



Grado en Ingeniería Informática

Sistemas distribuidos

Sincronización de tiempo y coordinación

Coulouris: Disitributed systems Ch14

Departamento de Tecnología Informática y Computación

Curso 2023 - 2024

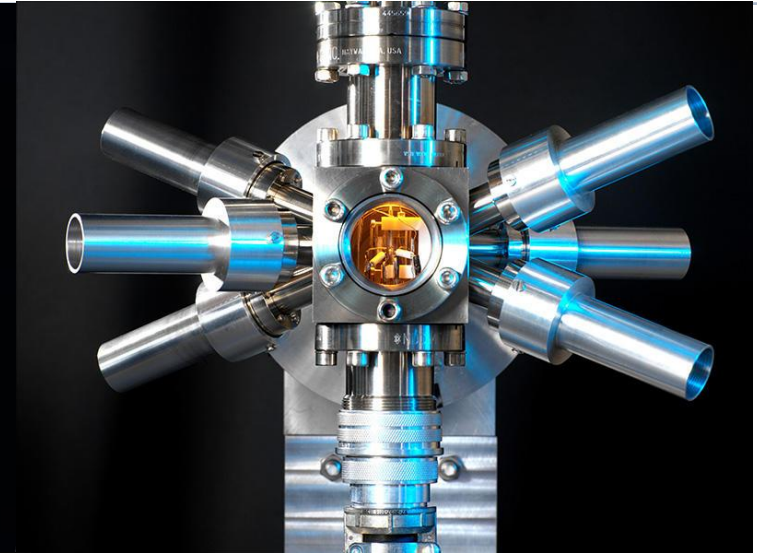
sincronización en los sistemas distribuidos

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador



sincronización en los sistemas distribuidos

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

- Para poder ejecutar tareas en entornos distribuidos de forma coherente es importante que el **orden** en las secuencias de las operaciones de procesos sea **estricto** y **universal**

Ejemplo:

Timestamps de comercio electrónico (facturas, orden de compra, ...)

sincronización en los sistemas distribuidos

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

- Para poder ejecutar tareas en entornos distribuidos de forma coherente es importante que el **orden** en las secuencias de las operaciones de procesos sea **estricto** y **universal**
- La **sincronización** se define como **la forma de forzar un orden parcial o total para cualquier conjunto de eventos** y se utiliza para abordar tres problemas distintos y relacionados:

sincronización en los sistemas distribuidos

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

- Para poder ejecutar tareas en entornos distribuidos de forma coherente es importante que el **orden** en las secuencias de las operaciones de procesos sea **estricto** y **universal**
- La **sincronización** se define como **la forma de forzar un orden parcial o total para cualquier conjunto de eventos** y se utiliza para abordar tres problemas distintos y relacionados:
 - La sincronización entre un emisor y un receptor

sincronización en los sistemas distribuidos

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

- Para poder ejecutar tareas en entornos distribuidos de forma coherente es importante que el **orden** en las secuencias de las operaciones de procesos sea **estricto** y **universal**
- La **sincronización** se define como **la forma de forzar un orden parcial o total para cualquier conjunto de eventos** y se utiliza para abordar tres problemas distintos y relacionados:
 - La sincronización entre un emisor y un receptor
 - La especificación y el control de la actividad común entre procesos cooperativos

sincronización en los sistemas distribuidos

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

- Para poder ejecutar tareas en entornos distribuidos de forma coherente es importante que el **orden** en las secuencias de las operaciones de procesos sea **estricto** y **universal**
- La **sincronización** se define como **la forma de forzar un orden parcial o total para cualquier conjunto de eventos** y se utiliza para abordar tres problemas distintos y relacionados:
 - La sincronización entre un emisor y un receptor
 - La especificación y el control de la actividad común entre procesos cooperativos
 - La serialización de accesos concurrentes a objetos compartidos

sincronización en los sistemas distribuidos

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

- Para poder ejecutar tareas en entornos distribuidos de forma coherente es importante que el **orden** en las secuencias de las operaciones de procesos sea **estricto** y **universal**
- La **sincronización** se define como **la forma de forzar un orden parcial o total para cualquier conjunto de eventos** y se utiliza para abordar tres problemas distintos y relacionados:
 - La sincronización entre un emisor y un receptor
 - La especificación y el control de la actividad común entre procesos cooperativos
 - La serialización de accesos concurrentes a objetos compartidos
- La **sincronización de relojes** en un sistema distribuido cobra especial importancia:

sincronización en los sistemas distribuidos

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

- Para poder ejecutar tareas en entornos distribuidos de forma coherente es importante que el **orden** en las secuencias de las operaciones de procesos sea **estricto** y **universal**
- La **sincronización** se define como **la forma de forzar un orden parcial o total para cualquier conjunto de eventos** y se utiliza para abordar tres problemas distintos y relacionados:
 - La sincronización entre un emisor y un receptor
 - La especificación y el control de la actividad común entre procesos cooperativos
 - La serialización de accesos concurrentes a objetos compartidos
- La **sincronización de relojes** en un sistema distribuido cobra especial importancia:
 - Se debe establecer la misma referencia de tiempo para todas las entidades

sincronización en los sistemas distribuidos

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

- Para poder ejecutar tareas en entornos distribuidos de forma coherente es importante que el **orden** en las secuencias de las operaciones de procesos sea **estricto** y **universal**
- La **sincronización** se define como **la forma de forzar un orden parcial o total para cualquier conjunto de eventos** y se utiliza para abordar tres problemas distintos y relacionados:
 - La sincronización entre un emisor y un receptor
 - La especificación y el control de la actividad común entre procesos cooperativos
 - La serialización de accesos concurrentes a objetos compartidos
- La **sincronización de relojes** en un sistema distribuido cobra especial importancia:
 - Se debe establecer la misma referencia de tiempo para todas las entidades
 - Se debe garantizar que los procesos se ejecuten de forma coordinada y respetando el orden original de los eventos, en la medida de lo posible

introducción

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

introducción

- Monitorización y temporización de ejecuciones

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

introducción

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

- Monitorización y temporización de ejecuciones
- Necesidad de conocer cuando ocurrió un evento

introducción

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

- Monitorización y temporización de ejecuciones
- Necesidad de conocer cuando ocurrió un evento
 - Algoritmos de sincronización

introducción

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

- Monitorización y temporización de ejecuciones
- Necesidad de conocer cuando ocurrió un evento
 - Algoritmos de sincronización
 - Manteniendo de consistencia en transacciones

introducción

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

- Monitorización y temporización de ejecuciones
- Necesidad de conocer cuando ocurrió un evento
 - Algoritmos de sincronización
 - Manteniendo de consistencia en transacciones
 - Protocolos de autenticación
 - ...

introducción

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

- Monitorización y temporización de ejecuciones
- Necesidad de conocer cuando ocurrió un evento
 - Algoritmos de sincronización
 - Manteniendo de consistencia en transacciones
 - Protocolos de autenticación
 - ...
- ¿Existe un reloj universal de referencia?

introducción

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

- Monitorización y temporización de ejecuciones
- Necesidad de conocer cuando ocurrió un evento
 - Algoritmos de sincronización
 - Manteniendo de consistencia en transacciones
 - Protocolos de autenticación
 - ...
- ¿Existe un reloj universal de referencia?
 - Teoría Especial de la Relatividad de Einstein
 - Causa física y efecto físico
 - Temporización de la causa y el efecto
 - Tiempo físico absoluto de Newton

introducción

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

introducción

Noción de tiempo es también problemática en un sistema distribuido:

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

introducción

Noción de tiempo es también problemática en un sistema distribuido:

- No existe un reloj global al sistema

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

introducción

Noción de tiempo es también problemática en un sistema distribuido:

- No existe un reloj global al sistema
- Cada computador de la red tiene su propio reloj interno:

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

introducción

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

Noción de tiempo es también problemática en un sistema distribuido:

- No existe un reloj global al sistema
- Cada computador de la red tiene su propio reloj interno:
 - usado por los procesos locales para obtener el valor del tiempo actual. P. ej. uso en el *make* de *unix*
 - procesos en computadores distintos pueden tener marcas de tiempo distintas
 - los relojes derivan con respecto al tiempo perfecto y las tasas de deriva también difieren entre ellos

introducción

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

Noción de tiempo es también problemática en un sistema distribuido:

- No existe un reloj global al sistema
- Cada computador de la red tiene su propio reloj interno:
 - usado por los procesos locales para obtener el valor del tiempo actual. P. ej. uso en el *make* de *unix*
 - procesos en computadores distintos pueden tener marcas de tiempo distintas
 - los relojes derivan con respecto al tiempo perfecto y las tasas de deriva también difieren entre ellos
- Aunque todos los relojes del SD se sincronicen, estos variarán significativamente con el tiempo

introducción

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

Noción de tiempo es también problemática en un sistema distribuido:

- No existe un reloj global al sistema
- Cada computador de la red tiene su propio reloj interno:
 - usado por los procesos locales para obtener el valor del tiempo actual. P. ej. uso en el *make* de *unix*
 - procesos en computadores distintos pueden tener marcas de tiempo distintas
 - los relojes derivan con respecto al tiempo perfecto y las tasas de deriva también difieren entre ellos
- Aunque todos los relojes del SD se sincronicen, estos variarán significativamente con el tiempo

Aproximaciones a la solución:

- Algoritmos de sincronización de los relojes de los computadores
- Relojes lógicos y relojes vectoriales

eventos y relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

eventos y relojes

- **Caracterización** de un sistema distribuido (SD):

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

eventos y relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

- **Caracterización** de un sistema distribuido (SD):
 - Un SD se define como una colección P de N procesos p_i donde $i = 1, 2, \dots, N$
 - Cada proceso p_i tiene un estado si formado por todas sus variables u objetos y que puede cambiar en ejecución
 - Se comunican a través de la red mediante mensajes
 - Las acciones que puede realizar un proceso son: enviar, recibir o cambiar estado

eventos y relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

- **Caracterización** de un sistema distribuido (SD):
 - Un SD se define como una colección P de N procesos p_i donde $i = 1, 2, \dots, N$
 - Cada proceso p_i tiene un estado si formado por todas sus variables u objetos y que puede cambiar en ejecución
 - Se comunican a través de la red mediante mensajes
 - Las acciones que puede realizar un proceso son: enviar, recibir o cambiar estado
- **Evento:** ocurrencia de una acción que lleva a cabo un proceso al ejecutar:
p.e. un envío, una recepción, un cálculo, ...

eventos y relojes

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

- **Caracterización** de un sistema distribuido (SD):
 - Un SD se define como una colección P de N procesos p_i donde $i = 1, 2, \dots, N$
 - Cada proceso p_i tiene un estado si formado por todas sus variables u objetos y que puede cambiar en ejecución
 - Se comunican a través de la red mediante mensajes
 - Las acciones que puede realizar un proceso son: enviar, recibir o cambiar estado
- **Evento:** ocurrencia de una acción que lleva a cabo un proceso al ejecutar:
p.e. un envío, una recepción, un cálculo, ...
- Los eventos en el proceso p_i , pueden ordenarse de forma total por la **relación \rightarrow_i “suceder antes en p_i ”**

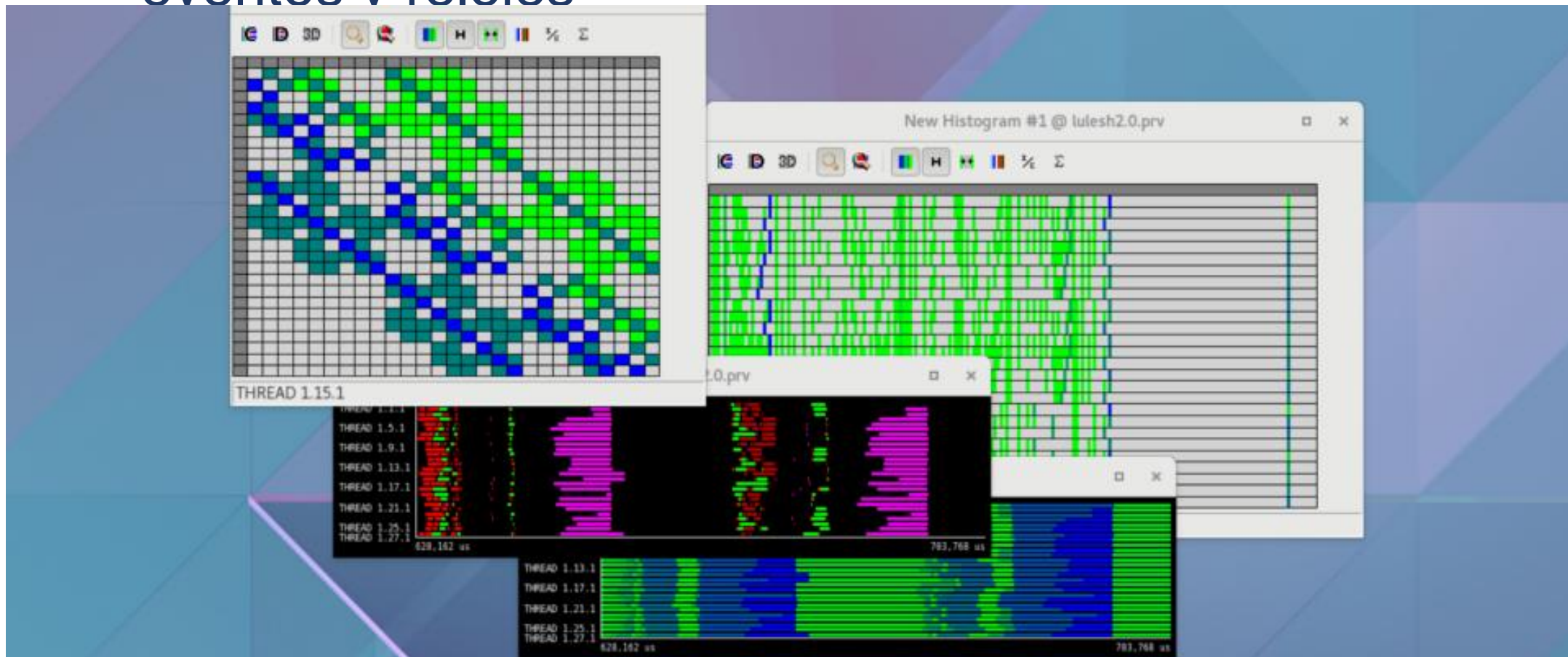
eventos v relojes

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador



- Los eventos en el proceso p_i , pueden ordenarse de forma total por la **relación \rightarrow_i “suceder antes en p_i ”**

eventos y relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

eventos y relojes

Relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

eventos y relojes

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Relojes

- Para establecer las marcas temporales se usa el reloj del computador
- En un instante t el SO lee el valor del reloj hardware del computador: $H_i(t)$ y calcula el tiempo mediante software (reloj software):

eventos y relojes

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Relojes

- Para establecer las marcas temporales se usa el reloj del computador
- En un instante t el SO lee el valor del reloj hardware del computador: $H_i(t)$ y calcula el tiempo mediante software (reloj software):

$$C_i(t) = \alpha H_i(t) + \beta \text{ (es decir, escala y compensa)}$$

- p.e. un número de 64-bit dando los nanosegundos desde un tiempo base
- En general no es completamente exacto, pero si C_i se comporta suficientemente bien, puede ser usado como marcador de los eventos de p_i

eventos y relojes

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Relojes

- Para establecer las marcas temporales se usa el reloj del computador
- En un instante t el SO lee el valor del reloj hardware del computador: $H_i(t)$ y calcula el tiempo mediante software (reloj software):

$$C_i(t) = \alpha H_i(t) + \beta \text{ (es decir, escala y compensa)}$$

- p.e. un número de 64-bit dando los nanosegundos desde un tiempo base
 - En general no es completamente exacto, pero si C_i se comporta suficientemente bien, puede ser usado como marcador de los eventos de p_i
- **Resolución del reloj:** periodo entre dos actualizaciones consecutivas del reloj y debe ser menor que el intervalo de tiempo entre dos eventos consecutivos

eventos y relojes

Relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

eventos y relojes

Relojes

- Los relojes de un SD no siempre están en perfecto acuerdo

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

eventos y relojes

Relojes

- Los relojes de un SD no siempre están en perfecto acuerdo
- **Sesgo**: diferencia de tiempo entre dos relojes en un instante determinado

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

eventos y relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

Relojes

- Los relojes de un SD no siempre están en perfecto acuerdo
- **Sesgo**: diferencia de tiempo entre dos relojes en un instante determinado
- **Tasa de deriva**: diferencia por unidad de tiempo en que el reloj del computador difiere del reloj de referencia

eventos y relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

Relojes

- Los relojes de un SD no siempre están en perfecto acuerdo
 - **Sesgo:** diferencia de tiempo entre dos relojes en un instante determinado
 - **Tasa de deriva:** diferencia por unidad de tiempo en que el reloj del computador difiere del reloj de referencia
 - Los relojes de cuarzo ordinarios derivan 1 seg. cada 11-12 días (10^{-6} seg/seg)
 - Los relojes de alta precisión derivan 10^{-7} ó 10^{-8} seg/seg
- ¿Qué puede influir en la tasa de deriva? p. ej. el estado de las baterías, ¿se te ocurren otras?

eventos y relojes

Relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

eventos y relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

Relojes

- **Corrección de un reloj:** Se dice que un reloj hardware (H) es correcto, si su límite de deriva es conocido $\rho > 0$. (p.e. 10^{-6} sec/sec):
 - El error en la en la marca de tiempo de dos eventos en t y t' está acotado: $(1 - \rho)(t' - t) \leq H(t') - H(t) \leq (1 + \rho)(t' - t)$, donde $t' > t$
 - Se impide que se produzcan saltos traumáticos en el valor de tiempo

eventos y relojes

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Relojes

- **Corrección de un reloj:** Se dice que un reloj hardware (H) es correcto, si su límite de deriva es conocido $\rho > 0$. (p.e. 10^{-6} sec/sec):
 - El error en la en la marca de tiempo de dos eventos en t y t' está acotado: $(1 - \rho)(t' - t) \leq H(t') - H(t) \leq (1 + \rho)(t' - t)$, donde $t' > t$
 - Se impide que se produzcan saltos traumáticos en el valor de tiempo
- Se puede relajar la condición, exigiendo tan solo **monotonicidad**:
 - $t' > t \rightarrow C(t') > C(t)$ [p.e. exigencia del *make* de Unix]
 - se puede alcanzar la monotonicidad en un reloj hardware que a mayor frecuencia, ajustando los valores de α β
 $C_i(t) = \alpha H_i(t) + \beta$

eventos y relojes

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Relojes

- **Corrección de un reloj:** Se dice que un reloj hardware (H) es correcto, si su límite de deriva es conocido $\rho > 0$. (p.e. 10^{-6} sec/sec):
 - El error en la en la marca de tiempo de dos eventos en t y t' está acotado: $(1 - \rho)(t' - t) \leq H(t') - H(t) \leq (1 + \rho)(t' - t)$, donde $t' > t$
 - Se impide que se produzcan saltos traumáticos en el valor de tiempo
- Se puede relajar la condición, exigiendo tan solo **monotonicidad**:
 - $t' > t \rightarrow C(t') > C(t)$ [p.e. exigencia del *make* de Unix]
 - se puede alcanzar la monotonicidad en un reloj hardware que a mayor frecuencia, ajustando los valores de α y β
 $C_i(t) = \alpha H_i(t) + \beta$
- Un **reloj defectuoso** es aquel que no cumple ninguna de las condiciones de corrección
- Un **fallo de ruptura** de reloj: el reloj se para, no emite tics
- Un **fallo arbitrario**: cualquier otro fallo... p. ej. efecto 2000 (**Y2K**) 2038 (**Y2K38**)

eventos y relojes

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Relojes

- **Corrección de un reloj:** Se dice que un reloj hardware (H) es correcto, si su límite de

Binary : 01111111 11111111 11111111 11110000

nde $t' > t$

Decimal : 2147483632

Date : 2038-01-19 03:13:52 (UTC)

Date : 2038-01-19 03:13:52 (UTC)

cción

- Un **fallo de ruptura** de reloj: el reloj se para, no emite tics
- Un **fallo arbitrario**: cualquier otro fallo... p. ej. efecto 2000 (**Y2K**) 2038 (**Y2K38**)

eventos y relojes

Tiempo Universal Coordinado (UTC)

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

eventos y relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

Tiempo Universal Coordinado (UTC)

- UTC es un estándar internacional de establecimiento y mantenimiento del tiempo transcurrido

eventos y relojes

introducción

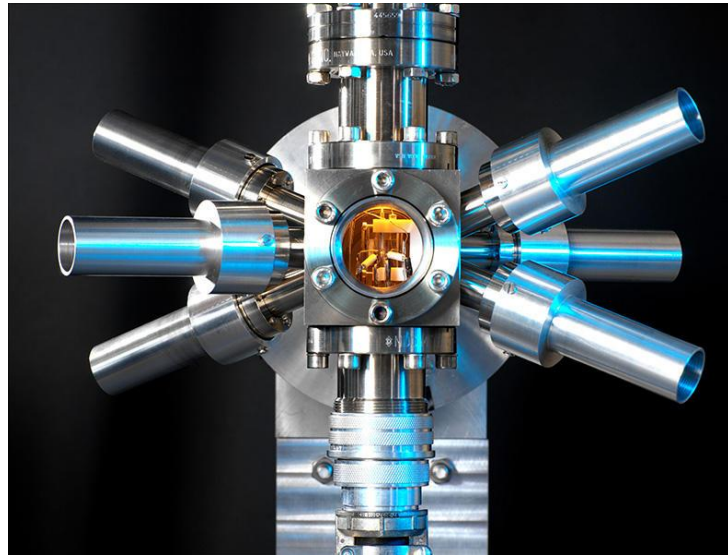
**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

Tiempo Universal Coordinado (UTC)

- UTC es un estándar internacional de establecimiento y mantenimiento del tiempo transcurrido
- Basado en el tiempo atómico y ocasionalmente ajustado al tiempo astronómico*. La referencia de tiempo la obtiene de un reloj con tasa de deriva de 10^{-13} (aprox. de un seg. cada 300.000 años) que establece el Tiempo Atómico Internacional



eventos y relojes

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Tiempo Universal Coordinado (UTC)

- UTC es un estándar internacional de establecimiento y mantenimiento del tiempo transcurrido
- Basado en el tiempo atómico y ocasionalmente ajustado al tiempo astronómico*. La referencia de tiempo la obtiene de un reloj con tasa de deriva de 10^{-13} (aprox. de un seg. cada 300.000 años) que establece el Tiempo Atómico Internacional
- La señal se difunde mediante estaciones de radio por tierra y mediante satélites, permitiendo que los relojes de los computadores se sincronizan con estas fuentes externas tan precisas, con receptores adecuados ¿factible? ¿útil?:
 - Las estaciones terrestres tienen una precisión entre 0.1-10 miliseg.
 - El GPS tiene una precisión de 1 microseg.

eventos y relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

eventos y relojes

Sincronización de los relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

eventos y relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

Sincronización de los relojes

- **Sincronización externa:** un reloj C_i se sincroniza con una fuente UTC exacta S , si se cumple:
 - $|S(t) - C_i(t)| < D$ para $i = 1, 2, \dots, N$ en un intervalo de tiempo
 - Los relojes C_i son precisos con el límite D

eventos y relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

Sincronización de los relojes

- **Sincronización externa:** un reloj C_i se sincroniza con una fuente UTC exacta S , si se cumple:
 - $|S(t) - C_i(t)| < D$ para $i = 1, 2, \dots, N$ en un intervalo de tiempo
 - Los relojes C_i son precisos con el límite D
- **Sincronización interna:** cualquier par de computadores están sincronizados, si sus relojes cumplen:
 - $|C_i(t) - C_j(t)| < Z$ para $i, j = 1, 2, \dots, N; i \neq j$ en un intervalo de tiempo
 - Los relojes C_i y C_j concuerdan con el límite Z

eventos y relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

Sincronización de los relojes

- **Sincronización externa:** un reloj C_i se sincroniza con una fuente UTC exacta S , si se cumple:
 - $|S(t) - C_i(t)| < D$ para $i = 1, 2, \dots, N$ en un intervalo de tiempo
 - Los relojes C_i son precisos con el límite D
- **Sincronización interna:** cualquier par de computadores están sincronizados, si sus relojes cumplen:
 - $|C_i(t) - C_j(t)| < Z$ para $i, j = 1, 2, \dots, N; i \neq j$ en un intervalo de tiempo
 - Los relojes C_i y C_j concuerdan con el límite Z
- Relojes sincronizados internamente no necesariamente lo están externamente, puesto que pueden derivar juntos
- *Si el conjunto P está sincronizado externamente, ¿también lo está internamente?*

eventos y relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

Sincronización de los relojes

- **Sincronización externa:** un reloj C_i se sincroniza con una fuente UTC exacta S , si se cumple:
 - $|S(t) - C_i(t)| < D$ para $i = 1, 2, \dots, N$ en un intervalo de tiempo
 - Los relojes C_i son precisos con el límite D
- **Sincronización interna:** cualquier par de computadores están sincronizados, si sus relojes cumplen:
 - $|C_i(t) - C_j(t)| < Z$ para $i, j = 1, 2, \dots, N; i \neq j$ en un intervalo de tiempo
 - Los relojes C_i y C_j concuerdan con el límite Z
- Relojes sincronizados internamente no necesariamente lo están externamente, puesto que pueden derivar juntos
- *Si el conjunto P está sincronizado externamente, ¿también lo está internamente?*

$$Z = 2D$$

sincronización de relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

sincronización de relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

La sincronización de relojes en un SD depende de la naturaleza del mismo y determinará el algoritmo de sincronización más adecuado:

- SD síncrono
- SD asíncrono

sincronización de relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

La sincronización de relojes en un SD depende de la naturaleza del mismo y determinará el algoritmo de sincronización más adecuado:

- SD síncrono
- SD asíncrono

Un **sistema distribuido** es **síncrono** si están definidos o se pueden determinar los parámetros siguientes:

sincronización de relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

La sincronización de relojes en un SD depende de la naturaleza del mismo y determinará el algoritmo de sincronización más adecuado:

- SD síncrono
- SD asíncrono

Un **sistema distribuido** es **síncrono** si están definidos o se pueden determinar los parámetros siguientes:

- Tiempo máximo y mínimo para ejecutar cada instrucción de un proceso
- Tiempo máximo y mínimo de recepción de un mensaje
- Los límites de deriva de cada reloj local donde se ejecuta cada proceso son conocidos

sincronización de relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

La sincronización de relojes en un SD depende de la naturaleza del mismo y determinará el algoritmo de sincronización más adecuado:

- SD síncrono
- SD asíncrono

Un **sistema distribuido** es **síncrono** si están definidos o se pueden determinar los parámetros siguientes:

- Tiempo máximo y mínimo para ejecutar cada instrucción de un proceso
- Tiempo máximo y mínimo de recepción de un mensaje
- Los límites de deriva de cada reloj local donde se ejecuta cada proceso son conocidos

Un **sistema distribuido** es **asíncrono** si no se pueden determinar los parámetros anteriores (es decir, **si no es síncrono**)

sincronización de relojes

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

La sincronización de relojes en un SD depende de la naturaleza del mismo y determinará el algoritmo de sincronización más adecuado:

- SD síncrono
- SD asíncrono

Un **sistema distribuido** es **síncrono** si están definidos o se pueden determinar los parámetros siguientes:

- Tiempo máximo y mínimo para ejecutar cada instrucción de un proceso
- Tiempo máximo y mínimo de recepción de un mensaje
- Los límites de deriva de cada reloj local donde se ejecuta cada proceso son conocidos

Un **sistema distribuido** es **asíncrono** si no se pueden determinar los parámetros anteriores (es decir, **si no es síncrono**)

¿Ejemplos de uno y otro? ¿computadores en una LAN? ¿Internet?

sincronización de relojes

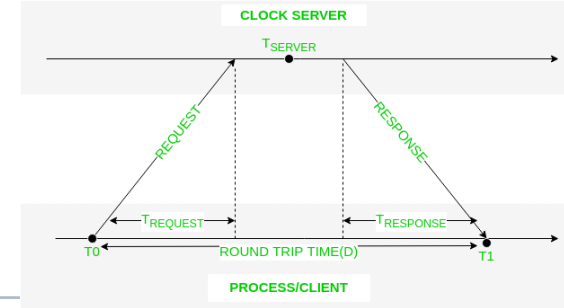
Método de Cristian (para SD síncronos y sincronización externa)

introducción

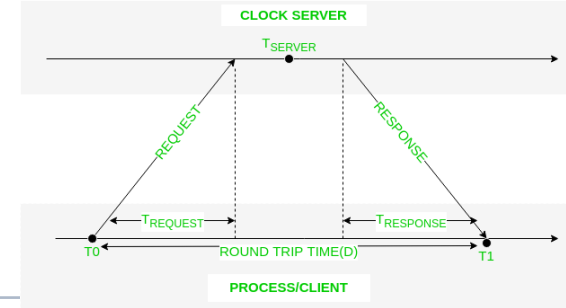
establecimiento
de tiempo

exclusión mutua

elección
coordinador



sincronización de relojes



Método de Cristian (para **SD** síncronos y sincronización externa)

Un servidor de tiempo S recibe señales UTC

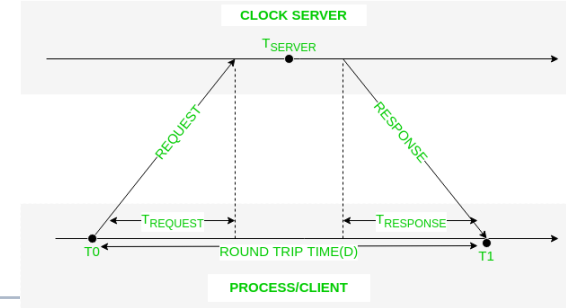
introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

sincronización de relojes



Método de Cristian (para SD síncronos y sincronización externa)

Un servidor de tiempo S recibe señales UTC

- El proceso p solicita el tiempo en un mensaje m_r y recibe t en m_t de S

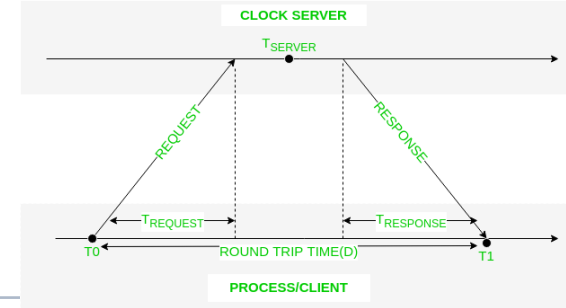
introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

sincronización de relojes



Método de Cristian (para SD síncronos y sincronización externa)

Un servidor de tiempo S recibe señales UTC

- El proceso p solicita el tiempo en un mensaje m_r y recibe t en m_t de S
- p establece su tiempo a $t + T_{round}/2$ [T_{round} es el tiempo de ida y vuelta]

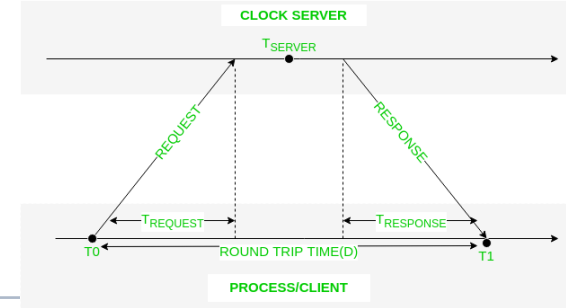
introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

sincronización de relojes



Método de Cristian (para SD síncronos y sincronización externa)

Un servidor de tiempo S recibe señales UTC

- El proceso p solicita el tiempo en un mensaje m_r y recibe t en m_t de S
- p establece su tiempo a $t + T_{round}/2$ [T_{round} es el tiempo de ida y vuelta]
- Precisión: $\pm (T_{round}/2 - min)$ [min es el mínimo estimado de transmisión]

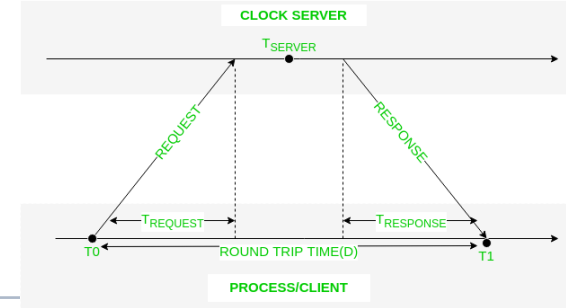
introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

sincronización de relojes



Método de Cristian (para SD síncronos y sincronización externa)

Un servidor de tiempo S recibe señales UTC

- El proceso p solicita el tiempo en un mensaje m_r y recibe t en m_t de S
- p establece su tiempo a $t + T_{round}/2$ [T_{round} es el tiempo de ida y vuelta]
- Precisión: $\pm (T_{round}/2 - min)$ [*min* es el mínimo estimado de transmisión]
 - El momento más temprano en que S pone t en m_t es *min* después de que p enviara m_r .
 - El momento más tardío es *min* antes de que m_t llegue a p
 - El tiempo de S cuando m_t llega está en el rango: $[t + min, t + T_{round} - min]$

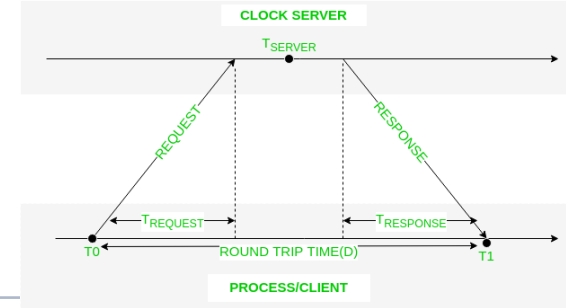
introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

sincronización de relojes



Método de Cristian (para SD síncronos y sincronización externa)

Un servidor de tiempo S recibe señales UTC

- El proceso p solicita el tiempo en un mensaje m_r y recibe t en m_t de S
- p establece su tiempo a $t + T_{round}/2$ [T_{round} es el tiempo de ida y vuelta]
- Precisión: $\pm (T_{round}/2 - min)$ [min es el mínimo estimado de transmisión]
 - El momento más temprano en que S pone t en m_t es min después de que p enviara m_r .
 - El momento más tardío es min antes de que m_t llegue a p
 - El tiempo de S cuando m_t llega está en el rango: $[t + min, t + T_{round} - min]$

Problemas:

- Es probabilístico
- Se soporta mediante un único servidor de tiempo → No tolerante a fallos

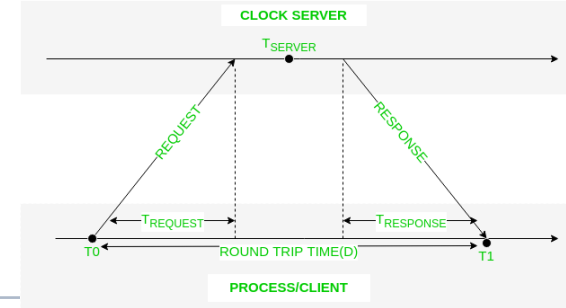
introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

sincronización de relojes



Método de Cristian (para SD síncronos y sincronización externa)

Un servidor de tiempo S recibe señales UTC

- El proceso p solicita el tiempo en un mensaje m_r y recibe t en m_t de S
- p establece su tiempo a $t + T_{round}/2$ [T_{round} es el tiempo de ida y vuelta]
- Precisión: $\pm (T_{round}/2 - min)$ [min es el mínimo estimado de transmisión]
 - El momento más temprano en que S pone t en m_t es min después de que p enviara m_r .
 - El momento más tardío es min antes de que m_t llegue a p
 - El tiempo de S cuando m_t llega está en el rango: $[t + min, t + T_{round} - min]$

Problemas:

- Es probabilístico
- Se soporta mediante un único servidor de tiempo → No tolerante a fallos

Solución a la tolerancia a fallos: tener varios servidores de tiempo y aceptar el primer mensaje válido

introducción

establecimiento
de tiempo

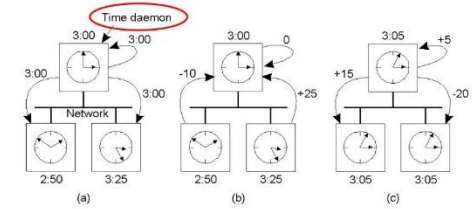
exclusión mutua

elección
coordinador

sincronización de relojes

The Berkeley Algorithm

synchronization **without time server**



- a) The time daemon asks all the other machines for their clock value discrepancies
 - b) The machines answer
 - c) The time daemon tells everyone how to adjust their clock to the average
- no Universal Coordinated Time available

Método de Berkeley (para SD síncronos y sincronización interna)

introducción

establecimiento
de tiempo

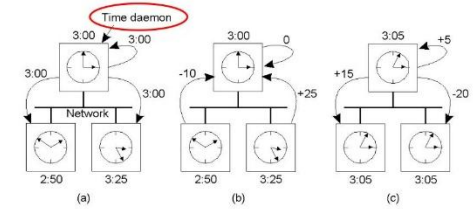
exclusión mutua

elección
coordinador

sincronización de relojes

The Berkeley Algorithm

synchronization **without time server**



- a) The time daemon asks all the other machines for their clock value discrepancies
 - b) The machines answer
 - c) The time daemon tells everyone how to adjust their clock to the average
- no Universal Coordinated Time available

Método de Berkeley (para SD síncronos y sincronización interna)

- Un maestro consulta y recoge valores de reloj del resto de computadores, esclavos

introducción

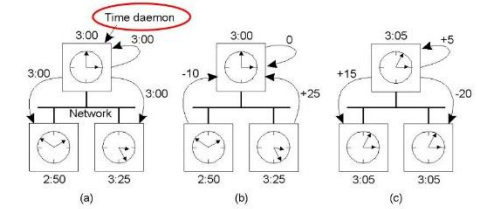
establecimiento
de tiempo

exclusión mutua

elección
coordinador

sincronización de relojes

The Berkeley Algorithm synchronization **without time server**



- a) The time daemon asks all the other machines for their clock value discrepancies
 - b) The machines answer
 - c) The time daemon tells everyone how to adjust their clock to the average
- no Universal Coordinated Time available

Método de Berkeley (para SD síncronos y sincronización interna)

- Un maestro consulta y recoge valores de reloj del resto de computadores, esclavos
- El maestro utiliza los tiempos de ida y vuelta de los mensajes para estimar el valor de los relojes esclavos (similar a **Cristian**)

introducción

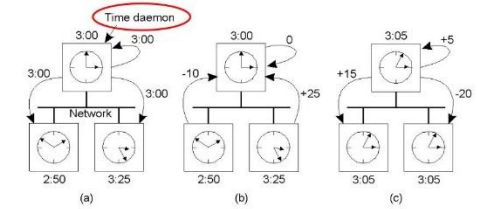
establecimiento
de tiempo

exclusión mutua

elección
coordinador

sincronización de relojes

The Berkeley Algorithm synchronization **without** time server



- a) The time daemon asks all the other machines for their clock value discrepancies
 - b) The machines answer
 - c) The time daemon tells everyone how to adjust their clock to the average
- no Universal Coordinated Time available

Método de Berkeley (para **SD** síncronos y sincronización interna)

- Un maestro consulta y recoge valores de reloj del resto de computadores, esclavos
- El maestro utiliza los tiempos de ida y vuelta de los mensajes para estimar el valor de los relojes esclavos (similar a **Cristian**)
- Promedia los resultados incluyéndose y eliminando cualquier valor que no sea consistente

introducción

establecimiento
de tiempo

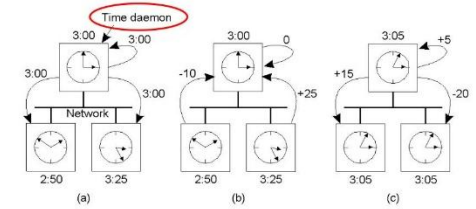
exclusión mutua

elección
coordinador

sincronización de relojes

The Berkeley Algorithm

synchronization **without** time server



- a) The time daemon asks all the other machines for their clock value discrepancies
 - b) The machines answer
 - c) The time daemon tells everyone how to adjust their clock to the average
- no Universal Coordinated Time available

Método de Berkeley (para SD síncronos y sincronización interna)

- Un maestro consulta y recoge valores de reloj del resto de computadores, esclavos
- El maestro utiliza los tiempos de ida y vuelta de los mensajes para estimar el valor de los relojes esclavos (similar a **Cristian**)
- Promedia los resultados incluyéndose y eliminando cualquier valor que no sea consistente
- Envía la magnitud de ajuste de cada reloj, puede ser positivo o **negativo**

introducción

establecimiento
de tiempo

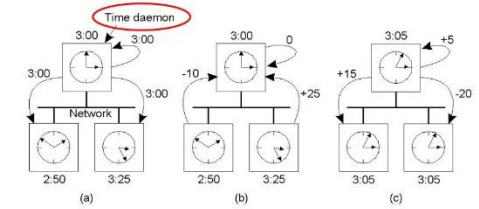
exclusión mutua

elección
coordinador

sincronización de relojes

The Berkeley Algorithm

synchronization **without** time server



- a) The time daemon asks all the other machines for their clock value discrepancies
 - b) The machines answer
 - c) The time daemon tells everyone how to adjust their clock to the average
- no Universal Coordinated Time available

Método de Berkeley (para SD síncronos y sincronización interna)

- Un maestro consulta y recoge valores de reloj del resto de computadores, esclavos
- El maestro utiliza los tiempos de ida y vuelta de los mensajes para estimar el valor de los relojes esclavos (similar a **Cristian**)
- Promedia los resultados incluyéndose y eliminando cualquier valor que no sea consistente
- Envía la magnitud de ajuste de cada reloj, puede ser positivo o **negativo**

Experimentos:

- 15 computadoras, tiempo de sincronización 20-25 milisegs. Tasa de deriva de relojes locales $< 2 \times 10^{-5}$

introducción

establecimiento
de tiempo

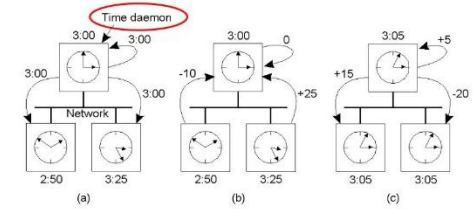
exclusión mutua

elección
coordinador

sincronización de relojes

The Berkeley Algorithm

synchronization **without** time server



- a) The time daemon asks all the other machines for their clock value discrepancies
 - b) The machines answer
 - c) The time daemon tells everyone how to adjust their clock to the average
- no Universal Coordinated Time available

Método de Berkeley (para SD síncronos y sincronización interna)

- Un maestro consulta y recoge valores de reloj del resto de computadores, esclavos
- El maestro utiliza los tiempos de ida y vuelta de los mensajes para estimar el valor de los relojes esclavos (similar a **Cristian**)
- Promedia los resultados incluyéndose y eliminando cualquier valor que no sea consistente
- Envía la magnitud de ajuste de cada reloj, puede ser positivo o **negativo**

Experimentos:

- 15 computadoras, tiempo de sincronización 20-25 milisegs. Tasa de deriva de relojes locales $< 2 \times 10^{-5}$

Problema: si se produce el fallo del maestro

introducción

establecimiento
de tiempo

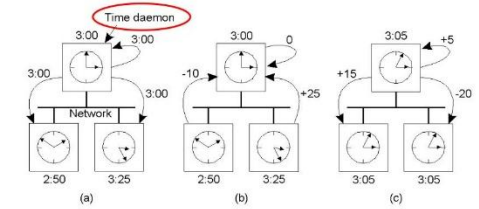
exclusión mutua

elección
coordinador

sincronización de relojes

The Berkeley Algorithm

synchronization **without** time server



- a) The time daemon asks all the other machines for their clock value discrepancies
 - b) The machines answer
 - c) The time daemon tells everyone how to adjust their clock to the average
- no Universal Coordinated Time available

Método de Berkeley (para SD síncronos y sincronización interna)

- Un maestro consulta y recoge valores de reloj del resto de computadores, esclavos
- El maestro utiliza los tiempos de ida y vuelta de los mensajes para estimar el valor de los relojes esclavos (similar a **Cristian**)
- Promedia los resultados incluyéndose y eliminando cualquier valor que no sea consistente
- Envía la magnitud de ajuste de cada reloj, puede ser positivo o **negativo**

Experimentos:

- 15 computadoras, tiempo de sincronización 20-25 milisegs. Tasa de deriva de relojes locales $< 2 \times 10^{-5}$

Problema: si se produce el fallo del maestro

Solución: activar algoritmo de elección para establecer un nuevo maestro

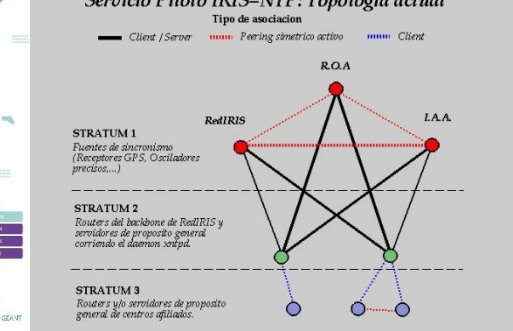
introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

sincronización de relojes



Protocolo de Tiempo de Red – NTP (para SD asíncronos y sincronización externa)

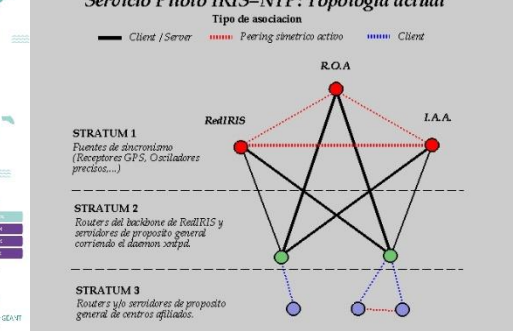
introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

sincronización de relojes



Protocolo de Tiempo de Red – NTP (para SD asíncronos y sincronización externa)

Servicio de sincronización de tiempo en Internet. Sincroniza a los clientes con UTC

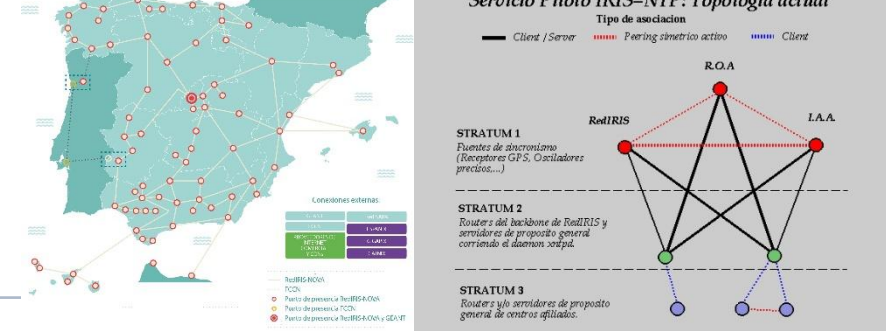
introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

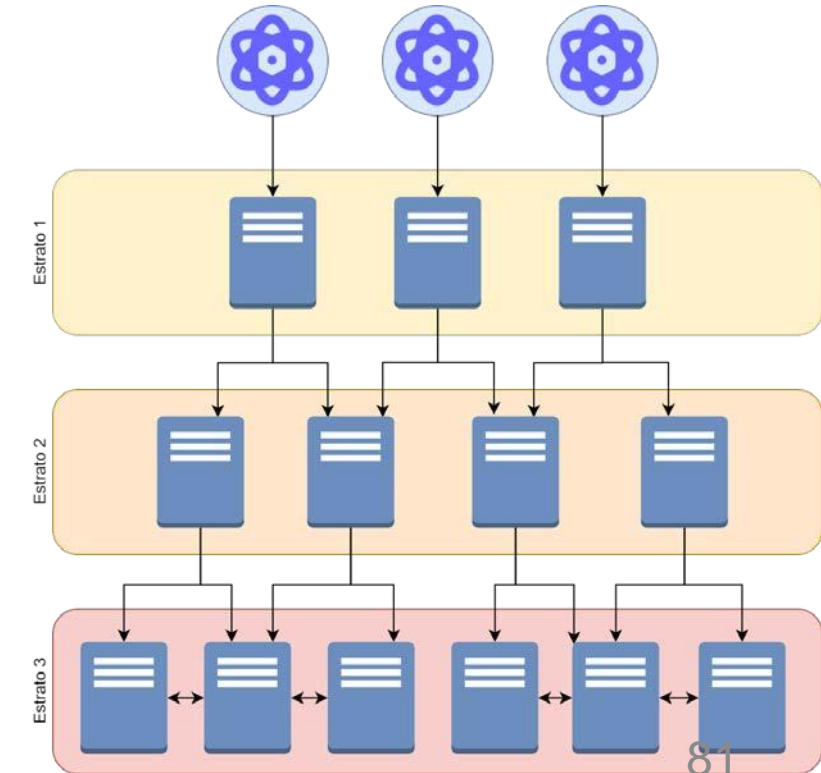
sincronización de relojes



Protocolo de Tiempo de Red – NTP (para SD asíncronos y sincronización externa)

Servicio de sincronización de tiempo en Internet. Sincroniza a los clientes con UTC

Es un servicio fiable, redundante, reconfigurable si alguno cae, escalable y permite autenticación de las fuentes de tiempo



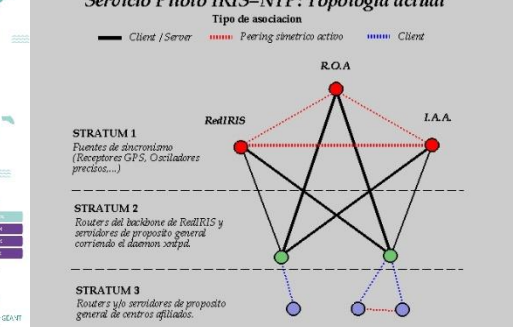
introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

sincronización de relojes



Protocolo de Tiempo de Red – NTP (para SD asíncronos y sincronización externa)

Servicio de sincronización de tiempo en Internet. Sincroniza a los clientes con UTC

Es un servicio fiable, redundante, reconfigurable si alguno cae, escalable y permite autenticación de las fuentes de tiempo

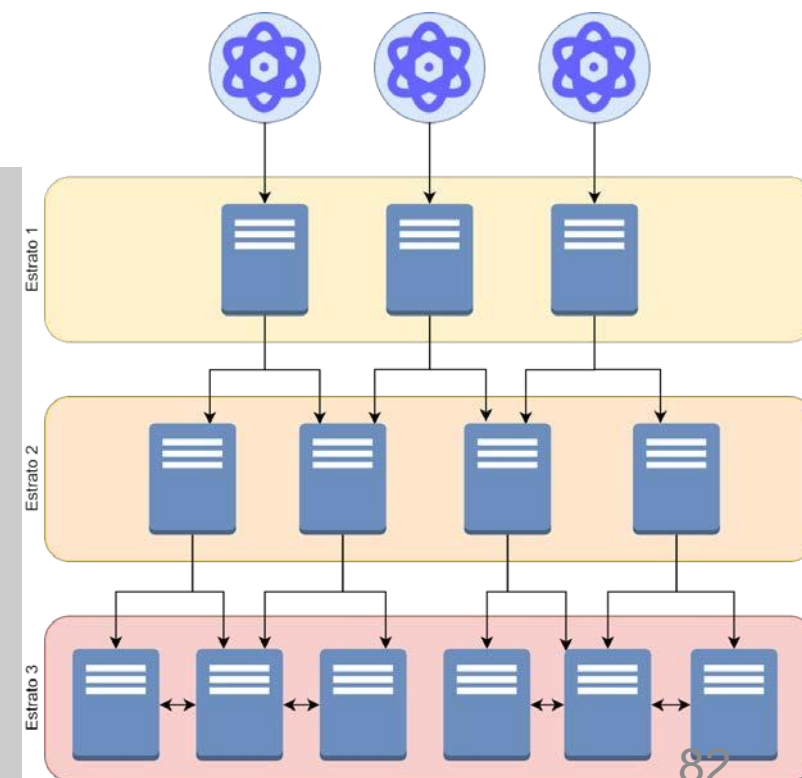
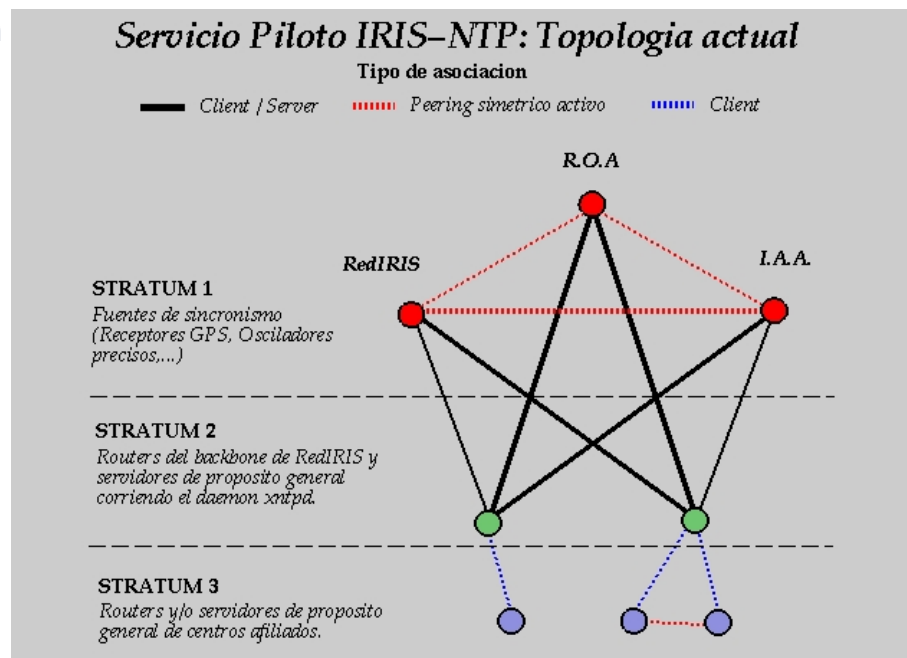
Este sistema conforma una jerarquía lógica llamada **subred de sincronización**

introducción

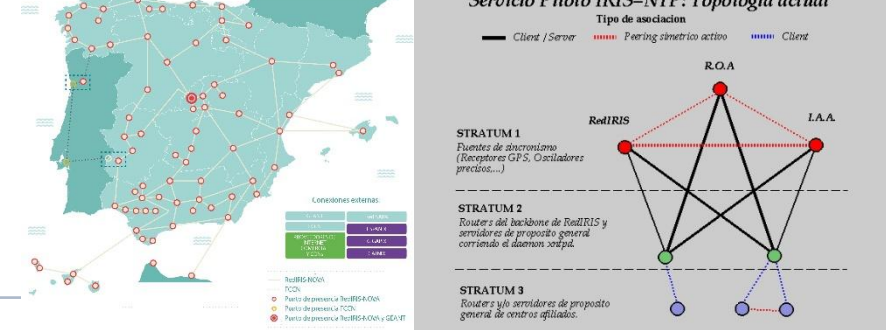
establecimiento de tiempo

exclusión mutua

elección coordinador



sincronización de relojes



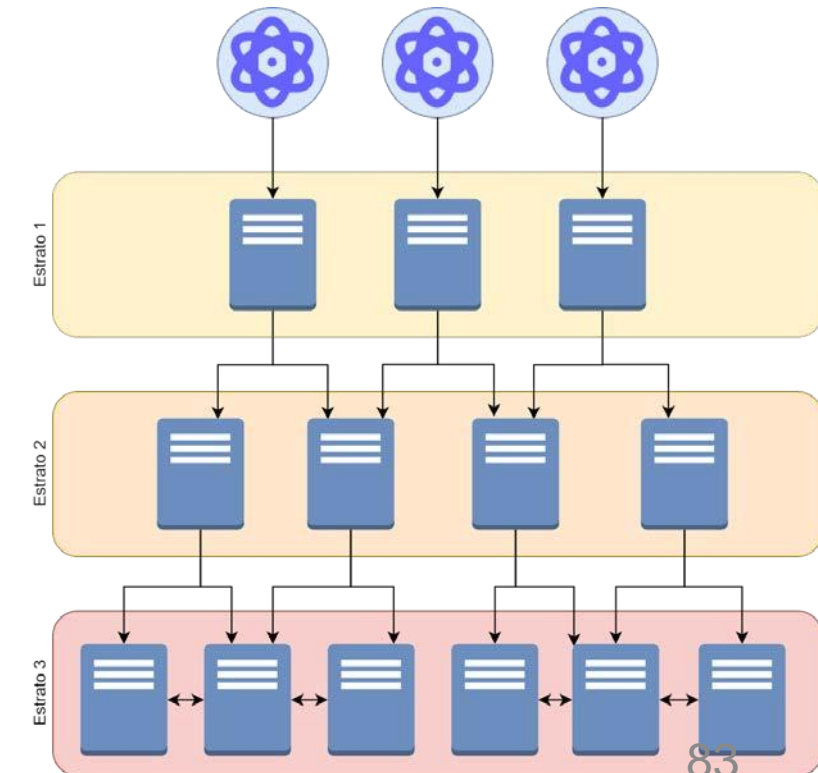
Protocolo de Tiempo de Red – NTP (para SD asíncronos y sincronización externa)

Servicio de sincronización de tiempo en Internet. Sincroniza a los clientes con UTC

Es un servicio fiable, redundante, reconfigurable si alguno cae, escalable y permite autenticación de las fuentes de tiempo

Este sistema conforma una jerarquía lógica llamada **subred de sincronización**

- Los relojes UTC están en el estrato 0.
- Los servidores primarios ocupan el estrato 1 y están conectados directamente a fuentes UTC
- Los servidores secundarios están sincronizados directamente con los primarios
- Los servidores hoja o de nivel más bajo se ejecutan en las máquinas de trabajo de los usuarios



introducción

establecimiento de tiempo

exclusión mutua

elección coordinador

sincronización de relojes

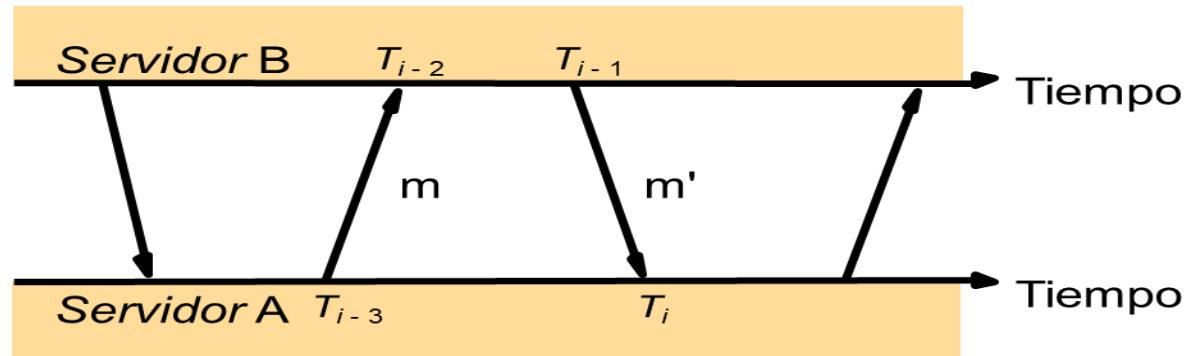
Protocolo de Tiempo de Red – NTP: Funcionamiento

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador



sincronización de relojes

Protocolo de Tiempo de Red – NTP: Funcionamiento

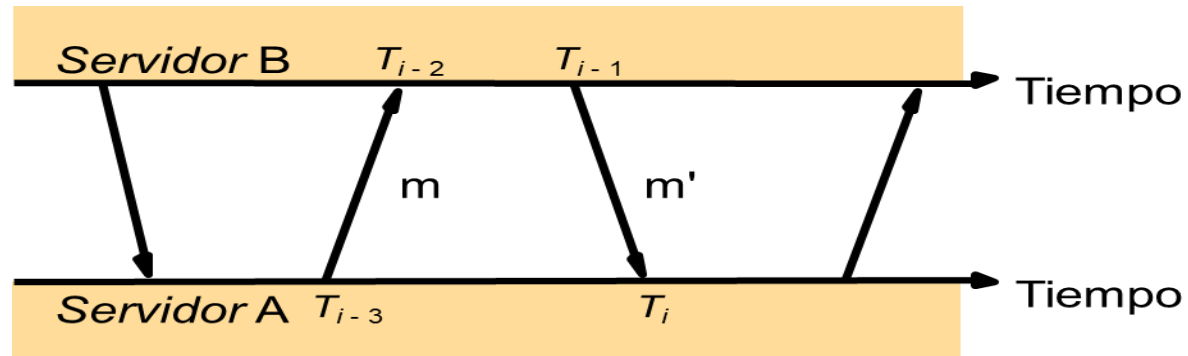
Intercambio de mensajes entre pares de servidores

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador



sincronización de relojes

introducción

**establecimiento
de tiempo**

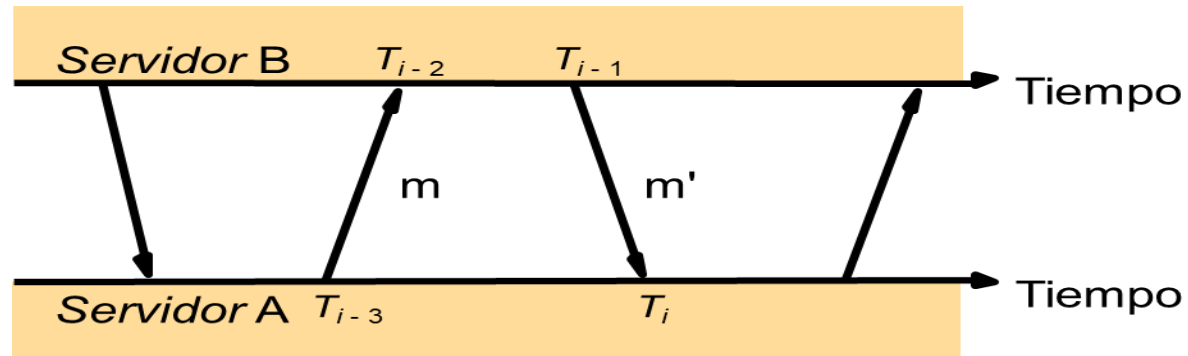
exclusión mutua

elección
coordinador

Protocolo de Tiempo de Red – NTP: Funcionamiento

Intercambio de mensajes entre pares de servidores

- Se usan mensajes UDP
- Cada mensaje lleva marcas de tiempo de los eventos recientes:
 - Tiempos locales de Envío y Recepción del mensaje anterior m
 - Tiempo local de envío del mensaje actual m'



sincronización de relojes

introducción

**establecimiento
de tiempo**

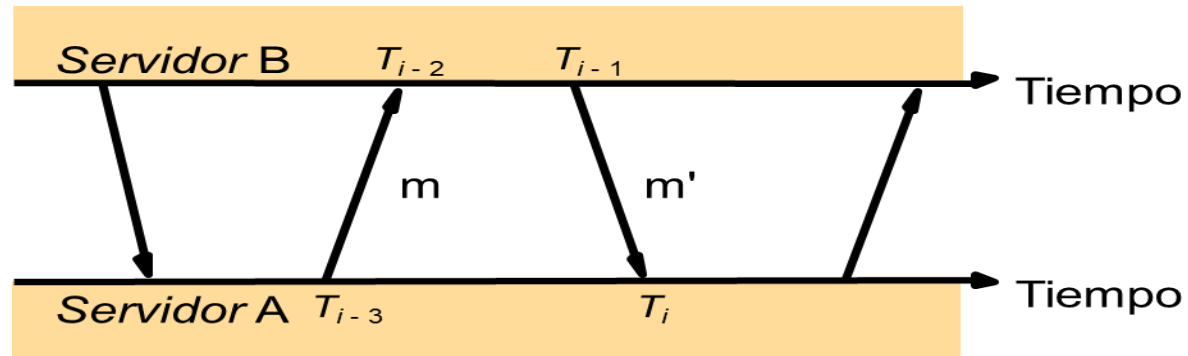
exclusión mutua

elección
coordinador

Protocolo de Tiempo de Red – NTP: Funcionamiento

Intercambio de mensajes entre pares de servidores

- Se usan mensajes UDP
- Cada mensaje lleva marcas de tiempo de los eventos recientes:
 - Tiempos locales de Envío y Recepción del mensaje anterior m
 - Tiempo local de envío del mensaje actual m'
- El receptor anota el tiempo local T_i cuando al recibir (o sea, tenemos $T_{i-3}, T_{i-2}, T_{i-1}, T_i$)



sincronización de relojes

introducción

**establecimiento
de tiempo**

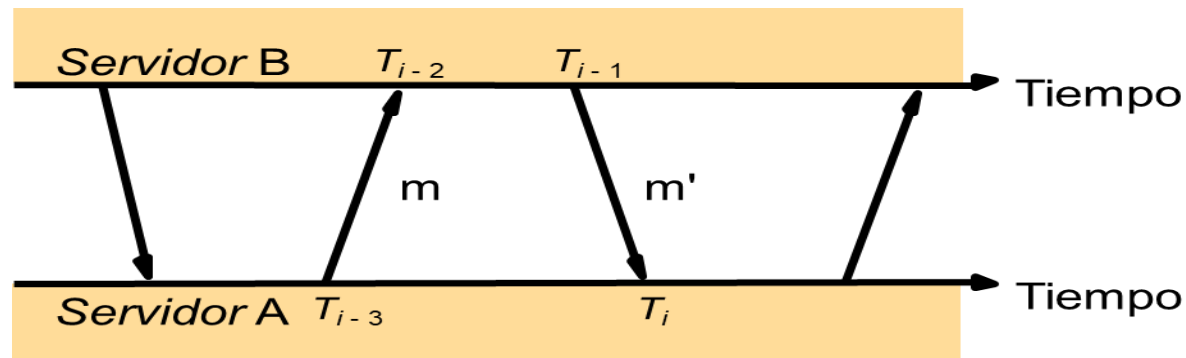
exclusión mutua

elección
coordinador

Protocolo de Tiempo de Red – NTP: Funcionamiento

Intercambio de mensajes entre pares de servidores

- Se usan mensajes UDP
- Cada mensaje lleva marcas de tiempo de los eventos recientes:
 - Tiempos locales de Envío y Recepción del mensaje anterior m
 - Tiempo local de envío del mensaje actual m'
- El receptor anota el tiempo local T_i cuando al recibir (o sea, tenemos $T_{i-3}, T_{i-2}, T_{i-1}, T_i$)
- Puede haber un retraso entre la llegada de un mensaje y el envío del siguiente y se pueden perder mensajes...
 - Pares de servidores se intercambian mensajes conteniendo información de tiempo
 - Usado en los casos en que se necesita muy alta precisión (p.ej. en primeros niveles)



sincronización de relojes

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Protocolo de Tiempo de Red – NTP: Precisión y calidad de los valores estimados

- Para cada par de mensajes entre servidores, NTP estima una compensación \mathbf{o} , entre los dos relojes y un retardo $\mathbf{d_i}$ (tiempo total de transmisión para los dos mensajes \mathbf{t} y $\mathbf{t'}$)

$$T_{i-2} = T_{i-3} + t + o \text{ y } T_i = T_{i-1} + t' - o$$

- Sumando las ecuaciones:

$$d_i = t + t' = T_{i-2} - T_{i-3} + T_i - T_{i-1}$$

- Restando las ecuaciones:

$$o = o_i + (t' - t)/2 \text{ donde } o_i = (T_{i-2} - T_{i-3} - T_i + T_{i-1})/2$$

- Como $t, t' > 0$ se puede ver que:

$$o_i - d_i/2 \leq o \leq o_i + d_i/2$$

por tanto $\mathbf{o_i}$ es una estimación de la deriva y $\mathbf{d_i}$ es una medida de la precisión

Los servidores NTP mantienen pares del tipo $\langle \mathbf{o_i}, \mathbf{d_i} \rangle$, estimando la fiabilidad de las variaciones y permitiendo cambiar el propio par

Ej.: precisión del orden de decenas de mseg. sobre Internet y de 1 mseg. sobre LAN

sincronización de relojes

Protocolo de Tiempo de Red – NTP: Sincronización de servidores

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

sincronización de relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

Protocolo de Tiempo de Red – NTP: Sincronización de servidores

- La subred de sincronización se puede reconfigurar si se produce un fallo:
 - un primario que pierde su conexión con UTC puede pasar a secundario
 - un secundario que pierde a su primario puede elegir otro primario

sincronización de relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

Protocolo de Tiempo de Red – NTP: Sincronización de servidores

- La subred de sincronización se puede reconfigurar si se produce un fallo:
 - un primario que pierde su conexión con UTC puede pasar a secundario
 - un secundario que pierde a su primario puede elegir otro primario
- Modos de sincronización:

sincronización de relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

Protocolo de Tiempo de Red – NTP: Sincronización de servidores

- La subred de sincronización se puede reconfigurar si se produce un fallo:
 - un primario que pierde su conexión con UTC puede pasar a secundario
 - un secundario que pierde a su primario puede elegir otro primario
- Modos de sincronización:
 - Multidifusión (multicast)
 - En LAN de alta velocidad. Un servidor reparte el tiempo al resto que establecen su tiempo asumiendo un retraso de transmisión (no preciso)

sincronización de relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

Protocolo de Tiempo de Red – NTP: Sincronización de servidores

- La subred de sincronización se puede reconfigurar si se produce un fallo:
 - un primario que pierde su conexión con UTC puede pasar a secundario
 - un secundario que pierde a su primario puede elegir otro primario
- Modos de sincronización:
 - Multidifusión (multicast)
 - En LAN de alta velocidad. Un servidor reparte el tiempo al resto que establecen su tiempo asumiendo un retraso de transmisión (no preciso)
 - Llamada a procedimiento
 - Similar a de Cristian. El servidor acepta peticiones. Precisión más alta

sincronización de relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

Protocolo de Tiempo de Red – NTP: Sincronización de servidores

- La subred de sincronización se puede reconfigurar si se produce un fallo:
 - un primario que pierde su conexión con UTC puede pasar a secundario
 - un secundario que pierde a su primario puede elegir otro primario
- Modos de sincronización:
 - Multidifusión (multicast)
 - En LAN de alta velocidad. Un servidor reparte el tiempo al resto que establecen su tiempo asumiendo un retraso de transmisión (no preciso)
 - Llamada a procedimiento
 - Similar a de Cristian. El servidor acepta peticiones. Precisión más alta
 - Simétrica
 - Pares de servidores se intercambian mensajes conteniendo información de tiempo
 - Usado en los casos en que se necesita muy alta precisión (p.e. en primeros niveles)

sincronización de relojes

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

Protocolo de Tiempo de Red – NTP: Ejemplo en Java de obtención de la fecha y hora

```
import java.io.*;
import java.net.*;
public class AskTime {

    public static void main(String args[]) throws Exception {

        String machine = "time-nw.nist.gov";
        final int daytimeport = 13;
        Socket so = new Socket(machine, daytimeport);
        BufferedReader br =
            new BufferedReader( new InputStreamReader( so.getInputStream() ) );
        String timestamp = br.readLine();
        System.out.println( "Según el servidor UTC:" + machine + " la fecha y hora es: " + timestamp );
    }
}
```


relojes lógicos

introducción

**establecimiento
de tiempo**

exclusión mutua

elección
coordinador

- No se sincronizan relojes físicos, si no que se pretende establecer una ordenación coherente de los eventos del sistema distribuido
 - Relación de orden parcial → “suceder antes”
- Se basan en el establecimiento de contadores software que garantizan que el comportamiento será siempre monótono creciente (alternativa a la corrección de un reloj)
- Estudiaremos dos tipos de relojes lógicos:
 - Relojes de Lamport
 - Relojes vectoriales

relojes de Lamport

introducción

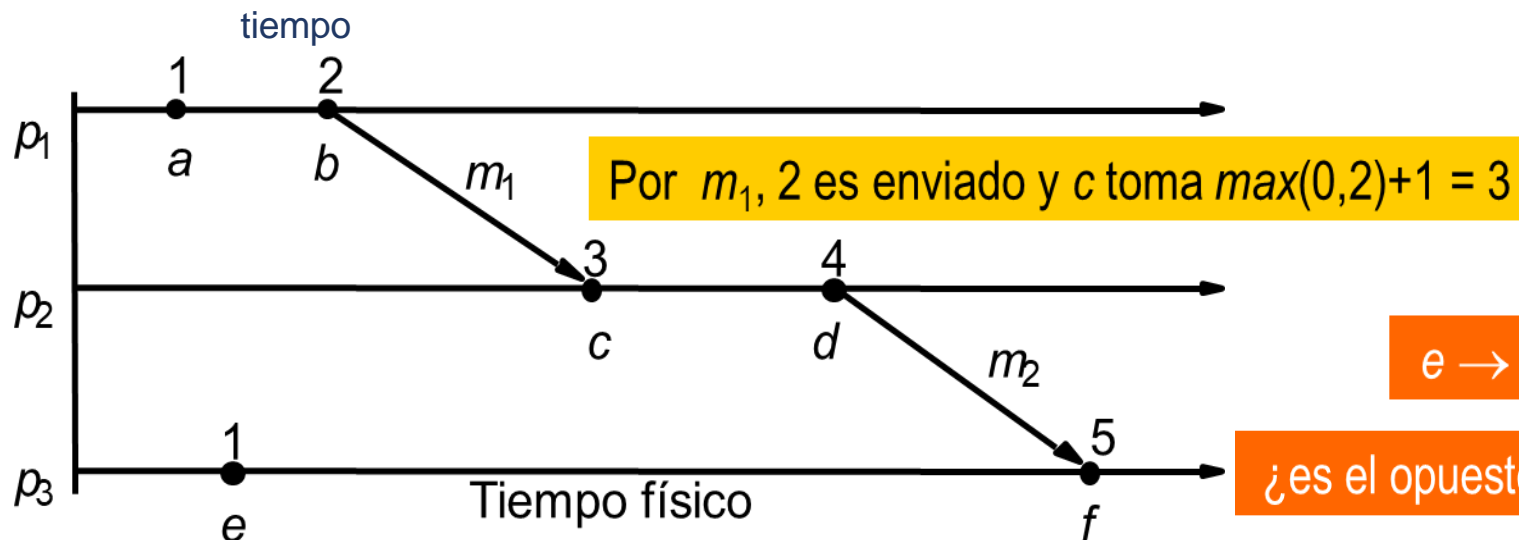
establecimiento
de tiempo

exclusión mutua

elección
coordinador

Desarrollados por Lamport en 1978

- Cada proceso p_i tiene su reloj lógico (L_i) inicializado a 0 que se utiliza para fijar las marcas temporales a los eventos según las siguientes reglas:
 - **R1:** L_i se incrementa en 1 antes de cada evento propio de p_i
 - **R2:** cuando p_i **envía** (m), adjunta al mensaje el valor propio $t = L_i$
 - **R3:** cuando p_j **recibe** (m, t) establece $L_j := \max(L_j, t)$ y aplica **R1**, antes de establecer la nueva marca de



$e \rightarrow e'$ implica $L(e) < L(e')$

¿es el opuesto cierto? $L(e) < L(e') \rightarrow (e \rightarrow e')$

relojes vectoriales

introducción

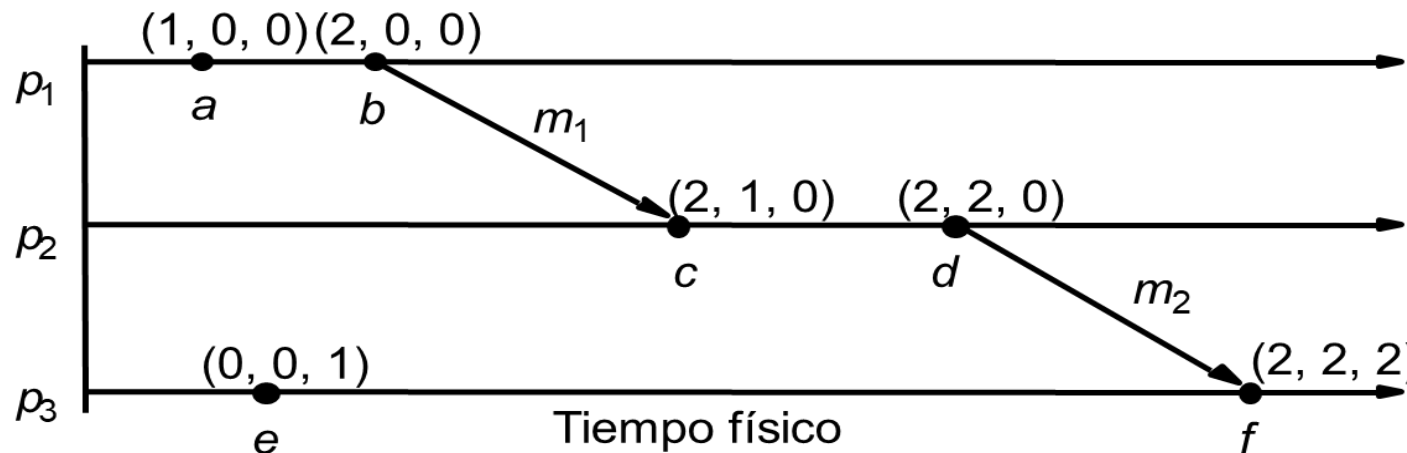
establecimiento
de tiempo

exclusión mutua

elección
coordinador

Mattern y Fidge (1989-91) los desarrollan para superar la deficiencia de los relojes lógicos de Lamport: $L(e) < L(e')$ no implica $e \rightarrow e'$

- Cada proceso p_i tiene un vector V_i con N componentes, tantos como procesos en el sistema. Las reglas de actualización son las siguientes:
 - **R1:** Antes de marcar un nuevo evento p_i incrementa $V_i[i] := V_i[i] + 1$
 - **R2:** cuando p_i **envía** (m), adjunta al mensaje el vector propio $t = V_i$
 - **R3:** cuando p_j **recibe** (m, t) establece establece $V_i[j] := \max(V_i[j], t[j])$ $j = 1, 2, \dots, N$ y se aplica **R1**



$$\begin{aligned} V = V' &\Leftrightarrow V[j] = V'[j] \quad j=1, 2, \dots, N \\ V \leq V' &\Leftrightarrow V[j] \leq V'[j] \quad j=1, 2, \dots, N \\ V < V' &\Leftrightarrow V \leq V' \wedge V \neq V' \end{aligned}$$

- $V(e) < V(e')$ implica $e \rightarrow e'$
- c y e paralelos, ni $V(c) \leq V(e)$ ni $V(e) \leq V(c)$



Grado en Ingeniería Informática

Sistemas distribuidos

Sincronización de tiempo y coordinación

Coulouris: Distributed systems Ch15

Departamento de Tecnología Informática y Computación

Curso 2023 - 2024

coordinación distribuida: introducción

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

coordinación distribuida: introducción

- Los procesos distribuidos necesitan a menudo coordinar sus actividades

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador



coordinación distribuida: introducción

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

- Los procesos distribuidos necesitan a menudo coordinar sus actividades
 - sistemas síncronos y asíncronos
 - tolerancia a fallos
 - exclusión mutua de los procesos distribuidos
 - ejemplo: reservas de billetes de avión

coordinación distribuida: introducción

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

- Los procesos distribuidos necesitan a menudo coordinar sus actividades
 - sistemas síncronos y asíncronos
 - tolerancia a fallos
 - exclusión mutua de los procesos distribuidos
 - ejemplo: reservas de billetes de avión
- En los SD para solucionar el problema de la exclusión mutua, no se pueden utilizar:
 - ni variables compartidas
 - ni facilidades dadas por un único núcleo central

coordinación distribuida: introducción

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

- Los procesos distribuidos necesitan a menudo coordinar sus actividades
 - sistemas síncronos y asíncronos
 - tolerancia a fallos
 - exclusión mutua de los procesos distribuidos
 - ejemplo: reservas de billetes de avión
- En los SD para solucionar el problema de la exclusión mutua, no se pueden utilizar:
 - ni variables compartidas
 - ni facilidades dadas por un único núcleo central

→ soluciones basadas en el **paso de mensajes**

coordinación distribuida: introducción

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

coordinación distribuida: introducción

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

- Dado un conjunto de procesos en un sistema distribuido se necesita:
 - Coordinar sus acciones
 - Llegar a un acuerdo en uno o más valores

coordinación distribuida: introducción

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

- Dado un conjunto de procesos en un sistema distribuido se necesita:
 - Coordinar sus acciones
 - Llegar a un acuerdo en uno o más valores
- Formas de coordinación y acuerdo:

coordinación distribuida: introducción

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

- Dado un conjunto de procesos en un sistema distribuido se necesita:
 - Coordinar sus acciones
 - Llegar a un acuerdo en uno o más valores
- Formas de coordinación y acuerdo:
 - Acceso a recursos: exclusión mutua distribuida
 - Selección de valores: algoritmos de elección
 - Comunicación distribuida: algoritmos de multidifusión
 - Toma de decisiones: algoritmos de consenso

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

algoritmos de exclusión mutua

Los algoritmos de exclusión mutua distribuida deben cumplir las siguientes propiedades:

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Los algoritmos de exclusión mutua distribuida deben cumplir las siguientes propiedades:

- EM1: **Seguridad** → en todo momento, como máximo hay un solo proceso ejecutando la región crítica

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Los algoritmos de exclusión mutua distribuida deben cumplir las siguientes propiedades:

- EM1: **Seguridad** → en todo momento, como máximo hay un solo proceso ejecutando la región crítica
- EM2: **Pervivencia** → a todo proceso que lo solicita se le concede la entrada/salida en la región crítica en algún momento:
 - Debe evitar el abrazo mortal (*deadlock*) y la inanición (*starvation*)

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Los algoritmos de exclusión mutua distribuida deben cumplir las siguientes propiedades:

- EM1: **Seguridad** → en todo momento, como máximo hay un solo proceso ejecutando la región crítica
- EM2: **Pervivencia** → a todo proceso que lo solicita se le concede la entrada/salida en la región crítica en algún momento:
 - Debe evitar el abrazo mortal (*deadlock*) y la inanición (*starvation*)
- EM3: **Ordenación** → la entrada en la región crítica debe concederse según la relación “suceder – antes”

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

algoritmos de exclusión mutua

Algoritmo basado en servidor central

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Algoritmo basado en servidor central

- El servidor central concede permisos en forma de **token** que concede acceso a la sección crítica (SC) y al salir de la SC, el proceso devuelve el **token** al servidor

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Algoritmo basado en servidor central

- El servidor central concede permisos en forma de **token** que concede acceso a la sección crítica (SC) y al salir de la SC, el proceso devuelve el **token** al servidor
- Suponiendo que no hay caídas y no se pierden mensajes:
 - se cumplen EM1 y EM2
 - EM3 está asegurada en el orden de llegada de los mensajes al servidor

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Algoritmo basado en servidor central

- El servidor central concede permisos en forma de **token** que concede acceso a la sección crítica (SC) y al salir de la SC, el proceso devuelve el **token** al servidor
- Suponiendo que no hay caídas y no se pierden mensajes:
 - se cumplen EM1 y EM2
 - EM3 está asegurada en el orden de llegada de los mensajes al servidor

Problemas

- todas las solicitudes se envían al servidor (recurso crítico) → cuello de botella
- caída o fallo del servidor → elección de nuevo servidor → EM3 no asegurada
- caída o fallo de un proceso en la SC → se detecta con temporizadores → se requisa el *token*

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Algoritmo basado en servidor central

- El servidor central concede permisos en forma de **token** que concede acceso a la sección crítica (SC) y al salir de la SC, el proceso devuelve el **token** al servidor
- Suponiendo que no hay caídas y no se pierden mensajes:
 - se cumplen EM1 y EM2
 - EM3 está asegurada en el orden de llegada de los mensajes al servidor

Problemas

- todas las solicitudes se envían al servidor (recurso crítico) → cuello de botella
- caída o fallo del servidor → elección de nuevo servidor → EM3 no asegurada
- caída o fallo de un proceso en la SC → se detecta con temporizadores → se requisa el *token*

Complejidad del algoritmo

- 2 mensajes para la entrada en la SC
- 1 mensaje para salir de la SC

algoritmos de exclusión mutua

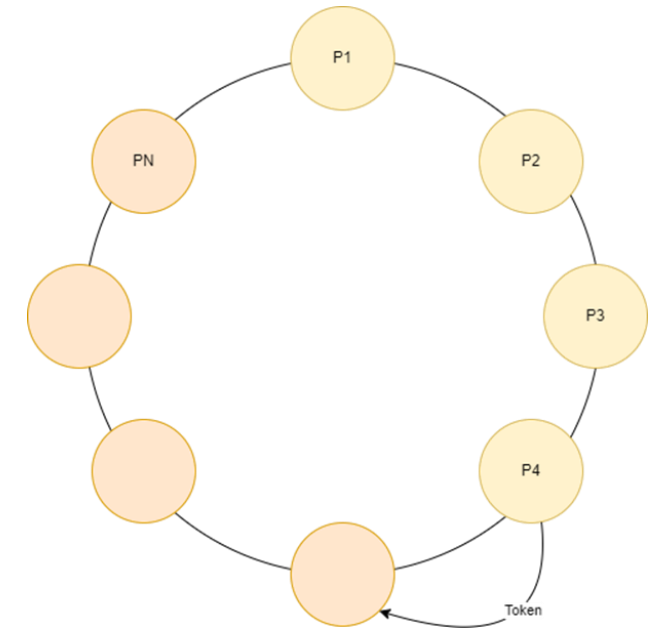
Algoritmo del anillo

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador



algoritmos de exclusión mutua

introducción

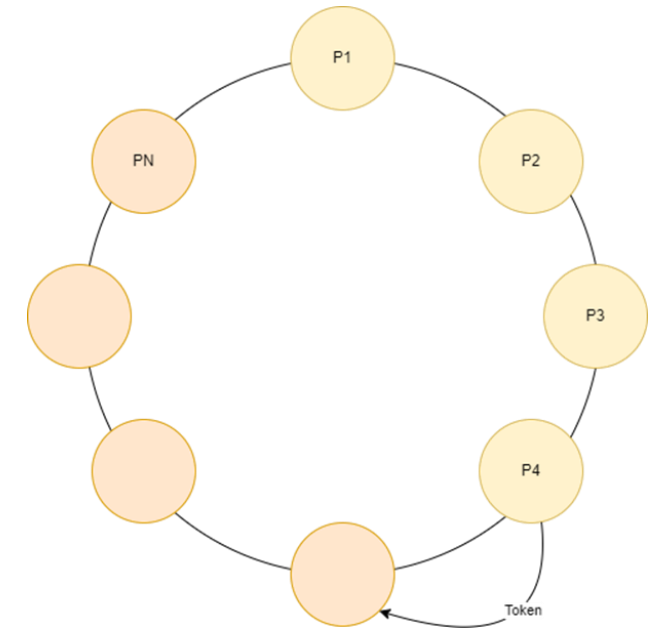
establecimiento
de tiempo

exclusión mutua

elección
coordinador

Algoritmo del anillo

- Se da a cada proceso la dirección de su vecino



algoritmos de exclusión mutua

introducción

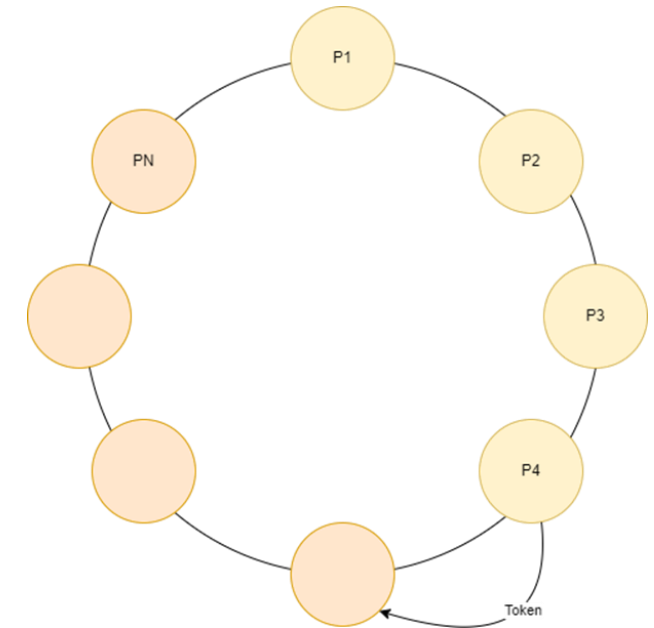
establecimiento
de tiempo

exclusión mutua

elección
coordinador

Algoritmo del anillo

- Se da a cada proceso la dirección de su vecino
- El *token* siempre está circulando por el anillo



algoritmos de exclusión mutua

introducción

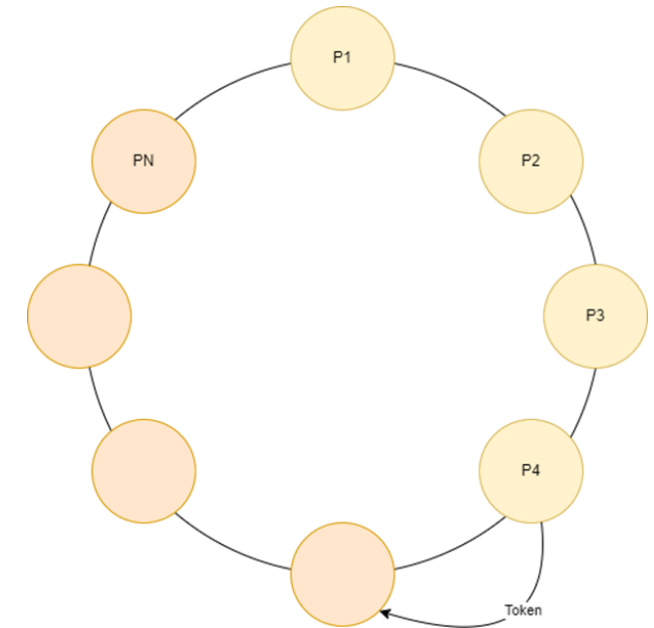
establecimiento
de tiempo

exclusión mutua

elección
coordinador

Algoritmo del anillo

- Se da a cada proceso la dirección de su vecino
- El *token* siempre está circulando por el anillo
- Cuando un proceso recibe el *token*:
 - Si no quiere entrar en la sección crítica lo envía a su vecino
 - Si quiere entrar en la sección crítica lo retiene y al terminar lo devuelve al vecino



algoritmos de exclusión mutua

introducción

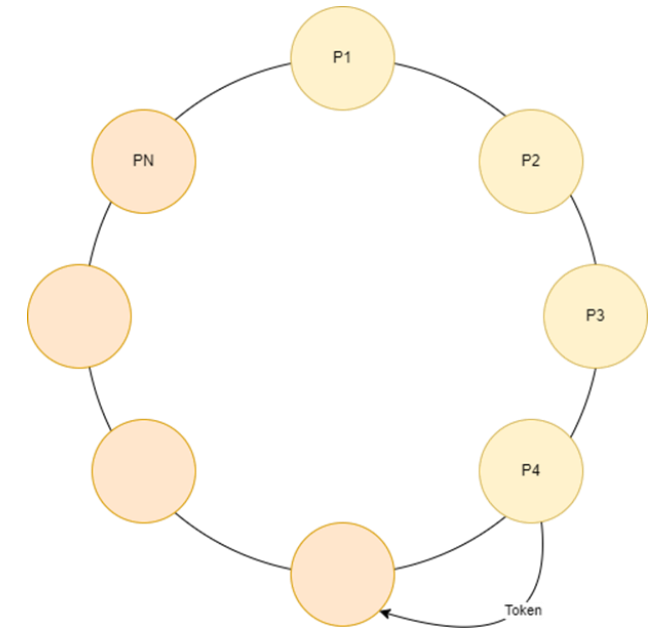
establecimiento
de tiempo

exclusión mutua

elección
coordinador

Algoritmo del anillo

- Se da a cada proceso la dirección de su vecino
- El *token* siempre está circulando por el anillo
- Cuando un proceso recibe el *token*:
 - Si no quiere entrar en la sección crítica lo envía a su vecino
 - Si quiere entrar en la sección crítica lo retiene y al terminar lo devuelve al vecino
- Tolerancia a fallos
 - Pérdida del token: detección y regeneración (temporizadores)
 - Caída de un proceso del anillo: reconfiguración del anillo



algoritmos de exclusión mutua

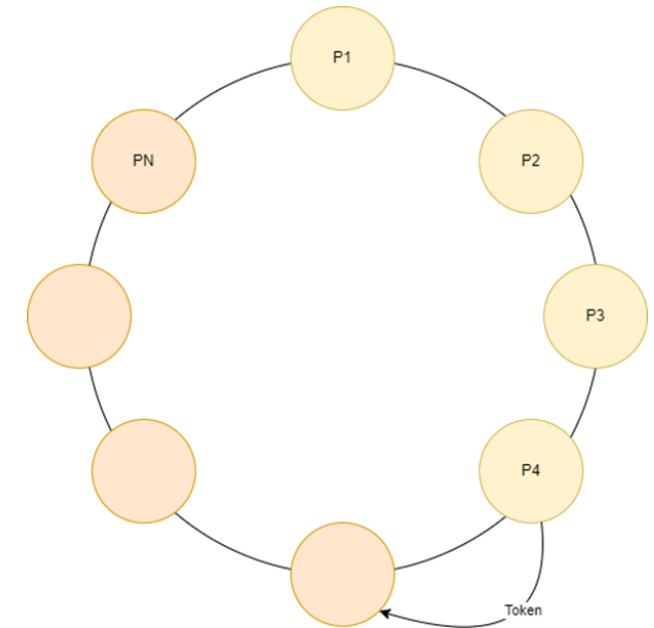
Algoritmo del anillo

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador



algoritmos de exclusión mutua

introducción

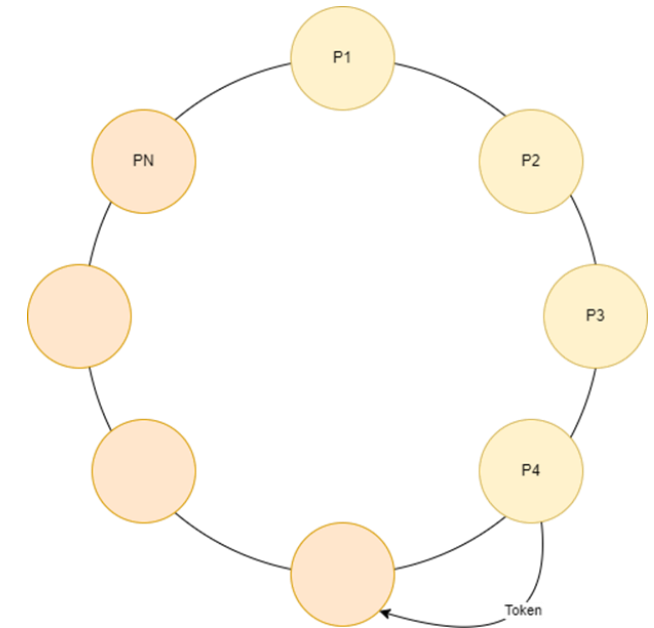
establecimiento
de tiempo

exclusión mutua

elección
coordinador

Algoritmo del anillo

- Se da a cada proceso la dirección de su vecino.



algoritmos de exclusión mutua

introducción

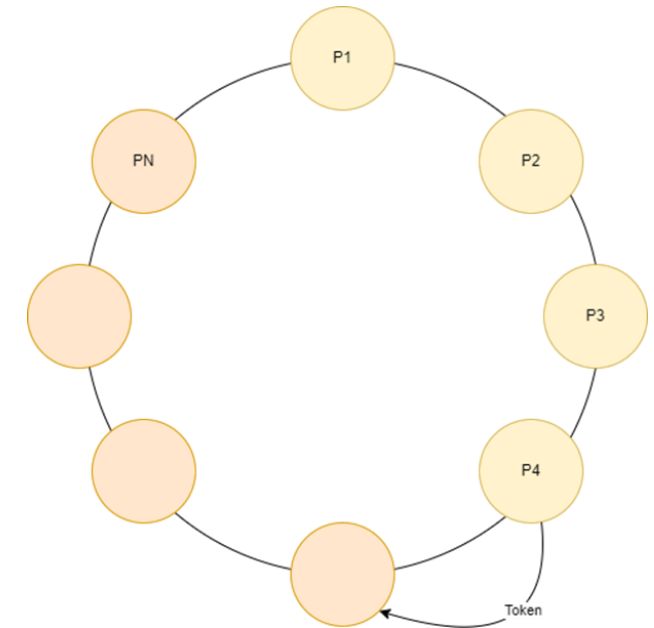
establecimiento
de tiempo

exclusión mutua

elección
coordinador

Algoritmo del anillo

- Se da a cada proceso la dirección de su vecino.
- Problemas
 - se sobrecarga la red aun cuando ningún proceso quiera entrar en la SC
 - si un proceso cae → necesita reconfiguración
 - si además tenía el testigo → elección para regenerar el testigo
 - asegurarse de que el proceso ha caído → varios testigos
 - si se produce desconexión o ruptura de la red



algoritmos de exclusión mutua

introducción

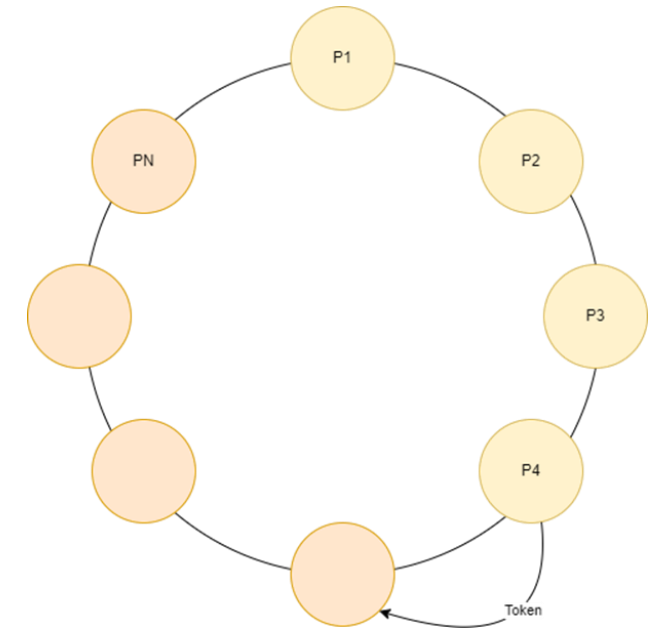
establecimiento
de tiempo

exclusión mutua

elección
coordinador

Algoritmo del anillo

Cumplimiento de propiedades



algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

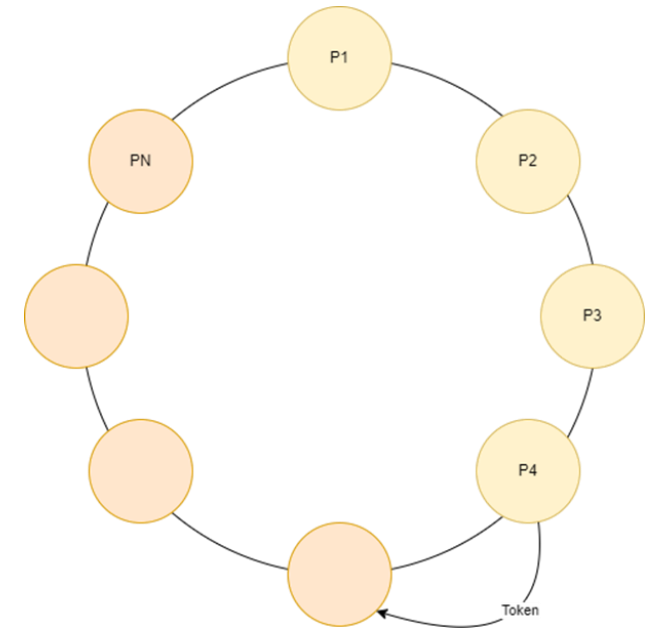
exclusión mutua

elección
coordinador

Algoritmo del anillo

Cumplimiento de propiedades

- EM1: Sí, el servidor se encarga de ello a través de un token único



algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

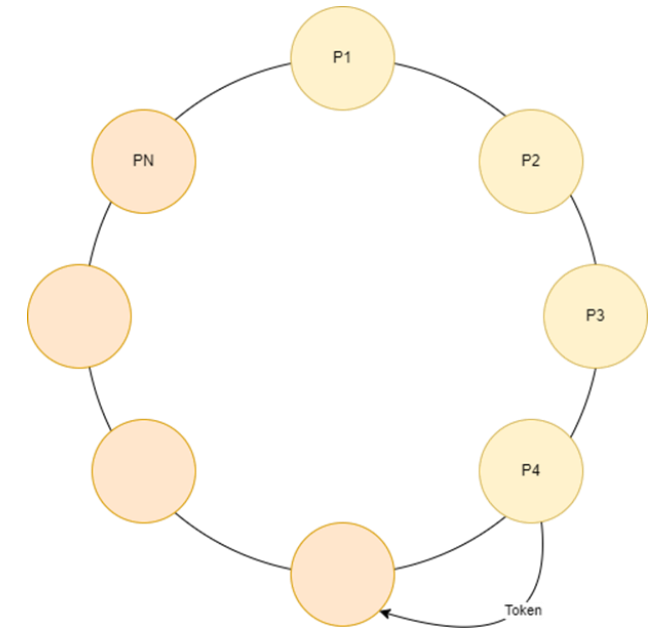
exclusión mutua

elección
coordinador

Algoritmo del anillo

Cumplimiento de propiedades

- EM1: Sí, el servidor se encarga de ello a través de un token único
- EM2: Sí, todas las peticiones se registran en la cola



algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

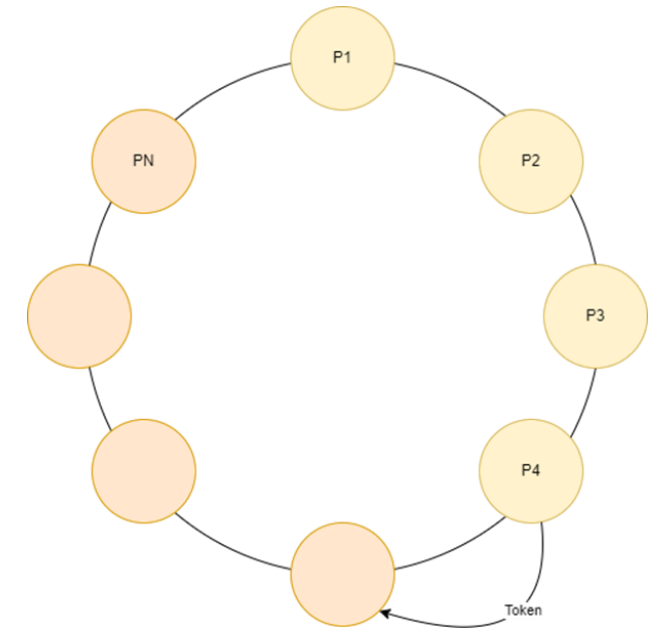
exclusión mutua

elección
coordinador

Algoritmo del anillo

Cumplimiento de propiedades

- EM1: Sí, el servidor se encarga de ello a través de un token único
- EM2: Sí, todas las peticiones se registran en la cola
- EM3: No, cada proceso recibe el *token* en el orden establecido por el anillo



algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

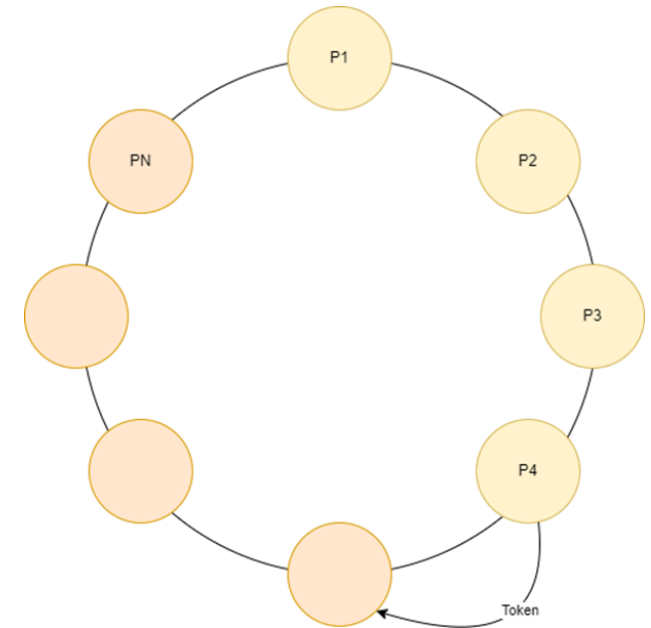
Algoritmo del anillo

Cumplimiento de propiedades

- EM1: Sí, el servidor se encarga de ello a través de un token único
- EM2: Sí, todas las peticiones se registran en la cola
- EM3: No, cada proceso recibe el *token* en el orden establecido por el anillo

Complejidad del algoritmo

- Acceso a SC: entre 0 y N mensajes
- Acceso SC: entre 1 y (N – 1) mensajes



algoritmos de exclusión mutua

Algoritmo Ricart - Agrawala

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

algoritmos de exclusión mutua

Algoritmo Ricart - Agrawala

- Algoritmo basado en relojes lógicos y multidifusión:

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Algoritmo Ricart - Agrawala

- Algoritmo basado en relojes lógicos y multidifusión:
 - Cada proceso conoce la dirección de los demás procesos
 - Basado en relojes de Lamport (cada proceso posee un reloj lógico)
 - Algoritmo descentralizado: evita cuellos de botella

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Algoritmo Ricart - Agrawala

- Algoritmo basado en relojes lógicos y multidifusión:
 - Cada proceso conoce la dirección de los demás procesos
 - Basado en relojes de Lamport (cada proceso posee un reloj lógico)
 - Algoritmo descentralizado: evita cuellos de botella
 - Pretende asegurar EM1, EM2 y EM3 (*¿lo consigue?*)

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Algoritmo Ricart - Agrawala

- Algoritmo basado en relojes lógicos y multidifusión:
 - Cada proceso conoce la dirección de los demás procesos
 - Basado en relojes de Lamport (cada proceso posee un reloj lógico)
 - Algoritmo descentralizado: evita cuellos de botella
 - Pretende asegurar EM1, EM2 y EM3 (*¿lo consigue?*)
- Idea básica del algoritmo

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Algoritmo Ricart - Agrawala

- Algoritmo basado en relojes lógicos y multidifusión:
 - Cada proceso conoce la dirección de los demás procesos
 - Basado en relojes de Lamport (cada proceso posee un reloj lógico)
 - Algoritmo descentralizado: evita cuellos de botella
 - Pretende asegurar EM1, EM2 y EM3 (*¿lo consigue?*)
- Idea básica del algoritmo
 - Cuando un proceso quiere entrar en la sección crítica (SC) → les pregunta a los demás si puede entrar
 - Cuando todos los demás le contesten afirmativamente → entra
 - El acceso a la SC se obtiene a través de un **token**

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Algoritmo Ricart - Agrawala

- Algoritmo basado en relojes lógicos y multidifusión:
 - Cada proceso conoce la dirección de los demás procesos
 - Basado en relojes de Lamport (cada proceso posee un reloj lógico)
 - Algoritmo descentralizado: evita cuellos de botella
 - Pretende asegurar EM1, EM2 y EM3 (*¿lo consigue?*)
- Idea básica del algoritmo
 - Cuando un proceso quiere entrar en la sección crítica (SC) → les pregunta a los demás si puede entrar
 - Cuando todos los demás le contesten afirmativamente → entra
 - El acceso a la SC se obtiene a través de un **token**
- Cada proceso guarda el estado en relación a la SC: liberada, buscada o tomada
 - Cola de solicitudes en cada proceso

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección
coordinador

Algoritmo Ricart - Agrawala

- Algoritmo basado en relojes lógicos y multidifusión:
 - Cada proceso conoce la dirección de los demás procesos
 - Basado en relojes de Lamport (cada proceso posee un reloj lógico)
 - Algoritmo descentralizado: evita cuellos de botella
 - Pretende asegurar EM1, EM2 y EM3 (*¿lo consigue?*)
- Idea básica del algoritmo
 - Cuando un proceso quiere entrar en la sección crítica (SC) → les pregunta a los demás si puede entrar
 - Cuando todos los demás le contesten afirmativamente → entra
 - El acceso a la SC se obtiene a través de un **token**
- Cada proceso guarda el estado en relación a la SC: liberada, buscada o tomada
 - Cola de solicitudes en cada proceso
- Estructura de los mensajes → Tupla $\langle T_i, P_i, SC_i \rangle$ (*en lo sucesivo nos referiremos solo una SC*)

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección y
consenso

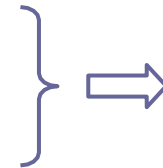
Algoritmo Ricart – Agrawala (pseudocódigo)

Inicialización

- estado SC := LIBERADA

Para entrar en la sección crítica

- estado SC := BUSCADA
- Multitransmite petición a todos los procesos
- $T :=$ marca temporal de la petición propia
- espera hasta que número de respuestas recibidas = $N - 1$
- estado SC := TOMADA;



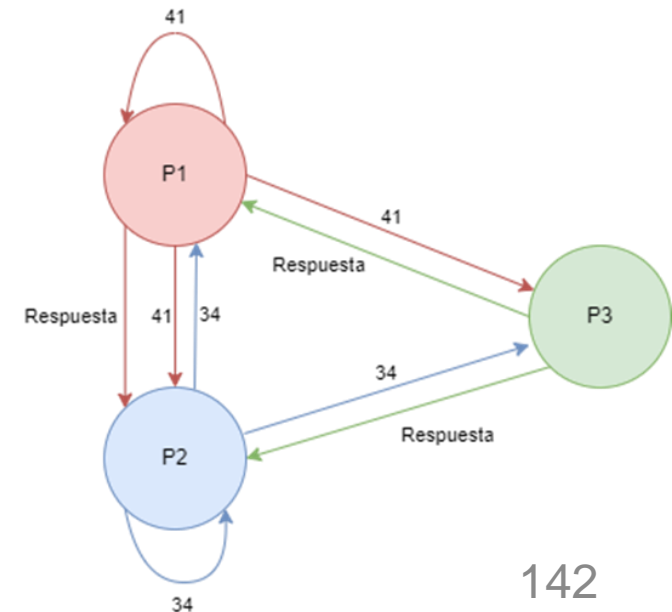
Si el proceso desea entrar en la SC, aplaza el procesamiento del resto de peticiones recibidas

Al recibir una petición $\langle T_i, p_i \rangle$ en el proceso $p_j (i \neq j)$

- **si** (estado = TOMADA o (estado = BUSCADA y $(T, p_i) < (T_i, p_i)$)) **entonces**
pone en la cola la petición por parte de p_i sin responder
- **sino**
responde inmediatamente a p_i
- **fin si**

Para salir de la sección crítica

- estado := LIBERADA
- responde a todas y cada una de las peticiones en la cola propia



algoritmos de exclusión mutua

Algoritmo Ricart - Agrawala

Complejidad del algoritmo

introducción

establecimiento
de tiempo

exclusión mutua

elección y
consenso

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección y
consenso

Algoritmo Ricart - Agrawala

Complejidad del algoritmo

Número de mensajes necesarios para obtener el recurso:

- sin soporte multicast: $2(n-1)$
- con soporte multicast: n

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección y
consenso

Algoritmo Ricart - Agrawala

Complejidad del algoritmo

Número de mensajes necesarios para obtener el recurso:

- sin soporte multicast: $2(n-1)$
- con soporte multicast: n

Problemas:

- Algoritmo más costoso que el del servidor central

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección y
consenso

Algoritmo Ricart - Agrawala

Complejidad del algoritmo

Número de mensajes necesarios para obtener el recurso:

- sin soporte multicast: $2(n-1)$
- con soporte multicast: n

Problemas:

- Algoritmo más costoso que el del servidor central
- Pese a ser algoritmos distribuidos, el fallo de cualquier proceso bloquea el sistema

algoritmos de exclusión mutua

introducción

establecimiento
de tiempo

exclusión mutua

elección y
consenso

Algoritmo Ricart - Agrawala

Complejidad del algoritmo

Número de mensajes necesarios para obtener el recurso:

- sin soporte multicast: $2(n-1)$
- con soporte multicast: n

Problemas:

- Algoritmo más costoso que el del servidor central
- Pese a ser algoritmos distribuidos, el fallo de cualquier proceso bloquea el sistema
- Los procesos implicados reciben y procesan cada solicitud:
 - igual o peor congestión que el servidor central

elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

r

elección de coordinador

- Procedimiento para elegir a un proceso dentro de un grupo para desempeñar un rol determinado

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

- Procedimiento para elegir a un proceso dentro de un grupo para desempeñar un rol determinado
- Se activa tanto en la inicialización del SD, como si no se detecta la presencia de coordinador (¿caída?)

elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

- Procedimiento para elegir a un proceso dentro de un grupo para desempeñar un rol determinado
- Se activa tanto en la inicialización del SD, como si no se detecta la presencia de coordinador (¿caída?)
- Una vez elegido el coordinador, los demás miembros del grupo lo asumen

elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

- Procedimiento para elegir a un proceso dentro de un grupo para desempeñar un rol determinado
- Se activa tanto en la inicialización del SD, como si no se detecta la presencia de coordinador (¿caída?)
- Una vez elegido el coordinador, los demás miembros del grupo lo asumen
- *Ejemplo:* elección del “maestro” en el algoritmo de Berkeley de sincronización de relojes

elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

- Procedimiento para elegir a un proceso dentro de un grupo para desempeñar un rol determinado
- Se activa tanto en la inicialización del SD, como si no se detecta la presencia de coordinador (¿caída?)
- Una vez elegido el coordinador, los demás miembros del grupo lo asumen
- *Ejemplo:* elección del “maestro” en el algoritmo de Berkeley de sincronización de relojes

Los algoritmos de elección deben cumplir las propiedades siguientes:

- E1 → Seguridad: Elección única de coordinador, aunque varios procesos inicien la elección
- E2 → Vivacidad: Si un proceso solicita coordinador, lo obtendrá en algún momento

elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

- Procedimiento para elegir a un proceso dentro de un grupo para desempeñar un rol determinado
- Se activa tanto en la inicialización del SD, como si no se detecta la presencia de coordinador (¿caída?)
- Una vez elegido el coordinador, los demás miembros del grupo lo asumen
- *Ejemplo:* elección del “maestro” en el algoritmo de Berkeley de sincronización de relojes

Los algoritmos de elección deben cumplir las propiedades siguientes:

- E1 → Seguridad: Elección única de coordinador, aunque varios procesos inicien la elección
- E2 → Vivacidad: Si un proceso solicita coordinador, lo obtendrá en algún momento

Tres algoritmos:

- Algoritmo basado en anillo: Chang y Roberts (1979)
- Algoritmo del matón (*bully*): H. García-Molina (1982)
- Algoritmo de invitación: H. García-Molina (1982)

algoritmos de elección de coordinador

Algoritmo de elección basado en **anillo**

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección basado en **anillo**

Objetivo: Elegir como coordinador al proceso con identificador más alto

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección basado en **anillo**

Objetivo: Elegir como coordinador al proceso con identificador más alto

Los procesos conforman un anillo lógico (cada uno conectado con su vecino)

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección basado en **anillo**

Objetivo: Elegir como coordinador al proceso con identificador más alto

Los procesos conforman un anillo lógico (cada uno conectado con su vecino)

- Inicialmente todos los procesos son no-candidatos → cualquiera puede empezar una elección:
 - (1) se marca como candidato y (2) envía mensaje de elección con su identificador al vecino

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección basado en **anillo**

Objetivo: Elegir como coordinador al proceso con identificador más alto

Los procesos conforman un anillo lógico (cada uno conectado con su vecino)

- Inicialmente todos los procesos son no-candidatos → cualquiera puede empezar una elección:
 - (1) se marca como candidato y (2) envía mensaje de elección con su identificador al vecino
- Cuando un proceso recibe un mensaje de elección:
 - si identificador del mensaje es mayor que el suyo → lo reenvía a sus vecinos
 - si es menor:
 - si es no-candidato → (1) sustituye el identificador por el suyo, (2) lo envía al vecino y (3) se marca como candidato
 - si es el suyo → (1) se marca como **elegido** y (2) envía mensaje de **elegido** a su vecino añadiendo su identidad

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección basado en **anillo**

Objetivo: Elegir como coordinador al proceso con identificador más alto

Los procesos conforman un anillo lógico (cada uno conectado con su vecino)

- Inicialmente todos los procesos son no-candidatos → cualquiera puede empezar una elección:
 - (1) se marca como candidato y (2) envía mensaje de elección con su identificador al vecino
- Cuando un proceso recibe un mensaje de elección:
 - si identificador del mensaje es mayor que el suyo → lo reenvía a sus vecinos
 - si es menor:
 - si es no-candidato → (1) sustituye el identificador por el suyo, (2) lo envía al vecino y (3) se marca como candidato
 - si es el suyo → (1) se marca como **elegido** y (2) envía mensaje de **elegido** a su vecino añadiendo su identidad
- Cuando un proceso recibe un mensaje de elegido:
 - (1) se marca como no-candidato y (2) lo envía a su vecino

algoritmos de elección de coordinador

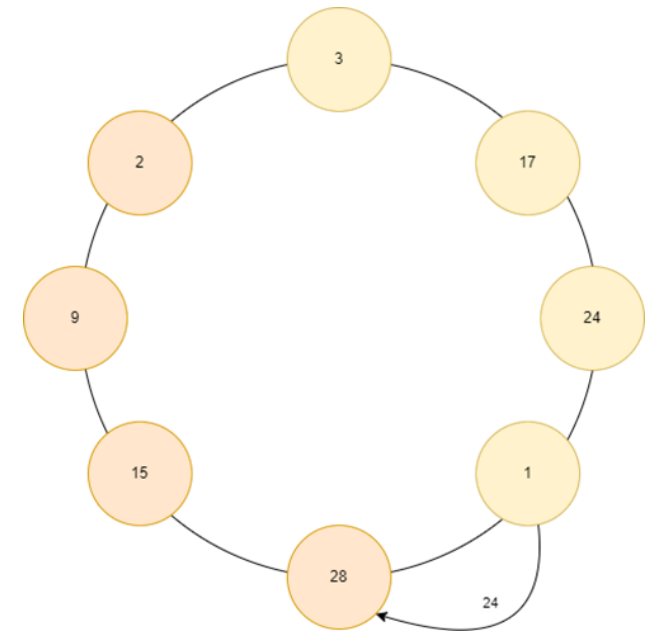
Algoritmo de elección basado en **anillo**

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**



algoritmos de elección de coordinador

introducción

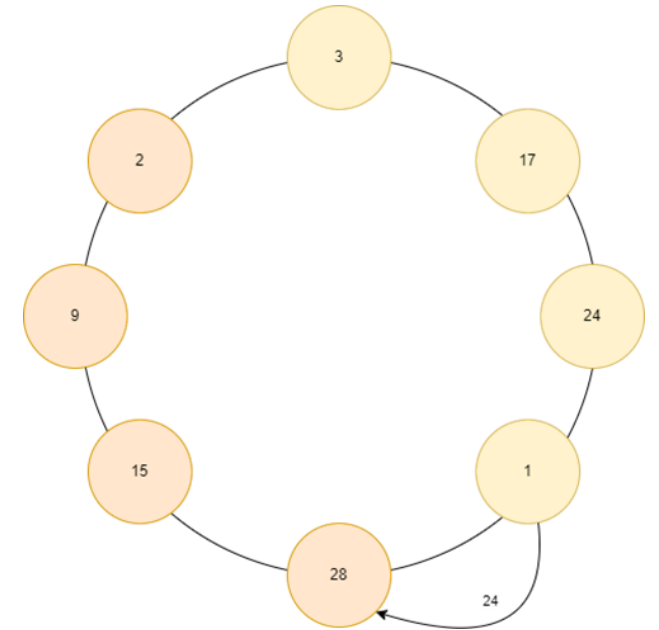
establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección basado en **anillo**

No detecta fallos, se supone procesos estables durante la elección



algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

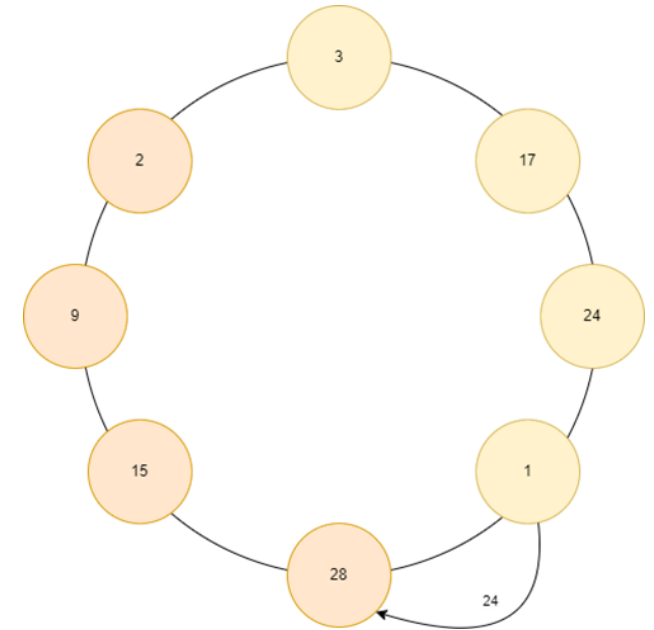
**elección
coordinador**

Algoritmo de elección basado en **anillo**

No detecta fallos, se supone procesos estables durante la elección

Cumplimiento de propiedades

- E1: Sí, se garantiza que el proceso activo con identificador más alto
- E2: Sí, la última vuelta garantiza la propagación del nuevo coordinador a todos



algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección basado en **anillo**

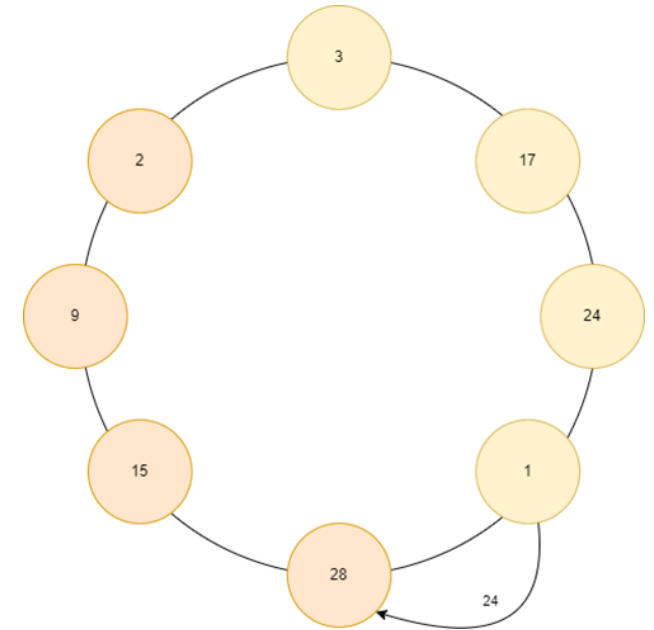
No detecta fallos, se supone procesos estables durante la elección

Cumplimiento de propiedades

- E1: Sí, se garantiza que el proceso activo con identificador más alto
- E2: Sí, la última vuelta garantiza la propagación del nuevo coordinador a todos

Complejidad del algoritmo

- peor caso: lanza elección sólo el vecino al futuro coordinador $\rightarrow 3n-1$ mensajes
- mejor caso: lanza elección el futuro coordinador $\rightarrow 2n$ mensajes



algoritmos de elección de coordinador

Algoritmo de elección del **abusón** o *bully*

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección del **abusón** o ***bully***

- Todos los procesos deben conocer el identificador (ID) y la dirección del resto de procesos

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección del **abusón** o *bully*

- Todos los procesos deben conocer el identificador (ID) y la dirección del resto de procesos
- Se presupone una comunicación fiable

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección del **abusón** o ***bully***

- Todos los procesos deben conocer el identificador (ID) y la dirección del resto de procesos
- Se presupone una comunicación fiable
- El algoritmo selecciona al proceso superviviente con **mayor ID / prioridad**

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección del **abusón** o **bully**

- Todos los procesos deben conocer el identificador (ID) y la dirección del resto de procesos
- Se presupone una comunicación fiable
- El algoritmo selecciona al proceso superviviente con **mayor ID / prioridad**
- Permite la caída de procesos durante la elección: utiliza **temporizadores** para detectar fallos de procesos

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección del **abusón** o **bully**

- Todos los procesos deben conocer el identificador (ID) y la dirección del resto de procesos
- Se presupone una comunicación fiable
- El algoritmo selecciona al proceso superviviente con **mayor ID / prioridad**
- Permite la caída de procesos durante la elección: utiliza **temporizadores** para detectar fallos de procesos

Existen 3 tipos de mensajes:

- Mensaje de **elección**: para anunciar una elección
- Mensaje de **respuesta**: respuesta a un mensaje de elección
- Mensaje de **coordinador**: para anunciar el ID del nuevo coordinador

algoritmos de elección de coordinador

Algoritmo de elección del **abusón** o ***bully***

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección del **abusón** o *bully*

- Un proceso inicia una elección al darse cuenta de que el coordinador ha caído:

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección del **abusón** o **bully**

- Un proceso inicia una elección al darse cuenta de que el coordinador ha caído:
 - envía mensaje de elección a los procesos con ID mayor que el suyo
 - espera algún mensaje de respuesta:
 - si vence temporizador (sin respuesta) → el proceso se erige como coordinador y envía mensaje de coordinador a todos los procesos con IDs más bajos
 - si recibe alguna respuesta → espera mensaje de coordinador
 - si vence temporizador → lanza una nueva elección

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección del **abusón** o *bully*

- Un proceso inicia una elección al darse cuenta de que el coordinador ha caído:
 - envía mensaje de elección a los procesos con ID mayor que el suyo
 - espera algún mensaje de respuesta:
 - si vence temporizador (sin respuesta) → el proceso se erige como coordinador y envía mensaje de coordinador a todos los procesos con IDs más bajos
 - si recibe alguna respuesta → espera mensaje de coordinador
 - si vence temporizador → lanza una nueva elección
- Si un proceso recibe un mensaje de elección:
 - contesta con un mensaje de respuesta y lanza una elección (si no ha lanzado ya antes una)

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección del **abusón** o *bully*

- Un proceso inicia una elección al darse cuenta de que el coordinador ha caído:
 - envía mensaje de elección a los procesos con ID mayor que el suyo
 - espera algún mensaje de respuesta:
 - si vence temporizador (sin respuesta) → el proceso se erige como coordinador y envía mensaje de coordinador a todos los procesos con IDs más bajos
 - si recibe alguna respuesta → espera mensaje de coordinador
 - si vence temporizador → lanza una nueva elección
- Si un proceso recibe un mensaje de elección:
 - contesta con un mensaje de respuesta y lanza una elección (si no ha lanzado ya antes una)
- Si un proceso recibe un mensaje de coordinador:
 - guarda el ID y trata a ese proceso como nuevo coordinador

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección del **abusón** o *bully*

- Un proceso inicia una elección al darse cuenta de que el coordinador ha caído:
 - envía mensaje de elección a los procesos con ID mayor que el suyo
 - espera algún mensaje de respuesta:
 - si vence temporizador (sin respuesta) → el proceso se erige como coordinador y envía mensaje de coordinador a todos los procesos con IDs más bajos
 - si recibe alguna respuesta → espera mensaje de coordinador
 - si vence temporizador → lanza una nueva elección
- Si un proceso recibe un mensaje de elección:
 - contesta con un mensaje de respuesta y lanza una elección (si no ha lanzado ya antes una)
- Si un proceso recibe un mensaje de coordinador:
 - guarda el ID y trata a ese proceso como nuevo coordinador
- Cuando un proceso se reinicia:
 - lanza una elección a menos que sea el de ID más alto (en cuyo caso se erigiría como nuevo coordinador → **abusón**)

algoritmos de elección de coordinador

Algoritmo de elección del **abusón** o ***bully***

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección del **abusón** o *bully*

Cumplimiento de propiedades:

- E1: Sí, salvo partición de la red
- E2: Sí, en último extremo, el que detecta el fallo del coordinador, acaba siéndolo

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección del **abusón** o *bully*

Cumplimiento de propiedades:

- E1: Sí, salvo partición de la red
- E2: Sí, en último extremo, el que detecta el fallo del coordinador, acaba siéndolo

Complejidad del algoritmo

- caso mejor: se da cuenta el segundo ID más alto $\rightarrow n - 2$ mensajes
- caso peor: se da cuenta el ID más bajo $\rightarrow O(n^2)$ mensajes

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección del **abusón** o *bully*

Cumplimiento de propiedades:

- E1: Sí, salvo partición de la red
- E2: Sí, en último extremo, el que detecta el fallo del coordinador, acaba siéndolo

Complejidad del algoritmo

- caso mejor: se da cuenta el segundo ID más alto $\rightarrow n - 2$ mensajes
- caso peor: se da cuenta el ID más bajo $\rightarrow O(n^2)$ mensajes

Discusión

¿Qué sentido puede tener que después de elegir nuevo coordinador, si se reinicia el de mayor ID se vuelva a asignar como coordinador?

algoritmos de elección de coordinador

Problemática de los algoritmos de elección anteriores

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Problemática de los algoritmos de elección anteriores

- Se basan en **temporizadores**: Retrasos de transmisión pueden causar la elección de múltiples líderes.

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Problemática de los algoritmos de elección anteriores

- Se basan en **temporizadores**: Retrasos de transmisión pueden causar la elección de múltiples líderes.
- La pérdida de conexión entre dos grupos de procesadores puede aislar permanentemente los nodos (p. ej. la partición de una red formando grupos de nodos aislados)

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Problemática de los algoritmos de elección anteriores

- Se basan en **temporizadores**: Retrasos de transmisión pueden causar la elección de múltiples líderes.
- La pérdida de conexión entre dos grupos de procesadores puede aislar permanentemente los nodos (p. ej. la partición de una red formando grupos de nodos aislados)

Los algoritmos de **invitación** pueden solventar los problemas

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Problemática de los algoritmos de elección anteriores

- Se basan en **temporizadores**: Retrasos de transmisión pueden causar la elección de múltiples líderes.
- La pérdida de conexión entre dos grupos de procesadores puede aislar permanentemente los nodos (p. ej. la partición de una red formando grupos de nodos aislados)

Los algoritmos de **invitación** pueden solventar los problemas

Características del algoritmo de **invitación**:

- Definición de grupos de procesadores con líder único
- Detección y agregación de grupos
- Reconocimiento por parte del líder de los miembros del grupo

algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

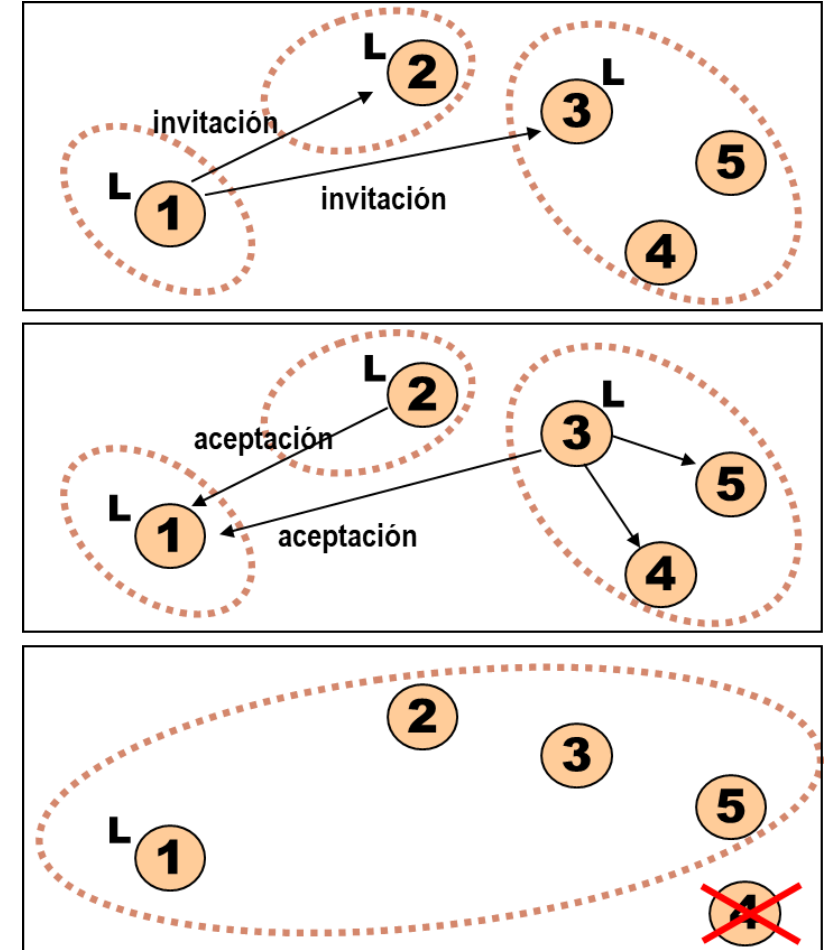
exclusión mutua

**elección
coordinador**

Algoritmo de elección por **invitación**

Funcionamiento

- Si un procesador detecta la pérdida del líder, entonces se declara líder y forma su propio grupo
- Periódicamente el líder de cada grupo invita a líderes de otros grupos
- Dos grupos se unen por medio de mensajes de aceptación:
 - De forma explícita
 - Como respuesta a mensajes de invitación (a líderes más prioritarios)



algoritmos de elección de coordinador

introducción

establecimiento
de tiempo

exclusión mutua

**elección
coordinador**

Algoritmo de elección por **invitación**

Cumplimiento de propiedades

- E1: Sí, al final del proceso se tiende a la existencia de un solo líder (tras desparticionarse el sistema y volver la conectividad)
- E2: Sí, todos los grupos tienen líder (incluso los de un solo miembro)

