Practica 1:

Colas:

Generar tiempos entre llegadas y tiempos de servicio

RndExp[rate_]:=-Log[RandomReal[]]/rate //N

interArrivals=Table[RndExp[mu],nm]

Acumulamos llegadas:

Arrivals = Accumulate[interArrivals]

Función FIFO (returns departures)

Fifo[arrivals_,service_]:=Module[{n, checktime},

n=0;

checktime=arrivals[[1]];

Map[(n++;If[checktime>=#,checktime+=service[[n]],checktime=#+service[[n]]])&,arrivals]

]

Representación del Resultado

ListStepPlot[{arrivals,departures}]

➔ Use Manipulate[f(x),{x,0,5,1}]

Creamos "Montaña"

1. Marcamos arrays Llegadas y salidas (1,-1) → Mark[arrivals,1]
2. Ordenamos → Event=Short[Join[arrivals1,departures1]]
3. Creamos la función ({tInidicio,tFinal,Nº Paquetes}):
   Mont[lst_]:=Module[{n,time},
   n=0;
   time=0;
   Map[(time,time=#[[1]],If[#[[2]]==1,n++,n--])&,lst]
   ]
4. ***Recordatorio que con **##&[]** será como un return en blanco***

Podemos realizar el calculo de las probabilidades seleccionando los tiempos de inicio y final de la función anterior para un valor de n concreto, sumar esa diferencia de tiempos, y dividirla entre el tiempo total del experimento.

S&W , GBN, etc …

Recordatorios iniciales:

1. Carga de la librería: Get[“path”]
2. Visualización de las funciones: ?drawTxPRM`*

Set Inicial + Funciones Importantes:

1. SetIniParDraw[tPropagacion,tProcesadoACK]
2. FIFO básico S&W , GBN:

   Igual que el Fifo original, solo que deberemos de tener en cuenta el sumar al checktime un valor de 2*tp+ts antes del condicional, y acordarse de inicializar el checktime antes del map con un valor de –(2*tp+ts) y así comenzar en checktime =0 en el primer calculo.

Practica 3:

Tratamiento de imagen:

1. Cargamos imagen → img = . . .; (Se puede hacer arrastrando el archivo al Mathematica)
2. Guardamos las dimensiones → size=ImageDimensions[img];
3. Obtenemos los bits → Flatten[ImageData[Binarize[img]]]

Canales:

Canal Z:

$$\text{CanalZ}[\text{bits\_}, \text{p\_}] := \text{Map}[(\text{If}[\# == 1, \text{If}[\text{Random}[] < p, 0,1],0])\&, \text{bits}]$$

Canal BSC:

$$\text{Bsc}[\text{bits\_}, \text{p\_}] := \text{Map}[(\text{If}[\text{Random}[] <= \text{p}, \text{Mod}[\# + 1,2], \#])\&, \text{bits}]$$

Codificación por repetición:

$$\text{Codif}[\text{bits\_}, \text{n\_}] := \text{Flatten}[\text{Map}[(\text{Table}[\#,2 * n + 1])\&, \text{bits}]]$$

Decodificación por mayoría:

$$\text{DecoMay}[\text{bits\_}, \text{n\_}] := \text{Map}[(\text{If}[\text{Count}[\#,1] <= n, 0,1])\&, \text{Partition}[\text{bits}, 2n + 1]]$$

Probabilidad de error:

$$\text{Pe2}[\text{n\_}, \text{p\_}] := \text{Binomial}[2n + 1, n + 1] * p\text{^}(n + 1)$$

Dudas resueltas clase repaso:

1. Demostrar si es de carácter exponencial / Puassoniano