

```

Archivo = 'BC-pAug89.TL'; % Nombre del archivo
identificador = fopen(Archivo, 'r'); % Lo abre para lectura
traza = fscanf(identificador, '%f', [2 inf]); % Lee la traza: dos columnas, todas las filas
fclose(identificador); % Cierra el archivo
% Ahora la variable traza contiene 2 columnas de 1000000 de filas que
% registran el tráfico en una red ethernet de BellCore del 29 de agosto de
% 1989 desde las 11:25 am durante cerca de una hora. La primera columna de
% la i-ésima fila da el tiempo en segundos desde el inicio de la traza
% hasta la llegada del i-ésimo paquete. La segunda columna de la i-ésima
% fila da la longitud de la i-ésima trama ethernet en bytes, sin incluir el
% preámbulo, el encabezado ni el CRC (http://ita.ee.lbl.gov/html/contrib/BC.html)
TiempoPromedioEntreLlegadas = mean(traza(2:end,1) - traza(1:end-1,1));
TamanoPromedioPaquete = mean(traza(:,2));
MM1 = false; % Si MM1 es "true", simula una cola MM1 con los
if MM1 % mismos promedios de la traza leída
    traza = zeros(1000000,2);
    traza(:,1) = cumsum(-TiempoPromedioEntreLlegadas*log(rand(1000000,1)));
    traza(:,2) = -TamanoPromedioPaquete*log(rand(1000000,1));
end
NP=max(size(traza)); % Número de paquetes
Tfinal = traza(end,1); % Tiempo de simulación
C=1544000/8; % Capacidad del enlace T1 (bytes/segundo)
NumEnlacesLibres = 1; % Número de enlaces
Tsalida = inf; % Tiempo de la próxima llegada (enlace desocupado)
SiguientePaquete=1; % Paquete con el que comienza (índice en traza)
Tllegada = traza(SiguientePaquete,1); % Tiempo de la próxima llegada
CupoCola = 120000; % Número máximo de paquetes en la cola
PrimeroCola = 1; % Puntero al primero de la cola
UltimoCola = 1; % Puntero al último de la cola
Cola=zeros(CupoCola,1); % Arreglo de la cola (almacena el tamaño del paquete)
Reloj=0; % Reloj de simulación
LongitudCola=zeros(50000,1); T=LongitudCola; k=0; % Registra algunos eventos
while Reloj < Tfinal % Simulación de eventos discretos
    RelojA = Reloj;
    [Reloj,i]=min([Tllegada Tsalida]); % Actualiza el reloj al siguiente evento
    if i==1 % Procesa una llegada
        if NumEnlacesLibres>0 % Si el enlace está libre
            NumEnlacesLibres = NumEnlacesLibres - 1; % lo ocupa
            Tsalida = Reloj + traza(SiguientePaquete,2)/C; % y programa su salida
        else
            Cola(UltimoCola)=traza(SiguientePaquete,2); % Si no, lo almacena
            UltimoCola=UltimoCola+1; % en la cola
            if UltimoCola>CupoCola, UltimoCola=1; end
        end
        if SiguientePaquete<NP % En todo caso, programa
            SiguientePaquete=SiguientePaquete+1; % la próxima llegada
            Tllegada = traza(SiguientePaquete,1);
        end
    else % Procesa una salida
        if PrimeroCola==UltimoCola % Si la cola queda desocupada
            NumEnlacesLibres=NumEnlacesLibres+1; % libera el enlace
            Tsalida=inf; % y no se espera ninguna salida
        else % pero, si hay paquetes en la cola,
            Tsalida = Reloj + Cola(PrimeroCola)/C; % Programa la próxima
            Cola(PrimeroCola)=0; % salida y saca el primer paquete
            PrimeroCola=PrimeroCola+1; % de la cola
            if PrimeroCola>CupoCola, PrimeroCola=1; end
        end
    end
    if floor(10*RelojA)<floor(10*Reloj), % Registra el estado de la cola
        k=k+1; % cada 10 segundos.
        LongitudCola(k)=sum(Cola); % Longitud en bytes
        T(k)=Reloj;
    end
end
while PrimeroCola~=UltimoCola % desocupa la cola
    RelojA = Reloj;
    Reloj = Reloj + Cola(PrimeroCola)/C;
    Cola(PrimeroCola)=0;
    PrimeroCola=PrimeroCola+1;
end

```

```

if PrimeroCola>CupoCola, PrimeroCola=1; end
if floor(10*RelojA)<floor(10*Reloj), % Registra el estado de la cola
    k=k+1; % cada 10 segundos.
    LongitudCola(k)=sum(Cola); % Longitud en bytes
    T(k)=Reloj;
end
end
T=T(1:k); LongitudCola=LongitudCola(1:k); % Muestras de la longitud de la cola
plot(T,LongitudCola,'b-'); drawnow
axis tight
title('Longitud de la cola en bytes')
xlabel('Tiempo en segundos')
ylabel('Numero de bytes en cola')

```

