Práctica 2: evaluación de arquitectura de conmutadores ATM

Sistemas de conmutación

Alberto Mateos Checa | Manuel Granero Valenzuela

Índice de contenidos

1. OBJETIVO:

Evaluación de las prestaciones de un conmutador ATM con una arquitectura por división espacial sin memoria, y otro con una arquitectura con memoria a la salida.

En particular, se persigue el análisis del throughput del conmutador, el retardo sufrido por las células al atravesar dicho conmutador, y además la probabilidad de pérdida de células en el conmutador.

2. FUNDAMENTO TEÓRICO Y SUPOSICIONES:

Las redes basadas en el Modo de Transferencia asíncrono se caracterizan fundamentalmente por:

-**Conmutación de paquetes**: La información se transmite en pequeños paquetes de tamaño fijo llamados células y divididos en 5 bytes de cabecera y 48 de información.

-**Orientado a conexión**: Las células se transmiten a través de circuitos virtuales y que se identifican mediante una etiqueta contenida en la cabecera.

-**La estructura es la siguiente**:

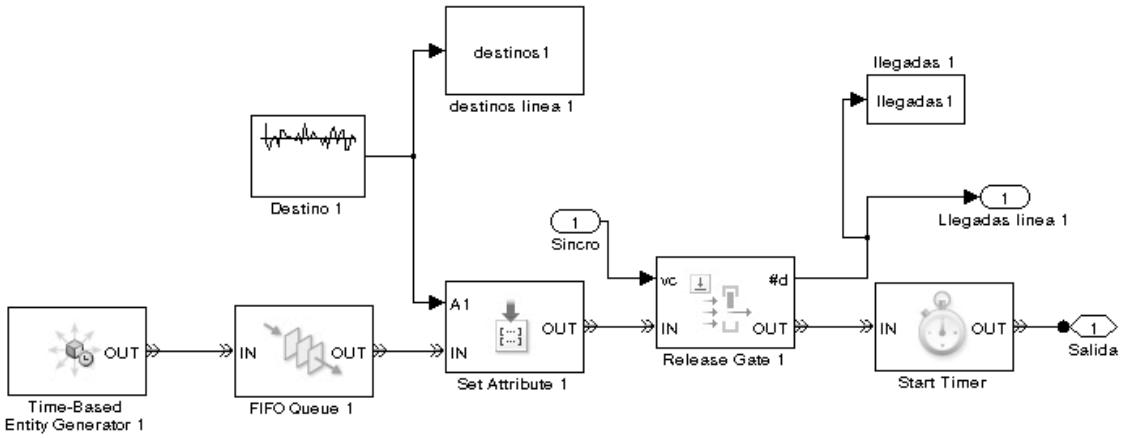
1. Módulos de entrada.
2. Módulos de salida.
3. Red de interconexión
4. La unidad de control.

Ahora vamos a comentar **cada parte del conmutador y cómo ha sido implementada.** Se muestran sólo los módulos correspondientes para el conmutador 2x2 ya que los conmutadores de mayor orden tendrán los mismos módulo solo que en mayor número.

2.1 Líneas de entrada.

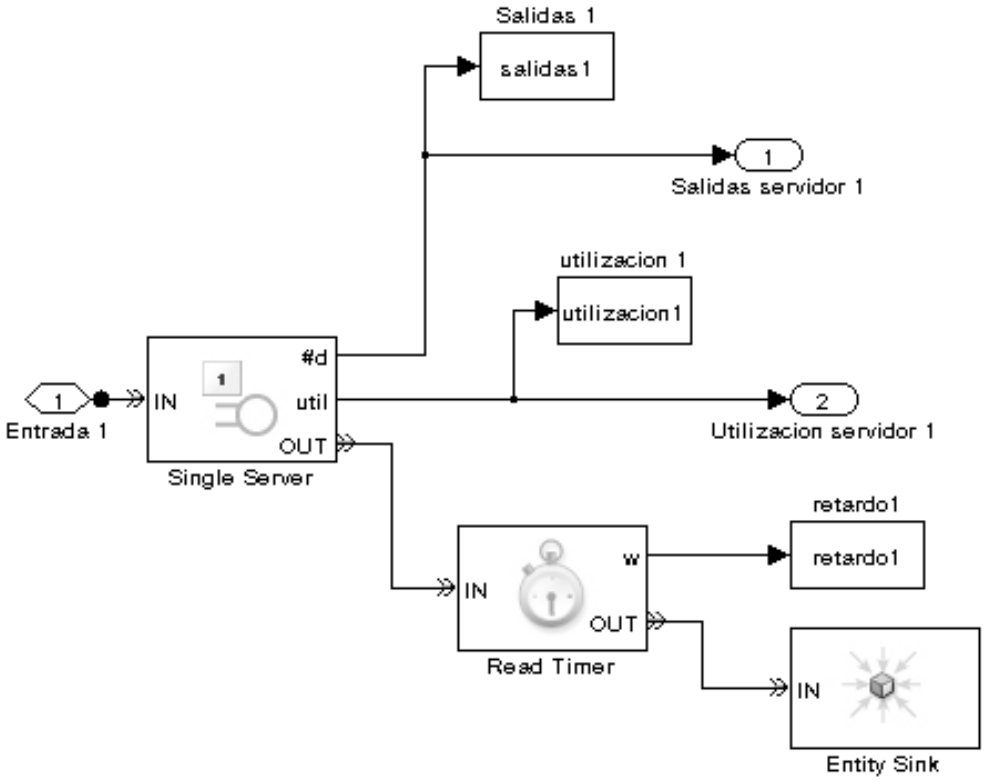
En las líneas de entrada hemos supuesto que el tráfico de llegada es de Bernouilli y que la probabilidad de que llegue una célula a una ranura es p, por lo que la probabilidad de que la ranura esté vacía es 1-p. Las variables aleatorias que modelan la llegada de células a los puertos de entrada son independientes e idénticamente distribuidas. Mediante el bloque de asignación de atributos vamos a ir etiquetando las salidas de las celdas mediante un vector de probabilidades. Las etiquetas irán saliendo de la línea de entrada de forma síncrona gracias a un generador de pulsos.

Las líneas de entradas la vamos a modelar en simulink de la siguiente forma:



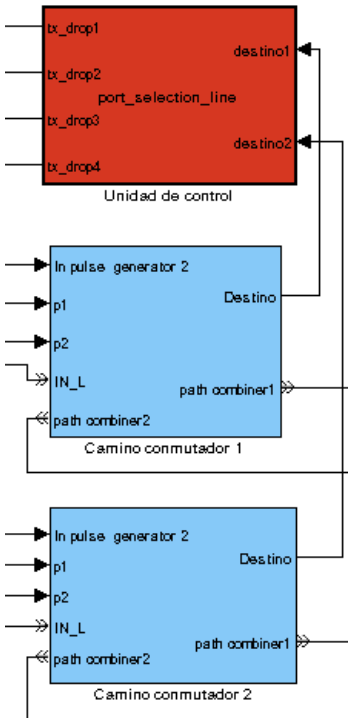
2.2 Líneas de salida.

En simulink tendrán el siguiente aspecto:

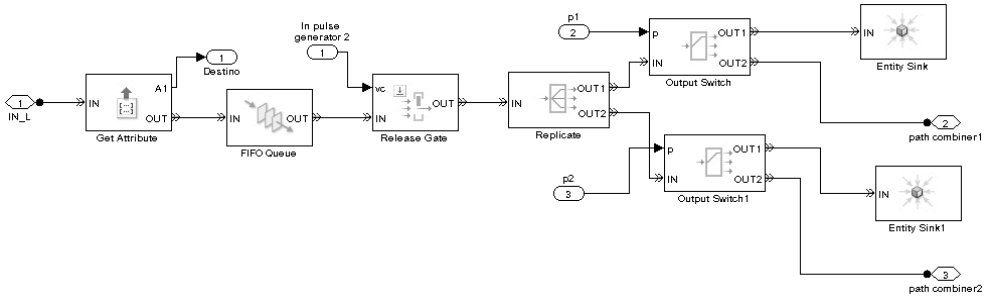


2.3 Red de interconexión.

La red de interconexión (conmutador) tiene el siguiente aspecto en el modelo usado para Simulink:



Donde el módulo camino conmutador tiene en su interior los siguientes bloques:



La red de interconexión proporciona la interconexión necesaria para la conmutación entre dos módulos de entrada y salida. En primer lugar vamos a extraer el destino de cada célula y vamos encolarlas para procesar la cabecera (modelado con una cola FIFO de una posición).

En segundo lugar implementaremos unas puertas de salida para introducir un retardo en la llegada de las células a la red de interconexión. El procesamiento de la cabecera presenta un retardo modelado mediante una diferencia de fases el generador de pulsos del conmutador y el generador de pulsos de las líneas de entrada. En nuestro caso la diferencia de fase entre los dos generadores de pulsos será de 0,5s.

Posteriormente introducimos un bloque de duplicadores de células para cada puerto de salida seguidos de los bloques que envían la célula al puerto de salida o a un sumidero dependiendo de las señales que envíe la unidad de control.

Por último habrá un combinador de caminos que combina las células provenientes de las N líneas de entrada y hacia las líneas de salida.

La unidad de control contiene la inteligencia que permite controlar los bloques transmite o descarta. En función de las direcciones de las células, se generan señales de control que gestionan la operación de los bloques transmite o descarta. Está imprementada como una función embebida de Matlab *(ver apéndice A)*.

Para hacer menos engorrosa la memoria, vamos a introducir un apéndice en el que introduciremos el codigo Matlab de la unidad de control y el script que hemos utilizado para simular el modelo para distintos valores de utilización para el conmutador 2x2.

Para el resto de conmutadores, solo hay que añadir tantas entradas y salidas como sean necesarias para llevar a cabo la lógica de control para la conmutación.

2.4 Otras suposiciones.

Todos los destinos (líneas de salida) son equiprobables para cada una de las células a la entrada. Además si varias células a la entrada tienen el mismo destino en un instante de tiempo, todas ellas tienen la misma probabilidad de alcanzar dicho destino (precedencia equiprobable para cada uno de los puertos de entrada).

Para todas las simulaciones se ha utilizado un tiempo de simulación de 10000 segundos. Además, el **periodo de todos los generadores de pulsos es 1**.

2.5 Parámetros a calcular.

- **Throughput:** es el número de células que el conmutador puede encaminar por éxito por unidad de tiempo o ranura. Por éxito se entiende que la célula alcance el puerto de salida deseado. Para obtener el throughput, calculamos en cada puerto de salida el número de células que son servidas y dividimos por el tiempo de simulación.

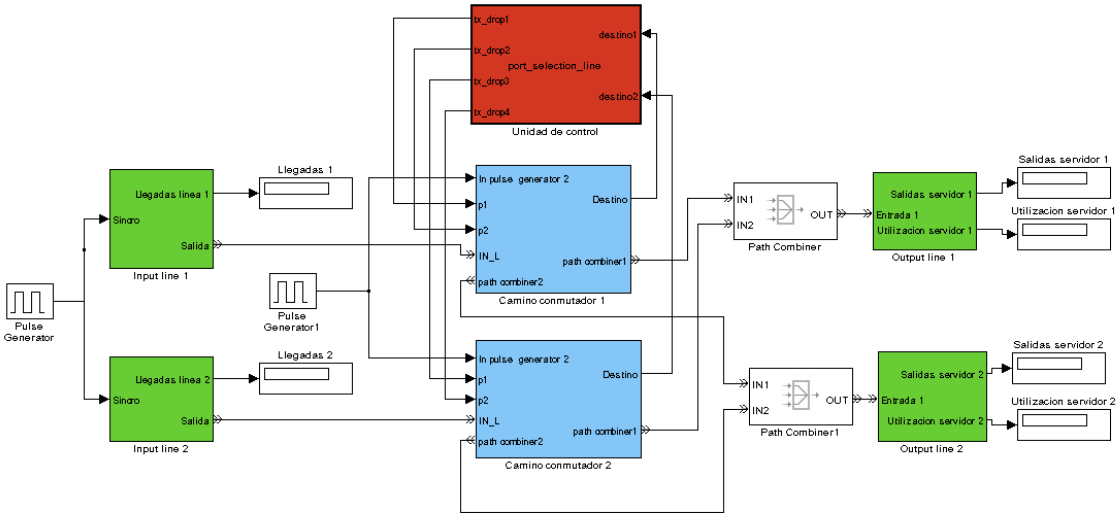
- **Probabilidad de error:** la perdida de células se produce cuando varias células compiten por un mismo recurso y algunas deben ser descartadas, es decir, se produce una colisión. . Para obtener este parámetro obtenemos las células descartadas mediante la diferencia de las células que entran y las que alcanzan su destino, y el resultado lo dividimos por las células que entran.
  - **Retardo:** es el retardo desde que se recibe el último bit de la célula por un puerto de entrada hasta que es retransmitido el último bit de la célula por un puerto de salida. En este caso vamos a introducir un reloj en la entrada y un reloj en todas las salidas para ver el retardo medio que experimenta un paquete en el conmutador.

En todos los casos vamos a representar estos parámetros en función de la utilización.

3. REALIZACIÓN PRÁCTICA:

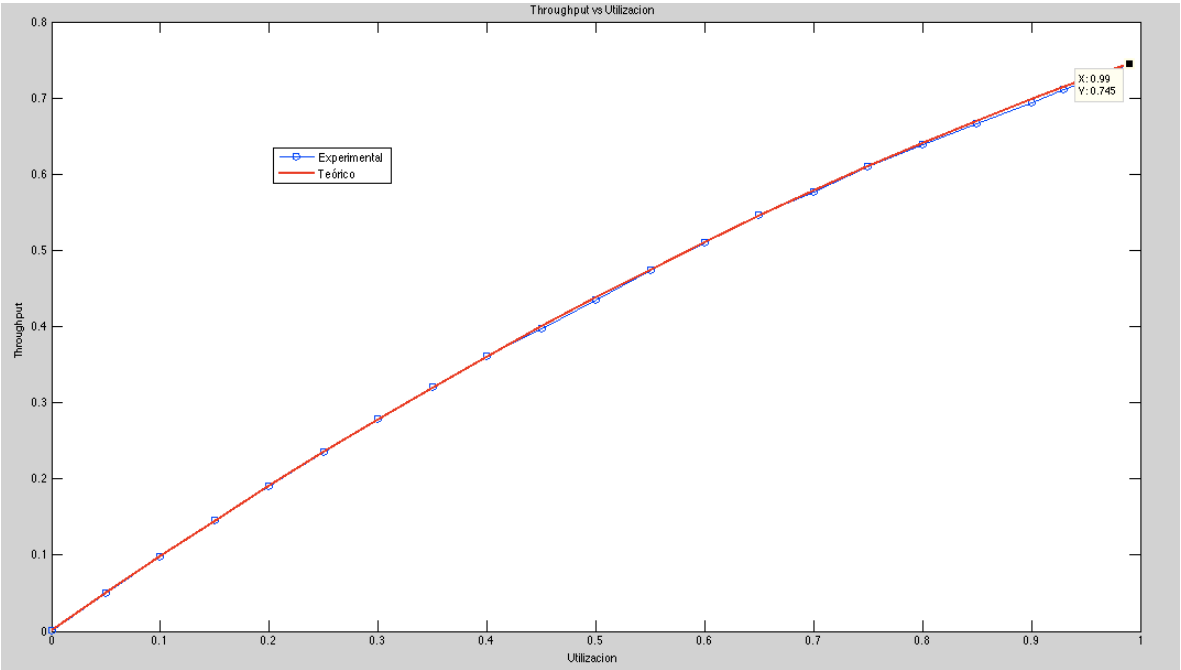
Apartado 1: Evaluación de Prestaciones de Conmutador Sin Memoria 2 x 2

En primer lugar diseñamos el modelo que tendrá la siguiente forma:



3.1.1. Throughput

Representamos gráficamente el throughput del conmutador en función de la utilización de las líneas de entradas (suponiendo que todas las líneas de entrada tienen la misma utilización). Además se muestra también la gráfica obtenida teóricamente mediante la aproximación de Lee.



Como podemos ver en la gráfica el throughput aumenta con la utilización y el máximo se obtiene cuando la utilización de la línea es 1. El valor obtenido es el siguiente:

Thmax=0,745 celulas/s

El caudal de células de salida (throughput) va a depender de la cantidad de tráfico que accede al conmutador a través de las líneas de entrada, de manera que cuantas más células lleguen al conmutador (mayor utilización), el caudal de salida será mayor.

Dicho throughput se obtiene debido a que las células llegan de manera más frecuente al conmutador en cada ranura, ya que la probabilidad de llegada de células es mayor y por tanto trabaja conmuta a la máxima velocidad para dar salida de células a esa tasa.

Cálculo teórico:

El throughput de una línea de salida se puede obtener teóricamente mediante la siguiente fórmula (aproximación de Lee):

$$Th_i = \left[1 - \left(1 - \frac{p}{N}\right)^N\right] \cdot \frac{1}{T}$$

Siendo:

- Thi: Throughput de la línea de salida i en células/s
- p: Probabilidad de que una célula llegue en una ranura temporal
- N: Numero de puertos de salida. En este caso N=2
- T: Duración de la ranura temporal. En este caso T=1 s

El Throughput máximo será para el valor p=1, es decir, la probabilidad de llegada de una célula es 1 y por tanto la utilización es 1. Calculando obtenemos que:

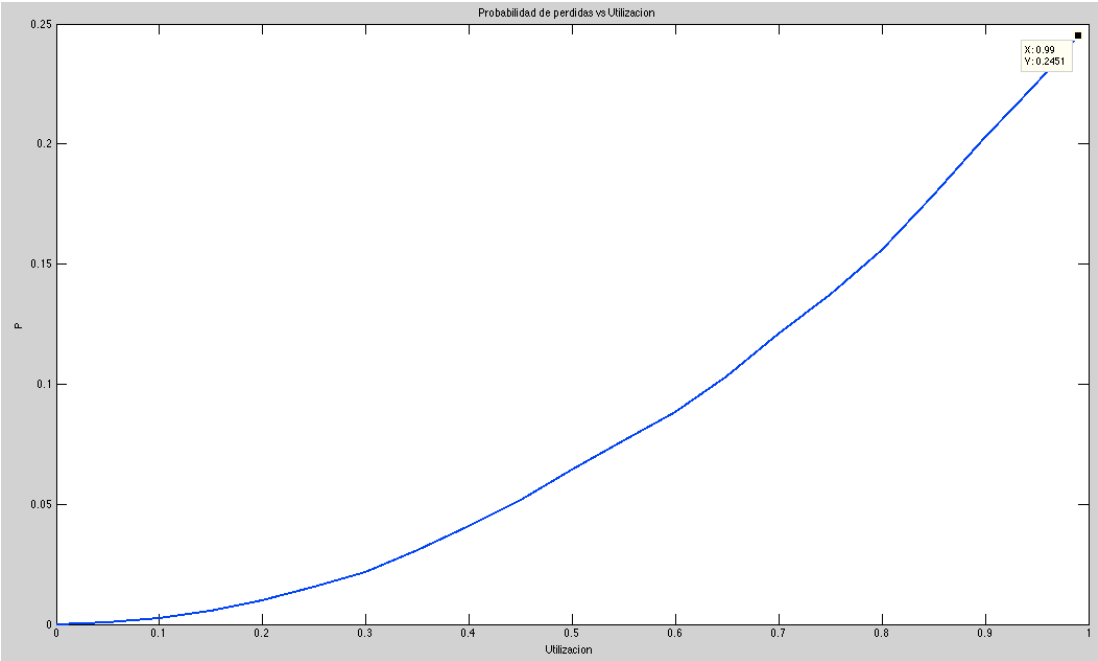
Thi =0,75 celulas/s

Podemos observar que el valor teórico está muy cerca del experimental. Calculamos el error entre ambos:

e= [(0,75-0,745)/0,75]*100=0,67%

3.1.2. Probabilidad de pérdidas

Si representamos la probabilidad de error en función de la utilización obtenemos la siguiente gráfica:



Podemos observar que cuando la utilización aumenta, llegan más células al conmutador por los puertos de entrada y por tanto aparecen más colisiones. Las células que colisionan son descartadas y por tanto aumenta la probabilidad de error.

La probabilidad de error máxima se obtiene de nuevo para la utilización del 100% por lo que hemos comentado antes, siendo dicho valor:

Prob de pérdidas=0,2451 =24,51%

Cálculo teórico:

Como sabemos, el throughput del puerto de salida se puede expresar como:

$$Thi=p*(1 - CLP)$$

Siendo CLP la probabilidad de pérdida de una célula. Despejando obtenemos:

$$CLP=(1-Thi)/p$$

Operando con p=1 para obtener el CLPmax obtenemos:

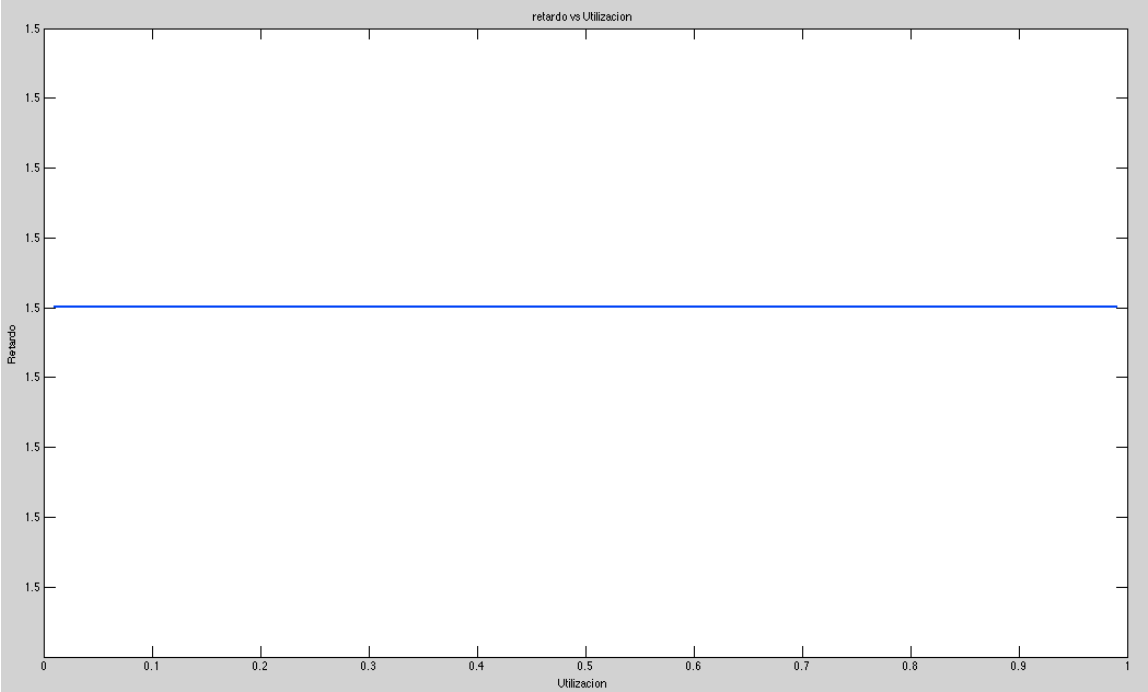
Clímax=0,25=25%

que se aproxima bastante al valor experimental. Vamos a obtener el error:

e=[(0,25-0.2451)/0,25]*100=1,96%

3.1.3. Retardo medio

Representamos el retardo en función de la utilización y obtenemos la siguiente gráfica:



Podemos observar que el retardo es constante y no depende de la utilización. El valor obtenido es:

Retardo=1,5 s

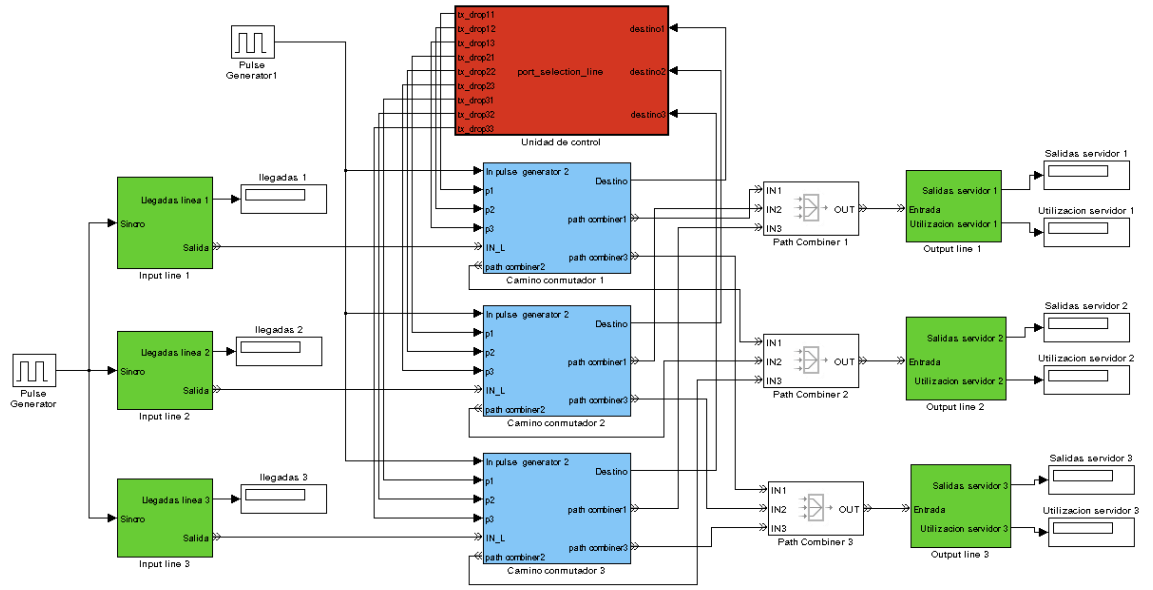
Este valor es constante ya que no hay colas ni en las entradas ni en las salidas del conmutador, por lo que los paquetes que no son desechados o no colisionan experimentan un retardo de 0,5 segundos de procesamiento de cabeceras más un 1 segundo en ser servidos a la línea de salida.

Se pide además que se calcule la desviación típica del retardo pero ya que es constante no es necesario calcularlo.

Apartado 2: Evaluación de Prestaciones de Conmutador Sin Memoria 3 x 3

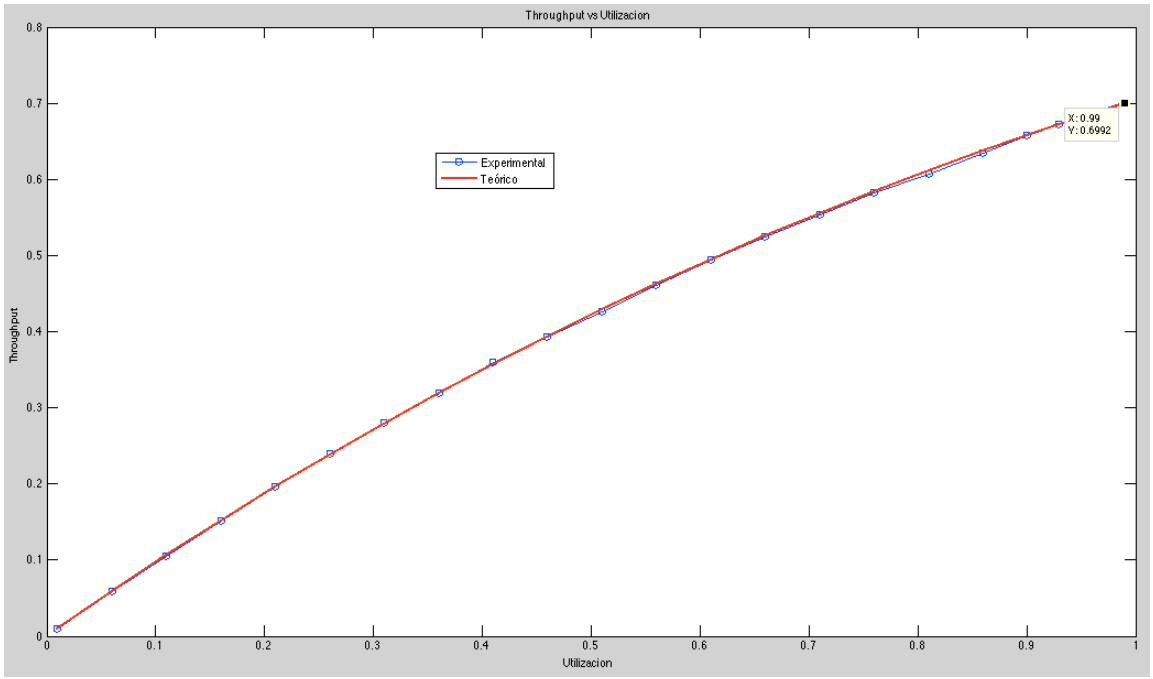
El conmutador conmuta células entre 3 líneas de entrada y 3 líneas de salida que son añadidas al modelo la red de interconexión y la unidad de control tiene el mismo funcionamiento que el explicado anteriormente con la diferencia de tener mayor complejidad debido al aumento del orden.

El modelo en Simulink es el mostrado en la siguiente figura:



3.2.1. Throughput

Representamos este parámetro junto con el calculado de manera teórica en función de la utilización obteniendo la siguiente figura:



Podemos ver cómo aumenta el throughput con la utilización de la misma forma que para el conmutador de 2x2. El valor máximo obtenido se alcanza cuando p=1, es decir, la utilización es del 100%.

THmax=0,6992 celulas/s

Cálculo teórico:

Calculamos de nuevo el throughput mediante el uso de la aproximación de Lee:

$$Th_i = \left[1 - \left(1 - \frac{p}{N}\right)^N\right] \cdot \frac{1}{T}$$

Siendo:

-Thi: Throughput del puerto de salida i.
- p: Probabilidad de que una célula llegue en una ranura
- N: Numero de puertos. En este caso N=3.
- T: Duración de la ranura temporal. En este caso T=1.

Aplicando la fórmula anterior obtenemos que:

Thimax=0,7037 células/s

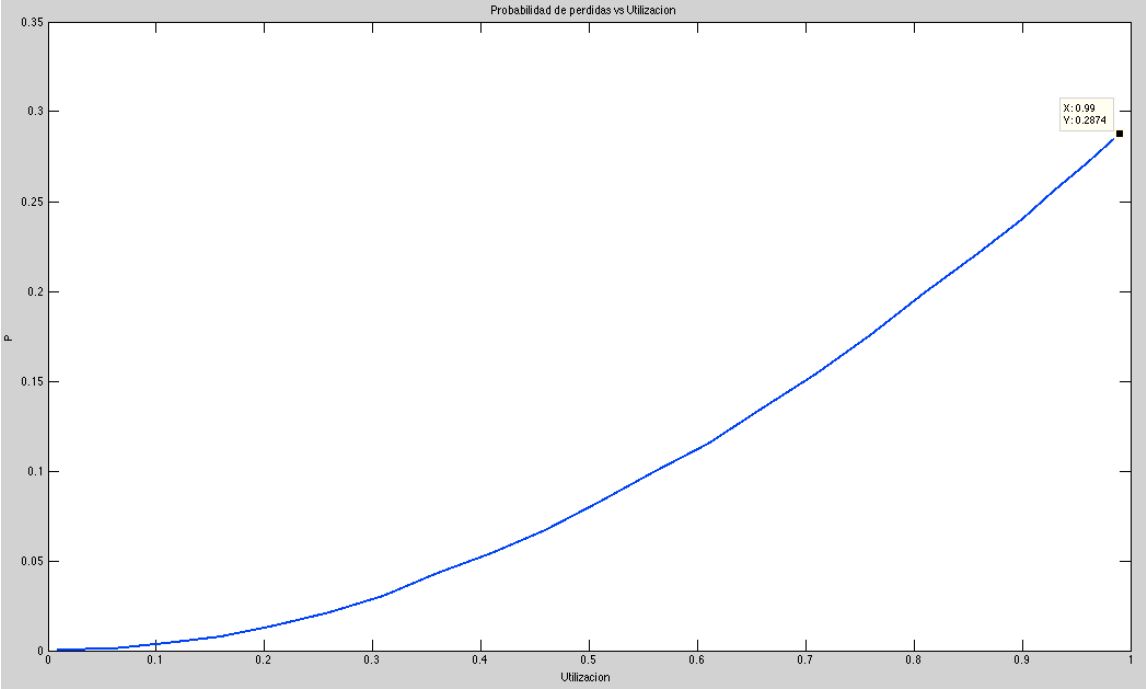
Obtenemos el error entre ambos valores:

e=[(0,7037-0,6992)/0,7037]*100=0,63%

Como podemos apreciar ambos resultados difieren muy poco por que la simulación es bastante buena.

3.2.2. Probabilidad de error

Representamos la probabilidad de error en función de la utilización.



Ocurre como en el caso del conmutador 2x2 que la probabilidad de error aumenta con la utilización. Vamos a obtener el valor máximo a partir de la gráfica:

Prob error max=0,2874=28,74%

Cálculo teórico:

A continuación vamos a calcular este valor mediante la fórmula:

CLP=(1-Thi)/p

El valor máximo se obtiene para p=1.

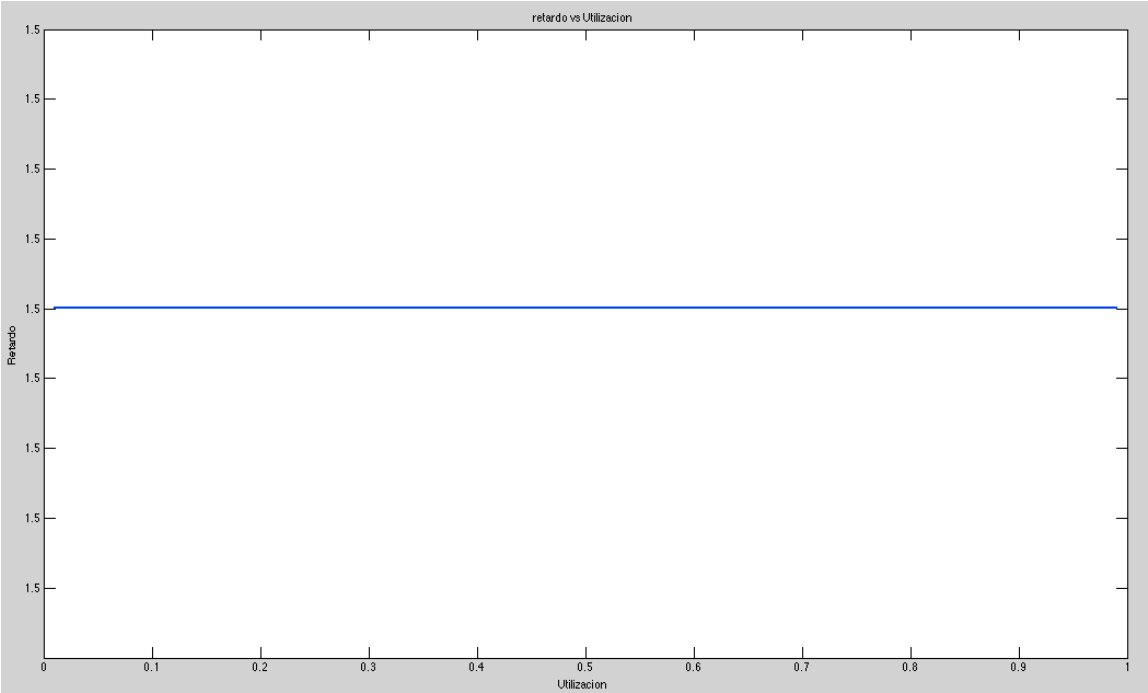
CLPmax=0,2963=29,63%

Podemos observar que se aproxima al valor experimental:

e= [(29,63-28,74)/ 29,63]*100=3,00%

3.2.3 Retardo medio

El retardo en función de la utilización se representa en la siguiente gráfica:



Ocurre exactamente lo mismo que en el conmutador 2x2. El valor del retardo medio por cada célula es constante y de valor:

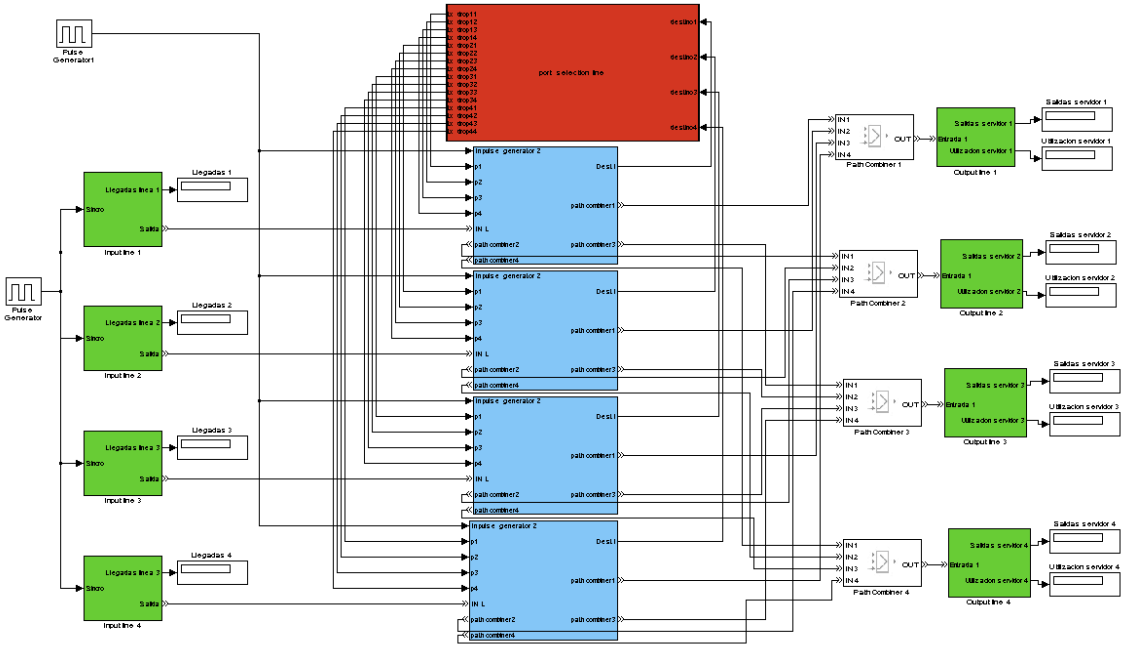
Retardo=1,5 segundos

De nuevo no es necesario calcular la desviación típica del retardo.

Apartado 3: Evaluación de Prestaciones de Conmutador Sin Memoria 4 x 4

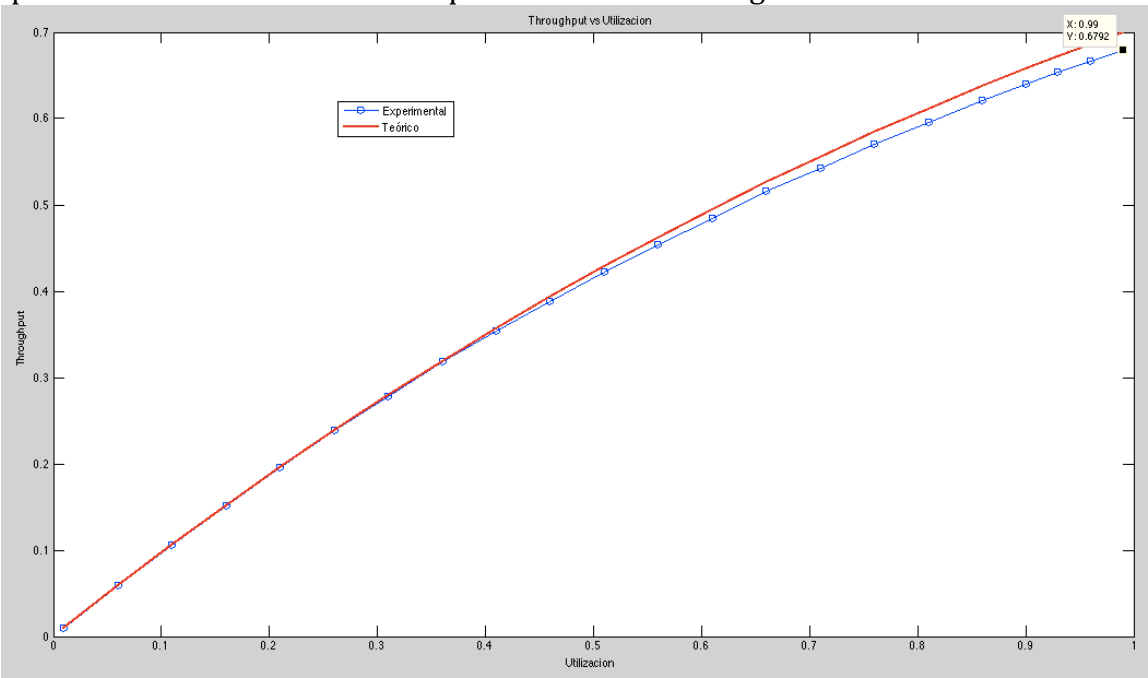
El conmutador conmuta células entre 4 líneas de entrada y 4 líneas de salida que son añadidas al modelo la red de interconexión y la unidad de control tiene el mismo funcionamiento que el explicado anteriormente con la diferencia de tener mayor complejidad debido al aumento del orden.

El modelo en Simulink es el mostrado en la siguiente figura:



3.3.1.1. Throughput.

Representamos este parámetro en función de la utilización. Al igual que en los apartados anteriores vamos a representar también la gráfica obtenida teóricamente:



El valor máximo del throughput alcanzado en la gráfica es:

Thmax=0,6792 células/s

Cálculo teórico:

Teóricamente sabemos que el throughput viene dado por la siguiente expresión:

$$Th_i = \left[1 - \left(1 - \frac{p}{N}\right)^N\right] \cdot \frac{1}{T}$$

Siendo:

- Thi: Throughput de la línea de salida i.
- p: Probabilidad de que una célula llegue en una ranura temporal.
- N: Numero de líneas de entrada o salida. En este caso N=4.
- T: Duración de la ranura temporal. En este caso T=1 segundo

El throughput máximo será para p=1, cuyo valor es el siguiente:

Thmax=0,6836 células/s

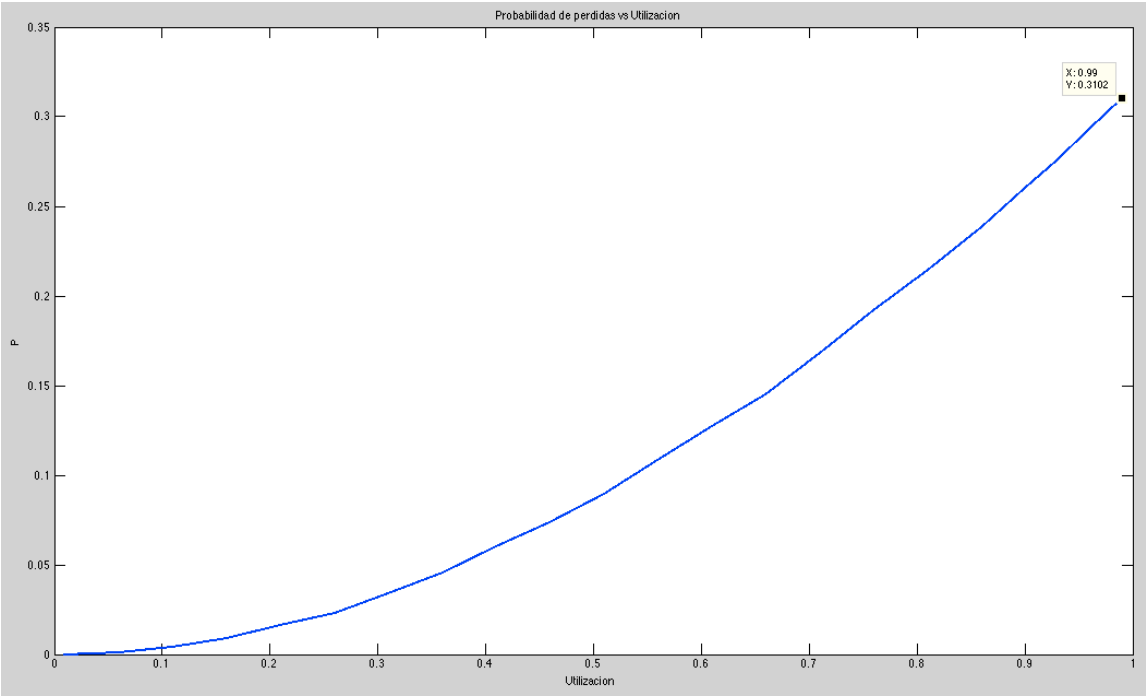
Vamos a obtener el error entre el resultado teórico y el resultado experimental:

e=[(0,6836-0,6792)/ 0,6836]*100=0,64%

Por tanto los resultados se asemejan mucho.

3.3.1.2. Probabilidad de error

Representamos la probabilidad de error en función de la utilización.



Ocurre como en el caso del conmutador 2x2 y 3x3 que la probabilidad de error aumenta con la utilización. Vamos a obtener el valor máximo a partir de la gráfica:

Prob error max=0,314=31,14%

Cálculo teórico:

CLP=(1-Thi)/p

El valor máximo se obtiene para p=1.

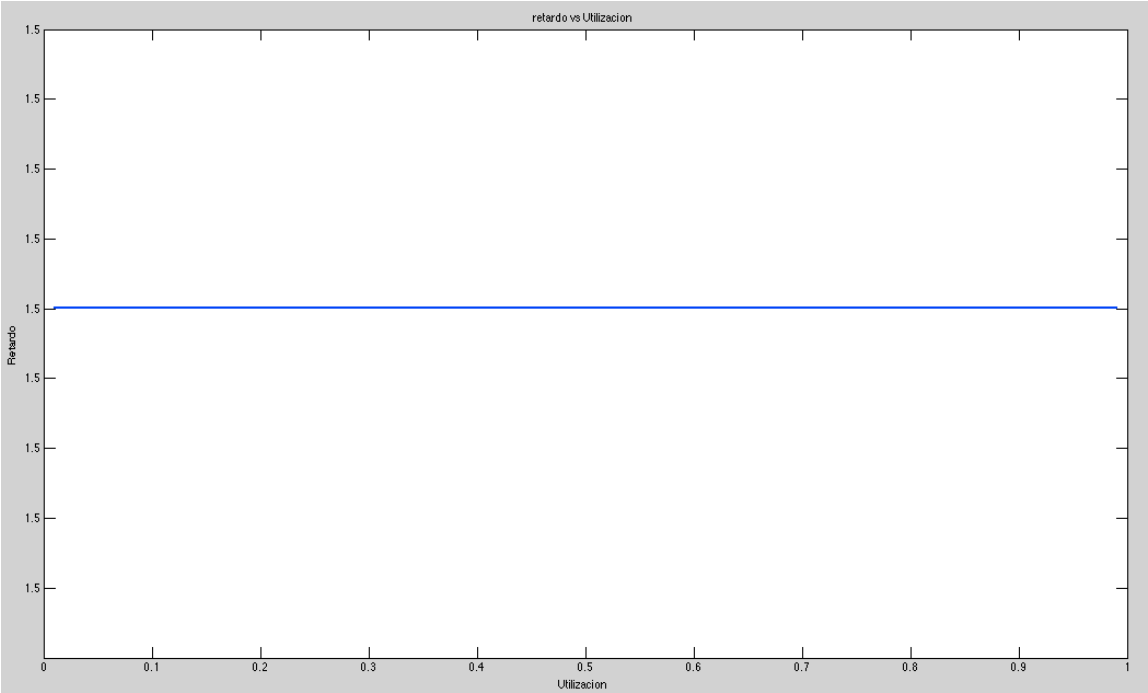
CLPmax=0,3164=31,64%

Podemos observar que se aproxima al valor experimental, calculemos el error:

e= [(31,64-31,14)/ 31,64]*100=1,58%

3.3.1.3 Retardo medio

El retardo en función de la utilización se representa en la siguiente gráfica:

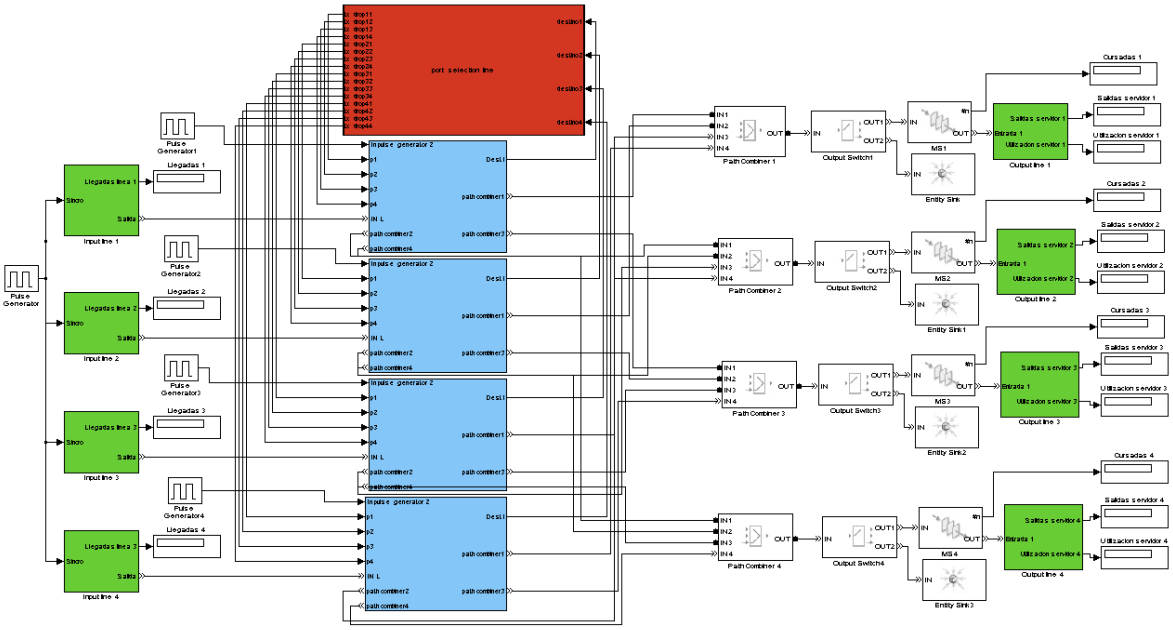


Ocurre exactamente lo mismo que en el conmutador 2x2 y 3x3. El valor del retardo medio por cada célula es:

Retardo=1,5 segundos

Apartado 3: Evaluación de Prestaciones de Conmutador Con Memoria a la Salida 4 x 4

El modelo usado para la simulación tiene el siguiente aspecto:



La diferencia con los otros modelos es que hemos introducido colas en los puertos de salida del conmutador 4x4. Esta arquitectura permite obtener un throughput del 100% en el conmutador.

En cambio, la red de interconexión debe funcionar 4 veces más rápido que en el conmutador sin memoria, ya que podría suceder que todas las células de entrada tuvieran que ser encaminadas por el mismo puerto de salida. Para ello, los generadores de pulsos van a tener un desfase de 0,25 segundos entre cada uno de ellos.

Además el buffer de salida ha de almacenar 4 células en cada instante de tiempo, con lo que la velocidad de la memoria impone un límite en el tamaño del conmutador.

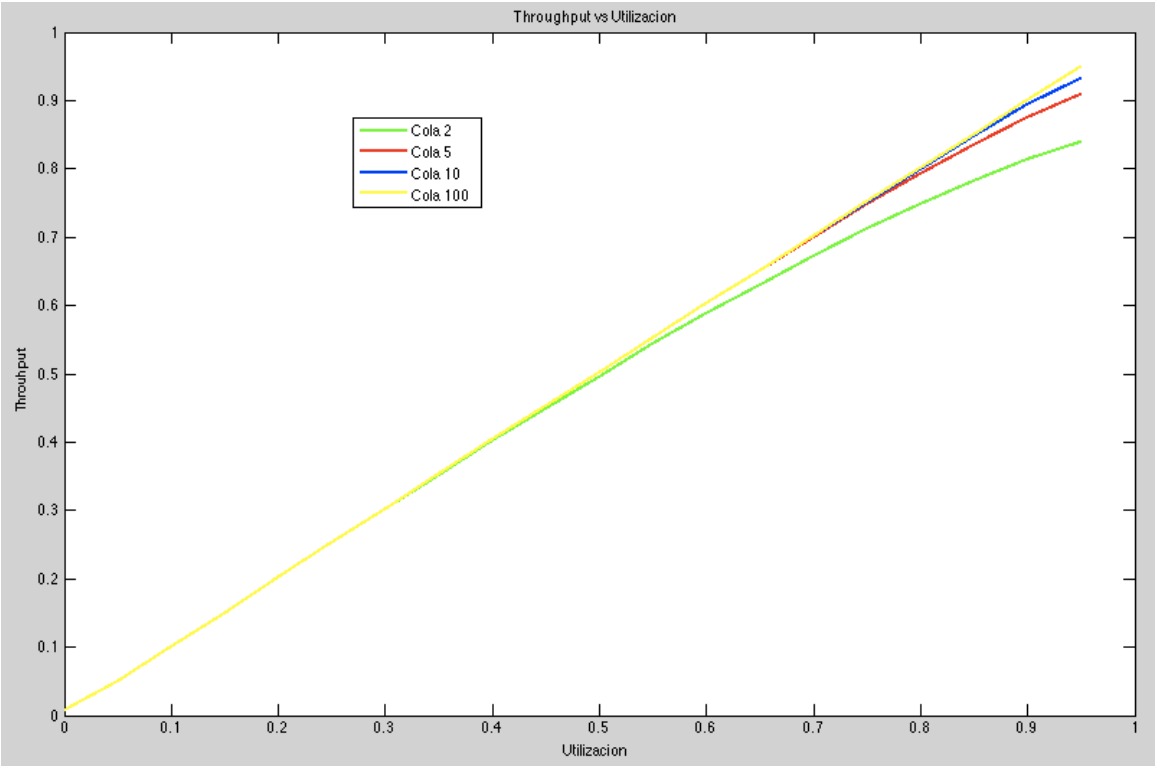
Si el tamaño de las memorias de salida son finitas, se pueden producir pérdidas de células si éstas encuentran los buffers llenos al salir de la red de interconexión, por ello vamos a introducir también sumideros a los que irán las células cuando las colas estén llenas.

En cuanto al tamaño de la cola, lo vamos a ir modificando para ir estudiando los parámetros de mérito. El script en el que calculamos los parámetros viene dado en el *apéndice B*.

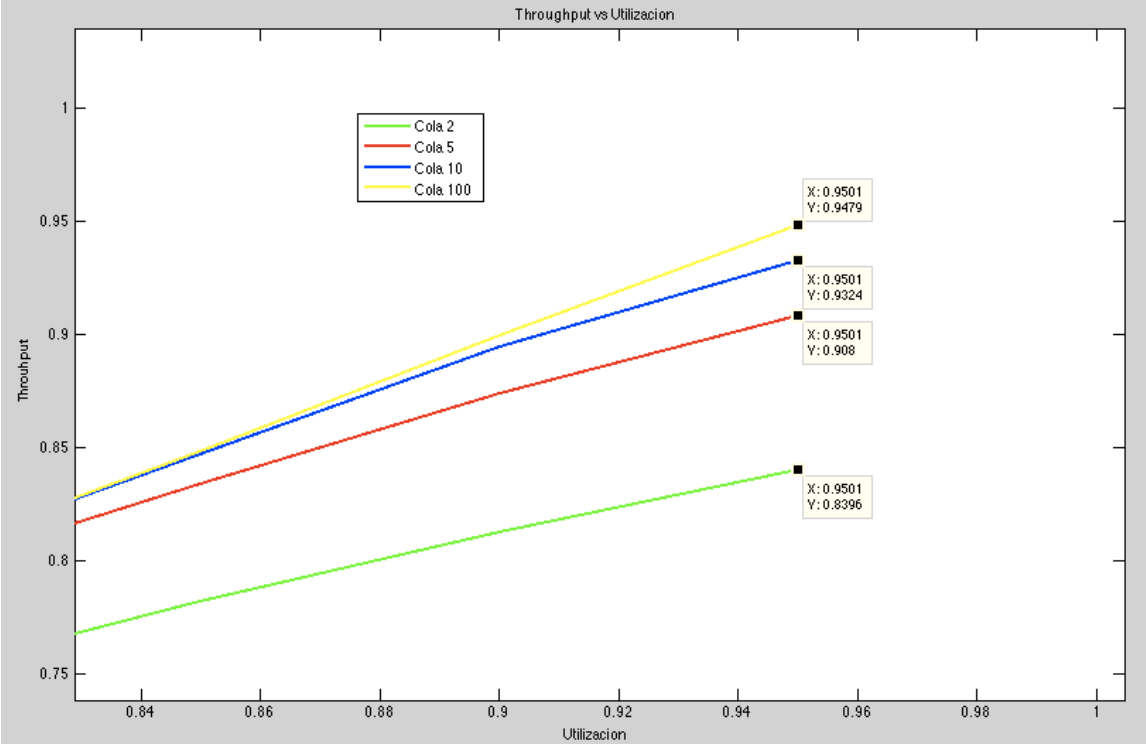
3.3.2.1 Throughput

Representamos gráficamente el throughput del conmutador en función de la utilización de las líneas de entrada (suponiendo que todas las líneas de entrada tienen la misma utilización) para los siguientes tamaños de memoria: 2, 5, 10 y 100 células.

La gráfica obtenida es la siguiente:



Vamos a verla más en detalle para saber el máximo de throughput que obtenemos:



Como vemos, el throughput aumenta con el tamaño de la cola debido a que se producen menos colisiones en las salidas ya que se almacenan en la cola de salida. En cambio, si la cola es muy pequeña, se desbordará y las células que colisionan serán desechadas reduciendo el throughput y teniendo un rendimiento más parecido al de los conmutadores sin memoria.

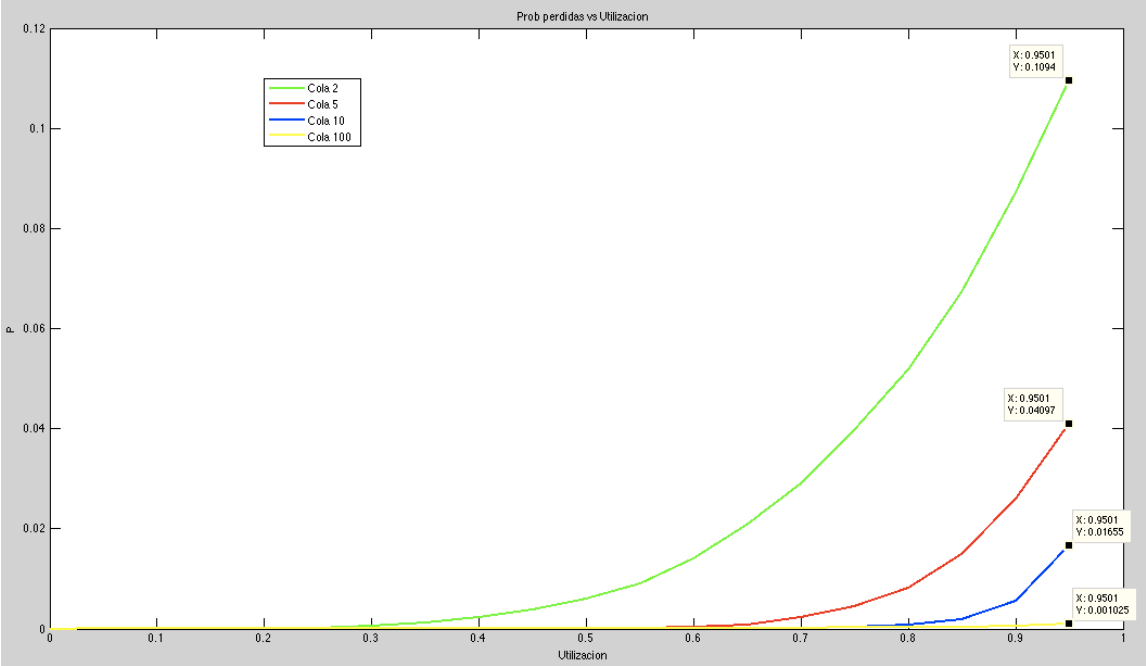
Los valores máximos vienen recogidos en la siguiente tabla:

Tamaño memoria salida	Throughput
2	0.8396
5	0.908
10	0.9324
100	0.9479

Podemos observar cómo aumenta el Throughput con el tamaño de la cola por las razones anteriores.

3.3.2.2. Probabilidad de pérdidas

Si representamos la probabilidad en función de la utilización. Obtenemos la siguiente gráfica:



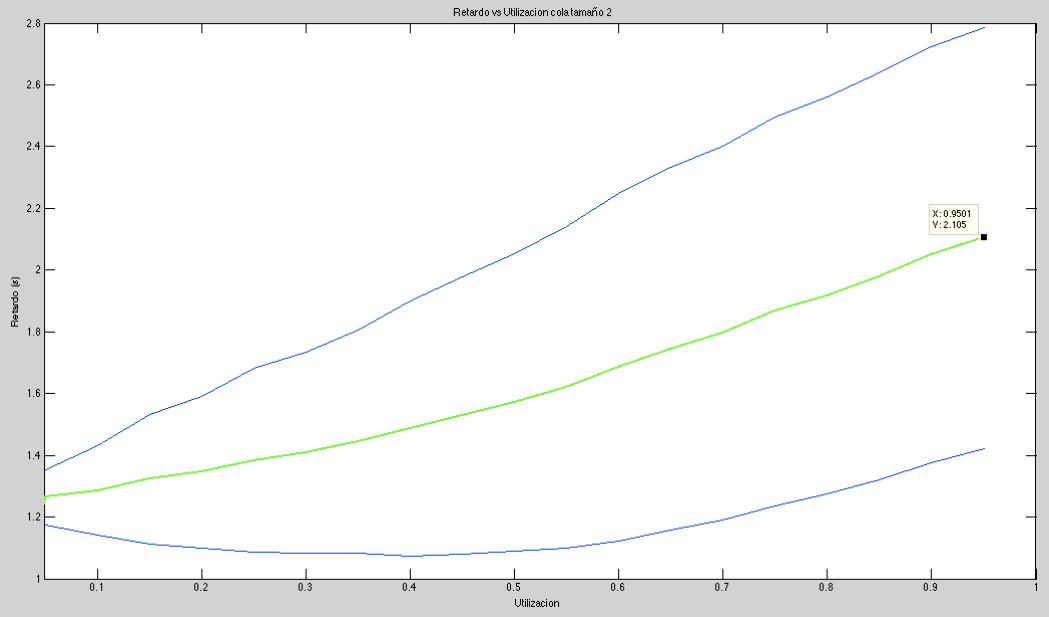
Tamaño memoria salida	Probabilidad de pérdidas
2	0.1094
5	0.04097
10	0.01655
100	0.001025

Podemos observar que la probabilidad de pérdidas disminuye al aumentar el tamaño de la memoria. Ello es debido a que cuando aumentamos el tamaño de la cola se pierden menos células ya que se encolan en la memoria de salida y menor será la probabilidad de que esté llena.

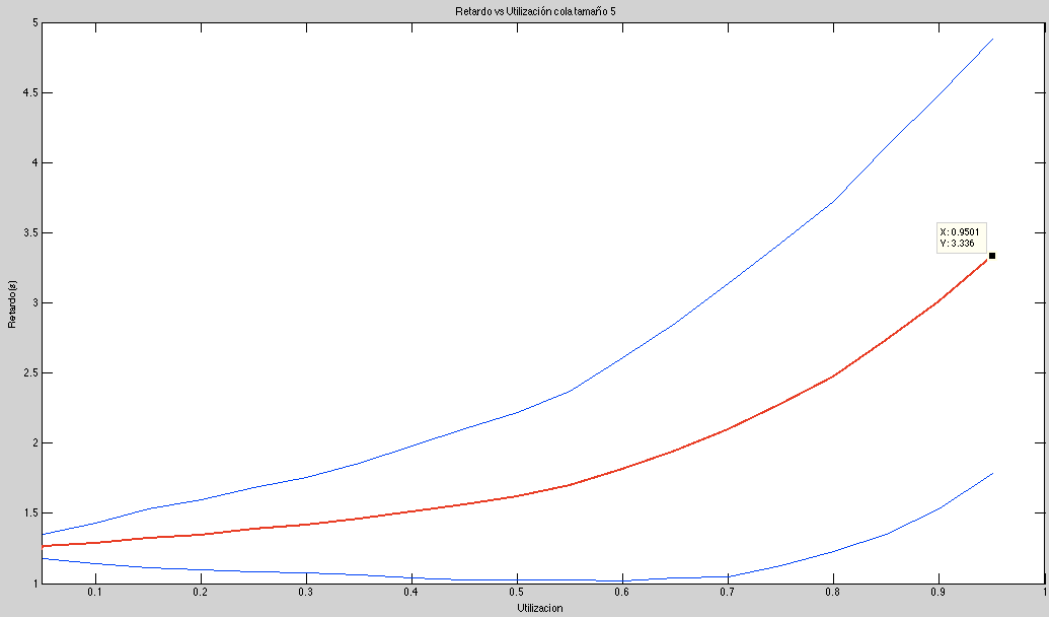
3.3.2.3 Retardo medio

Vamos a representar el retardo medio junto con la desviación típica del mismo para cada uno de los tamaños de cola:

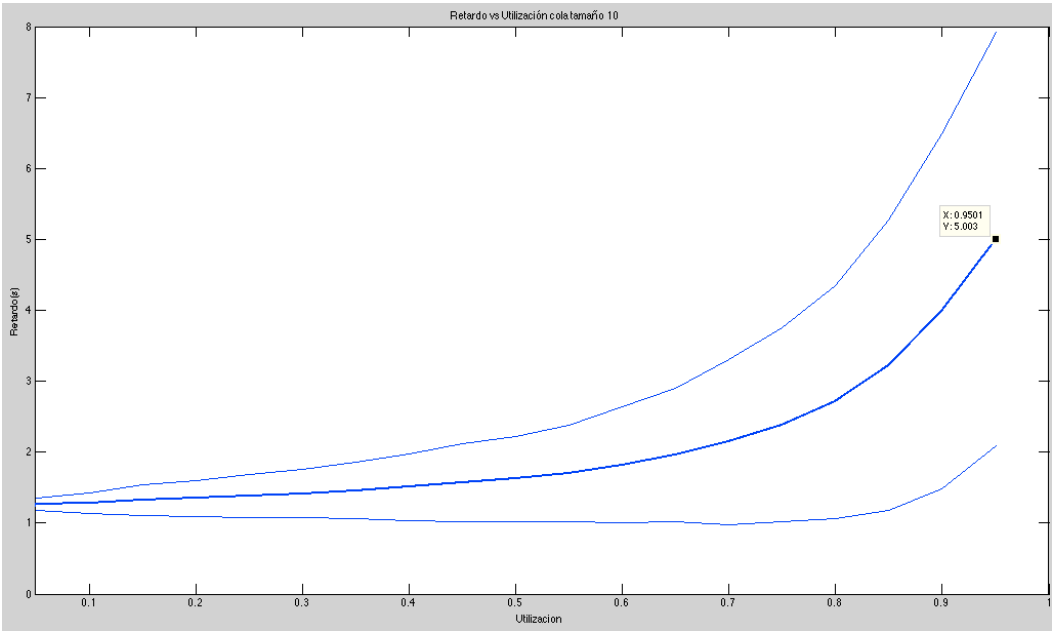
- **Tamaño de cola 2**



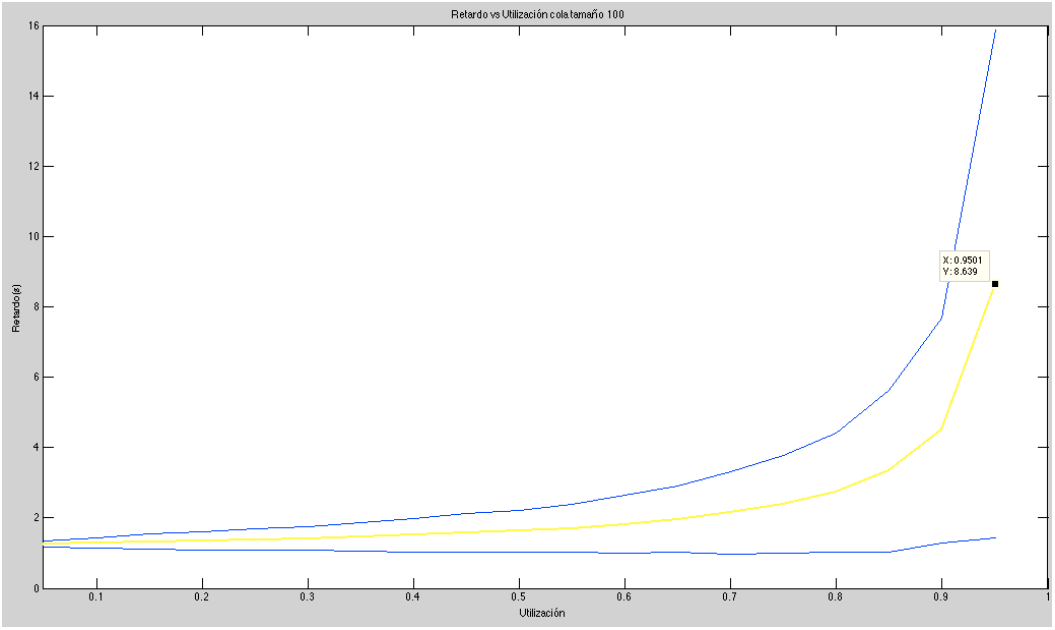
- **Tamaño de cola 5**



- **Tamaño de cola 10**

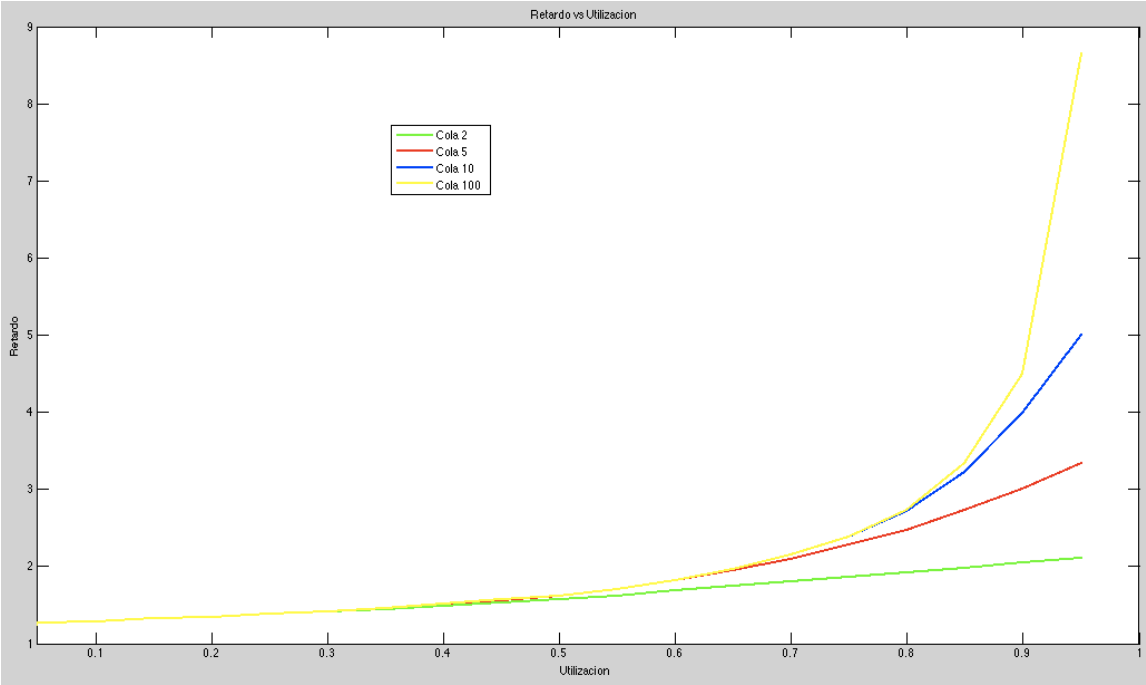


- **Tamaño de cola 100**



El retardo máximo corresponde a utilización máxima y aumenta con el tamaño de la cola, ya que cuanto más grande sea el tamaño más células habrá encoladas y el retardo será mayor.

Vamos a representar también los retardos para cada tamaño juntos sin sus desviaciones:



Tamaño memoria salida	Retardo medio	Desviación típica media del retardo
2	2.105	0.4519
5	3.336	0.7141
10	5.003	0.9174
100	8.639	1.1967

Cálculo teórico:

Cada una de las líneas de salida individualmente puede modelarse como un sistema de colas M/D/1, el cuál tiene una tasa de llegadas aleatorias, un tiempo de servicio constante con un servidor y una cola cuyo tamaño será variable.

Por tanto vamos a tener dos fuentes de retardo: un retardo debido al tiempo de servicio constante para servir la célula, y un retardo asociado al tiempo de espera en cola que será variable dependiendo del tamaño de la cola y las células que hayan encoladas.

La ecuación del retardo que rige el sistema es:

$$T = W_q + X = \frac{L_q}{\mu} + X$$

Siendo:

- T: Tiempo en el sistema
- Wq: Tiempo de espera en cola
- X: Tiempo de servicio
- Lq: Tamaño de la cola

4. CONCLUSIONES:

Vamos a realizar una comparativa de todos los sistemas analizados:

		Sin memoria			4x4 con memoria			
		2X2	3X3	4X4	Cola 2	Cola 5	Cola 10	Cola 100
Througput (células/s)	Teórico	0,75	0,7037	0,6836				
	Experimental	0,745	0,6992	0,6729	0,8396	0,908	0,9324	0,9479
Retardo (s)	Teórico	1,5	1,5	1,5				
	Experimental	1,5	1,5	1,5	2,105	3,336	5,003	8,639
Prob. Pérdida (%)	Teórica	25	29,63	31,64				
	Experimental	24,51	28,74	31,4	0,1094	0,04097	0,01655	0,001025

Como podemos observar, el throughput máximo disminuye cuando aumentamos el orden del conmutador en el caso de los conmutadores sin memoria. Para el conmutador con memoria, conforme mayor es la cola mayor es el throughput que se obtiene.

El retardo permanece constante en todos los sistemas sin memoria mientras que para el conmutador con memoria el retardo aumenta cuanto mayor es el tamaño de la cola. Además la desviación del mismo también crece con el tamaño de cola.

La probabilidad de pérdidas aumenta al aumentar la capacidad del sistema en sistemas sin memoria. Para el caso del conmutador con memoria la probabilidad de pérdida de una célula disminuye conforme se aumenta el tamaño de la cola.

Por tanto, parece claro que dependiendo de las necesidades que tengamos debemos elegir un sistema u otro ya que aunque el throughput y la probabilidad de

pérdidas son mejores para el conmutador con memoria, el hecho de que los conmutadores sin memoria tengan un retardo constante puede resultar muy beneficioso para aplicaciones en tiempo real. De ello se puede deducir que el sistema ideal sería una mezcla de ambos sistemas analizados.

Apéndice A: Matlab embedded

A continuación se muestran los códigos que van a hacer que sea posible la conmutación de células entre las líneas de entrada y salida. Para los conmutadores sin memoria se muestra sólo el correspondiente al conmutador de orden 4x4 ya que es el más complejo, siendo los códigos del 2x2 y 3x3 una mera simplificación del mismo.

Conmutadores sin memoria:

```
nction [tx_drop11,
_drop12,tx_drop13,tx_drop14,tx_drop21,tx_drop22,tx_drop23,tx_drop24
x_drop31,tx_drop32,tx_drop33,tx_drop34,tx_drop41,tx_drop42,tx_drop4
 tx_drop44] =
rt_selection_line(destino1,destino2,destino3,destino4)

atriz de puertos de salida (filas=entradas, columnas=salidas)
ertos_salida=ones(4,4); %Inicialmente todos los paquetes se
scartan.
ector de destinos de cada entrada
stinos=[destino1, destino2, destino3,destino4];

signamos el destino de cada célula a la línea de salida
rrespondiente

r i=1:1:length(destinos)
    puertos_salida(i,:)=designar_puertos(destinos(i));
d
omprobamos si existen colisiones
r i=1:length(puertos_salida(1,:))
   %creamos un vector para las colisiones
   entradas_en_conflicto=zeros(1,length(destinos));

   for j=1:length(puertos_salida(:,1))

       if puertos_salida(j,i)==2
          entradas_en_conflicto(j)=1;
       end
   end
```

```
     if sum(entradas_en_conflicto)>1
        entrada=elegir_entrada(entradas_en_conflicto) ;
        puertos_salida(:,i)=resolver_conflicto(entrada);
     end
end

%generamos las señales tx_drop
tx_drop11=puertos_salida(1,1);
tx_drop12=puertos_salida(1,2);
tx_drop13=puertos_salida(1,3);
tx_drop14=puertos_salida(1,4);
tx_drop21=puertos_salida(2,1);
tx_drop22=puertos_salida(2,2);
tx_drop23=puertos_salida(2,3);
tx_drop24=puertos_salida(2,4);
tx_drop31=puertos_salida(3,1);
tx_drop32=puertos_salida(3,2);
tx_drop33=puertos_salida(3,3);
tx_drop34=puertos_salida(3,4);
tx_drop41=puertos_salida(4,1);
tx_drop42=puertos_salida(4,2);
tx_drop43=puertos_salida(4,3);
tx_drop44=puertos_salida(4,4);

%%%%%%Funcion que asocia una salida con la configuracion de los
puertos de salida correspondiente
function [puertos]=designar_puertos(destino)
    puertos=ones(1,4);
    if destino~=0
        puertos(destino)=2;
     end
%%%%%%Funcion que escoge un elemento entre N elementos al azar. Se
usar· para resolver los conflictos
function [ent]=elegir_entrada(entradas)
    cont=0;
    ent=0;
    aux=sum(entradas);
    seleccion=ceil(aux.*rand(1));
    for i=1:length(entradas)

        if entradas(i)==1
            cont=cont+1;
            if cont==seleccion
            ent=i;
            end
        end
    end
%%%%%%Funcion que asigna los puertos de salida segun la entrada
function [columna]=resolver_conflicto(entrada)
    columna=ones(4,1);
    columna(entrada)=2;
```

```
function [tx_drop11,
tx_drop12,tx_drop13,tx_drop14,tx_drop21,tx_drop22,tx_drop23,tx_dro
p24,tx_drop31,tx_drop32,tx_drop33,tx_drop34,tx_drop41,tx_drop42,tx
_drop43,tx_drop44] =
port_selection_line(destino1,destino2,destino3,destino4)

%matriz de puertos de salida (filas=entradas, columnas=salidas)
puertos_salida=ones(4,4);
%vector de destinos de cada entrada

destinos=[destino1, destino2, destino3,destino4];


for i=1:1:length(destinos)
      puertos_salida(i,:)=designar_puertos(destinos(i));
end

%generamos las seÒales tx_drop
tx_drop11=puertos_salida(1,1);
tx_drop12=puertos_salida(1,2);
tx_drop13=puertos_salida(1,3);
tx_drop14=puertos_salida(1,4);
tx_drop21=puertos_salida(2,1);
tx_drop22=puertos_salida(2,2);
tx_drop23=puertos_salida(2,3);
tx_drop24=puertos_salida(2,4);
tx_drop31=puertos_salida(3,1);
tx_drop32=puertos_salida(3,2);
tx_drop33=puertos_salida(3,3);
tx_drop34=puertos_salida(3,4);
tx_drop41=puertos_salida(4,1);
tx_drop42=puertos_salida(4,2);
tx_drop43=puertos_salida(4,3);
tx_drop44=puertos_salida(4,4);

%%%%%Funcion que asocia una salida con la configuracion de los
puertos de salida correspondiente
function [puertos]=designar_puertos(destino)
    puertos=ones(1,4);
    if destino~=0
        puertos(destino)=2;
    end
```

Conmutador con memoria

Apéndice B: scripts de ejecución

Para facilitar la ejecución de los modelos y la extracción de datos y gráficas de los mismos se han implementado una serie de scripts para Matlab que se encargarán de variar la probabilidad de llegada de células y de simular tantas veces como sea necesario. De nuevo, al igual que en el apéndice A se muestra el correspondiente código para el conmutador 4x4 sin memoria y con memoria.

Conmutadores sin memoria:

```
Tsim=10000;              %tiempo de simulacion
throughput=[];           %vector que almacena el throughput del conmutador
del conmutador para cada valor de utilizacion
retardo=[];              %vector que almacena el retardo de una celula
para cada valor de utilizacion
prob_perdidas=[];        %vector que almacena la probabilidad de
descarte de una celula para cada valor de utilizacion
throug_teorico=[];       %vector que almacena el throughput teÛrico del
conmutador para cada valor de utilizacion

%%ahora definimos todos los valores de utilizacion que vamos a
evaluar  (22 en total)
util1=0.01:0.05:0.96;
util2=0.9:0.03:0.999;
utilizacion=[util1, util2];
throughput=zeros(1,length(utilizacion));
retardo=zeros(1,length(utilizacion));
prob_perdidas=zeros(1,length(utilizacion));
throug_teorico=zeros(1,length(utilizacion));

for i=1:length(utilizacion)

    p=utilizacion(i);
    %simulamos el modelo
    sim('conmutadorsinmemoria',Tsim);

 %%%%%calculamos los datos de interes
    %contamos el numero de celulas vacias para cada linea
    vacias1=0;
    vacias2=0;
    vacias3=0;
    vacias4=0;
    for k=1:length(destino1)
        if destino1(k)==0
            vacias1=vacias1+1;
        end
        if destino2(k)==0
            vacias2=vacias2+1;
        end
        if destino3(k)==0
            vacias3=vacias3+1;
        end
        if destino4(k)==0
            vacias4=vacias4+1;
        end
    end
```

```
 %obtenemos el throughput

throughput(i)=(utilizacion1(length(utilizacion1))+utilizacion2(
length(utilizacion2))+utilizacion3(length(utilizacion3))+utiliz
acion4(length(utilizacion4)))/4;
    %%obtenemos el throughput teorico
    throug_teorico(i)=(1-(1-(p/3))^3);

    %obtenemos el retardo

retardo(i)=(retardo1(length(retardo1))+retardo2(length(retardo2
))+retardo3(length(retardo3))+retardo4(length(retardo4)))/4;

    %calculamos la probabilidad de perdidas
    prob_perdidas(i)= (4*Tsim-
(vacias1+vacias2+vacias3+vacias4)-
(salidas1(length(salidas1))+salidas2(length(salidas2))+salidas3
(length(salidas3))+salidas4(length(salidas4))))/(Tsim*4);

end

%representamos los resultados obtenidos
figure;
plot(utilizacion,throughput);
title('Throughput vs Utilizacion','LineWidth',2);
xlabel('Utilizacion');
ylabel('Throughput');
hold on;
plot(utilizacion,throug_teorico,'r','LineWidth',2);

figure;
plot(utilizacion,retardo,'LineWidth',2);
title('retardo vs Utilizacion');
xlabel('Utilizacion');
ylabel('Retardo');

figure;
plot(utilizacion,prob_perdidas,'LineWidth',2);
title('Probabilidad de perdidas vs Utilizacion');
xlabel('Utilizacion');
ylabel('P');
```

Conmutador con memoria

```
Tsim=10000;
throughput=[];
retardo=[];
prob_perdidas=[];

%definimos todos los valores de utilizacion que vamos a evaluar
utilizacion=0.001:0.05:0.9999;

%vectores que contendran los resultados de cada tamaÒo de cola
throughputs=[];
retardos=[];
perdidas=[];
desviaciones=[];
throughput=zeros(1,length(utilizacion));
retardo=zeros(1,length(utilizacion));
prob_perdidas=zeros(1,length(utilizacion));
desviacion=zeros(1,length(utilizacion));

%definimos tambien los tamaÒos de cola que vamos a evaluar
tam_colas=[2 5 10 100];

for j=1:length(tam_colas)
    tam_fifo=tam_colas(j);
for i=1:length(utilizacion)

    p=utilizacion(i);
    sim('conmutadorconmemoria',Tsim);

 %%%%%calculamos los datos de interes
    %contamos las ranuras que estaban vacÌas
    vacias1=0;
    vacias2=0;
    vacias3=0;
    vacias4=0;
    for k=1:length(destinos1)
        if destinos1(k)==0
            vacias1=vacias1+1;
        end
        if destinos2(k)==0
            vacias2=vacias2+1;
        end
        if destinos3(k)==0
            vacias3=vacias3+1;
        end
        if destinos4(k)==0
            vacias4=vacias4+1;
        end
    end

     %obtenemos el throughput

throughput(i)=(utilizacion1(length(utilizacion1))+utilizacion2(
length(utilizacion2))+utilizacion3(length(utilizacion3))+utiliz
acion4(length(utilizacion4)))/4;
```

```
retardo(i)=(retardomedio1(length(retardomedio1))+retardomedio2(len
gth(retardomedio2))+retardomedio3(length(retardomedio3))+retardome
dio4(length(retardomedio4)))/4;

    %desviaciÛn tÌpica

desviacion(i)=(std(retardo1)+std(retardo2)+std(retardo3)+std(retar
do4))/4;

    %calculamos la probabilidad de perdidas
    prob_perdidas(i)= (4*Tsim-(vacias1+vacias2+vacias3+vacias4)-
(salidas1(length(salidas1))+salidas2(length(salidas2))+salidas3(le
ngth(salidas3))+salidas4(length(salidas4))))/(Tsim*4);

end

throughputs=[throughputs;throughput];
retardos=[retardos;retardo];
perdidas=[perdidas;prob_perdidas];
desviaciones=[desviaciones;desviacion];
end %for c(recorrido de los tamaÒos de cola)


%definimos los colores que usaremos para las gr·ficas
colores=['g' 'r' 'b' 'y'];

%representamos los datos obtenidos
figure;
for c=1:length(tam_colas)
    title('Throughput vs Utilizacion')
    plot(utilizacion,throughputs(c,:),colores(c),'LineWidth',2);
    hold on;
end

figure;
for c=1:length(tam_colas)
    title('retardo vs Utilizacion')
    plot(utilizacion,retardos(c,:),colores(c),'LineWidth',2);
    hold on;

plot(utilizacion,retardos(c,:)+desviaciones(c,:),'LineWidth',1);
    hold on;
    plot(utilizacion,retardos(c,:)-
desviaciones(c,:),'LineWidth',1);
    hold on
end

figure;
for c=1:length(tam_colas)
    title('Prob perdidas vs Utilizacion')
     plot(utilizacion,perdidas(c,:),colores(c),'LineWidth',2);
    hold on;
end
```