



Actividades UD 9. Algoritmos distribuidos



Concurrencia y Sistemas Distribuidos

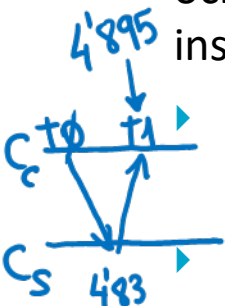


Actividad 1

- ▶ En un determinado sistema se utiliza el algoritmo de Cristian para sincronizar los relojes de los ordenadores. En un determinado momento uno de los nodos clientes envía el mensaje de petición al servidor y cuando éste lo recibe su reloj local marcaba 11:01:04,830 (es decir, las once, un minuto y cuatro segundos con ochocientos treinta milisegundos). El cliente envió su mensaje de petición en el instante 11:01:04,750 y recibió la respuesta en el instante 11:01:04,880.
 - ▶ ¿A qué valor establecerá su reloj local?
 - ▶ ¿Se tendrá que suspender el avance del reloj local del cliente durante algún tiempo? ¿Por qué?
 - ▶ Si además de lo dicho anteriormente se supiera que el mensaje de petición necesitó **50ms** en ser transmitido y el de respuesta **80 ms**, ¿se podría afirmar la sincronización efectuada por este algoritmo fue precisa? ¿Por qué?

Actividad 1

- En un determinado sistema se utiliza el algoritmo de Cristian para sincronizar los relojes de los ordenadores. En un determinado momento uno de los nodos clientes envía el mensaje de petición al servidor y cuando éste lo recibe su reloj local marcaba 11:01:04,830 (es decir, las once, un minuto y cuatro segundos con ochocientos treinta milisegundos). El cliente envió su mensaje de petición en el instante 11:01:04,750 y recibió la respuesta en el instante 11:01:04,880.

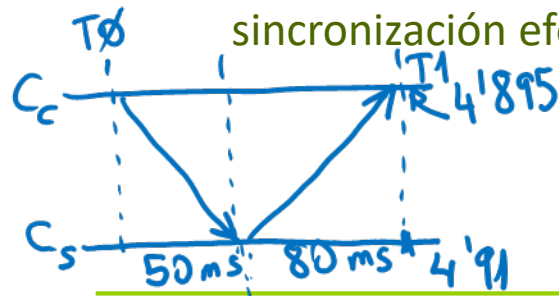


¿A qué valor establecerá su reloj local?

$$C = C_s + (T_1 - T_0) / 2 = 4'83 + (4'88 - 4'75) / 2 = 4'83 + 0'065 = 4'895$$

¿Se tendrá que suspender el avance del reloj local del cliente durante algún tiempo? ¿Por qué? No. Porque $C > C_c$ en T_1 $4'895 - 4'880 = 0'015$. Hay que avanzar el reloj del cliente en 15 ms.

- Si además de lo dicho anteriormente se supiera que el mensaje de petición necesitó 50ms en ser transmitido y el de respuesta 80 ms, ¿se podría afirmar la sincronización efectuada por este algoritmo fue precisa? ¿Por qué? No. Se asume



tiempos de transmisión de mensaje iguales. Como $(80 + 50) / 2 = 0'065$ C_c se ajustaría a 4'895, pero en C_s sería $4'83 + 0'08 = 4'91$ es decir 15ms. más. Lo hemos ajustado 15 ms. menos de lo que tocaba.



Actividad 2

- ▶ En un sistema distribuido con 4 nodos se emplea el algoritmo de Berkeley para sincronizar sus relojes. Uno de los nodos (nodo A) actúa como servidor del algoritmo, mientras que los otros nodos (nodos B, C y D) actúan como clientes. Supongamos que cada nodo tiene un reloj que indica el número de tics transcurridos desde la misma base temporal; y que en un momento determinado tienen el siguiente valor para sus relojes: $C_A=10000$, $C_B=10005$, $C_C=10005$ y $C_D=10010$ (siendo C_n el reloj de cada nodo n). En dicho momento el servidor (nodo A) inicia el algoritmo de Berkeley.
- ▶ Si asumimos que han transcurrido 30 tics de reloj durante todos esos pasos, ¿cuál será el valor final del reloj de cada nodo?

Actividad 2

- Si asumimos que han transcurrido 30 ticks de reloj durante todos esos pasos, ¿cuál será el valor final del reloj de cada nodo?

- En **b, c y d** se calculan las diferencias respecto a 10.000 y se envían a A
 - El servidor A ajusta cada diferencia notificada

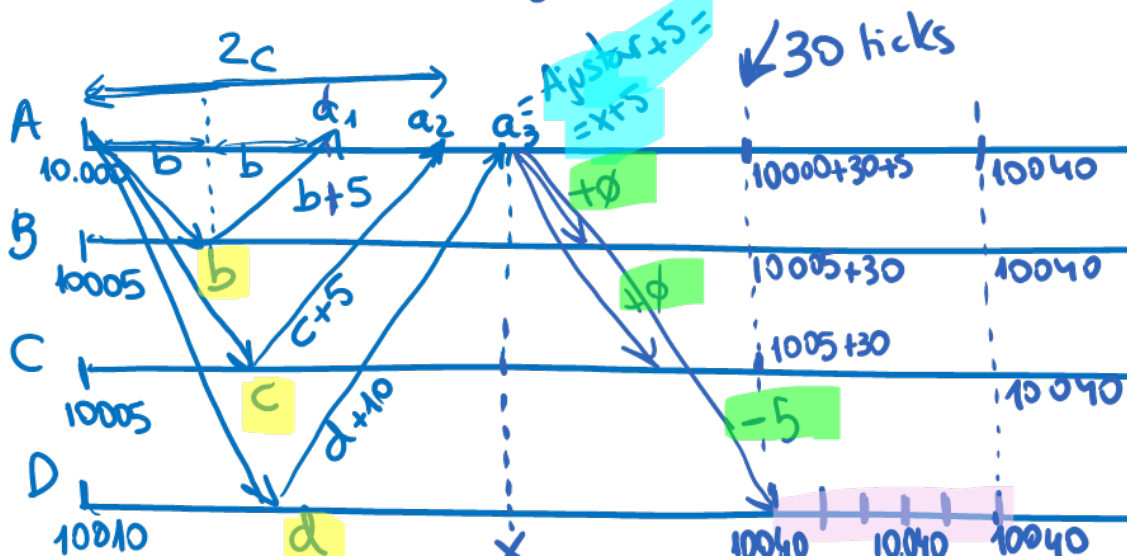
- en a_1 : $(b+5) - 2b/2 = +5 = D'_B$

- en a_2 : $(c+5) - 2c/2 = +5 = D'_C$

- en a_3 : $(d+10) - 2d/2 = +10 = D'_D$

Promedio incluyendo a A
 $D = \frac{(0+5+5+10)}{4} = 5$

El servidor ajusta su reloj incrementándolo en 5
 en a_3 valdrá $x+5$



Se notifica el ajuste: $A_B = D - D'_B = 5 - 5 = \phi$
 $A_C = D - D'_C = 5 - 5 = \phi$
 $A_D = D - D'_D = 5 - 10 = -5$

Parar el reloj durante 5 unidades de tiempo

Se sincronizarán los relojes en 10,040



Actividad 4

- | | |
|---|--|
| 1. En el algoritmo de Cristian, si un cliente C pregunta al servidor su hora en el instante 20.000 (según el reloj de C), recibe la respuesta del servidor en el instante 20.010 (según el reloj de C) y calcula que el nuevo valor para su reloj debe ser 20.024, entonces se puede deducir que la respuesta del servidor habrá sido 14. | |
| 2. En el algoritmo de Berkeley, el nodo que actúa como coordinador nunca tendrá que ajustar su reloj. Solamente ajustan su reloj los otros nodos que participan en el algoritmo. | |
| 3. El algoritmo de Berkeley asume que el envío de un mensaje desde el nodo A al nodo B consume el doble de tiempo que la respuesta desde B hasta A. | |
| 4. En el algoritmo de Cristian, para fijar la hora del cliente se calcula el promedio entre el reloj del cliente y el del servidor. | |
| 5. En el algoritmo de Berkeley, cada cliente puede sincronizar su reloj en cualquier momento, con independencia de cuándo sincronizan su reloj otros clientes. | |



Actividad 4

<p>1. En el algoritmo de Cristian, si un cliente C pregunta al servidor su hora en el instante 20.000 (según el reloj de C), recibe la respuesta del servidor en el instante 20.010 (según el reloj de C) y calcula que el nuevo valor para su reloj debe ser 20.024, entonces se puede deducir que la respuesta del servidor habrá sido 14.</p> <p><i>C_c 20000 20010</i> <i>C_s 20024</i> <i>El servidor responde el valor de su reloj = $20024 - \frac{(20010 - 20000)}{2} = 20019$</i></p>	F
<p>2. En el algoritmo de Berkeley, el nodo que actúa como coordinador nunca tendrá que ajustar su reloj. Solamente ajustan su reloj los otros nodos que participan en el algoritmo.</p> <p><i>al igual que</i> <i>puede que tenga</i></p>	F
<p>3. El algoritmo de Berkeley asume que el envío de un mensaje desde el nodo A al nodo B consume el doble de tiempo que la respuesta desde B hasta A.</p> <p><i>mismo</i></p>	F
<p>4. En el algoritmo de Cristian, para fijar la hora del cliente se calcula el promedio entre el reloj del cliente y el del servidor.</p>	F
<p>5. En el algoritmo de <u>Berkeley</u> <i>Cristian</i> cada cliente puede sincronizar su reloj en cualquier momento, con independencia de cuándo sincronizan su reloj otros clientes.</p>	F



Actividad 4

6. Suponga que seis nodos utilizan el algoritmo de Berkeley para sincronizar sus relojes. El nodo N1 inicia el algoritmo cuando los valores de los relojes de los seis nodos son respectivamente N1=2002, N2=2008, N3=2005, N4=2010, N5=2000 y N6=2005 y recibe las respuestas de los otros nodos al mismo tiempo. Dados estos supuestos, el nodo N1 determina que debe adelantar su propio reloj en 30 unidades de tiempo.	F
7. En el algoritmo de Berkeley, si asumimos un total de N nodos, se requiere el envío de $3 \cdot N$ mensajes en total.	F
8. El algoritmo de Cristian requiere conocer a priori el retardo medio de los mensajes que intercambian los distintos nodos.	F
9. En el algoritmo de Cristian, si un cliente C pregunta al servidor su hora en el instante 10.000 y recibe como respuesta del servidor el valor 10.006 en el instante 10.008, entonces deberá detener su reloj un total de 2 unidades de tiempo.	F

$$3 = (2 + 6 + 3 + 8 + (-2) + 3) / 6$$

$$\underline{3 \cdot (N-1)}$$

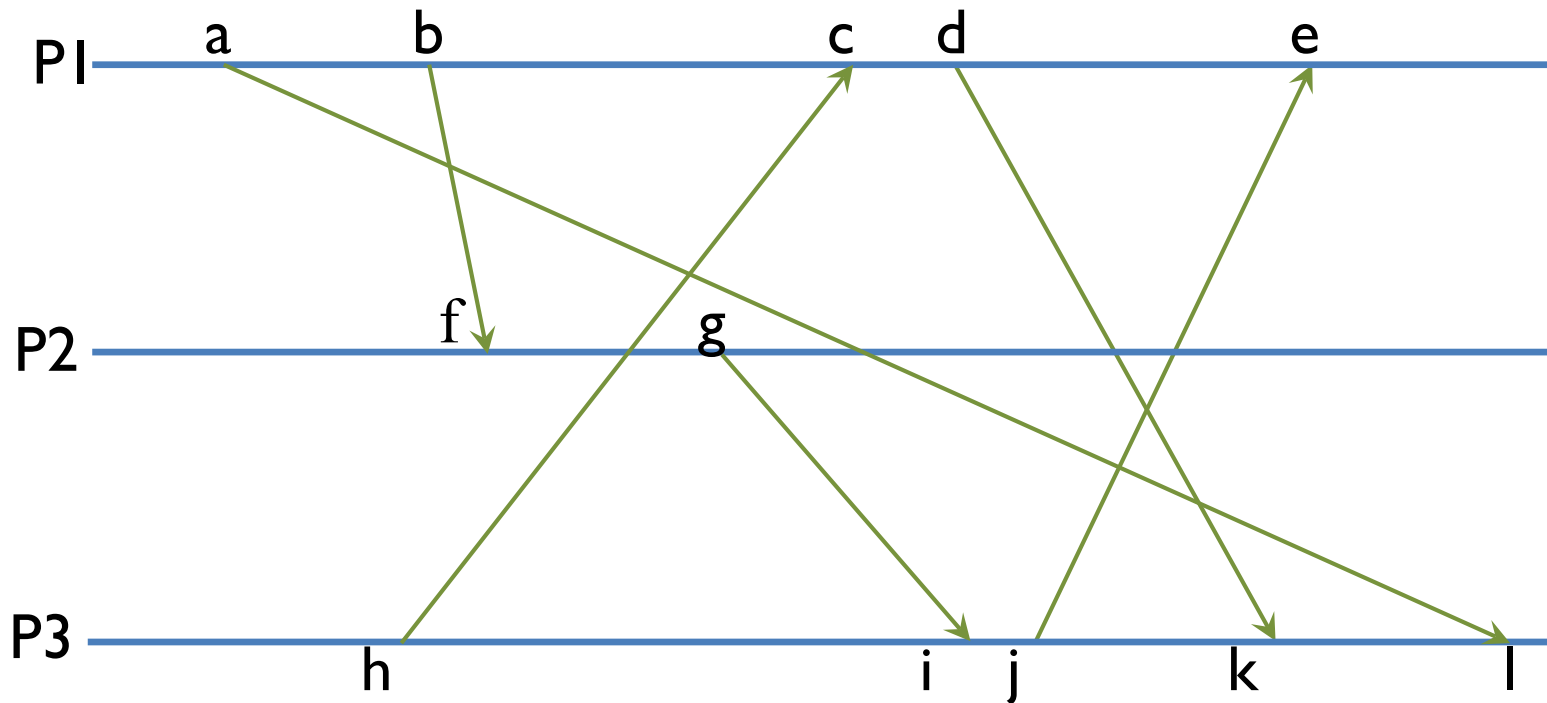
↓ incluyendo al coordinador

Hay que ajustarlo a $10.006 + 4 = 10.010 \Rightarrow$

Hay que adelantarlo no pararlo

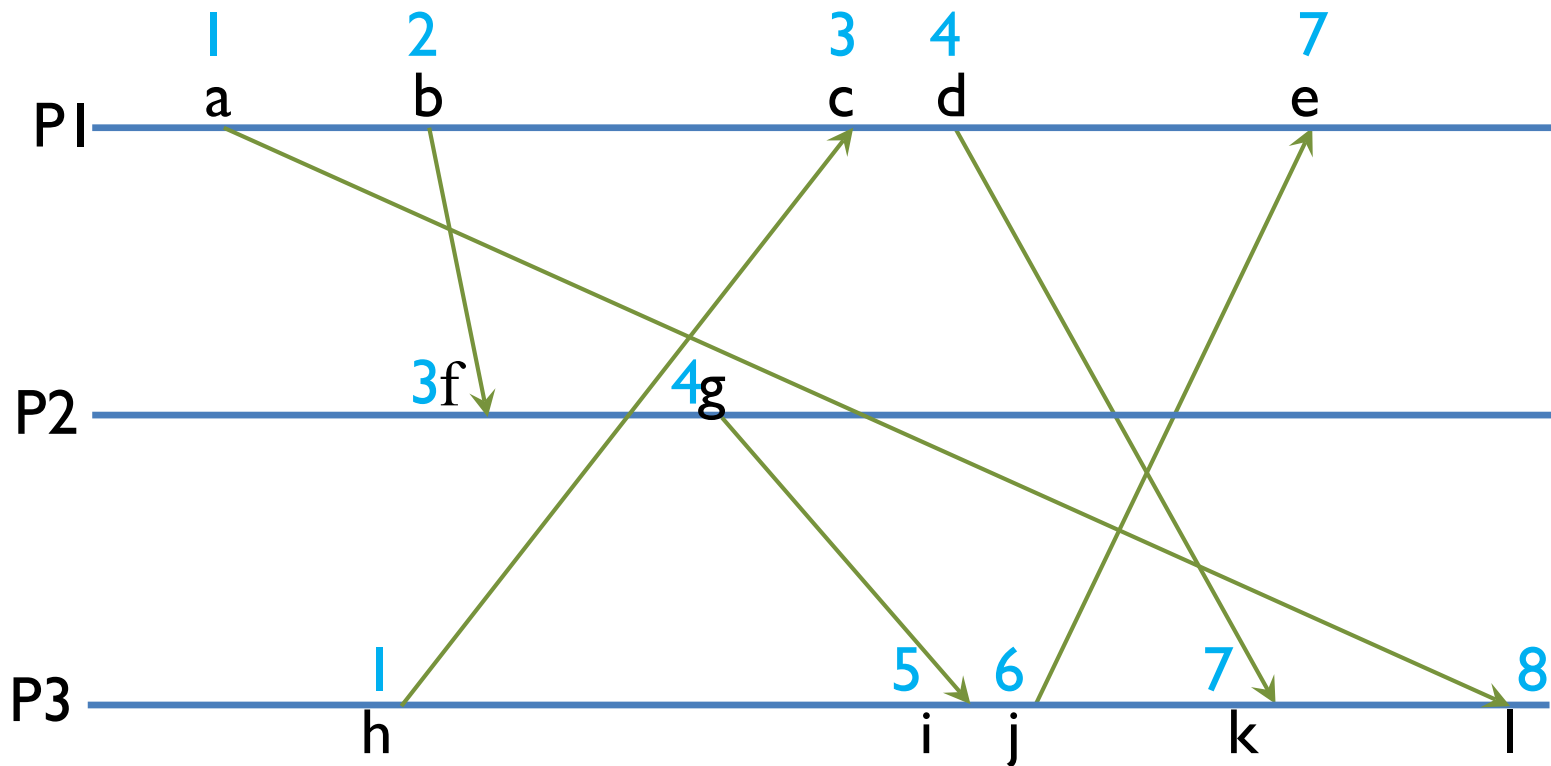
Actividad 5

- Indique para cada uno de los eventos el valor que asignaría el algoritmo de Lamport a tales eventos



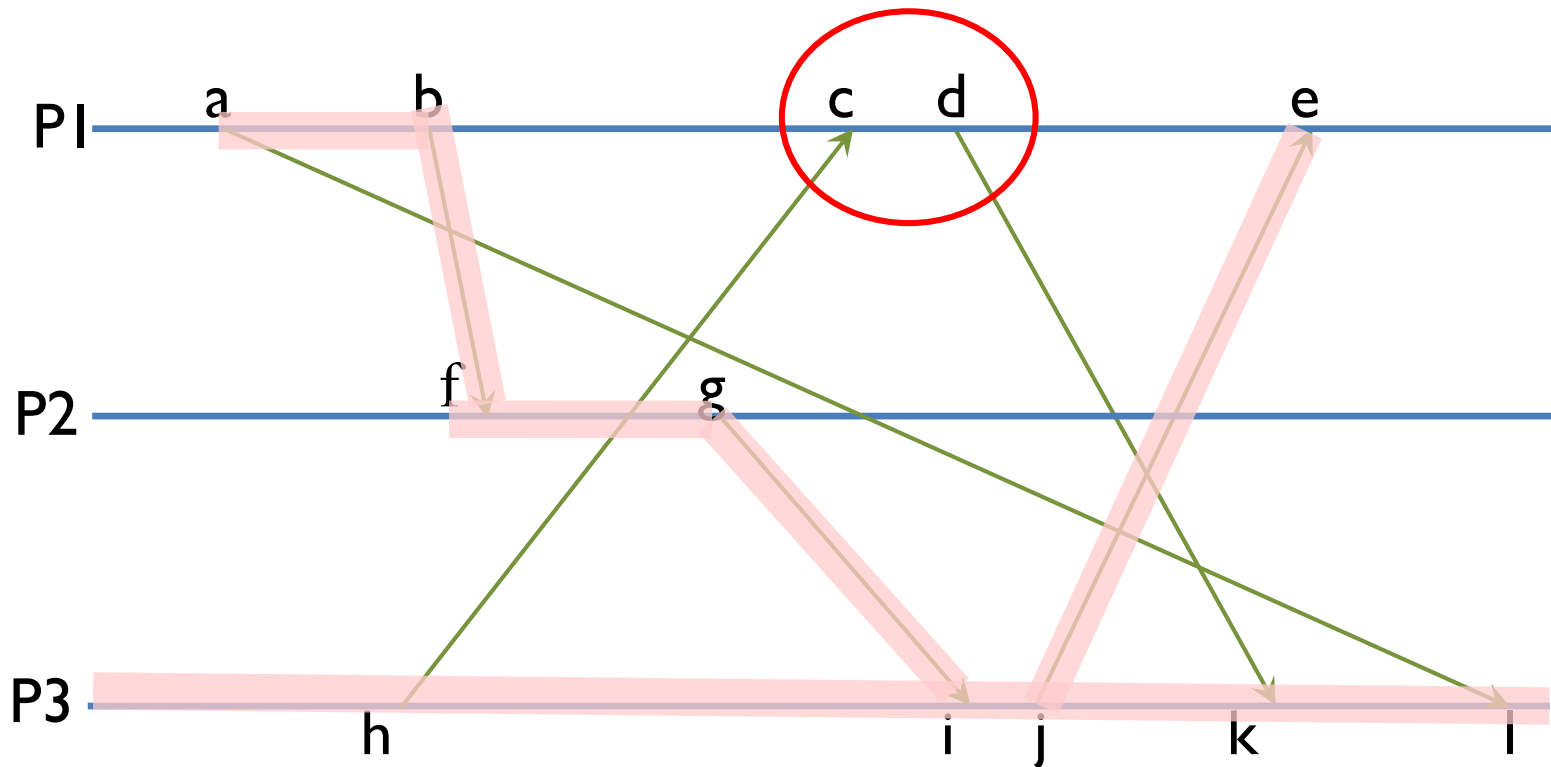
Actividad 5

- Indique para cada uno de los eventos el valor que asignaría el algoritmo de Lamport a tales eventos



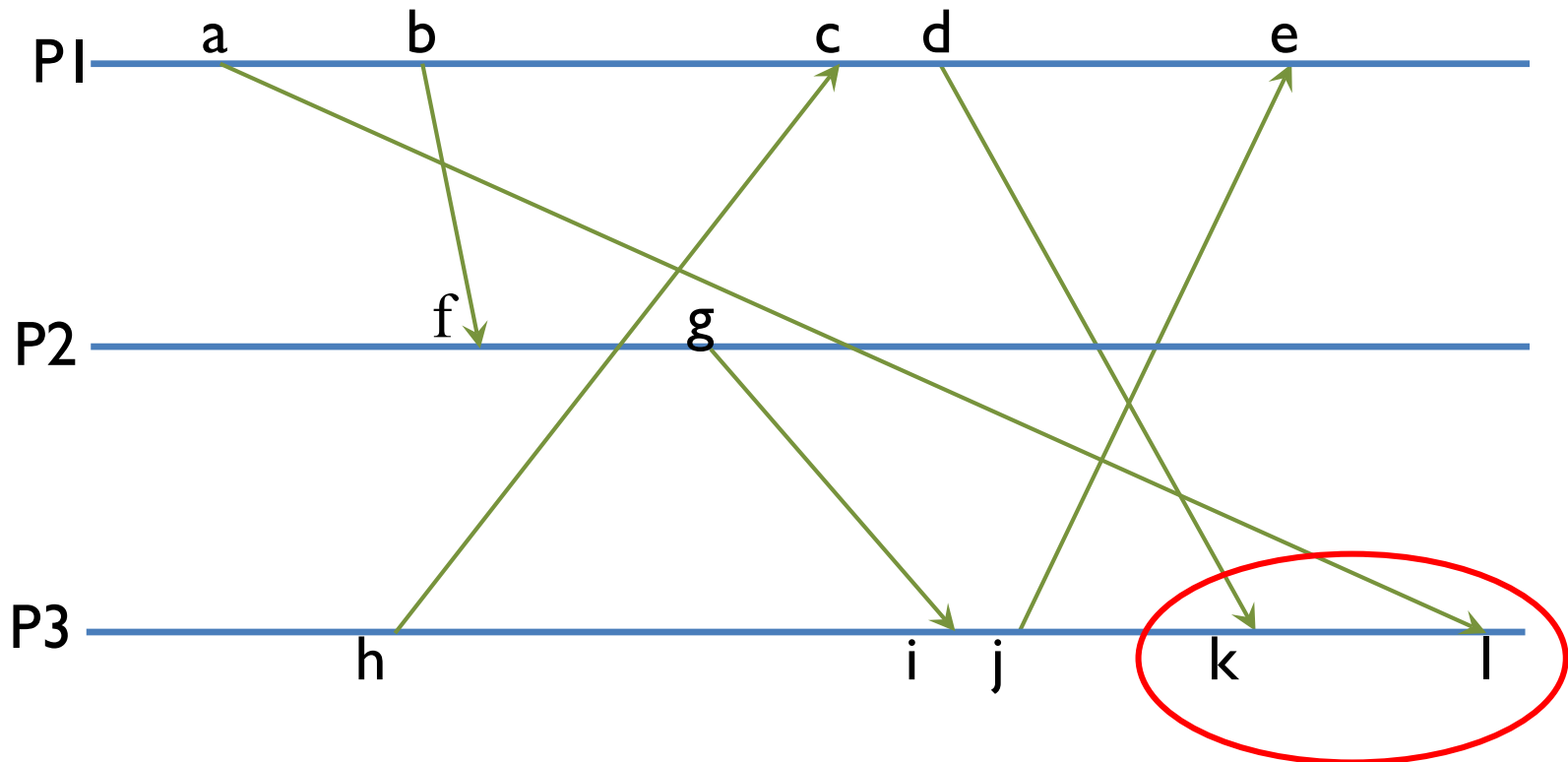
Actividad 5

- Indique qué eventos serán concurrentes con el evento “i”



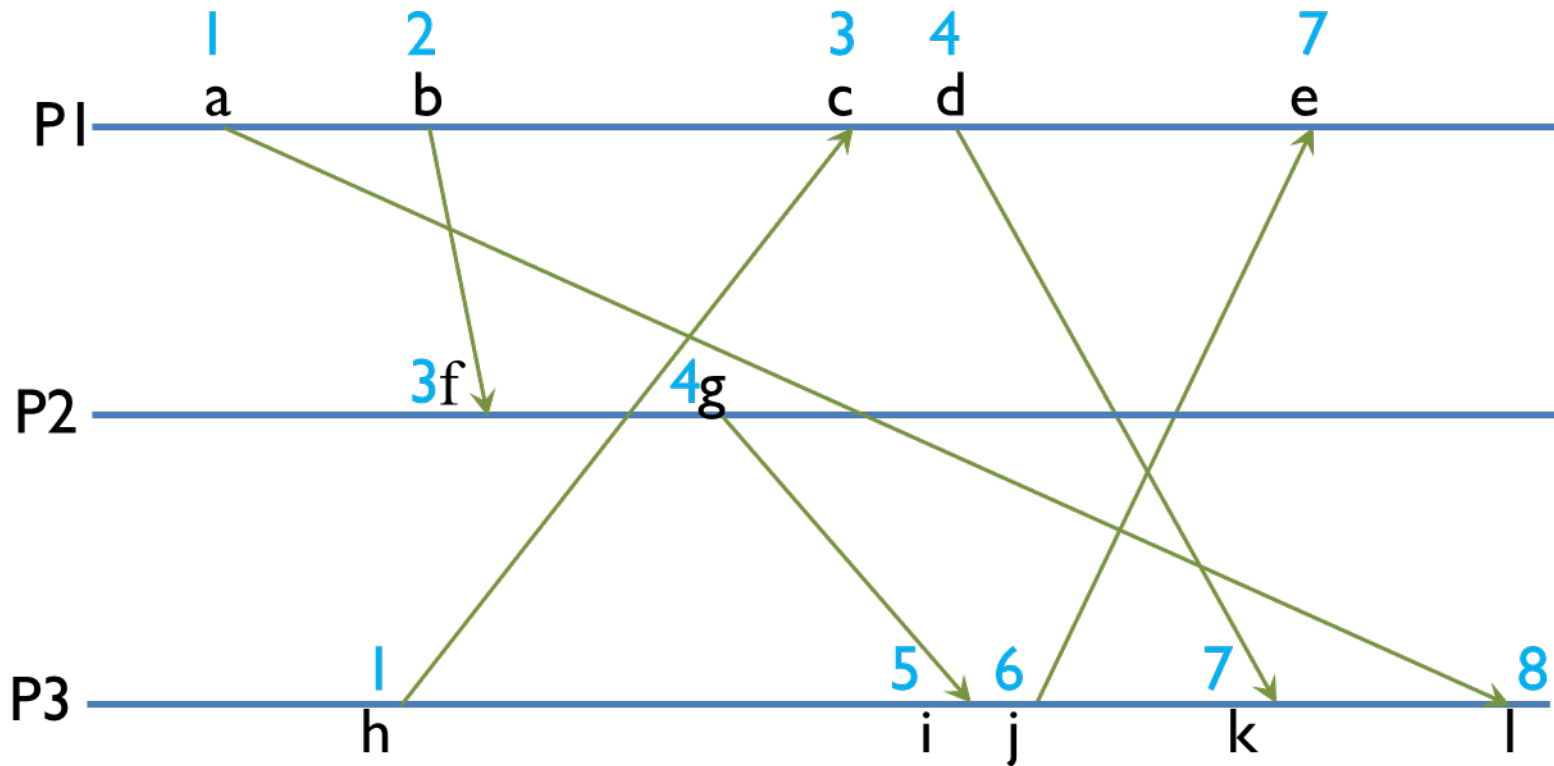
Actividad 5

- Indique qué eventos serán concurrentes con el evento “e”



Actividad 5

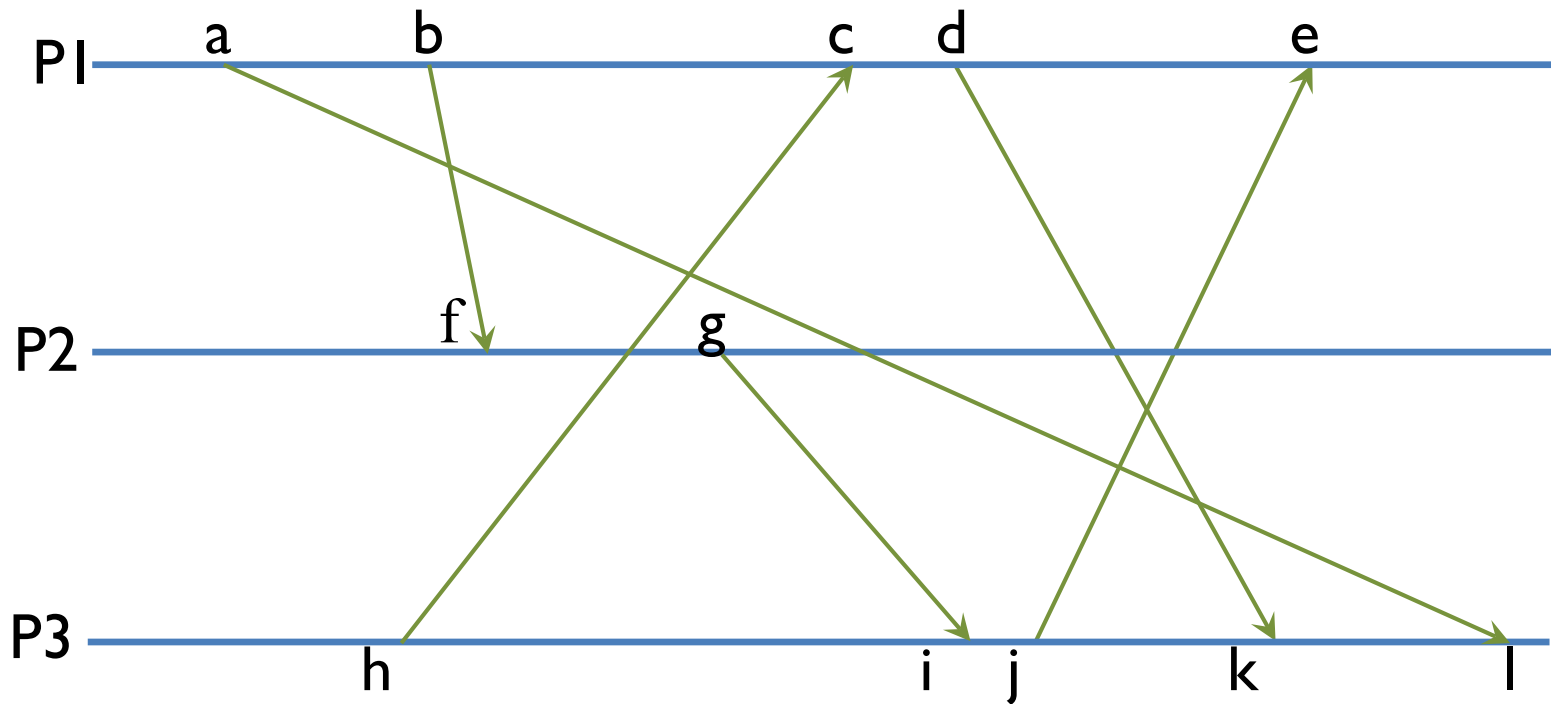
- Proporcione un orden total de los eventos



- Añadiendo el sufijo del nodo al reloj de Lamport: a (1.1), h(1.3), b(2.1), c(3.1), f(3.2), d(4.1), g(4.2), i(5.3), j(6.3), e(7.1), k(7.3), l(8.3)

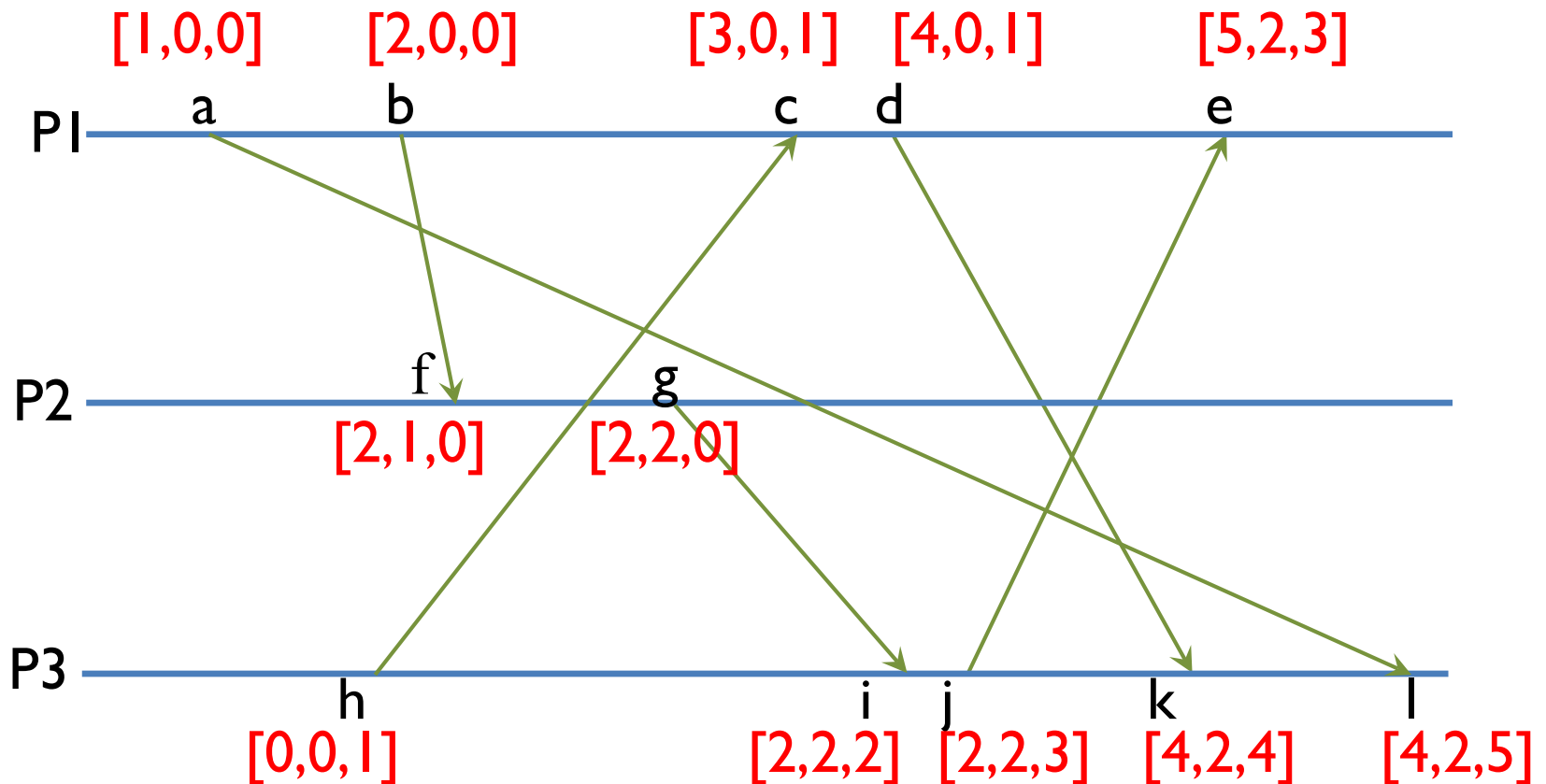
Actividad 6

- Indique qué reloj vectorial recibiría cada uno de los eventos



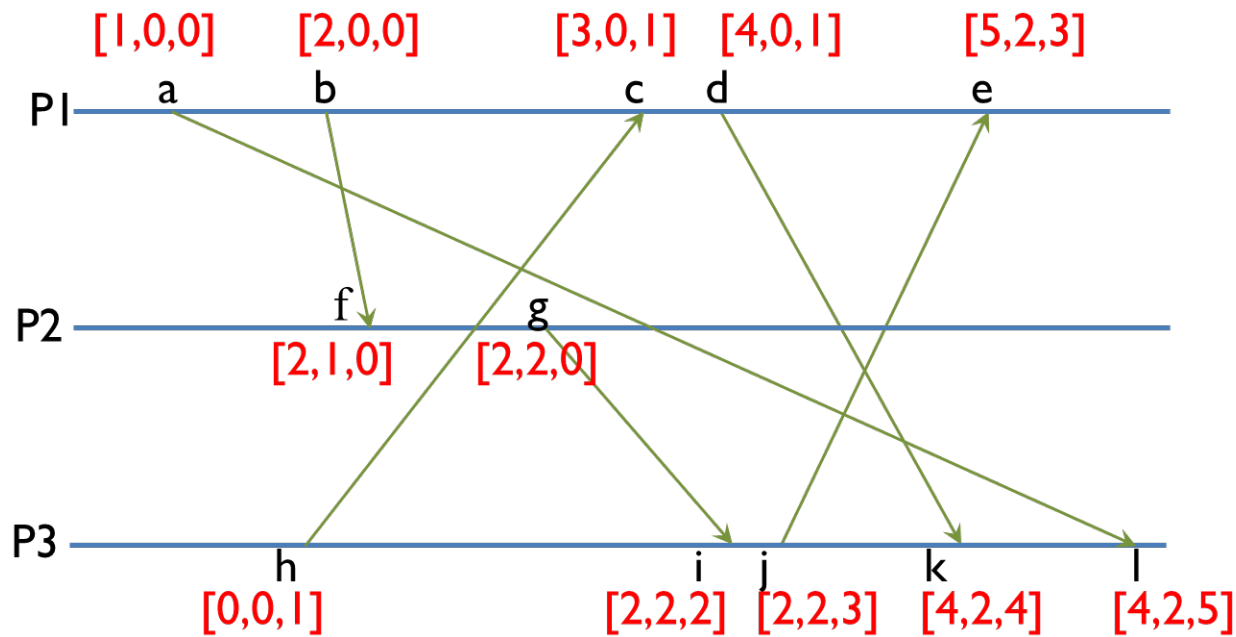
Actividad 6

- Indique qué reloj vectorial recibiría cada uno de los eventos



Actividad 6

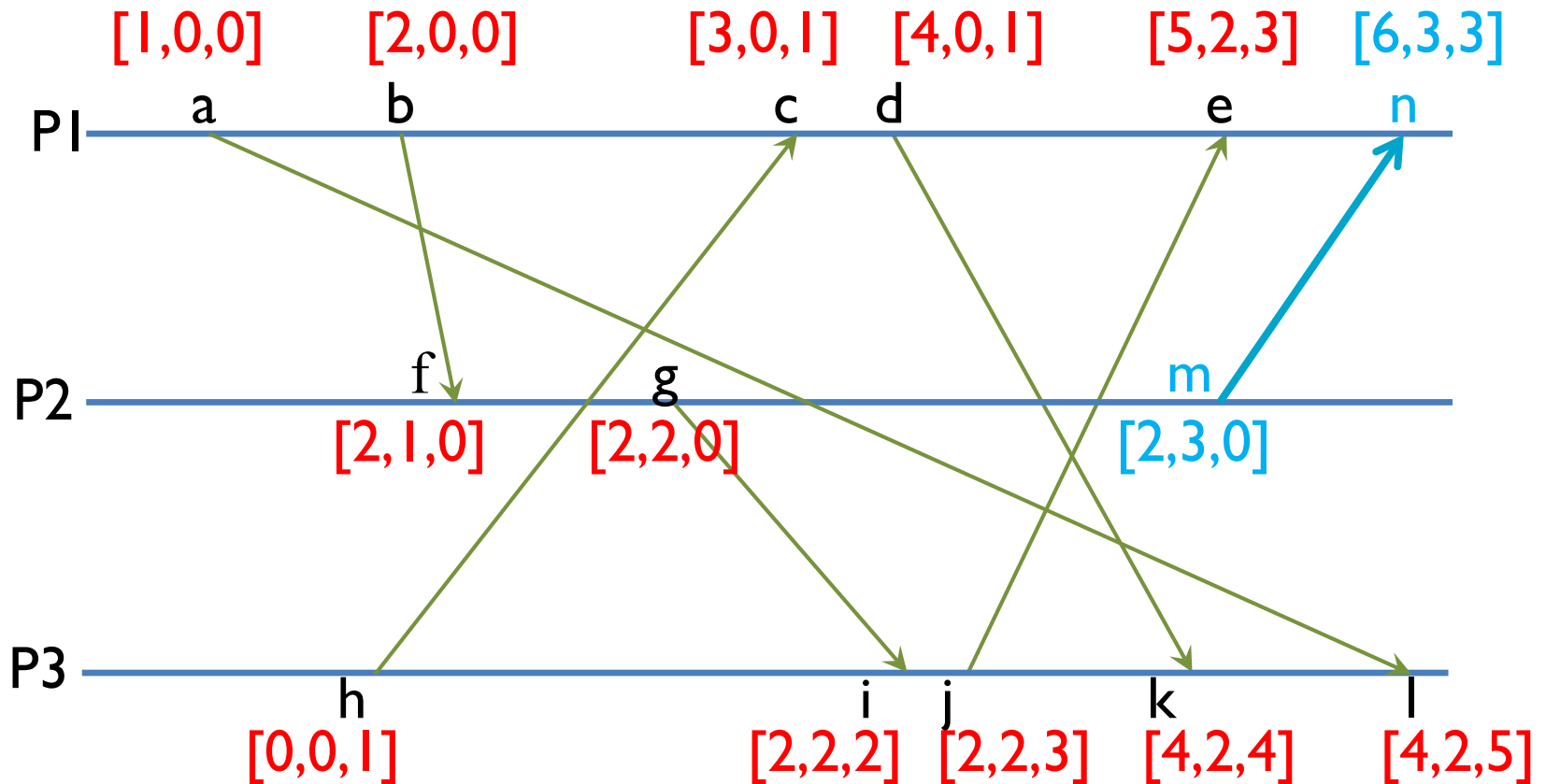
- Proporcione la lista completa de eventos concurrentes



Evento	Concurrente con:
a	h
b	h
c	f, g, i, j
d	f, g, i, j
e	k, l
f	c, d, h
g	c, d, h

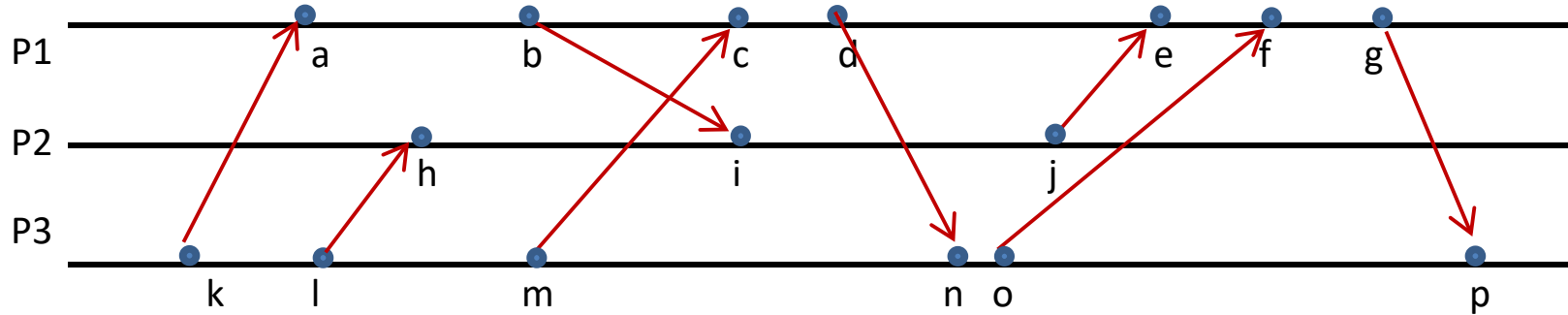
Actividad 6

- Si tras la traza mostrada en dicha ejecución el proceso P2 enviara un mensaje al proceso P1 y no hubiera ningún otro evento en ese sistema... ¿qué relojes vectoriales serían asignados al evento de envío y al evento de recepción de dicho mensaje?



Actividad 7

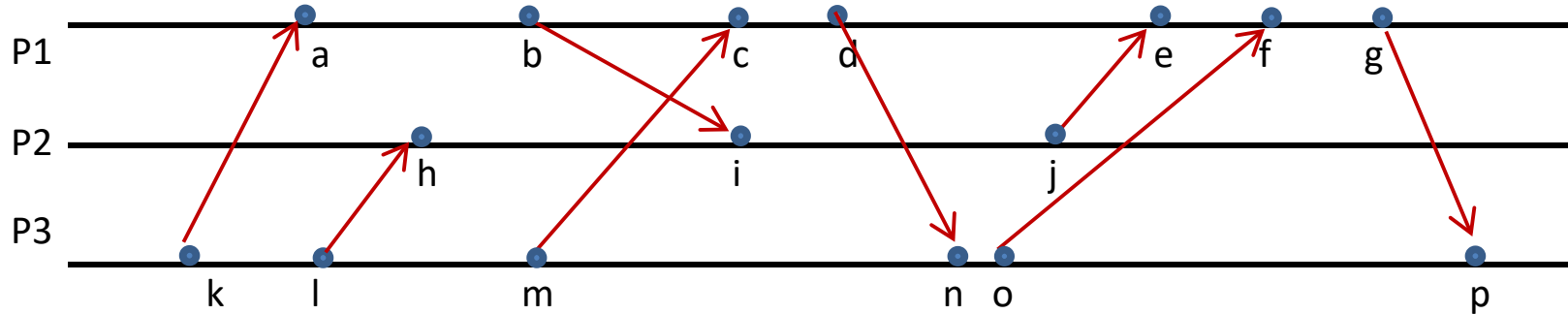
El cronograma que muestra la ejecución de tres procesos en un sistema distribuido, cada uno en un nodo distinto:



1.	Se determina que los eventos "d " y "j" son concurrentes.	
2.	$h \parallel n$	
3.	El reloj de Lamport de "g" es $C(g)=9$ y su reloj vectorial es $V(g)=[7, 3, 5]$	
4.	El reloj vectorial de "m" es $V(m)=[2,1,3]$, el reloj vectorial de "e" es $V(e)=[5,3,4]$ y podemos afirmar que $m \rightarrow e$	
5.	Se requiere utilizar relojes vectoriales para ordenar de forma total los eventos de este sistema.	

Actividad 7

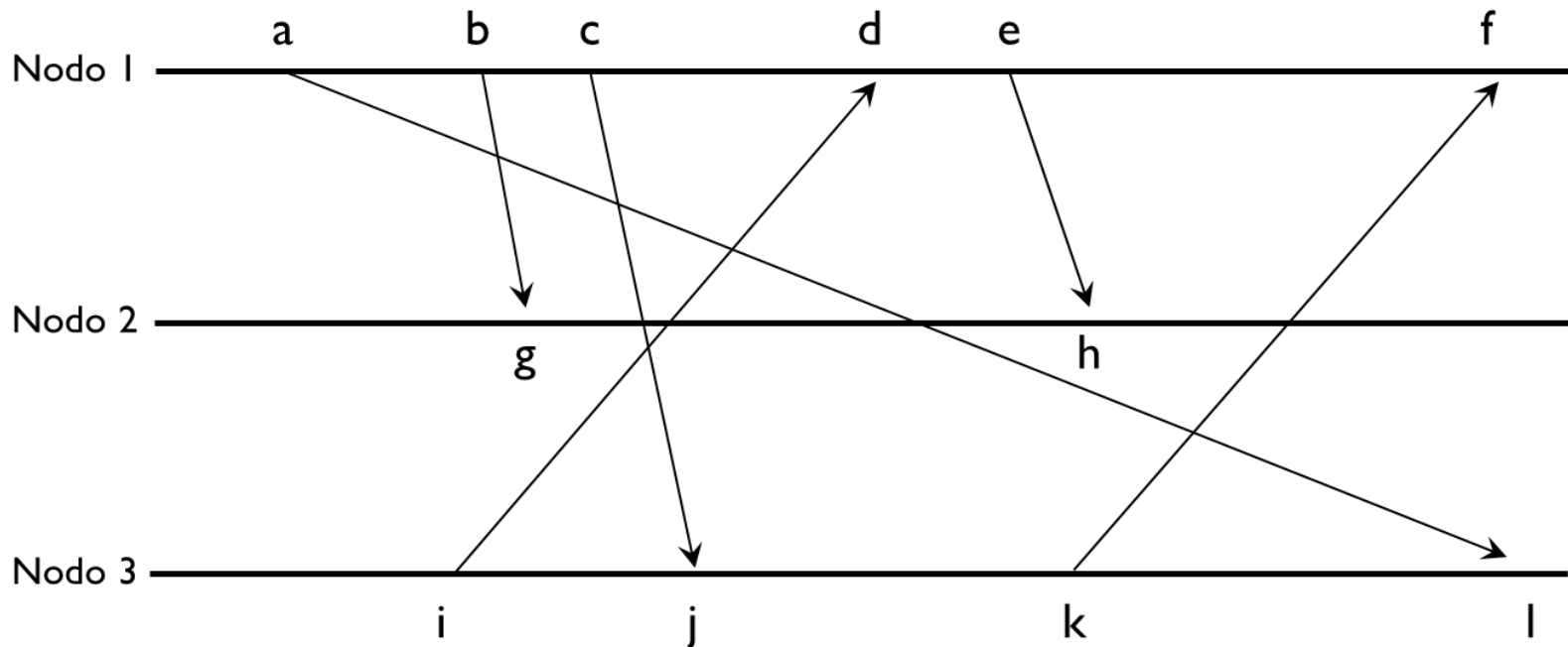
El cronograma que muestra la ejecución de tres procesos en un sistema distribuido, cada uno en un nodo distinto:



1.	Se determina que los eventos "d " y "j" son concurrentes.	V
2.	$h \parallel n$	V
3.	El reloj de Lamport de "g" es $C(g)=9$ y su reloj vectorial es $V(g)=[7, 3, 5]$	V
4.	El reloj vectorial de "m" es $V(m)=[2,1,3]$, el reloj vectorial de "e" es $V(e)=[5,3,4]$ y podemos afirmar que $m \rightarrow e$	F
5.	Se requiere utilizar relojes vectoriales para ordenar de forma total los eventos de este sistema.	F

Actividad 8

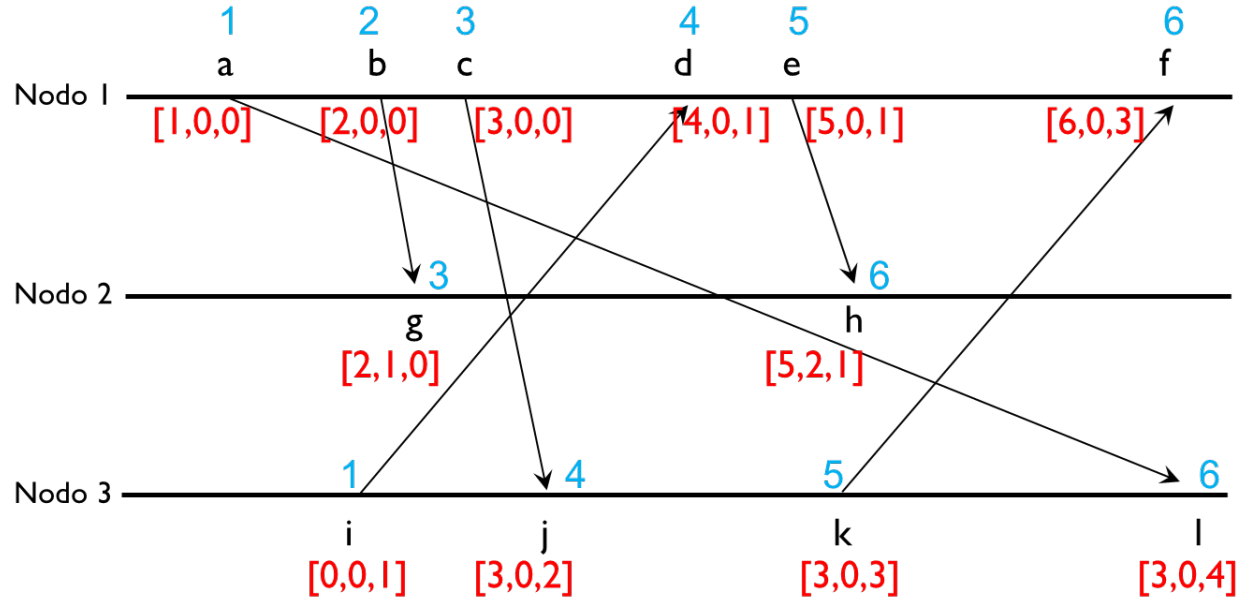
- ▶ La figura muestra todos los envíos de mensajes entre tres nodos:



- ▶ Indique los valores de los relojes de Lamport y vectoriales de cada evento.
- ▶ Conteste a las cuestiones V/F



Actividad 8



1.	Se cumple que $c \rightarrow l$ y que $e l$	V
2.	El valor en l del reloj lógico de Lamport es igual a 6	V
3.	El valor en d del reloj vectorial es igual a $V(d)=[4,0,1]$ y el valor en j del reloj vectorial es igual a $V(j)= [3,1,2]$ <i>JUSTIFICACIÓN: $V(d)=[4,0,1]$, $V(j)=[3,0,2]$</i>	F



Actividad 9

Respecto a los relojes lógicos de Lamport y la relación causal "ocurre-antes", indique si las siguientes afirmaciones son VERDADERAS (V) o FALSAS (F).

1.	Decimos que el evento "a" precede al evento "b" ($a \rightarrow b$) si todos los nodos están de acuerdo en que primero ven el evento "a" y después ven el evento "b".	
2.	Los relojes lógicos de Lamport garantizan que dentro de un mismo nodo los valores $L(x)$ son siempre estrictamente crecientes.	
3.	Dos eventos "a" y "b" de distintos nodos no pueden tener valores lógicos asociados $L(a)$ y $L(b)$ tales que $L(a)=L(b)$.	
4.	El reloj vectorial asocia un valor vectorial a cada evento (ej. para evento "x" hablamos de $V(x)$), de forma que si para dos eventos "a" y "b" cualesquiera se cumple $V(a) < V(b)$, entonces $a \rightarrow b$.	
5.	En los relojes vectoriales, para cualquier par de vectores distintos $V(a)$ y $V(b)$, se cumple que $V(a) < V(b)$ o $V(b) < V(a)$.	
6.	En los relojes vectoriales, en cada nodo necesitamos un vector de valores enteros, con un elemento por evento posible.	



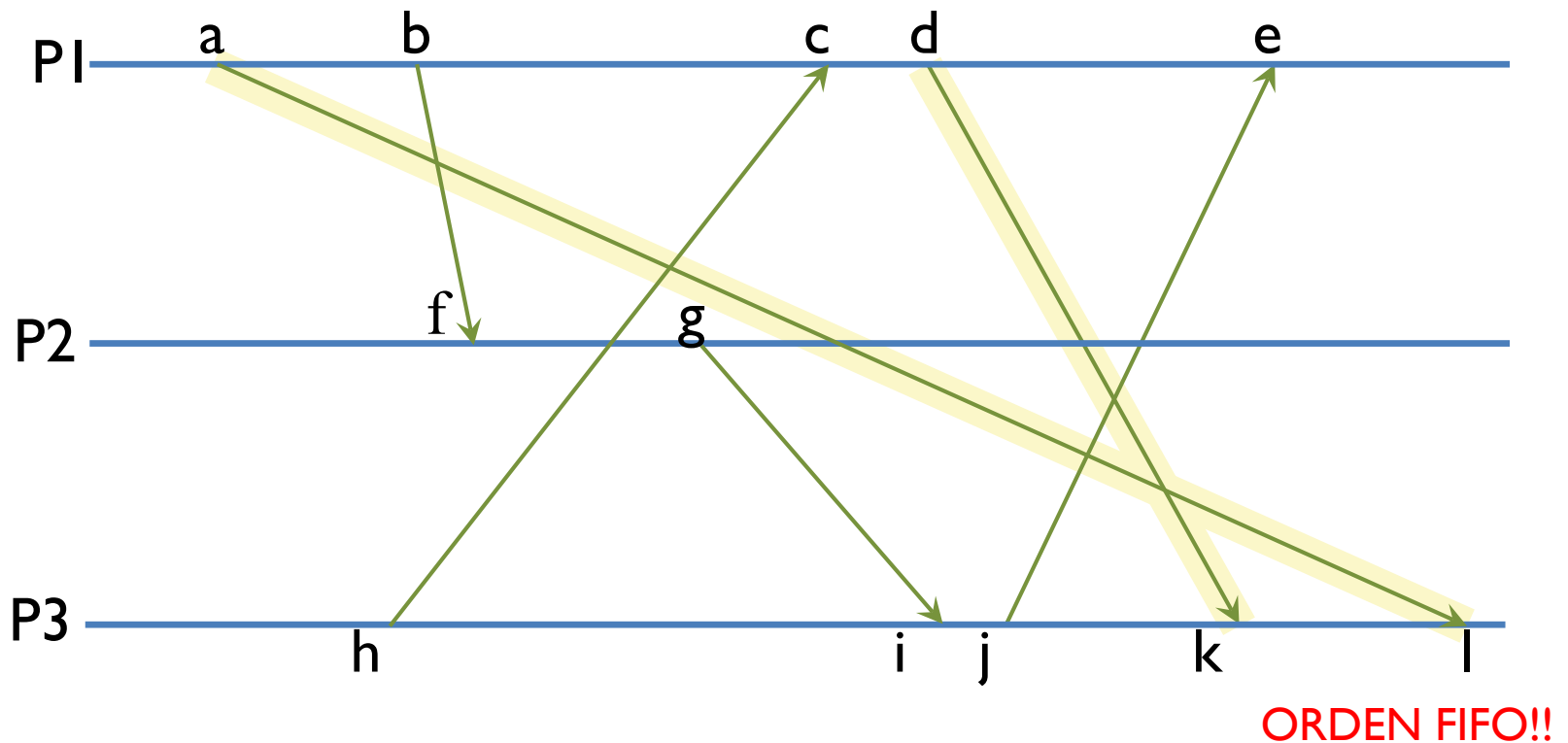
Actividad 9

Respecto a los relojes lógicos de Lamport y la relación causal "ocurre-antes", indique si las siguientes afirmaciones son VERDADERAS (V) o FALSAS (F).

1. Decimos que el evento "a" precede al evento "b" ($a \rightarrow b$) si todos los nodos están de acuerdo en que primero ven el evento "a" y después ven el evento "b".	V
2. Los relojes lógicos de Lamport garantizan que dentro de un mismo nodo los valores $L(x)$ son siempre estrictamente crecientes.	V
3. Dos eventos "a" y "b" de distintos nodos no pueden tener valores lógicos asociados $L(a)$ y $L(b)$ tales que $L(a)=L(b)$.	F
4. El reloj vectorial asocia un valor vectorial a cada evento (ej. para evento "x" hablamos de $V(x)$), de forma que si para dos eventos "a" y "b" cualesquiera se cumple $V(a) < V(b)$, entonces $a \rightarrow b$.	V
5. En los relojes vectoriales, para cualquier par de vectores distintos $V(a)$ y $V(b)$, se cumple que $V(a) < V(b)$ o $V(b) < V(a)$.	F
6. En los relojes vectoriales, en cada nodo necesitamos un vector de valores enteros, con un elemento por evento posible.	F

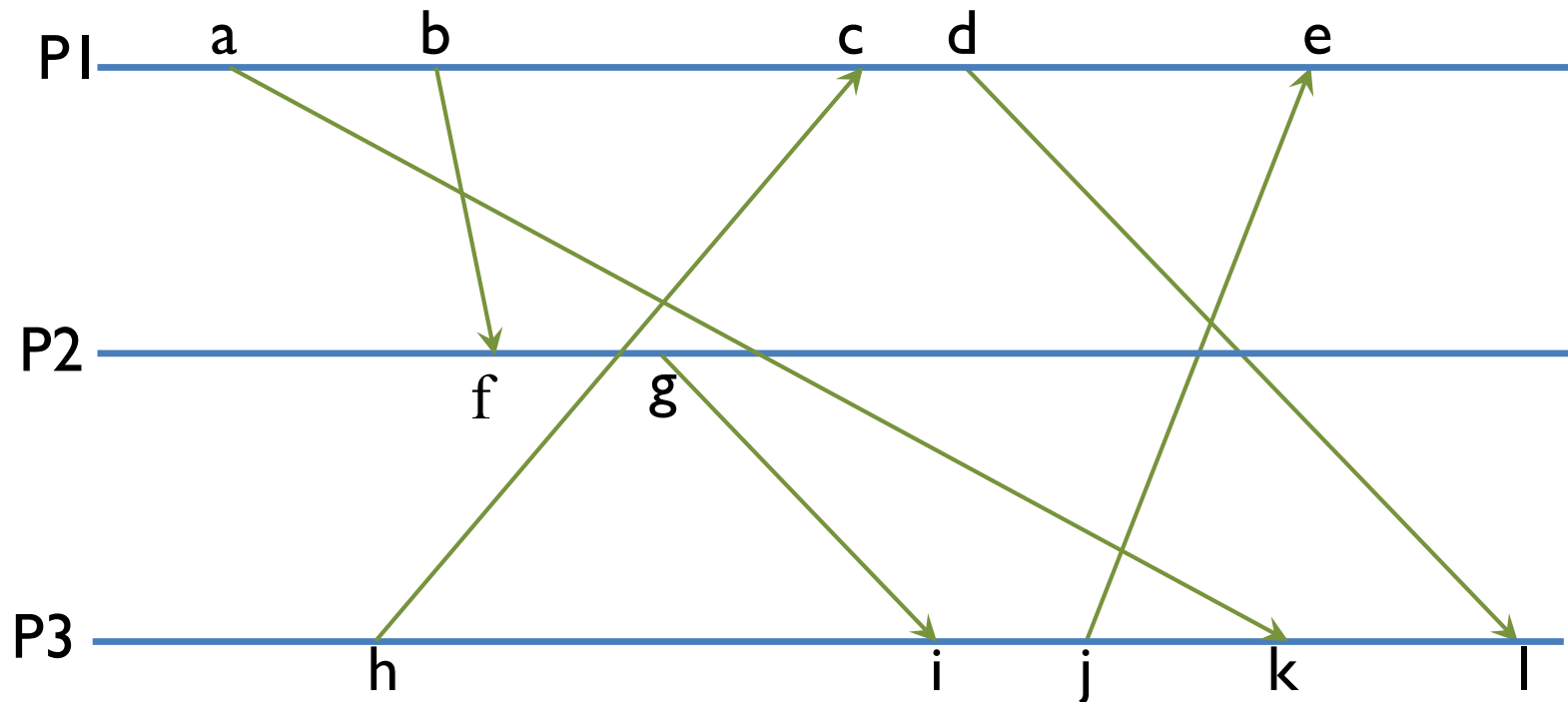
Actividad 10

- Razone si se podría utilizar el algoritmo de Chandy y Lamport

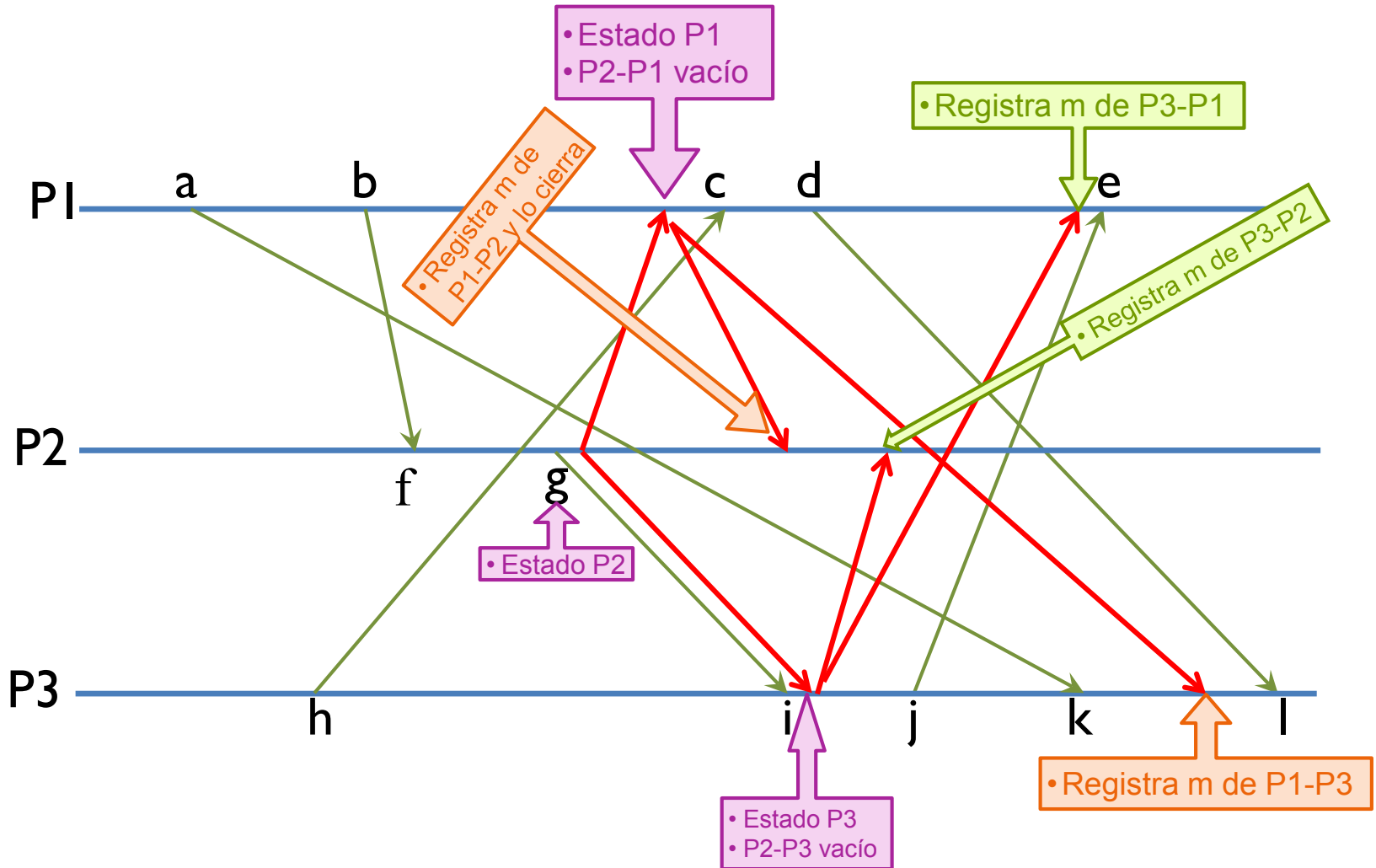


Actividad 11

- ▶ P2 inicia el algoritmo de Chandy y Lamport inmediatamente después de haber generado el evento “g” y que el mensaje MARCA que difunde se entrega en P1 y P3 justo antes de sus respectivos eventos “c” y “j”, generando a su vez difusiones de nuevos mensajes MARCA antes de los eventos “d” y “j”, respectivamente.



Actividad 11



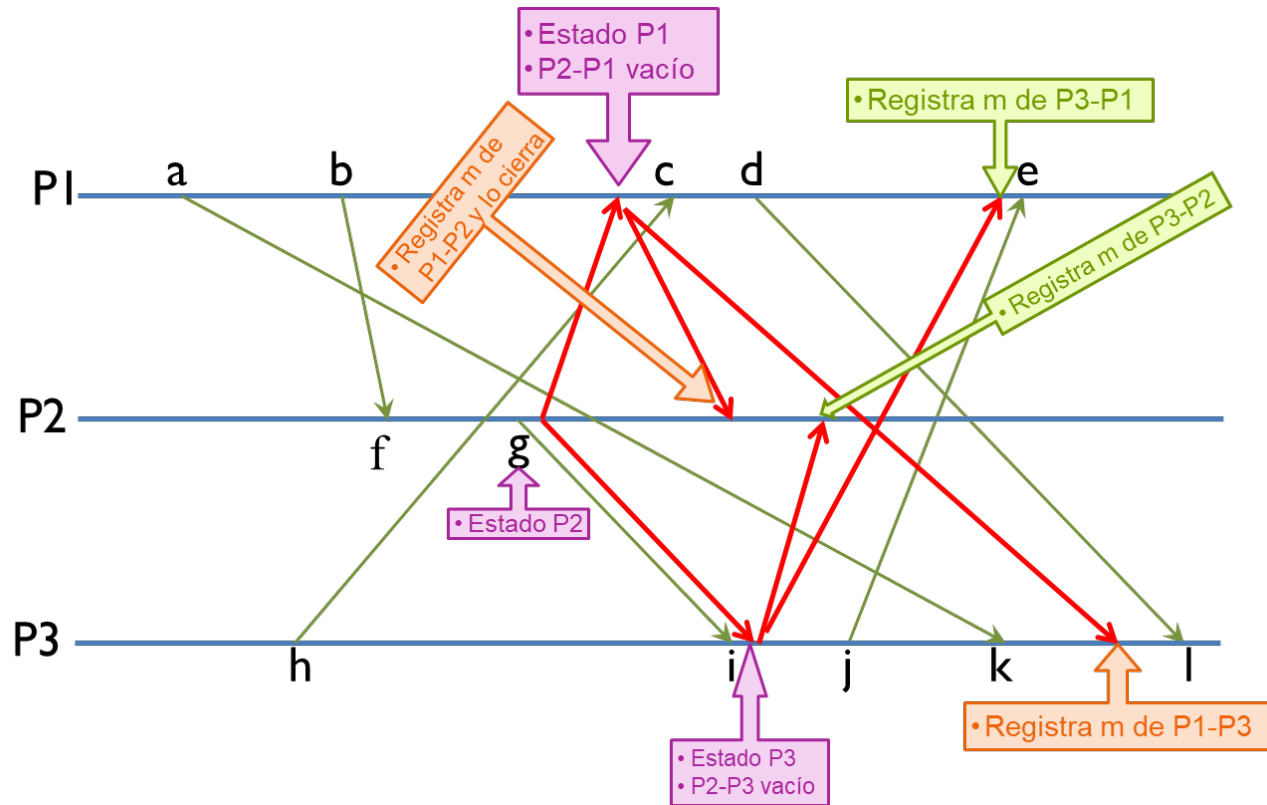
Actividad 11

► Mensajes en tránsito que se llegan a registrar son:

- Canal P3-→P1, mensaje emitido en el evento “h”.
- Canal P1-→P3, mensaje emitido en el evento “a”.

► Y los estados locales de cada proceso son:

- P1: estado que mantenía justo antes del evento “c”.
- P2: estado que mantenía inmediatamente después del evento “g”.
- P3: estado que mantenía entre los eventos “i” y “j”.

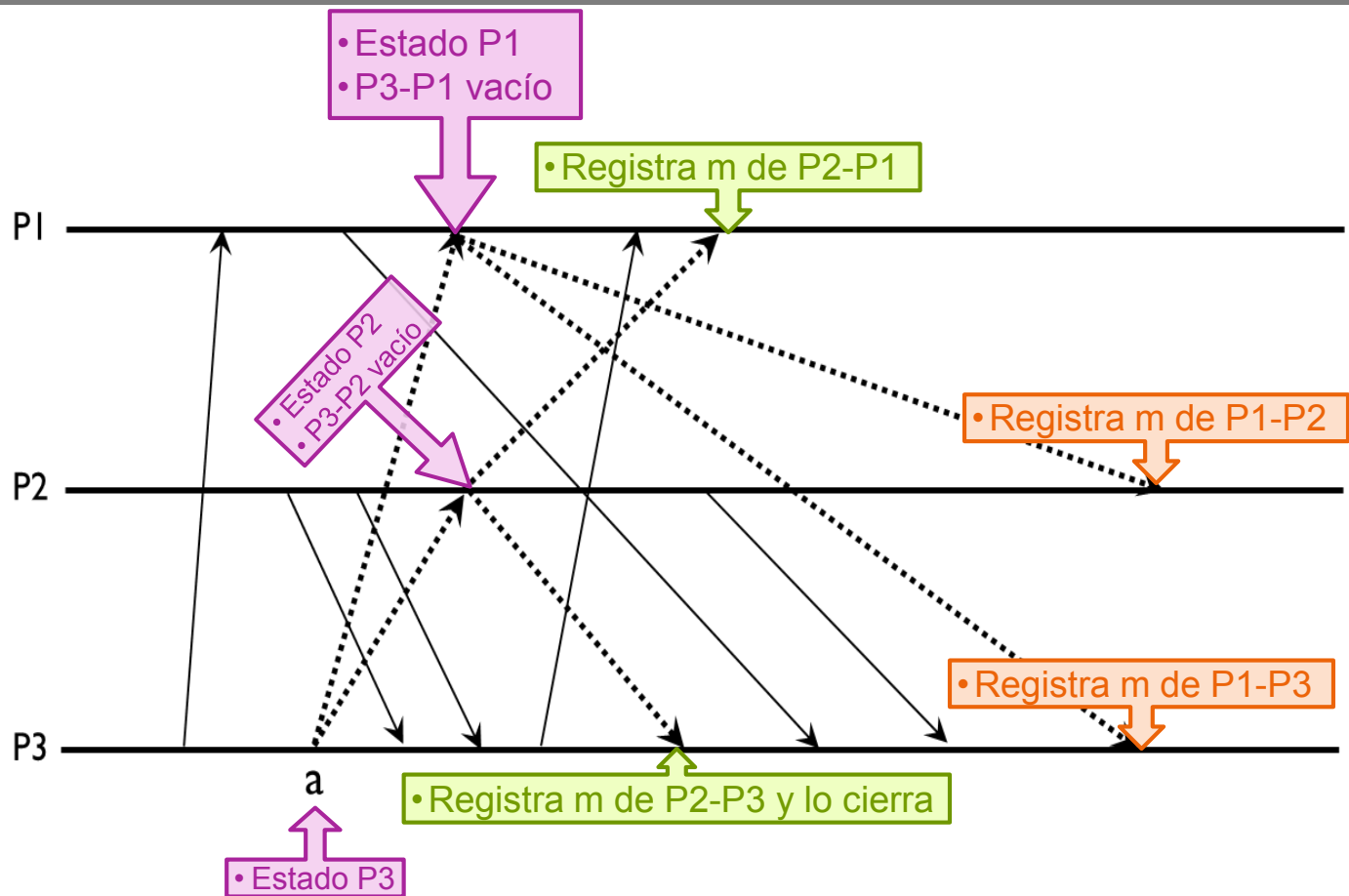




-

- | | | |
|----|---|--|
| 1. | en el canal (P3,P1) se habrán registrado 0 mensajes | |
| 2. | en el canal (P2,P3) se habrán registrado 3 mensajes | |
| 3. | en el canal (P1,P3) se habrá registrado 1 mensaje | |

Actividad 12



1.	en el canal (P3,P1) se habrán registrado 0 mensajes	V
2.	en el canal (P2,P3) se habrán registrado 3 mensajes	F
3.	en el canal (P1,P3) se habrá registrado 1 mensaje	V



Actividad 13

1. El algoritmo de Chandy-Lamport permite que los canales desordenen mensajes.

FORMULACIÓN CORRECTA:

2. El algoritmo de Chandy y Lamport permite que los canales de comunicación no respeten un orden FIFO. **FORMULACIÓN CORRECTA:**

3. El algoritmo de Chandy-Lamport permite establecer un orden total entre los eventos de un sistema distribuido. **FORMULACIÓN CORRECTA:**

4. En el algoritmo de Chandy-Lamport los procesos avanzan tomando sus decisiones en base a información global. **FORMULACIÓN CORRECTA:**

5. El algoritmo de Chandy-Lamport falla si más de un proceso inicia el algoritmo de forma quasi-simultánea (o sea, antes de recibir el mensaje MARCA del otro iniciador).

FORMULACIÓN CORRECTA:



Actividad 13

1. El algoritmo de Chandy-Lamport permite que los canales desordenen mensajes.
FORMULACIÓN CORRECTA: El algoritmo de Chandy-Lamport **requiere** que los canales **no** desordenen **ni pierdan** mensajes.
2. El algoritmo de Chandy y Lamport permite que los canales de comunicación no respeten un orden FIFO. **FORMULACIÓN CORRECTA:** El algoritmo de Chandy y Lamport **requiere** que los canales de comunicación respeten un orden FIFO.
3. El algoritmo de Chandy-Lamport permite establecer un orden total entre los eventos de un sistema distribuido. **FORMULACIÓN CORRECTA:** El algoritmo de Chandy-Lamport permite **obtener una imagen global consistente del sistema**.
4. En el algoritmo de Chandy-Lamport los procesos avanzan tomando sus decisiones en base a información global. **FORMULACIÓN CORRECTA:** En el algoritmo de Chandy-Lamport los procesos avanzan tomando sus decisiones en base a información **local**.
5. El algoritmo de Chandy-Lamport falla si más de un proceso inicia el algoritmo de forma quasi-simultánea (o sea, antes de recibir el mensaje MARCA del otro iniciador).
FORMULACIÓN CORRECTA: El algoritmo de Chandy-Lamport **no** falla si más de un proceso inicia el algoritmo de forma quasi-simultánea (o sea, antes de recibir el mensaje MARCA del otro iniciador).



Actividad 14

1. El algoritmo de Bully falla si dos o más nodos inician el algoritmo de forma simultánea (es decir, si hay más de un iniciador).	
2. El algoritmo de elección de líder sobre anillos falla si dos o más nodos inician el algoritmo de forma simultánea (es decir, si hay más de un iniciador).	
3. El algoritmo centralizado de exclusión mutua visto en clase limita la escalabilidad y la tolerancia a fallos porque el coordinador supone un cuello de botella y un punto de fallo único.	
4. El algoritmo distribuido de exclusión mutua visto en clase utiliza relojes lógicos para ordenar algunas solicitudes de entrada a la sección crítica.	
5. El algoritmo de Berkeley es uno de los algoritmos de exclusión mutua más eficientes.	
6. El algoritmo distribuido de exclusión mutua no necesita realizar ninguna acción en su protocolo de salida.	
7. El algoritmo de Cristian es un algoritmo de elección de líder muy eficiente.	



Actividad 14

1. El algoritmo de Bully falla si dos o más nodos inician el algoritmo de forma simultánea (es decir, si hay más de un iniciador).	F
2. El algoritmo de elección de líder sobre anillos falla si dos o más nodos inician el algoritmo de forma simultánea (es decir, si hay más de un iniciador).	F
3. El algoritmo centralizado de exclusión mutua visto en clase limita la escalabilidad y la tolerancia a fallos porque el coordinador supone un cuello de botella y un punto de fallo único.	V
4. El algoritmo distribuido de exclusión mutua visto en clase utiliza relojes lógicos para ordenar algunas solicitudes de entrada a la sección crítica.	V
5. El algoritmo de Berkeley es uno de los algoritmos de exclusión mutua más eficientes.	F
6. El algoritmo distribuido de exclusión mutua no necesita realizar ninguna acción en su protocolo de salida.	F
7. El algoritmo de Cristian es un algoritmo de elección de líder muy eficiente.	F



Actividad 15

- Sea un sistema con 8 nodos (N_1, N_2, \dots, N_8) donde en un momento determinado solamente están activos los nodos impares (y esta información aún no es conocida por los nodos). **COMPLETE LAS SIGUIENTES FRASES:**

1. En el algoritmo de elección de líder en anillo, si un nodo cualquiera recibe un mensaje de tipo ELECCIÓN con el contenido $(5, \{5, 3\})$, este mensaje indica:

iniciador participantes

2. Si se utiliza el algoritmo Bully de elección de líder, si el nodo N_7 envía un mensaje de tipo ELECCIÓN, esto indica que:

como no recibirá respuesta del N_8 , será el nuevo líder

3. Para la elección del líder, si el nodo N_1 inicia el algoritmo Bully, entonces enviará un mensaje ELECCIÓN a los nodos:

a todos los nodos cuyo id. sea mayor $N_2, N_3 \dots N_8$

4. Si el nodo N_5 inicia el algoritmo de elección de líder en anillo, en algún momento recibirá un mensaje de tipo ELECCIÓN con el siguiente contenido:

$(5, \{1, 3, 5, 7\})$



Actividad 15

- Sea un sistema con 8 nodos (N_1, N_2, \dots, N_8) donde en un momento determinado solamente están activos los nodos impares (y esta información aún no es conocida por los nodos).

1. Para la elección del líder, el algoritmo Bully implica que el nodo N_1 envía un mensaje ELECCIÓN únicamente a los siguientes nodos: N_3, N_5, N_7 .	
2. Si el nodo N_5 inicia el algoritmo de elección de líder en anillo, en algún momento recibirá un mensaje de tipo ELECCIÓN con el siguiente contenido: $(5, \{5, 6, 7, 8, 1, 2, 3, 4\})$	
3. En el algoritmo de elección de líder en anillo, si un nodo cualquiera recibe un mensaje de tipo ELECCIÓN con el contenido $(5, \{5, 3\})$, este mensaje anuncia que el nuevo líder es el nodo N_5 .	
4. Si se utiliza el algoritmo Bully de elección de líder, el nodo N_7 no enviará ningún mensaje de tipo ELECCIÓN.	



Actividad 15

- Sea un sistema con 8 nodos (N_1, N_2, \dots, N_8) donde en un momento determinado solamente están activos los nodos impares (y esta información aún no es conocida por los nodos).

1. Para la elección del líder, el algoritmo Bully implica que el nodo N_1 envía un mensaje ELECCIÓN únicamente a los siguientes nodos: N_3, N_5, N_7 .	F
2. Si el nodo N_5 inicia el algoritmo de elección de líder en anillo, en algún momento recibirá un mensaje de tipo ELECCIÓN con el siguiente contenido: $(5, \{5, 6, 7, 8, 1, 2, 3, 4\})$	F
3. En el algoritmo de elección de líder en anillo, si un nodo cualquiera recibe un mensaje de tipo ELECCIÓN con el contenido $(5, \{5, 3\})$, este mensaje anuncia que el nuevo líder es el nodo N_5 .	F
4. Si se utiliza el algoritmo Bully de elección de líder, el nodo N_7 no enviará ningún mensaje de tipo ELECCIÓN.	F



Actividad 16

1. El algoritmo de Chandy-Lamport requiere que los canales no pierdan ni desordenen mensajes.	V
2. En general es problemático hacer que el reloj de un ordenador retroceda.	V
3. En el algoritmo de Berkeley no se asume que exista un ordenador con un reloj preciso.	V
4. El algoritmo de Berkeley sincroniza los relojes de los nodos de un determinado sistema distribuido, sin importarle la divergencia que pueda haber entre estos relojes y la hora “oficial”.	V
5. El algoritmo de Cristian se utiliza para gestionar relojes lógicos	F
6. El algoritmo de Cristian proporciona la base necesaria para implantar cualquier algoritmo descentralizado.	F
7. El algoritmo de Chandy y Lamport requiere que los canales de comunicación respeten un orden FIFO	V
8. El algoritmo de Cristian permite identificar los mensajes en tránsito por cada uno de los canales de comunicación de un sistema distribuido.	F



Actividad 17

1. Se trata de un problema de consenso cuando varios nodos adoptan como valor de consenso el promedio de los valores estimados por cada uno de ellos.	F
2. Se trata de un problema de consenso cuando varios nodos adoptan como valor de consenso el máximo de los valores estimados por cada uno de ellos.	V
3. Después de ejecutarse el algoritmo de consenso, solamente los nodos que inicialmente propusieron "estimate(V)" proporcionarán como salida "decision(V)".	F
4. Un solución correcta de consenso debe cumplir con las propiedades de exclusión mutua, progreso y no expulsión.	F
5. Al final del algoritmo de consenso, dos procesos correctos podrían dar lugar a decisiones diferentes, pues cada uno habrá propuesto inicialmente un "estimate(Vi)" y, por tanto, dará lugar a un "decision(Vi)" diferente	F
6. Se cumple la propiedad de "integridad uniforme" cuando si un proceso decide "v", entonces "v" fue propuesto por algún proceso.	F
7. Los algoritmos de Paxos y Raft son implementaciones de algoritmos de consenso ampliamente utilizadas.	V
8. Un ejemplo de algoritmo de consenso consiste en que todos los nodos difundan su "estimate(V)" y se decida como "decision(V)" el valor propuesto por el nodo con el identificador más alto.	V



Actividad 18

1. En un algoritmo de consenso distribuido, en el que se considere que los nodos pueden fallar, los nodos disponen de un detector de fallos para conocer si el nodo ha cometido un fallo al calcular su valor "estimate(V)" que debe proponer.	F
2. Se llaman "detectores de fallos eventualmente perfectos" porque son detectores que antes o después determinarán un fallo de manera perfecta.	F
3. El algoritmo funciona si hay un máximo de $N/2$ nodos correctos.	F
4. Si tenemos 7 nodos y fallan 2, el algoritmo no funcionará, pues el nodo coordinador se bloqueará.	F
5. En el algoritmo, los nodos van ejecutando rondas. El nodo termina el algoritmo cuando es elegido nodo coordinador.	F
6. En cada ronda, los nodos ordinarios envían al coordinador de la ronda un mensaje "propose".	F
7. Los algoritmos de Paxos y Raft son implementaciones de algoritmos de consenso ampliamente utilizadas.	V
8. El coordinador elige uno de los valores "estimate" que recibe de entre todos aquellos que tengan el máximo valor de ronda.	V