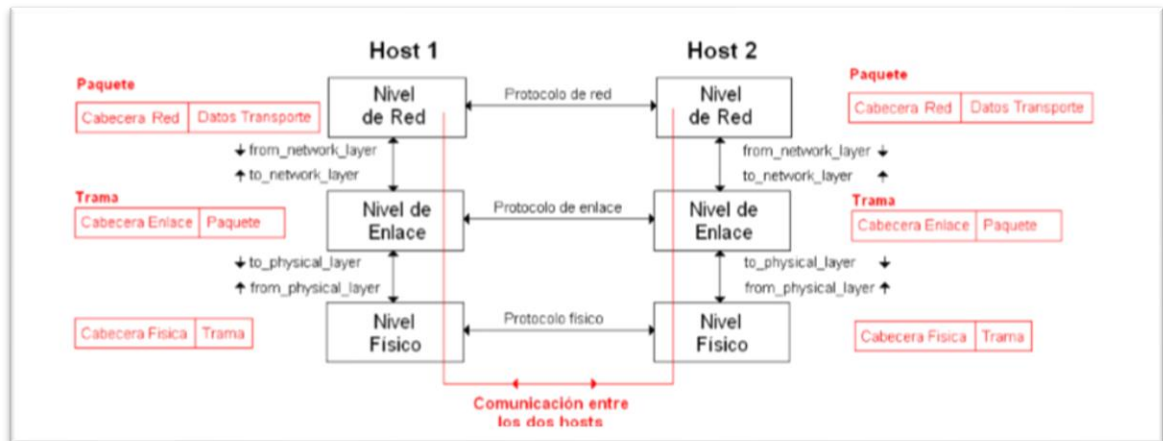


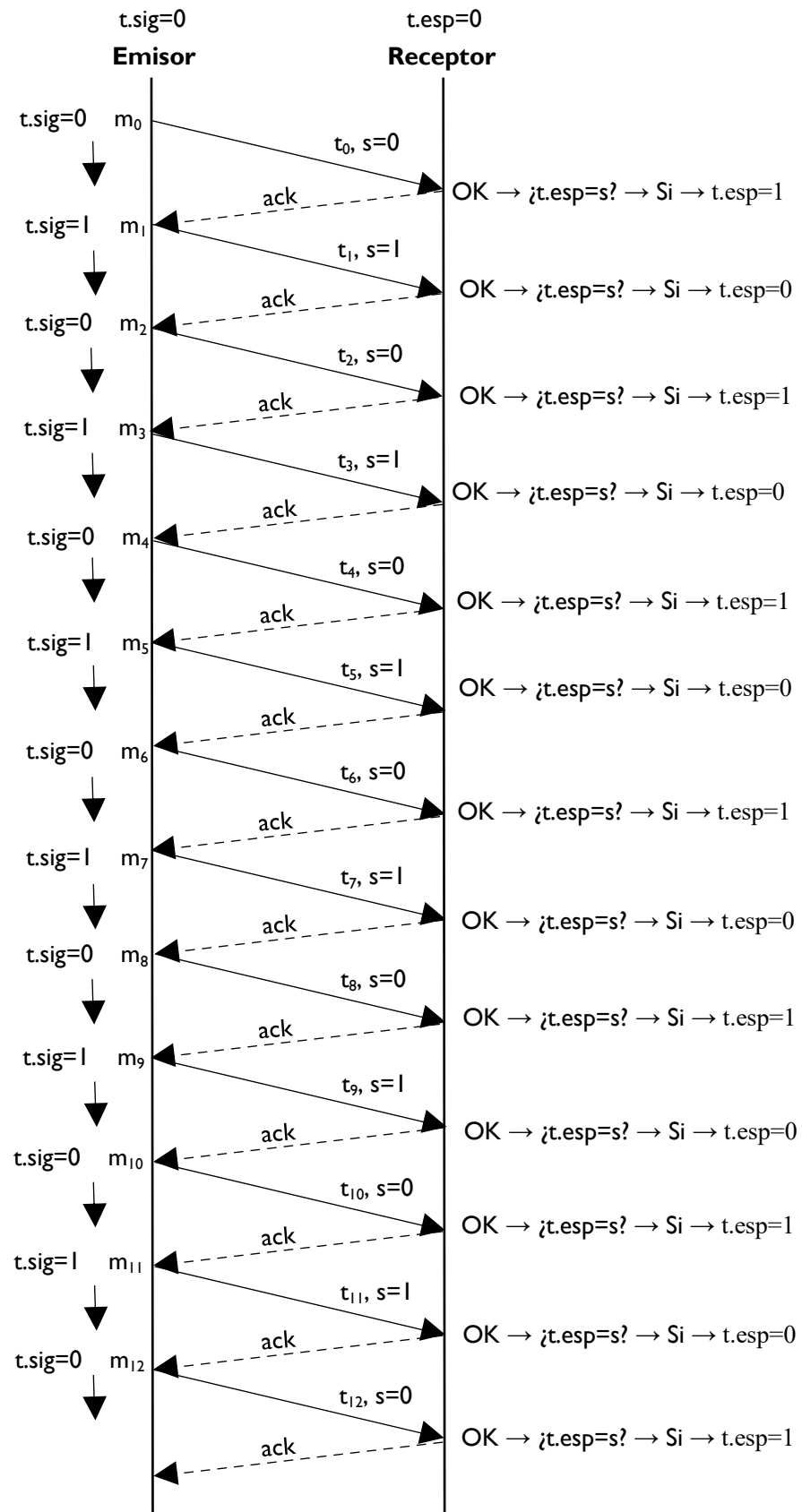
SIMULADOR DEL NIVEL DE ENLACE



LUIS DÍAZ

1.- Ejecutar el protocolo 2 con la siguiente orden ./protocol2 100 10 0 0 0. Dibujar el resultado obtenido sobre la gráfica siguiente: 0

t = trama
t.sig = trama siguiente
t.esp = trama esperada
m = mensaje
s = secuencia
ack = t. confirmación
↓ = temporizador



2.- Ejecutar el protocolo 3 con diferentes tasas de pérdida de paquetes, diferentes tasas de errores en trama. ¿Cómo afectan el incremento de la pérdida de paquetes a la eficiencia del protocolo 3? ¿Cómo afecta el incremento de la tasa de errores en tramas a la eficiencia del protocolo 3?

```
./protocol3 events timeout pct_loss pct_chsum flags
```

- events → Indica el tiempo que durará la simulación
- timeout → Indica el tiempo que se considerará como timeout
- pct_loss → Indica el porcentaje de tramas que se pierden (0 – 99)
- pct_sum → Indica el porcentaje de tramas que llegan con error (0 – 99)
- flags → Opciones de visualización de resultados

* NOTA: Ejecutar los protocolos 2 y 3 con flag = 0

Ejecución	events	timeout	pct_loss	pct_chsum	flags	Eficiencia
1	1000	100	0	0	0	99%
2	1000	100	10	0	0	85%
3	1000	100	20	0	0	66%
4	1000	100	30	0	0	18%
5	1000	100	40	0	0	18%
6	1000	100	0	10	0	72%
7	1000	100	0	20	0	50%
8	1000	100	0	30	0	41%
9	1000	100	0	40	0	27%
10	1000	100	0	50	0	6%
11	1000	100	10	10	0	60%
12	1000	100	20	20	0	25%
13	1000	100	30	30	0	9%
14	1000	100	40	40	0	10%

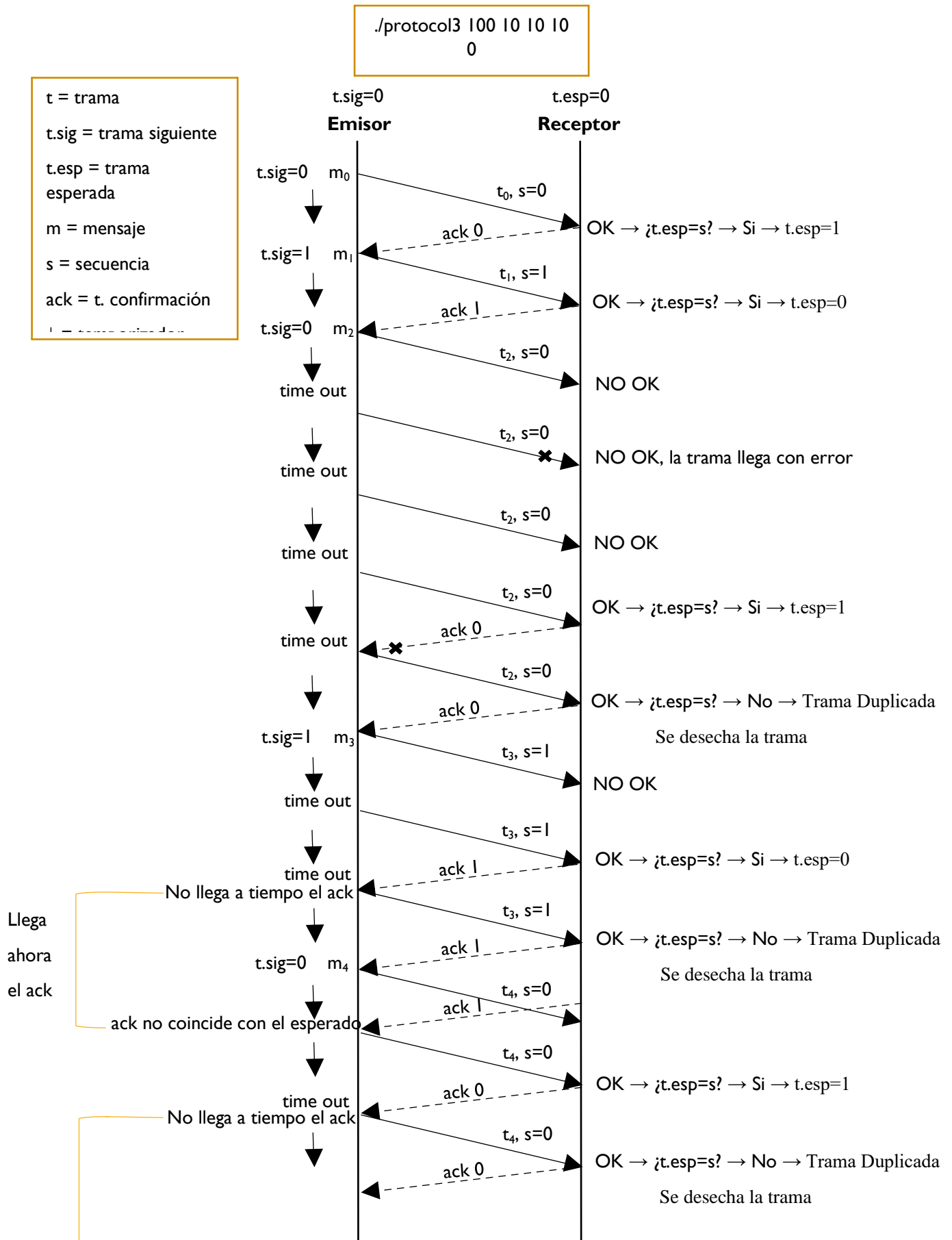
Todas las ejecuciones las he realizado con un tiempo de simulación y un time out iguales cambiando el porcentaje de tramas que se pierden y el porcentaje de tramas que llegan con error para estudiar como varia la eficiencia del protocolo 3.

Se puede observar, como era de esperar, cómo en las ejecuciones 2, 3, 4, y 5, en las que sólo se añade porcentaje de tramas que se pierden, la eficiencia baja conforme sube el porcentaje. Esto también ocurre en las ejecuciones 6, 7, 8, 9 y 10 cuando sólo vario el porcentaje de tramas que llegan con error, pero esta vez, la eficiencia baja considerablemente más que cuando sólo se varia el porcentaje de tramas que se pierden.

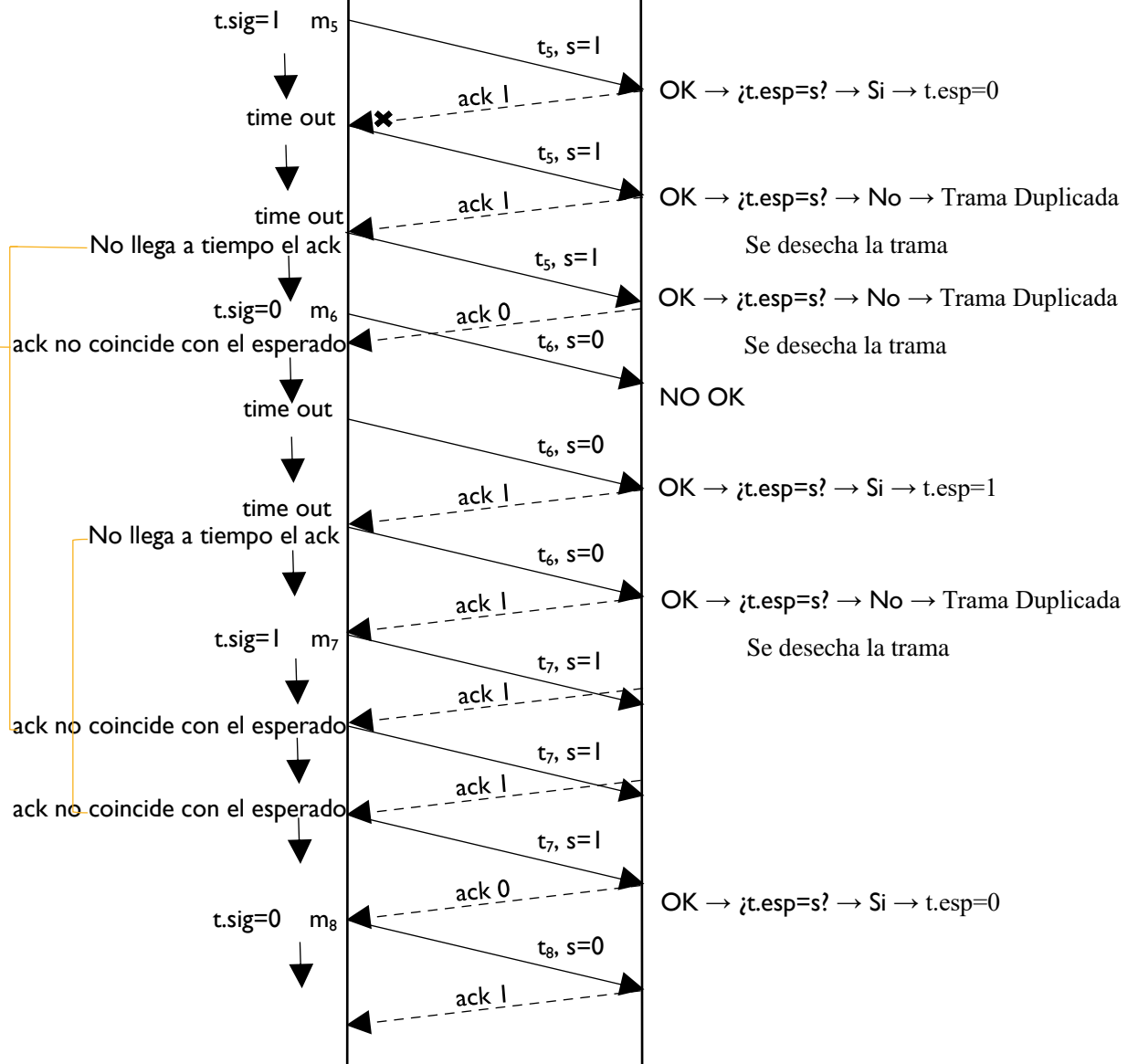
En las ejecuciones 11, 12, 13 y 14 añadido porcentajes tanto de tramas que se pierden como de tramas que llegan con error, y de nuevo, como era de esperar, la eficiencia baja más que en los casos anteriores.

Hay que destacar que a partir del 30% de tramas que se pierden, la eficiencia no baja de 18% hasta que llega a 0% (cuando se supera el 90% de tramas perdidas); esto también pasa a partir del 50% de tramas que llegan con error, en las que la eficiencia no baja del 6% hasta que llega a 0%. Resalto esto, porque, aunque en las ejecuciones 11, 12 y 13 las eficiencias siempre son menores que cuando sólo se añaden un porcentaje de tramas que se pierden o con llegada con error, pero, cuando hablamos del mínimo de eficiencia antes de llegar a al 0% en la trama 15, con un 10%, esta es un porcentaje intermedio entre los 18% de las tramas que se pierden y el 6% de las tramas que llegan con error.

3.- Ejecutar el protocolo 3 con la siguiente orden: `./protocol3 100 10 10 10 0`. Dibujar en la gráfica inferior el resultado obtenido. Interpretar y comentar cada una de las situaciones que se producen en la comunicación y sus causas (errores, timeouts, retransmisiones, etc.).



Llega
ahora
el ack



```
root@debian:/home/luisdiaz/Documents/ProtocolosNivelDeEnlace/simulador# ./protocol3 100 10 10 10 0
```

Simulating Protocol 3

Events: 100 Parameters: 10 10 10
Trama siguiente: 0

Trama esperada: 0

DATOS seq: 0 paquete: 0 ---->

<---- ACK seq: 0 paquete: 0
Trama esperada: 1

Trama siguiente: 1

DATOS seq: 1 paquete: 1 ---->

<---- ACK seq: 1 paquete: 1
Trama esperada: 0

Trama siguiente: 0

DATOS seq: 0 paquete: 2 ---->
TIMEOUT

DATOS seq: 0 paquete: 2 ---->

CKSUM_ERROR

TIMEOUT

DATOS seq: 0 paquete: 2 ---->
TIMEOUT

DATOS seq: 0 paquete: 2 ---->

<---- ACK seq: 0 paquete: 2
Trama esperada: 1

TIMEOUT

DATOS seq: 0 paquete: 2 ---->

SEQ no esperado
<---- ACK seq: 0 paquete: 2
Trama esperada: 1

Trama siguiente: 1

DATOS seq: 1 paquete: 3 ---->
TIMEOUT

DATOS seq: 1 paquete: 3 ---->

<---- ACK seq: 1 paquete: 3
Trama esperada: 0
SEQ no esperado
<---- ACK seq: 1 paquete: 3
Trama esperada: 0

Trama siguiente: 0

DATOS seq: 0 paquete: 4 ---->
ACK no coincide con el esperado

DATOS seq: 0 paquete: 4 ---->

<---- ACK seq: 0 paquete: 4
Trama esperada: 1
SEQ no esperado
<---- ACK seq: 0 paquete: 4
Trama esperada: 1

Trama siguiente: 1

DATOS seq: 1 paquete: 5 ---->

<---- ACK seq: 1 paquete: 5
Trama esperada: 0

CKSUM_ERROR

DATOS seq: 1 paquete: 5 ---->

SEQ no esperado
<---- ACK seq: 1 paquete: 5
Trama esperada: 0

TIMEOUT

DATOS seq: 1 paquete: 5 ---->
Trama siguiente: 0

DATOS seq: 0 paquete: 6 ---->

SEQ no esperado
<---- ACK seq: 1 paquete: 5
Trama esperada: 0

ACK no coincide con el esperado

DATOS seq: 0 paquete: 6 ---->
TIMEOUT

DATOS seq: 0 paquete: 6 ---->

<---- ACK seq: 0 paquete: 6
Trama esperada: 1
SEQ no esperado

ACK no coincide con el esperado

DATOS seq: 0 paquete: 6 ---->
TIMEOUT

DATOS seq: 0 paquete: 6 ---->

<---- ACK seq: 0 paquete: 6
Trama esperada: 1
SEQ no esperado
<---- ACK seq: 0 paquete: 6
Trama esperada: 1

Trama siguiente: 1

DATOS seq: 1 paquete: 7 ---->

SEQ no esperado
<---- ACK seq: 0 paquete: 6
Trama esperada: 1

ACK no coincide con el esperado

DATOS seq: 1 paquete: 7 ---->
ACK no coincide con el esperado

DATOS seq: 1 paquete: 7 ---->

<---- ACK seq: 1 paquete: 7
Trama esperada: 0

Trama siguiente: 0

DATOS seq: 0 paquete: 8 ---->

SEQ no esperado
<---- ACK seq: 1 paquete: 7
Trama esperada: 0

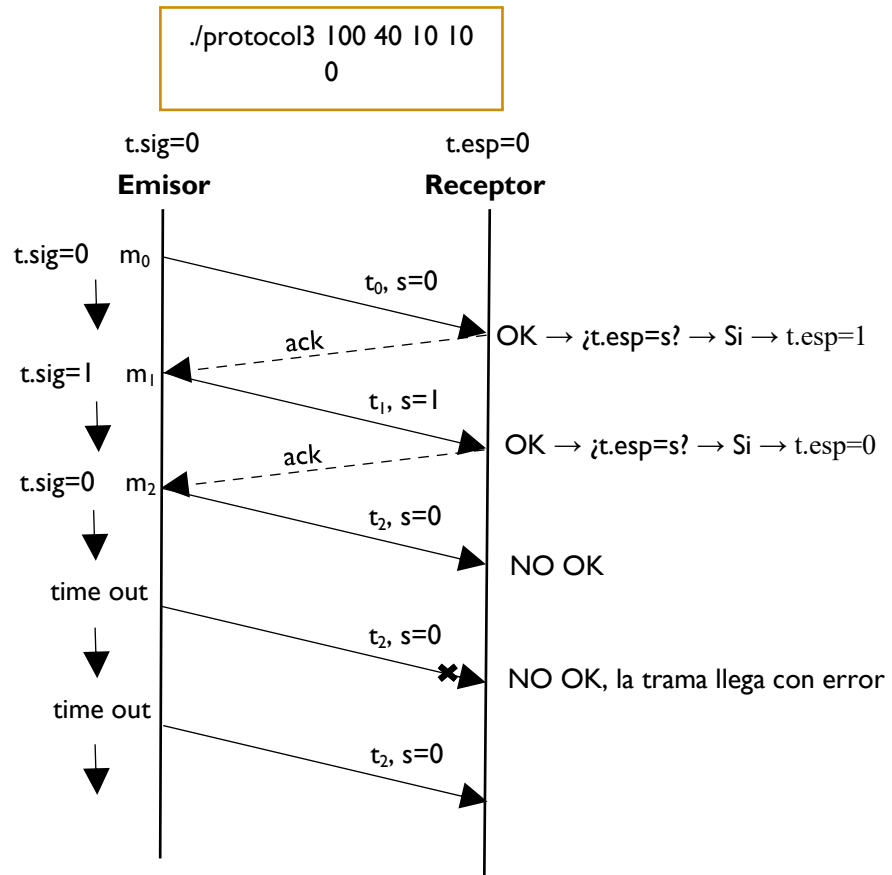
Process 0:

Process 1:

Total data frames sent:	21
Total data frames sent:	0
Data frames lost:	2
Data frames lost:	0
Data frames not lost:	19
Data frames not lost:	0
Frames retransmitted:	7
Frames retransmitted:	0
Good ack frames rec'd:	12
Good ack frames rec'd:	0
Bad ack frames rec'd:	1
Bad ack frames rec'd:	0
Good data frames rec'd:	0
Good data frames rec'd:	16
Bad data frames rec'd:	0
Bad data frames rec'd:	1
Payloads accepted:	0
Payloads accepted:	8
Total ack frames sent:	0
Total ack frames sent:	16
Ack frames lost:	0
Ack frames lost:	2
Ack frames not lost:	0
Ack frames not lost:	14
Timeouts:	7
Timeouts:	0
Ack timeouts:	0
Ack timeouts:	0

Efficiency (payloads accepted/data pkts sent) = 38%
End of simulation. Time=100

4.- Ejecutar el protocolo 3 con la siguiente orden: `./protocol3 100 40 10 10 0`. Dibujar en una gráfica como la anterior el resultado obtenido. ¿cuál es la diferencia con el apartado 3?.



```
root@debian:/home/luisdiaz/Documents/ProtocolosNivelDeEnlace/simulador# ./protocol3 100 40 10 10 0
```

Simulating Protocol 3

Events: 100 Parameters: 40 10 10
Trama siguiente: 0

Trama esperada: 0

DATOS seq: 0 paquete: 0 ---->

<---- ACK seq: 0 paquete: 0
Trama esperada: 1

Trama siguiente: 1

DATOS seq: 1 paquete: 1 ---->

<---- ACK seq: 1 paquete: 1
Trama esperada: 0

Trama siguiente: 0

DATOS seq: 0 paquete: 2 ---->
TIMEOUT

DATOS seq: 0 paquete: 2 ---->

CKSUM_ERROR

TIMEOUT

DATOS seq: 0 paquete: 2 ---->


```
Process 0:
  Total data frames sent:      5
  Data frames lost:           2
  Data frames not lost:       3

Process 1:
  Frames retransmitted:        2
  Total data frames sent:      0
  Good ack frames rec'd:      2
  Data frames lost:           0
  Bad ack frames rec'd:        0

  Data frames not lost:       0
  Good data frames rec'd:      0
  Frames retransmitted:        0
  Bad data frames rec'd:       0
  Good ack frames rec'd:       0
  Payloads accepted:           0
  Bad ack frames rec'd:        0

  Total ack frames sent:       0
  Good data frames rec'd:      2
  Ack frames lost:             0
  Bad data frames rec'd:       1
  Ack frames not lost:         0
  Payloads accepted:           2
  Timeouts:                    2
  Total ack frames sent:       2
  Ack timeouts:                0
  Ack frames lost:             0
  Ack frames not lost:         2
  Timeouts:                    0
  Ack timeouts:                0

Efficiency (payloads accepted/data pkts sent) = 40%
End of simulation. Time=100
```

En esta ejecución del protocolo 3 se aumenta el Time Out de 10 a 40 respecto al ejercicio anterior y se puede observar como ahora los mensajes de confirmación ack llegan de forma correcta al contrario que en el ejercicio 3 donde a veces no daba tiempo a que llegase el ack y saltaba el Time Out. Esto hacía que no hubiese una sincronización entre emisor y receptor y terminaban llegando fuera de tiempo sin saber cuál se confirmaba. Esto hace que la eficiencia haya aumentado de 38% a 40%.

5.- Ejecutar el protocolo 4. Indicar si aporta ventajas con respecto al protocolo 3. En términos de la eficiencia utilizada por el simulador, ¿resulta más adecuado utilizar el protocolo 4 o el protocolo 3 para un enlace de red?

```
./protocol4 events timeout pct_loss pct_chsum flags
```

- events → Indica el tiempo que durará la simulación
- timeout → Indica el tiempo que se considerará como timeout
- pct_loss → Indica el porcentaje de tramas que se pierden (0 – 99)
- pct_sum → Indica el porcentaje de tramas que llegan con error (0 – 99)
- flags → Opciones de visualización de resultados

* NOTA: Ejecutar los protocolos 2 y 3 con flag = 0

Ejecución	events	timeout	pct_loss	pct_chsum	flags	Eficiencia
1	1000	100	0	0	0	49%
2	1000	100	10	0	0	60%
3	1000	100	20	0	0	69%
4	1000	100	30	0	0	59%
5	1000	100	40	0	0	22%
6	1000	100	0	10	0	45%
7	1000	100	0	20	0	39%
8	1000	100	0	30	0	35%
9	1000	100	0	40	0	30%
10	1000	100	0	50	0	26%
11	1000	100	10	10	0	47%
12	1000	100	20	20	0	33%
13	1000	100	30	30	0	36%
14	1000	100	40	40	0	24%

En términos de eficiencia el protocolo 3 es más adecuado para un enlace de red. Cuando el porcentaje de tramas que se pierden y llegan con error es 0, el protocolo 3 tiene una eficiencia del **99%** y el protocolo 4 del **49%**, pero esto no es todo; en el protocolo 3 cuanto más se sube el porcentaje de tramas que se pierden o que llegadas con error, la eficiencia baja de forma de gradual, lo que facilita la estabilidad entre la conexión Emisor-Receptor, en cambio, en el protocolo 4 con un mayor porcentaje de tramas que se pierden o que llegan con error, puede tener una eficiencia mayor que con un porcentaje menor.

También se observa en el protocolo 4 que a mayor porcentaje de tramas que se pierden o que llegan con error, mayor eficiencia tiene. Esto no es productivo porque la eficiencia resultante es mayor que en protocolo 3 cuantas más tramas se pierdan o lleguen con error.

6.- Ejecutar los protocolos de ventana deslizante (protocolo5 y protocolo6) y realizar una comparación entre ellos. Para ello probar la ejecución de los 2 protocolos bajo las mismas circunstancias e ir variando los parámetros de entrada. ¿Bajo qué condiciones resulta más adecuado utilizar el protocolo5? ¿y el protocolo6?

- Protocolo 5: `./protocol5 events timeout pct_loss pct_chsum flags`

Ejecución	events	timeout	pct_loss	pct_chsum	flags	Eficiencia
1	500	50	0	0	0	97%
2	500	50	10	0	0	56%
3	500	50	20	0	0	21%
4	500	50	30	0	0	17%
5	500	50	40	0	0	14%
6	500	50	0	10	0	58%
7	500	50	0	20	0	34%
8	500	50	0	30	0	30%
9	500	50	0	40	0	19%
10	500	50	10	10	0	37%
11	500	50	20	20	0	19%
12	500	50	30	30	0	12%
13	500	50	40	40	0	5%

- Protocolo 6: `./protocol6 events timeout pct_loss pct_chsum flags`

Ejecución	events	timeout	pct_loss	pct_chsum	flags	Eficiencia
1	500	50	0	0	0	98%
2	500	50	10	0	0	82%
3	500	50	20	0	0	63%
4	500	50	30	0	0	39%
5	500	50	40	0	0	27%
6	500	50	0	10	0	83%
7	500	50	0	20	0	68%
8	500	50	0	30	0	41%
9	500	50	0	40	0	45%
10	500	50	10	10	0	61%
11	500	50	20	20	0	42%
12	500	50	30	30	0	29%
13	500	50	40	40	0	9%

El protocolo 6 es más eficiente en cada prueba de ejecución que el protocolo 5, esto se debe que el protocolo 6 usa la técnica de rechazo selectivo (RS), disponiendo de un buffer de memoria y pudiendo almacenar tramas correctas y que lleguen con error. El protocolo 5, en cambio, usa la técnica de rechazo no selectivo(NRS), en la que no hay un buffer de memoria y si ocurre un error en una trama, el emisor tendrá que retransmitir esa trama y todas las tramas siguientes que hubiese enviado.

Respondiendo a la pregunta que bajo qué condiciones sería más adecuado utilizar el protocolo 5, sería cuando el receptor no tenga memoria para almacenar las tramas correctas después de una con error.