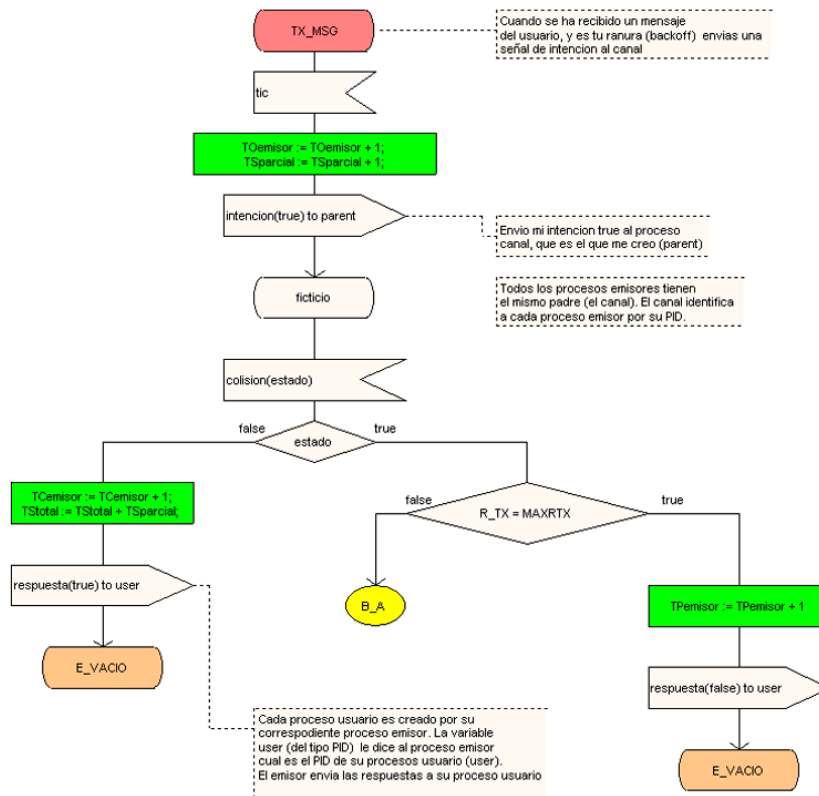
El usuario no puede enviar nada porque no tiene permiso, en este momento enviaremos una intención false, diciendo que no queremos enviar nada, reduciremos a la mitad la probabilidad de éxito al enviar el mensaje. Como resultado podemos obtener como resultado 0, que el mensaje haya llegado, o que se haya producido una colisión. Tras ello, se volverá a indicar que el emisor está vacío y puede volver a iniciarse el proceso

- Tenga el mensaje listo para enviar. (izquierda)

El usuario puede enviar el mensaje, por lo que inicia el número de reintentos a 0 (por ser la primera vez), el valor $k = 1$, y establece un tiempo parcial de 0. Posteriormente pasa al proceso para transmitir el mensaje.

process Emisor

2(3)



1

Este es el proceso que se encarga de transmitir un mensaje, el usuario comienza obteniendo el “tic”. Tras ello aumenta el Trafico Ofrecido y el Tiempo de Servicio. El usuario la intención al proceso que le ha creado y comprueba si ha habido o no una colisión.

- Hay colisión

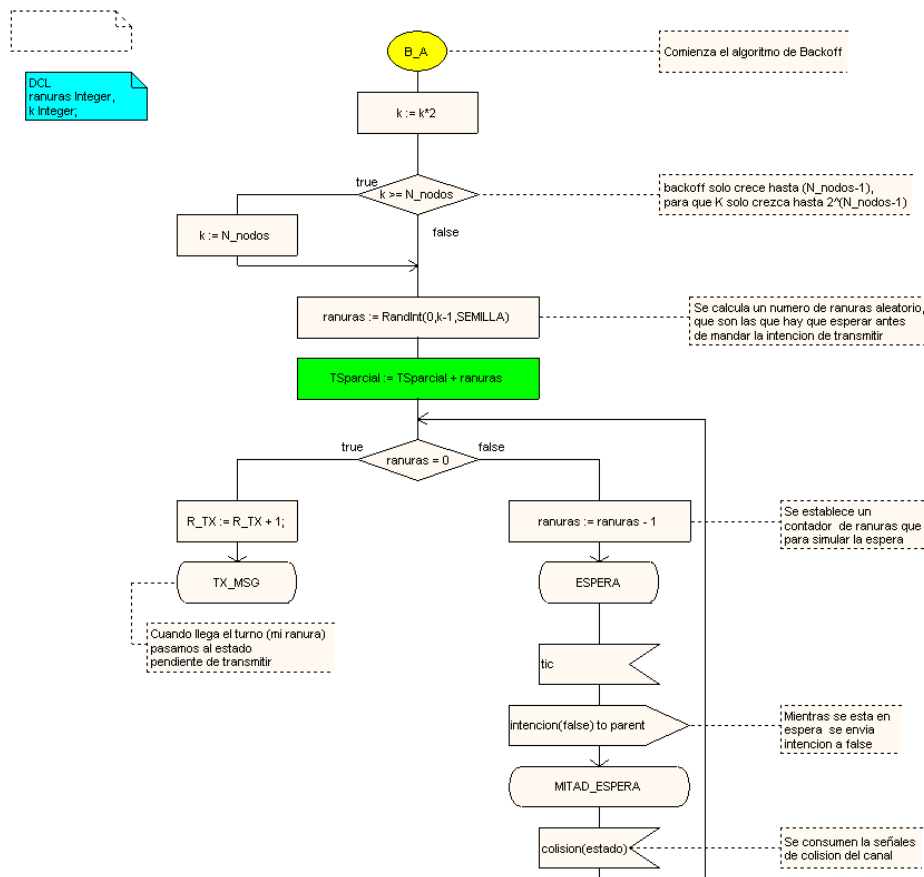
Comprueba si puede continuar reintentando enviar el mensaje, si puede continuar salta al algoritmo Back-Off que intentará solucionar el problema, en caso contrario avisaremos al proceso padre de que no podemos enviar y volveremos al estado en el que el emisor está vacío.

- No hay colisión

Indicamos al proceso padre que sí podemos enviar el mensaje, aumentamos el tráfico cursado en uno y calculamos la tasa de servicio que hemos obtenido. Tras ello, liberamos el emisor para que vuelvas a reiniciarse el bloque emisor

process Emisor

3(3)



Este algoritmo se encarga de establecer un valor de k. Nosotros iniciamos k en 2 por lo que comprobará si k es mayor o igual que los nodos que tenemos (5 en nuestro caso). Si el resultado es true, establece k = número de nodos, en caso contrario utilizará el valor de k. Tras ello, este algoritmo establecerá el tiempo en ranuras. Si el tiempo de ranura es 0 (ranura no utilizada), podrá reintentar el proceso de transmitir el mensaje, en otro caso, tendrá que esperar hasta recibir una ranura libre para reintentarlo.

b. Canal

process Canal

1(1)



```
NEWTYPE PidArray Array (Integer, Pid)
ENDNEWTYPE;
```

```
DCL
  Emisor_ID PidArray,
  i Integer := 1,
  num_i Integer,
  Intencion Boolean,
  Colision Boolean;
```

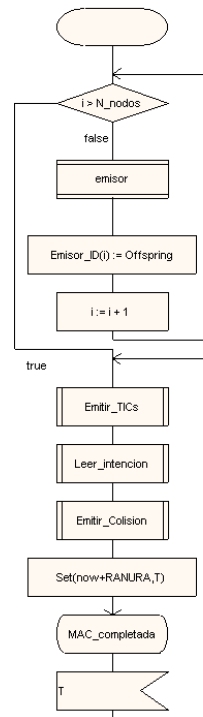
```
DCL
  tx_sim Integer := 0,
  G Integer := 0,
  S Integer := 0;
```

TIMER T;

Emitir_TICs

Leer_Intencion

Emitir_Colision



Instanciamos la señal T, del tipo TIMER, para que ocurra dentro de un tiempo (now+Ranura). Now es una variable de sistema que recoge el valor del tiempo actual en nuestro sistema.

Creacion de instancias de proceso emisor. El proceso Canal crea 'N_nodos' instancias del proceso Emisor y guarda el PID de cada proceso Emisor en Emisor_ID, que es un array de PID.

Cuando el proceso Canal emita una señal, debe indicar a que proceso va dirigida. La sintaxis es "señal TO PID". En nuestro caso sera "señal TO Emisor_ID(i)"

offspring es una variable de sistema y contiene el valor del ultimo PID creado. El array Emisor_ID recoge ese valor, para que no se pierda

1

Este bloque se encarga de crear un array con los ID de cada emisor y establecer el número de emisores que pueden trabajar, es decir, tendrá que haber como mínimo tantos nodos como emisores vayan a participar. Tras establecer el número de emisores que van a trabajar, iremos a los siguientes procesos:

1. Emitir Colisión:

Se encargará de informar al emisor o al usuario que haya enviado que, durante el mismo, se ha producido una colisión de su mensaje.

2. Emitir TICs:

Se encarga de enviar las señales a los emisores y usuarios para indicarles que pueden enviar o recibir mensajes.

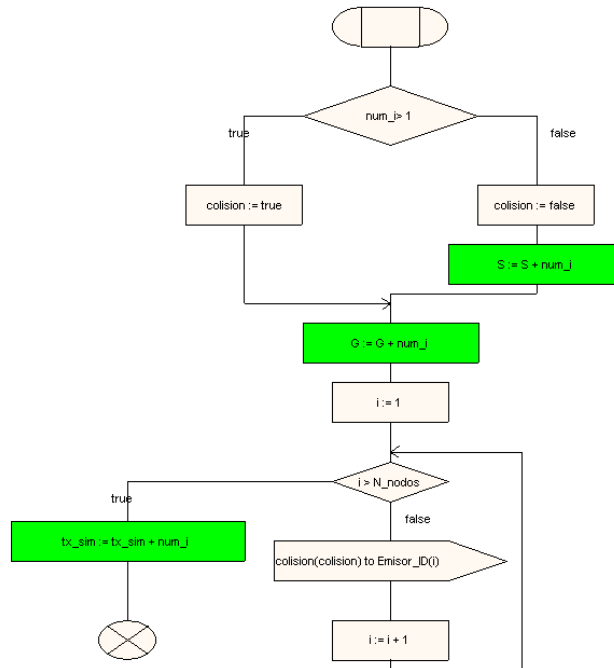
3. Leer Intención:

Se encarga de ver qué intención tiene cada emisor, si tiene la intención de transmitir mensaje o esperar tic.

i. Emitir Colisión

procedure Emitir_Colision

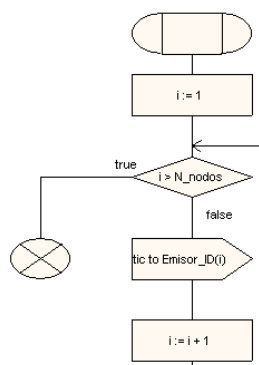
1(1)



ii. Emitir TICs

procedure Emitir_TICs

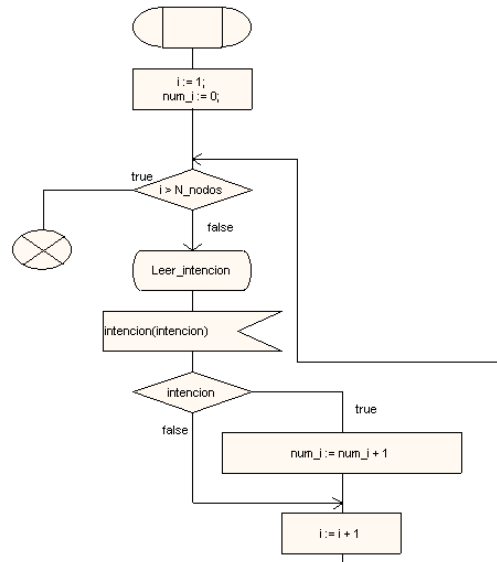
1(1)



iii. Leer intención

procedure Leer_Intencion

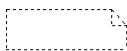
1(1)



c. Usuario

process Usuario

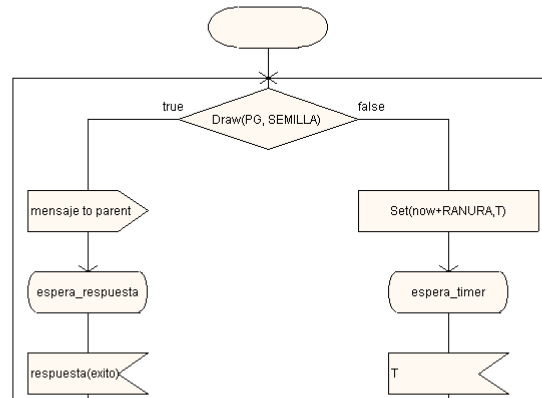
1(1)



DCL
exito Boolean;

Timer T;

El proceso usuario, en un periodo de tiempo igual al de una RANURA, es capaz de generar una mensaje, con probabilidad PG.



Cuando el proceso Usuario genera un mensaje, lo envía al proceso Emisor que lo creo (a su padre), para que este lo envíe al canal.
El proceso Usuario queda a la espera de una respuesta, que puede ser true o false

Este bloque representa al usuario. Se encarga de comprobar si puede recibir respuesta o no dada una probabilidad. Si el resultado de esta comprobación es true, enviará un mensaje al proceso anterior indicando que puede recibir respuesta y se queda la respuesta del proceso emisor y obtiene la respuesta de éxito del envío (ACK). Si el resultado de la comprobación es false, tendrá que establecer un nuevo tiempo para ver si puede recibir otra ranura, espera al timer T y tras ese tiempo, volverá a comprobarlo.

d. Parámetros y variables

A continuación, indicaremos los parámetros y variables que hemos utilizado:

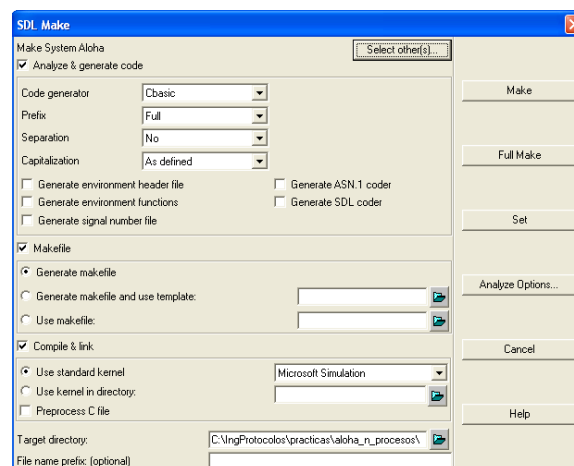
- **TIC:** Señal enviada por el canal que sirve para indicar que se puede enviar información a través de este.
- **R_TX:** Número de reintentos.
- **K:** Número de ranuras máximas que esperaría el protocolo para realizar un siguiente reintento.
- **RANURA:** Slot de tiempo en el cual se podría enviar un mensaje.
- **NODOS:** Instancia de un proceso emisor.
- **COLISION:** Transmisión de paquetes fallida debido al confrontamiento entre paquetes o ruido.
- **ACK:** Mensaje de confirmación de la trama.
- **OFFSPRING:** Variable del sistema que contiene el valor del último PID creado.
- **PID:** Identificador que se le asigna a cada emisor.
- **ARRAY ID:** Array que almacena los PID.
- **TO:** Tráfico Ofrecido.
- **TC:** Tráfico Cursado.
- **TP:** Tráfico Perdido.
- **TS:** Tiempo de Servicio.
- **B_A:** Algoritmo Back-Off.
- **MAC:** Control de Acceso al Medio físico.

3. SIMULACIÓN, VALIDACIÓN Y OBTENCIÓN DE DATOS

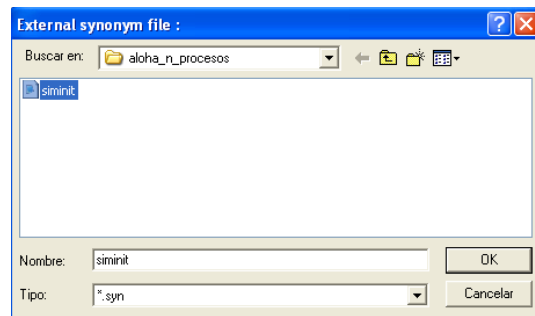
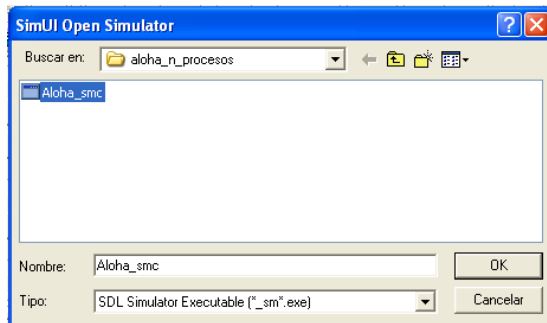
Una vez explicada la programación del protocolo procederemos a explicar como ha sido el proceso de simulación, validación de nuestro programa y la obtención de sus datos.

a. Simulación

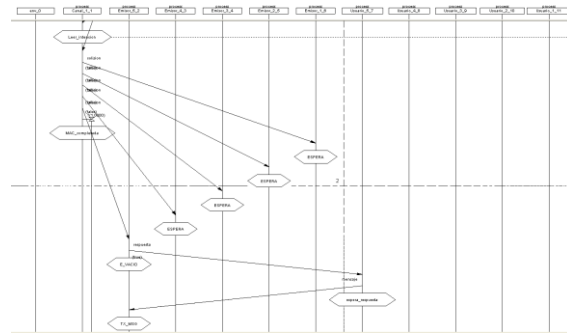
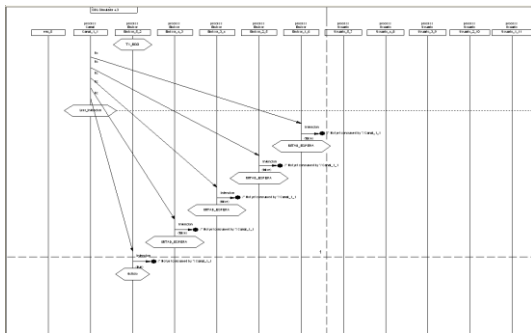
- Nos iremos a la pestaña “Generate” y pulsaremos sobre “Make” o usaremos el acceso directo *Ctrl + M*, pondremos las siguientes opciones y pulsaremos sobre “Full Make”. Si hay algún error se nos abrirá la ventana “Organizer Log” la cual nos indicará dónde está el fallo.



- ii. Iniciamos el simulador SDL con el comando rápido “CTRL + J” y abrimos el archivo “aloha_smc.exe”, a continuación, nos pedirá el archivo “siminit.syn”, si no está creado podemos introducir los datos manualmente pulsando sobre “Cancelar”.

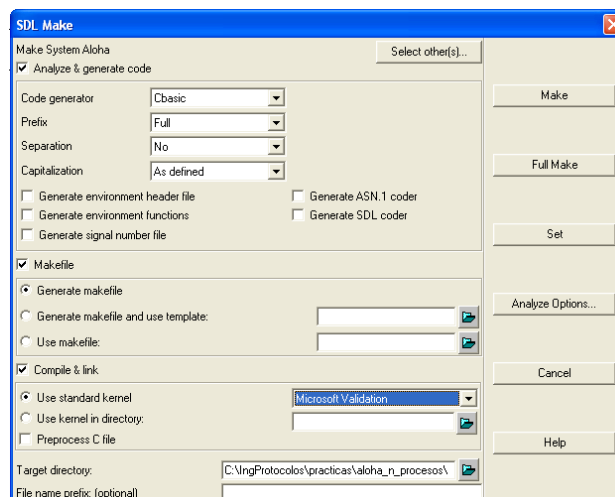


- iii. Procedemos a la simulación del protocolo Aloha Ranurado y nos aseguramos de que no exista ningún tipo de error en su funcionamiento. Esta simulación la realizamos siguiendo los pasos de los tutoriales previos vistos en clase. A continuación, se pueden ver algunas imágenes de la simulación

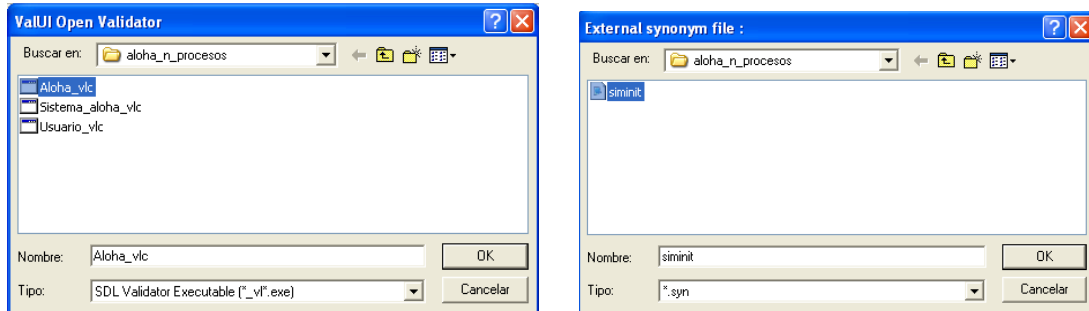


b. Validación

- i. Nos iremos a la pestaña “Generate” y pulsaremos sobre “Make” o usaremos el acceso directo Ctrl + M, pondremos las siguientes opciones y pulsaremos sobre “Full Make”. Pero esta vez en vez de poner Microsoft Simulation, seleccionaremos Microsoft Validation



- ii. Iniciamos el validador SDL con el comando rápido “CTRL + G” y abrimos el archivo “aloha_vlc.exe, a continuación, nos pedirá el archivo “siminit.syn”, si no está creado podemos introducir los datos manualmente pulsando sobre “Cancelar”.



- iii. Procedemos a la validación del protocolo Aloha Ranurado y nos aseguramos de que no exista ningún tipo de error en su funcionamiento. Esta simulación la realizamos siguiendo los pasos de los tutoriales previos vistos en clase.

c. Obtención de datos

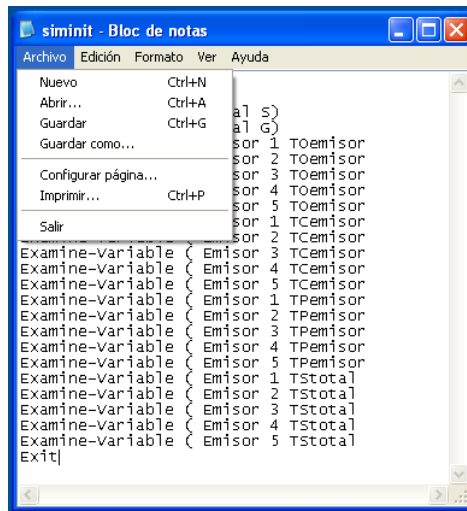
- i. Creamos el archivo “siminit.com”, esto lo haremos abriendo un bloc de notas de Windows, copiaremos el siguiente código:

```
Set-Trace 0
Proceed-Until 10000
Examine-Variable ( Canal S)
Examine-Variable ( Canal G)
Examine-Variable ( Emisor 1 TOemisor
Examine-Variable ( Emisor 2 TOemisor
Examine-Variable ( Emisor 3 TOemisor
Examine-Variable ( Emisor 4 TOemisor
Examine-Variable ( Emisor 5 TOemisor
Examine-Variable ( Emisor 1 TCemisor
Examine-Variable ( Emisor 2 TCemisor
Examine-Variable ( Emisor 3 TCemisor
Examine-Variable ( Emisor 4 TCemisor
Examine-Variable ( Emisor 5 TCemisor
Examine-Variable ( Emisor 1 TPemisor
Examine-Variable ( Emisor 2 TPemisor
Examine-Variable ( Emisor 3 TPemisor
Examine-Variable ( Emisor 4 TPemisor
Examine-Variable ( Emisor 5 TPemisor
Examine-Variable ( Emisor 1 TStotal
Examine-Variable ( Emisor 2 TStotal
Examine-Variable ( Emisor 3 TStotal
Examine-Variable ( Emisor 4 TStotal
Examine-Variable ( Emisor 5 TStotal
Exit
```

Ponemos *Set-Trace* a 0 para eliminar las trazas y que así los tamaños de los ficheros de resultados no crezcan mucho. *Proceed-Until 10000*

lo utilizamos para realizar la simulación para 10000 ranuras. Y *Examine-Variable* es para obtener los valores de dichas variables.

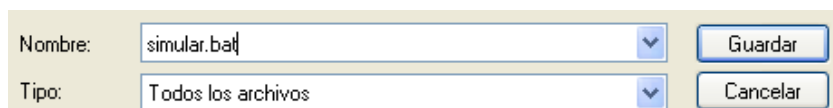
Este código servirá para definir que datos queremos obtener y las ranuras de la simulación. Una vez escrito tendremos que pulsar sobre “Archivo”, “Guardar como...” y escribiremos lo que tenemos en la siguiente imagen



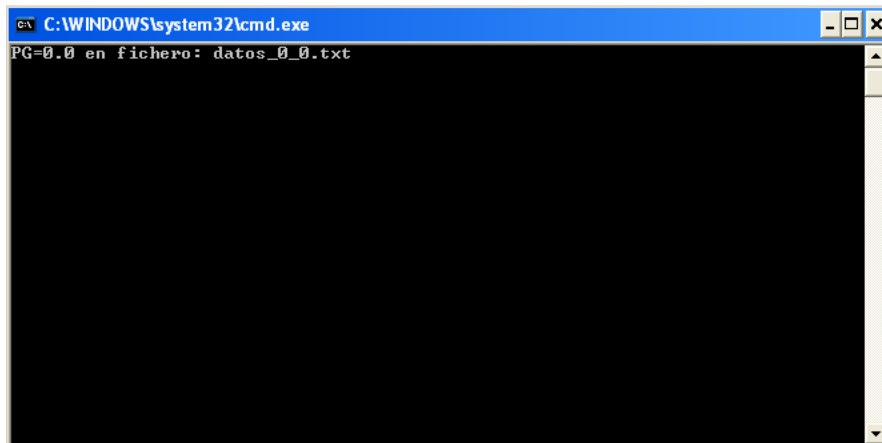
- ii. Creamos el archivo “*simular.bat*” para automatizar el proceso de simulación. Abriremos un bloc de notas de Windows y escribiremos el siguiente código:

```
@echo off
set SDTEXTSYNFILE=siminit.syn
for %i in (0 01 025 05 075 1 125 150 175 2 25 3
35 4 45 5 6 7 8 9
95) do (
echo PG=0.%i en fichero: resultado_0_%i.txt
echo N_nodos 5 > siminit.syn
echo PG 0.%i >> siminit.syn
aloha_smc.exe > resultado_0_%i.txt
)
```

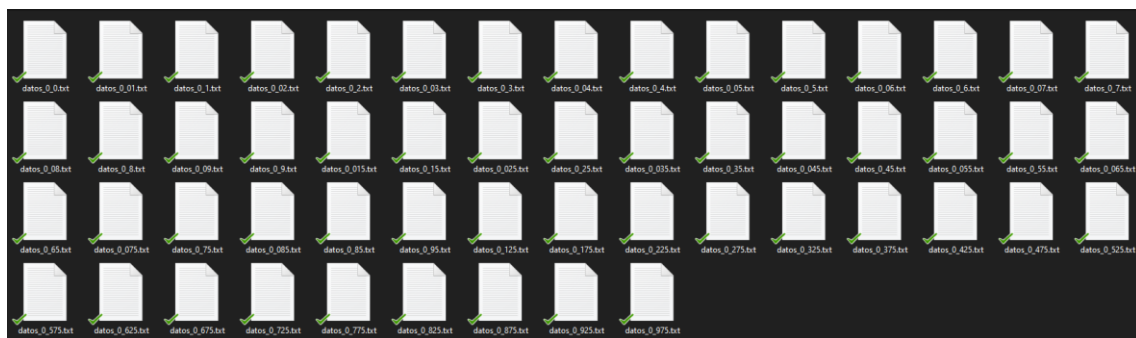
En el bucle “*for*” introducimos los puntos que queremos simular. Hay que tener en cuenta que se introducen suponiendo que llevan un 0, delante. Pulsaremos sobre “*Guardar como...*”, y lo guardaremos de la siguiente manera dentro de la carpeta del proyecto.



- iii. Hacemos doble click sobre el archivo “*simular.bat*” y se nos abrirá la siguiente ventana que realizará la simulación de nuestro programa.



Finalmente, se nos generarán los siguientes archivos:



```

Simu_5_2019h Bloc de notas
Archivo Edición Formato Ver Ayuda
No connection with the Postmaster. Running stand-alone.
Using external synonym file "siminit.syn" according to environment variable "SDTEXTSYNFILE".

Welcome to the SQL SIMULATOR. Simulating system Aloha.

:Command : Set-Trace 0
Default trace set to 0

:Command : Proceed-Until 10000

:Command : Examine-Variable ( Canal 5 )
: (Integer) = 3685

:Command : Examine-Variable ( Canal 6 )
: (Integer) = 8545

:Command : Examine-Variable ( Emisor 1 TOemisor
TOemisor (Integer) = 1720

:Command : Examine-Variable ( Emisor 2 TOemisor
TOemisor (Integer) = 1634

:Command : Examine-Variable ( Emisor 3 TOemisor
TOemisor (Integer) = 1726

:Command : Examine-Variable ( Emisor 4 TOemisor
TOemisor (Integer) = 1751

:Command : Examine-Variable ( Emisor 5 TOemisor
TOemisor (Integer) = 1714

:Command : Examine-Variable ( Emisor 1 TCemisor
TCemisor (Integer) = 732

:Command : Examine-Variable ( Emisor 2 TCemisor
TCemisor (Integer) = 705

:Command : Examine-Variable ( Emisor 3 TCemisor
TCemisor (Integer) = 723

:Command : Examine-Variable ( Emisor 4 TCemisor
TCemisor (Integer) = 752

:Command : Examine-Variable ( Emisor 5 TCemisor
TCemisor (Integer) = 693

:Command : Examine-Variable ( Emisor 1 TPemisor
TPemisor (Integer) = 36

:Command : Examine-Variable ( Emisor 2 TPemisor
TPemisor (Integer) = 38

:Command : Examine-Variable ( Emisor 3 TPemisor
TPemisor (Integer) = 38

Ln 1, Col 1      100% Windows (CRUF)  UTF-8

```

4. ANÁLISIS DE DATOS

Una vez que hemos realizado el montaje del protocolo, la simulación y posterior validación de este procedemos a la obtención de datos y elaboración de diferentes gráficas para observar a que probabilidad funciona mejor el protocolo elaborado.

Para la elaboración de las tablas y de las gráficas posteriores nos hemos servido del programa Excel de Google Drive. Hemos obtenido datos del tráfico ofrecido, tráfico cursado, tráfico perdido y tiempo de servicio.

PG	TOemisor1	TOemisor2	TOemisor3	TOemisor4	TOemisor5	TOemisorMedia
0	0	0	0	0	0	0
0,01	97	104	95	94	100	0,049
0,015	142	154	166	151	158	0,0771
0,02	249	204	200	217	210	0,108
0,025	302	297	301	272	284	0,1456
0,03	321	365	373	338	363	0,176
0,035	436	394	465	430	464	0,2189
0,04	541	542	563	524	499	0,2669
0,045	653	561	602	696	569	0,3081
0,05	687	735	735	739	745	0,3641
0,055	799	768	817	823	818	0,4025
0,06	927	886	898	921	957	0,4589
0,065	1019	1012	1031	1073	1028	0,5163
0,07	1223	1124	1197	1237	1203	0,5984
0,075	1384	1390	1282	1312	1314	0,6682
0,08	1453	1459	1435	1426	1461	0,7234
0,085	1598	1616	1514	1514	1557	0,7799
0,09	1720	1634	1726	1751	1714	0,8545
0,1	1890	1955	1922	1950	1793	0,951
0,125	2361	2326	2367	2418	2377	1,1849
0,15	2735	2858	2863	2843	2748	1,4047
0,175	3081	3038	3116	3158	3187	1,558
0,2	3377	3380	3368	3356	3308	1,6789
0,225	3543	3567	3430	3524	3525	1,7589
0,25	3662	3664	3689	3712	3661	1,8388
0,275	3771	3793	3795	3722	3785	1,8866
0,3	3964	3908	3854	3858	3800	1,9384
0,325	3952	3914	3912	4011	3941	1,973
0,35	4027	4041	4011	4003	3989	2,0071
0,375	4098	4050	4044	4094	4057	2,0343
0,4	4083	4160	4171	4084	4171	2,0669
0,425	4234	4122	4187	4186	4114	2,0843
0,45	4231	4184	4191	4232	4201	2,1039
0,475	4220	4333	4250	4208	4198	2,1209
0,5	4240	4285	4313	4228	4238	2,1304
0,525	4303	4320	4276	4265	4243	2,1407
0,55	4288	4300	4316	4298	4284	2,1486
0,575	4332	4368	4333	4264	4264	2,1561
0,6	4348	4262	4328	4344	4346	2,1628
0,625	4398	4367	4384	4315	4329	2,1793
0,65	4428	4318	4401	4369	4345	2,1861
0,675	4370	4395	4344	4393	4406	2,1908
0,7	4395	4428	4420	4407	4434	2,2084
0,725	4436	4389	4426	4378	4398	2,2027
0,75	4377	4439	4471	4421	4403	2,2111
0,775	4380	4495	4449	4424	4402	2,215
0,8	4530	4404	4427	4463	4403	2,2227
0,825	4452	4419	4434	4518	4434	2,2257
0,85	4409	4504	4476	4428	4476	2,2293
0,875	4460	4424	4457	4492	4518	2,2351
0,9	4439	4507	4484	4419	4502	2,2351
0,925	4493	4532	4400	4478	4537	2,244
0,95	4495	4531	4531	4497	4461	2,2515
0,975	4537	4569	4546	4445	4460	2,2557

PG	TCemisor1	TCemisor2	TCemisor3	TCemisor4	TCemisor5	TCemisorMedia
0	0	0	0	0	0	0
0,01	97	98	95	88	100	0,0478
0,015	131	143	153	140	150	0,0717
0,02	226	190	177	193	192	0,0978
0,025	267	258	259	234	240	0,1258
0,03	270	314	311	289	303	0,1487
0,035	356	315	375	332	360	0,1738
0,04	402	387	420	392	397	0,1998
0,045	462	433	421	490	432	0,2238
0,05	468	478	482	505	504	0,2437
0,055	516	532	530	546	534	0,2658
0,06	576	564	556	552	594	0,2842
0,065	622	569	603	613	619	0,3026
0,07	650	599	607	658	643	0,3157
0,075	682	690	618	623	654	0,3267
0,08	680	662	697	674	706	0,3419
0,085	708	721	677	686	723	0,3515
0,09	732	705	723	752	693	0,3605
0,1	718	746	781	748	713	0,3706
0,125	785	741	763	804	769	0,3862
0,15	674	740	737	746	703	0,36
0,175	658	683	697	699	740	0,3477
0,2	619	673	669	631	617	0,3209
0,225	639	595	590	630	604	0,3058
0,25	594	570	561	580	595	0,29
0,275	581	588	604	525	554	0,2852
0,3	554	555	548	565	519	0,2741
0,325	507	498	500	532	536	0,2573
0,35	506	523	548	539	477	0,2593
0,375	527	494	487	503	497	0,2508
0,4	463	515	517	458	505	0,2458
0,425	497	475	486	495	481	0,2434
0,45	487	483	465	449	466	0,235
0,475	475	510	459	452	469	0,2365
0,5	449	463	485	475	448	0,232
0,525	475	444	451	453	460	0,2283
0,55	428	456	465	452	446	0,2247
0,575	442	480	451	428	385	0,2186
0,6	446	431	474	469	426	0,2246
0,625	462	426	485	440	439	0,2252
0,65	460	448	439	436	435	0,2218
0,675	430	456	445	468	432	0,2231
0,7	451	471	411	393	460	0,2186
0,725	450	437	446	401	422	0,2156
0,75	390	455	482	441	427	0,2195
0,775	402	429	417	417	424	0,2089
0,8	475	387	441	454	398	0,2155
0,825	432	417	415	451	412	0,2127
0,85	393	434	389	433	441	0,209
0,875	405	410	390	449	465	0,2119
0,9	406	413	446	376	440	0,2081
0,925	417	436	389	434	446	0,2122
0,95	400	448	415	399	414	0,2076
0,975	403	447	424	402	389	0,2065

PG	TPemisor1	TPemisor2	TPemisor3	TPemisor4	TPemisor5	TPemisorMedia
0	0	0	0	0	0	0
0,01	0	0	0	0	0	0
0,015	0	0	0	0	0	0
0,02	0	0	0	0	0	0
0,025	0	0	0	0	0	0
0,03	0	1	1	0	0	0,0002
0,035	0	2	1	0	0	0,0003
0,04	0	1	2	0	0	0,0003
0,045	2	0	2	4	1	0,0009
0,05	4	6	4	6	3	0,0023
0,055	4	6	5	5	4	0,0024
0,06	10	6	8	11	8	0,0043
0,065	4	15	11	11	7	0,0048
0,07	19	22	22	18	21	0,0102
0,075	29	32	33	30	26	0,015
0,08	28	41	25	34	24	0,0152
0,085	39	36	34	29	29	0,0167
0,09	36	38	44	37	51	0,0206
0,1	72	60	44	58	60	0,0294
0,125	105	102	86	100	99	0,0492
0,15	161	156	166	147	153	0,0783
0,175	205	184	195	206	207	0,0997
0,2	266	255	242	251	253	0,1267
0,225	278	299	283	277	301	0,1438
0,25	307	325	323	326	307	0,1588
0,275	332	341	341	359	351	0,1724
0,3	376	367	361	355	363	0,1822
0,325	387	398	397	394	398	0,1974
0,35	414	395	388	390	415	0,2002
0,375	410	429	423	410	418	0,209
0,4	446	420	423	442	437	0,2168
0,425	445	437	446	434	435	0,2197
0,45	449	449	458	464	451	0,2271
0,475	440	454	473	463	455	0,2285
0,5	469	464	473	457	465	0,2328
0,525	460	482	481	477	453	0,2353
0,55	446	475	484	470	479	0,2354
0,575	484	474	480	487	511	0,2436
0,6	485	476	472	489	498	0,242
0,625	488	488	472	485	496	0,2429
0,65	496	487	495	493	498	0,2469
0,675	501	493	484	480	502	0,246
0,7	492	494	519	525	501	0,2531
0,725	500	491	509	522	509	0,2531
0,75	524	492	491	508	502	0,2517
0,775	512	521	518	515	507	0,2573
0,8	513	529	506	503	525	0,2576
0,825	504	512	521	515	526	0,2578
0,85	529	517	534	503	514	0,2597
0,875	533	517	545	509	513	0,2617
0,9	528	541	519	546	521	0,2655
0,925	520	520	526	523	526	0,2615
0,95	527	521	530	540	519	0,2637
0,975	552	525	546	532	542	0,2697

PG	TStotal1	TStotal2	TStotal3	TStotal4	TStotal5	TStotalMedia
0	0	0	0	0	0	0
0,01	97	113	95	95	100	0,05
0,015	152	166	183	161	166	0,0828
0,02	264	221	224	237	228	0,1174
0,025	335	342	350	314	338	0,1679
0,03	386	403	426	388	420	0,2023
0,035	514	459	560	539	570	0,2642
0,04	696	711	697	659	620	0,3383
0,045	713	856	790	677	837	0,3873
0,05	882	957	884	932	963	0,4618
0,055	1123	962	1059	1097	1094	0,5335
0,06	1236	1186	1207	1230	1262	0,6121
0,065	1459	1344	1424	1465	1399	0,7091
0,07	1680	1438	1608	1719	1583	0,8028
0,075	1873	1752	1634	1729	1746	0,8734
0,08	1985	1918	2058	1882	2022	0,9865
0,085	2140	2205	2073	2115	2168	1,0701
0,09	2424	2276	2353	2444	2239	1,1736
0,1	2341	2637	2733	2647	2295	1,2653
0,125	2899	2928	3179	3092	3036	1,5134
0,15	3232	3442	3236	3468	3298	1,6676
0,175	3362	3564	3533	3452	3469	1,738
0,2	3334	3313	3496	3378	3248	1,6769
0,225	3441	3373	3249	3372	3110	1,6545
0,25	3400	3163	3252	3208	3431	1,6454
0,275	3336	3289	3204	2886	3064	1,5779
0,3	3157	3143	3180	3218	3096	1,5794
0,325	3085	2839	2793	3057	2821	1,4595
0,35	2841	3110	3188	3125	2812	1,5076
0,375	3025	2755	2779	3049	2913	1,4521
0,4	2593	3042	3117	2702	2887	1,4341
0,425	2902	2869	2843	3012	2748	1,4374
0,45	2848	2704	2678	2816	2856	1,3902
0,475	3059	3011	2632	2690	2733	1,4125
0,5	2702	2809	2684	2861	2754	1,381
0,525	2956	2691	2576	2658	2871	1,3752
0,55	2802	2599	2852	2640	2621	1,3514
0,575	2675	2941	2729	2574	2265	1,3184
0,6	2709	2706	2893	2618	2545	1,3471
0,625	2760	2528	2906	2684	2511	1,3389
0,65	2680	2597	2712	2658	2597	1,3244
0,675	2580	2692	2727	2848	2596	1,3443
0,7	2757	2698	2446	2354	2640	1,2895
0,725	2685	2810	2555	2341	2561	1,2952
0,75	2340	2868	2844	2623	2622	1,3297
0,775	2518	2529	2533	2458	2532	1,257
0,8	2614	2328	2631	2725	2327	1,2625
0,825	2702	2570	2462	2666	2385	1,2785
0,85	2300	2704	2391	2645	2692	1,2732
0,875	2376	2516	2253	2733	2686	1,2564
0,9	2409	2356	2570	2108	2582	1,2025
0,925	2615	2657	2389	2506	2528	1,2695
0,95	2609	2652	2530	2344	2579	1,2714
0,975	2308	2637	2294	2383	2265	1,1887

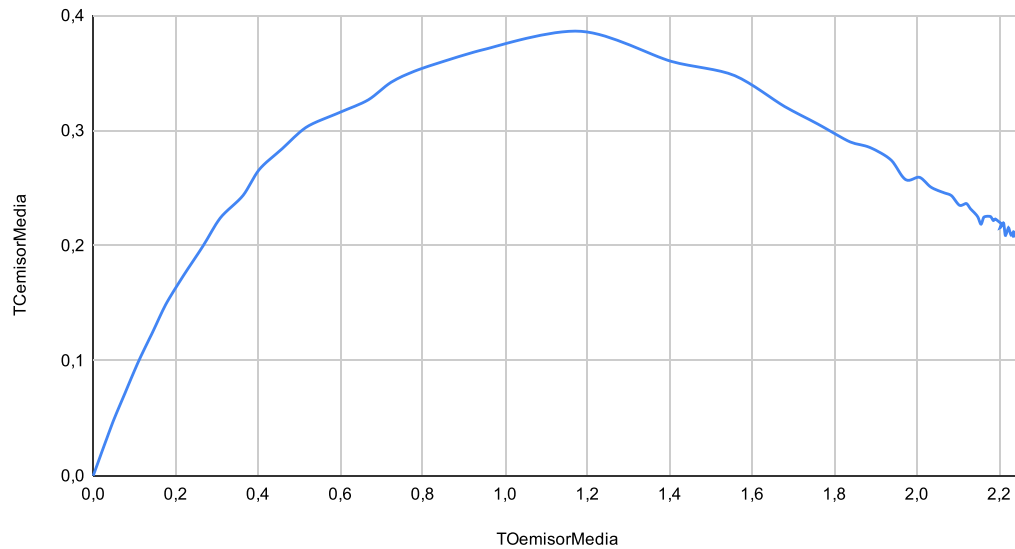
Hemos elaborado diferentes gráficas con los datos obtenidos, y para ellos nos hemos servido de los valores medios resultantes de los 5 emisores vistos en las tablas de arriba. En estas podremos observar la calidad del canal, el tiempo medio de servicio y el número de paquetes perdidos.

Gráfica 1: Tráfico Cursado vs Tráfico Ofrecido (Eficiencia del canal)

Gráfica 2: Tráfico Perdido vs Tráfico Cursado (Número de paquetes perdidos)

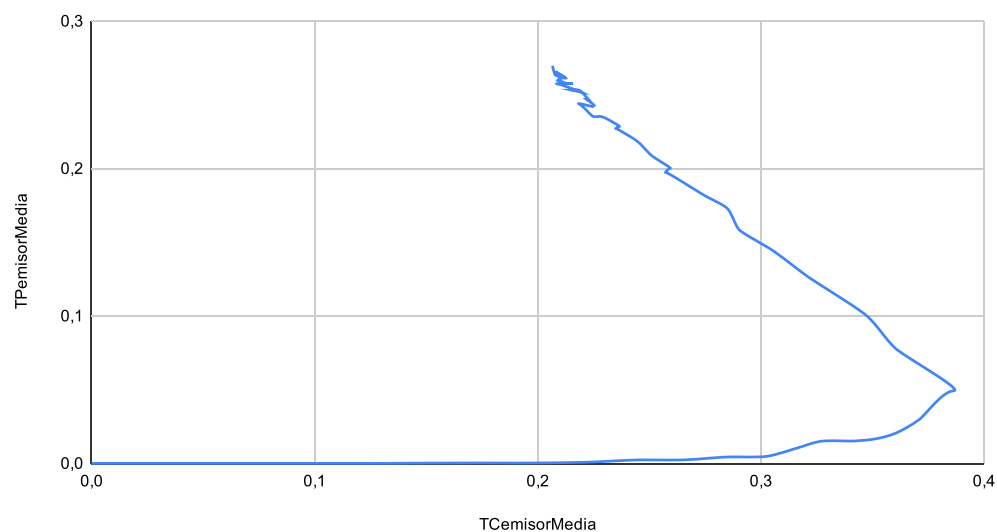
Gráfica 3: Tiempo de Servicio Total vs Tráfico Cursado (Tiempo medio de servicio)

TCemisorMedia frente a TOemisorMedia



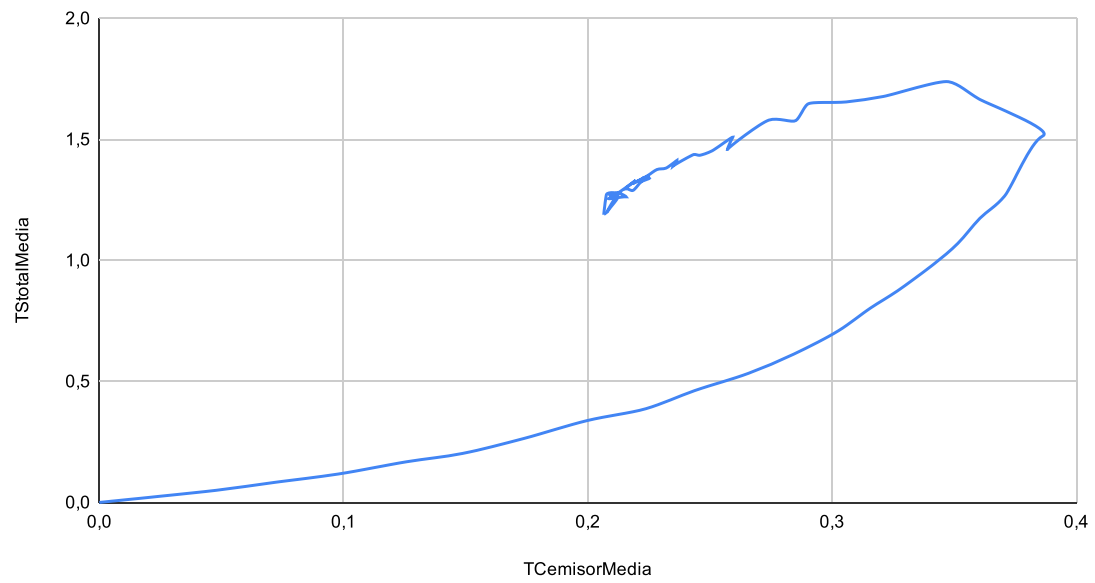
Gráfica (1) Eficiencia del canal

TPemisorMedia frente a TCemisorMedia



Gráfica (2) Número de paquetes perdidos

TStotalMedia frente a TCemisorMedia



Gráfica (3) Tiempo medio de servicio