Lección 11
Sincronización en los Sistemas Distribuidos

Índice

- Introducción
- Relojes lógicos
- Relojes físicos
  - Definición de segundo
  - Sincronización de relojes

Introducción

- Algunos algoritmos dependen de alguna forma del paso del tiempo para su funcionamiento:
  - Pueden requerir la hora exacta. Ej., para incluirla en forma de timestamp en los mensajes
  - O pueden necesitar sólo conocer en qué orden ocurrieron ciertos eventos. Ej, la utilidad make

Problema

En un sistema distribuido no hay un reloj único, sino uno en cada máquina y los relojes de los computadores no son exactos.

Introducción: Funcionamiento del reloj

- En un computador la hora se mantiene gracias a un oscilador de cuarzo.
  - Cada oscilación genera un 1 seguido de un 0.
  - Un circuito cuenta cuántos "unos" se han generado.
  - Cuando la cuenta alcance un valor prefijado, se generará una interrupción.
  - Cada vez que el sistema operativo recibe una interrupción, avanza un contador.
  - Este contador permite calcular el tiempo transcurrido desde el arranque, y por tanto la hora.

Problema

La frecuencia de oscilación del cuarzo puede ser ligeramente diferente entre máquinas, causando que el reloj *adelante* o *atrase*.

Reloj lógico

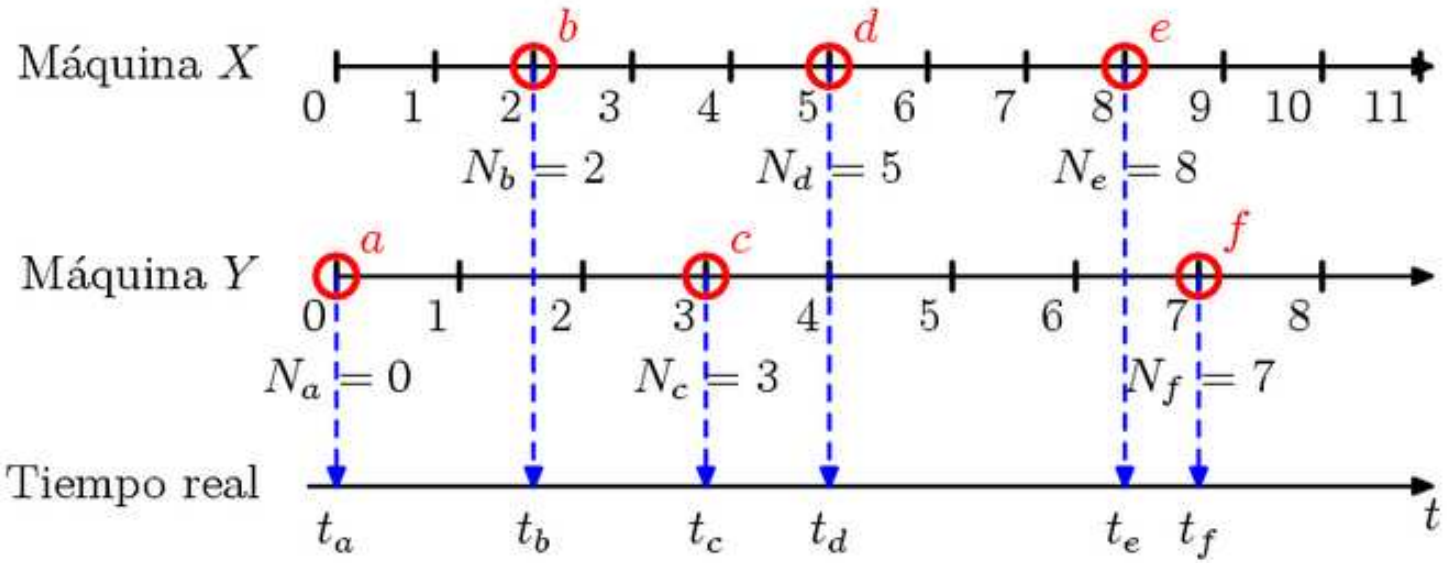
- Observar que:
  - Si el cuarzo no oscila a la frecuencia correcta, la hora que el computador "cree" que es es falsa
  - Pero incluso en ese caso, existe una consistencia lógica entre las horas que el computador genera, ya que

Si dos eventos *a* y *b* ocurren cuando el contador vale respectivamente N_a y N_b, siendo N_a mayor que N_b, esto implica que el evento *a* ocurrió después que *b. Es decir:*

$$N_a > N_b \quad \Rightarrow \quad t_a > t_b$$

- En cambio, si a y b son eventos en diferentes máquinas, la implicación anterior no es necesariamente cierta

Ejemplo



$N_a = 0$ $N_d = 5$
$N_b = 2$ $N_e = 8$
$N_c = 3$ $N_f = 7$

Observar

En los eventos *e* y *f*, se da que $t_e < t_f$ y sin embargo $N_e > N_f$

Observación [Lamport]

- La hora real, *ta*, a la que ocurre un evento a es desconocida. Sólo podemos conocer lo que marca el reloj de esa máquina en ese instante, *Na*

Definición

La relación $a \rightarrow b$ significa "**El evento *a* ocurrió antes que el evento *b***", por tanto $t_a < t_b$

- Sin conocer t_a ni t_b

  ¿Cuándo podremos afirmar que a $\longrightarrow$ b?

  - 1. Cuando *a* y *b* ocurren en la misma máquina y $N_a < N_b$
  - 2. Cuando *a* representa el envío de un mensaje y *b* su recepción (aunque sea en diferentes máquinas)

Idea [Lamport]

- ¿Será posible asignar a cada evento *e* un número *C(e)* que cumpla

$$C(a) < C(b) \Rightarrow a \longrightarrow b \quad ?$$

- Es decir:
  - Si *a* ocurre antes que *b* en la misma máquina, debe cumplir *C(a) < C(b)*.
    - Esto lo cumple el contador de interrupciones de la máquina con tal de que no se le haga retroceder
  - Si *a* es el envío de un mensaje y *b* es su recepción, debe cumplirse *C(a) < C(b)*
    - Para que esto se cumpliera debería incluirse *C(a)* en el mensaje, y actualizar el reloj en *b.*

Algoritmo de Lamport

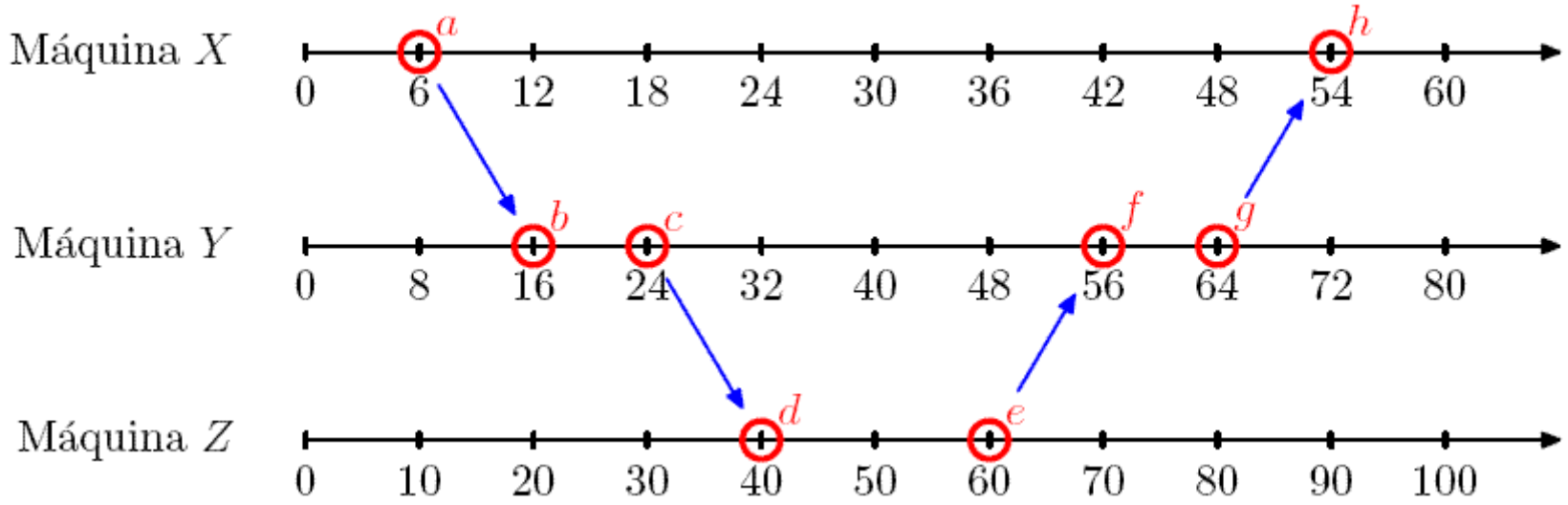
- Se usa como *C(a)* de un evento a el valor del contador de interrupciones de la máquina en que ocurre el evento.
- Al enviar un mensaje, se incluye en el mismo el contador de interrupciones en el momento del envío, *C(a)*.
- Al recibir un mensaje, se compara el contador de interrupciones local, *C(b)*, con el valor que viene en el mensaje, *C(a)*
  - Si $C(a) < C(b)$ no hay contradicción lógica.
    No se hace nada.
  - Si $C(a) \geq C(b)$ se adelanta el contador local,
    haciendo $C(b) = C(a) + 1$

Algoritmo de Lamport

- Observaciones:
  - Este algoritmo garantiza además que, si $a \rightarrow b$ y $b \rightarrow c$, entonces $a \rightarrow c$
  - Pueden existir pares de eventos a, b tales que no sea posible afirmar $a \rightarrow b$, ni tampoco $b \rightarrow a$. Se dice en este caso que a y b son concurrentes, y se denota por $a||b$.

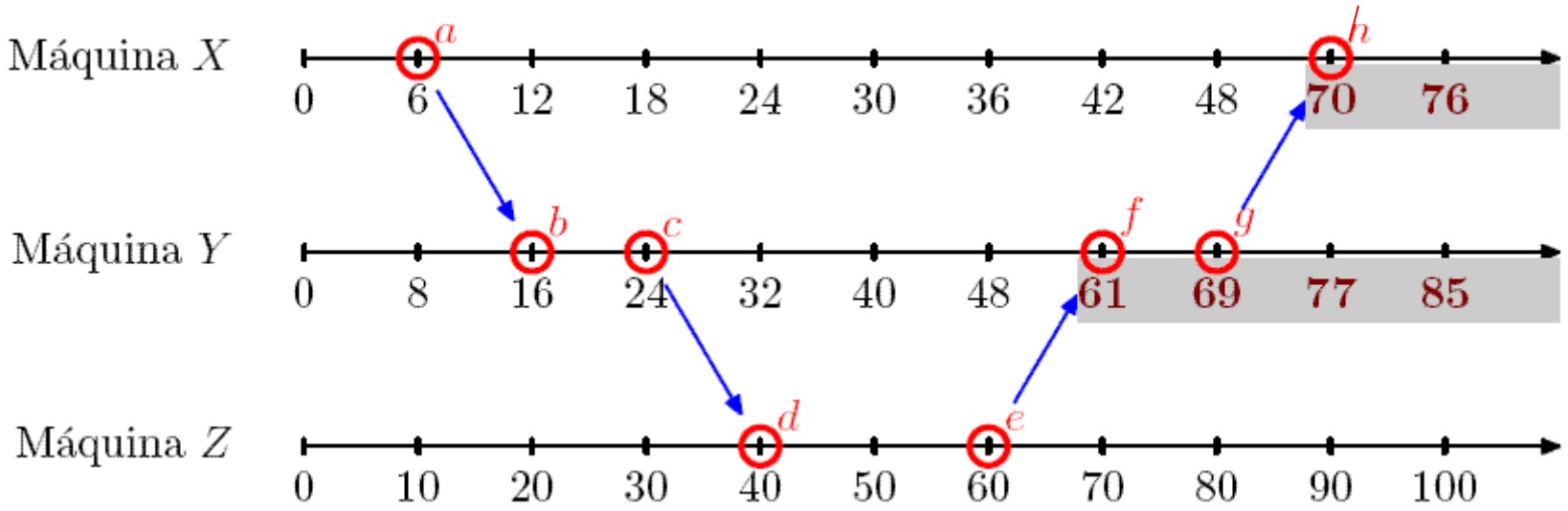
Ejemplo (algoritmo de Lamport)

- Supongamos tres máquinas, X, Y y Z cuyos osciladores tienen diferentes frecuencias de modo que mientras en Z se producen 10 interrupciones, en Y se producen 8 y en X sólo 6.



- La máquina X envía un mensaje a Y , poco después, ésta envía un mensaje a Z. Más tarde Z responde a Y y seguidamente Y responde a X.

Ejemplo (algoritmo de Lamport)



Cuando se detecta una contradicción al recibir un mensaje se adelanta el reloj en el receptor

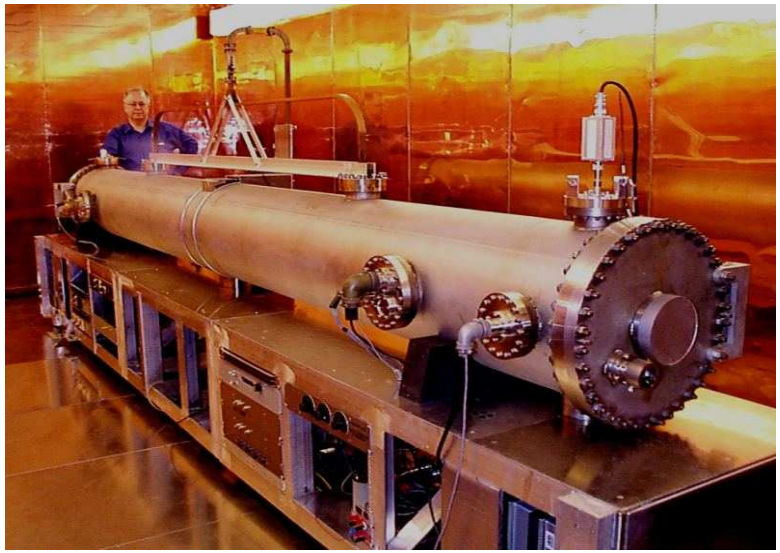
Relojes físicos

- Denominaremos **reloj físico** a uno que marca la hora exacta, y no una mera ordenación de eventos como en el caso de los relojes lógicos.
- La definición de "*hora exacta*" es más compleja de lo que parece a simple vista.
  - El segundo estaba definido como 1/86400 del día
  - Pero resulta que la longitud del día sufre pequeñas variaciones, quizás por alteraciones del núcleo de la tierra
  - Se define el ***segundo solar medio***, como 1/86400 del día promedio tras medir la longitud de un gran número de días
  - La rotación de la tierra sobre si misma se va ralentizando con el tiempo, aunque no el giro alrededor del sol. La definición de segundo solar por tanto no es constante. . .

Definición moderna de segundo

- En lugar de las vueltas de la tierra, se cuentan las transiciones de un átomo de cesio-133
- Un reloj atómico de cesio puede contar con precisión cuántas transiciones hace un átomo de cesio-133 (resulta que las hace con una frecuencia de F = 9 192 631 770 Hz.)
- Se define el **TAI** (*International Atomic Time*), como el número de transiciones que ha hecho un átomo de cesio-133 desde el 1 de Enero de 1958, dividido entre 9 192 631 770.

Le Bureau International de l'Heure (BIH) en Paris recibe información de relojes de cesio, y calcula el TAI.



La hora UTC

- Medir el tiempo usando TAI presenta un problema:
  - El "segundo de cesio" (TAI) es constante de un año a otro
  - El "segundo solar" en cambio va creciendo con el tiempo, debido a la ralentización de la rotación terrestre.
- Estas dos medidas, por tanto, se desincronizan.
  - Cada vez que la diferencia entre los "segundos TAI" y los "segundos solares" se hace mayor de 800ms, el BIH inserta un "segundo bisiesto", de modo que queden sincronizados de nuevo.
- El contador de segundos (incluyendo bisiestos) suministrado por el BIH, se denomina **UTC** (*Universal Coordinated Time*), y es la referencia mundial de "hora exacta".

Cómo obtener el valor de UTC

Mediante WWV

Es una emisora de radio de onda corta que emite un pulso cada vez que transcurre un segundo UTC.

- Un computador puede equiparse con un receptor WWV

Mediante GPS

El sistema de localización global (GPS) basado en satélites, también suministra el valor de UTC con precisión de 10ms

- Un computador puede equiparse con un receptor GPS

Sincronización de relojes

- Caso más simple: una de las máquinas tiene un receptor WWV o GPS, y las demás se deben sincronizar con ella.
- Idea básica: Cada cierto tiempo, las restantes máquinas piden la hora a la que tiene el receptor, y usan la respuesta para poner en hora sus relojes.

Problemas

- ¿Cada cuánto le preguntan la hora?
- ¿Cómo se tiene en cuenta el retardo de la red?
- ¿Cómo poner en hora el reloj local una vez recibida la respuesta? (Téngase en cuenta que no puede retroceder)

Frecuencia de sincronización

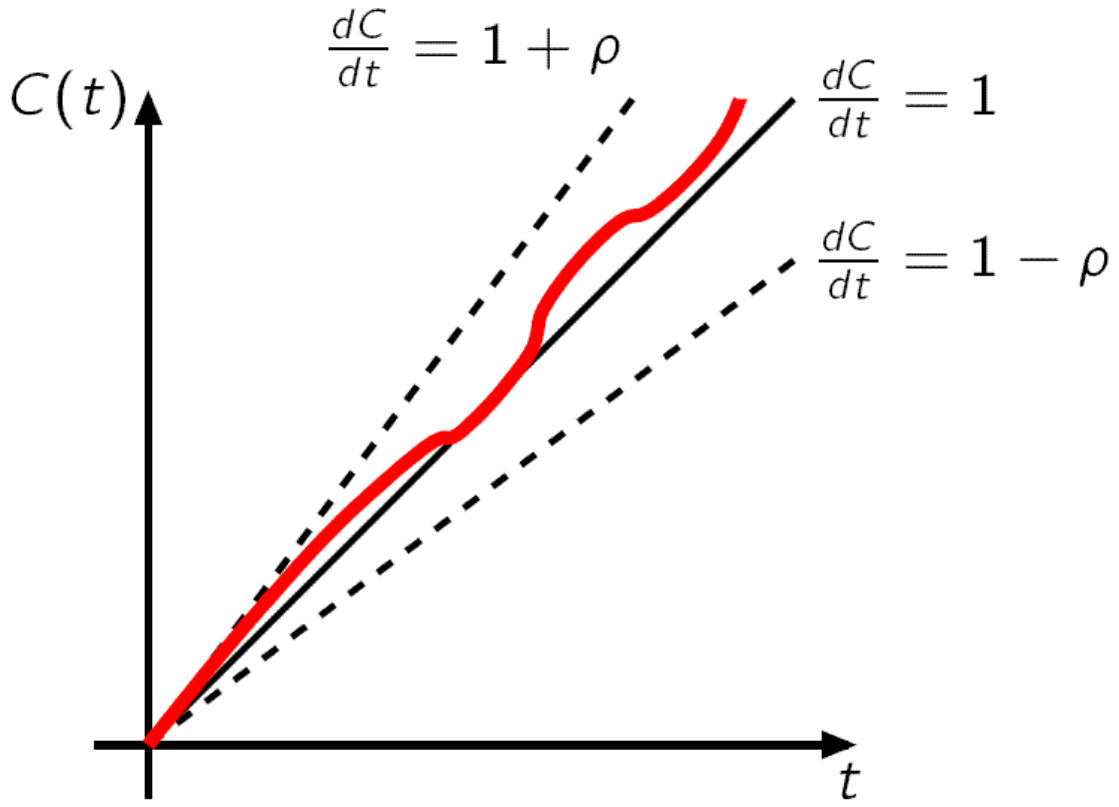
- Llamemos *C(t)* a la hora que tiene una máquina cuando la hora real es *t*.
  - Si su reloj fuera perfecto, cumpliría *C(t)* = *t* todo el tiempo, es decir, d*C*/d*t* = 1
  - Mientras el reloj adelante, d*C*/d*t* >1
  - Mientras el reloj atrase, d*C*/d*t* < 1
- El fabricante de un reloj debe garantizar que d*C*/d*t* no se aleja de 1 sin control, sino dentro de unos límites

$$\frac{dC}{dt} = 1 \pm \rho$$

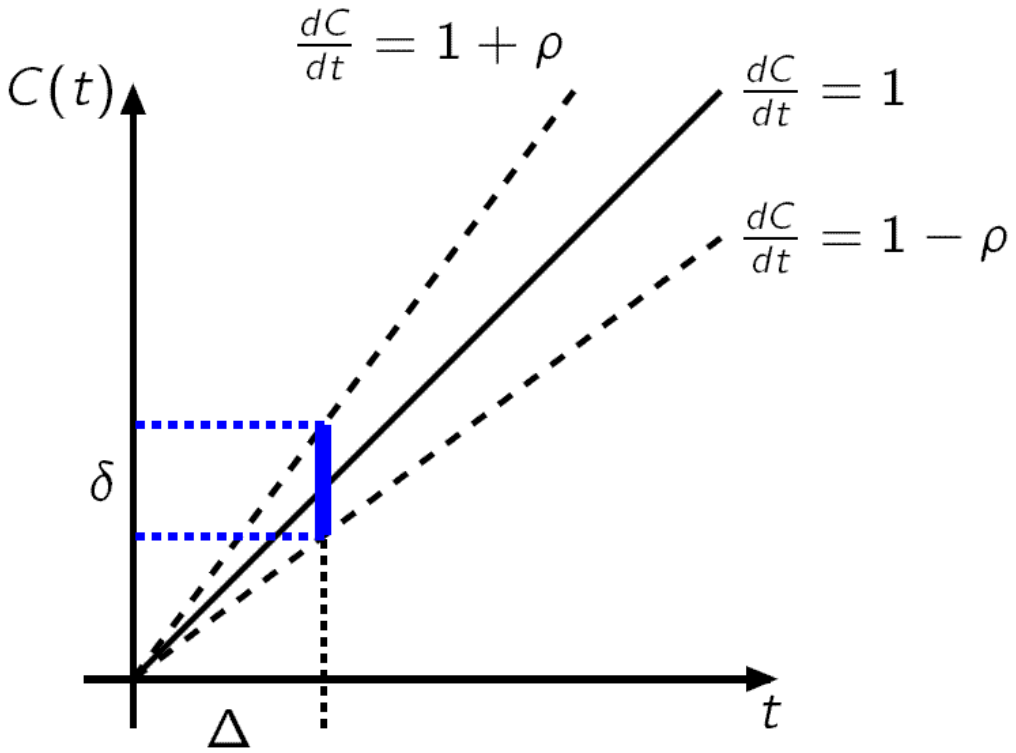
siendo ρ el **ratio máximo de deriva** del reloj (dato del fabricante)

Frecuencia de sincronización

- Variación de la hora local con respecto a la hora real



Frecuencia de sincronización

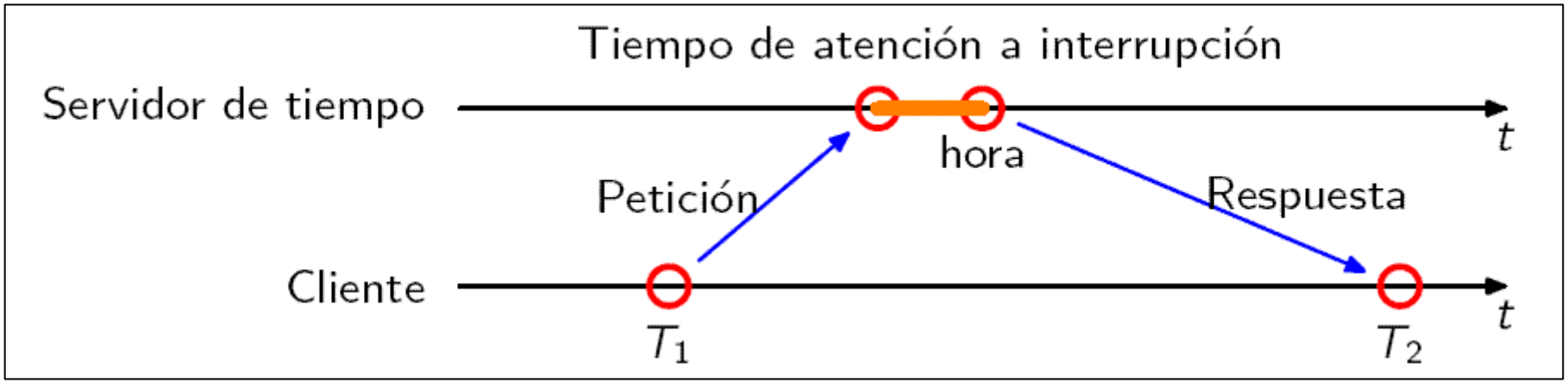


- Cuando haya transcurrido un tiempo Δ, dos relojes con el mismo ρ estarán desincronizados como máximo $\delta = 2\rho\Delta$

- Si no queremos que la discrepancia sea mayor de δ, hay que resincronizarlos cada $\delta/2\rho$ segundos.

El retardo de la red

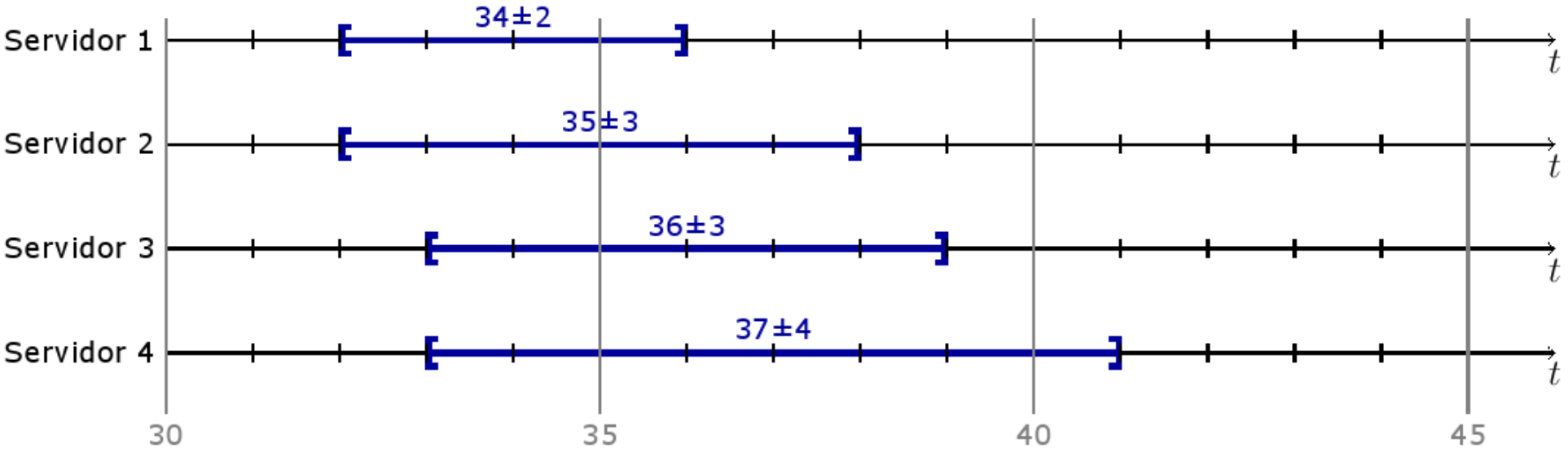
- La hora que recibimos no es la que tiene en ese momento el servidor, sino la que tenía cuando respondió.



- En ausencia de más datos podemos aproximar el retardo de la respuesta por $(T_2 - T_1)/2$, siendo T_1 la hora local cuando preguntamos la hora, y T_2 la hora local cuando la recibimos. [Algoritmo de Christian]

Algoritmos más elaborados

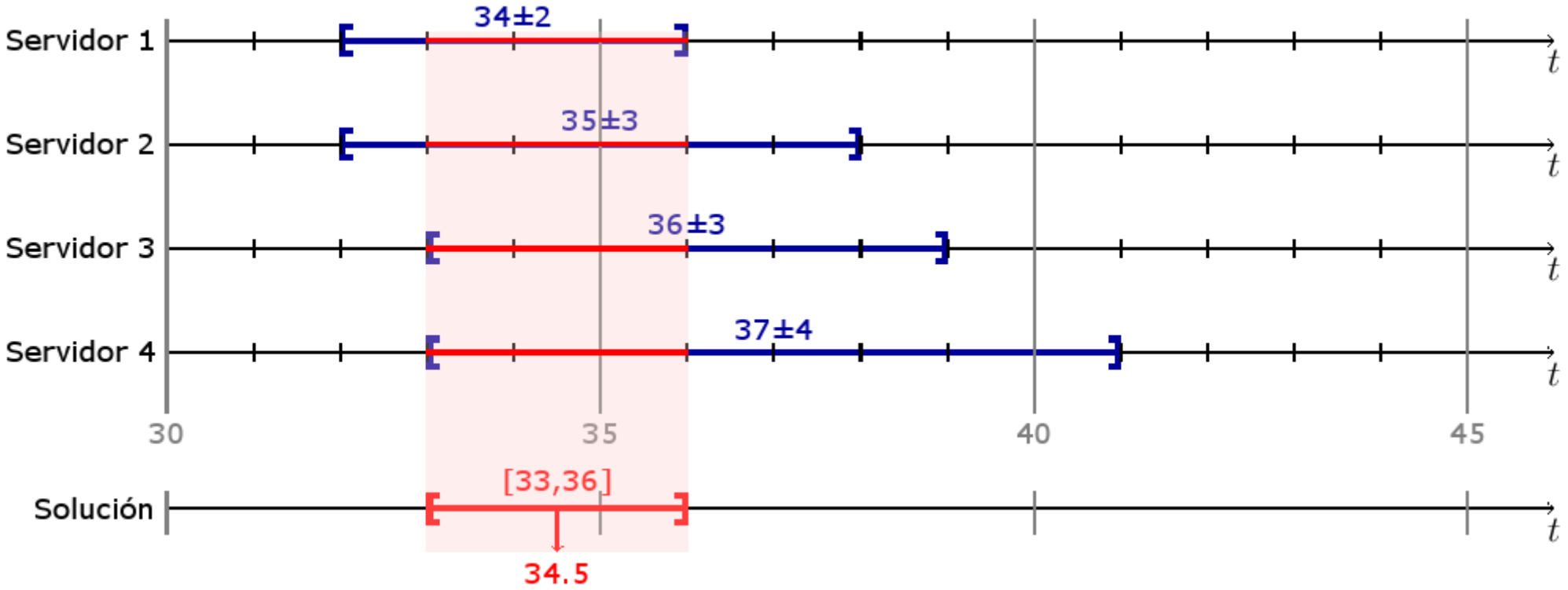
- Si en lugar de preguntar a un solo servidor preguntamos a varios, podremos tener una estimación más exacta de la hora.
- **Planteamiento:**
  - Cada servidor responde con su hora y un margen de error.



  - ¿Cómo estimar la hora correcta?

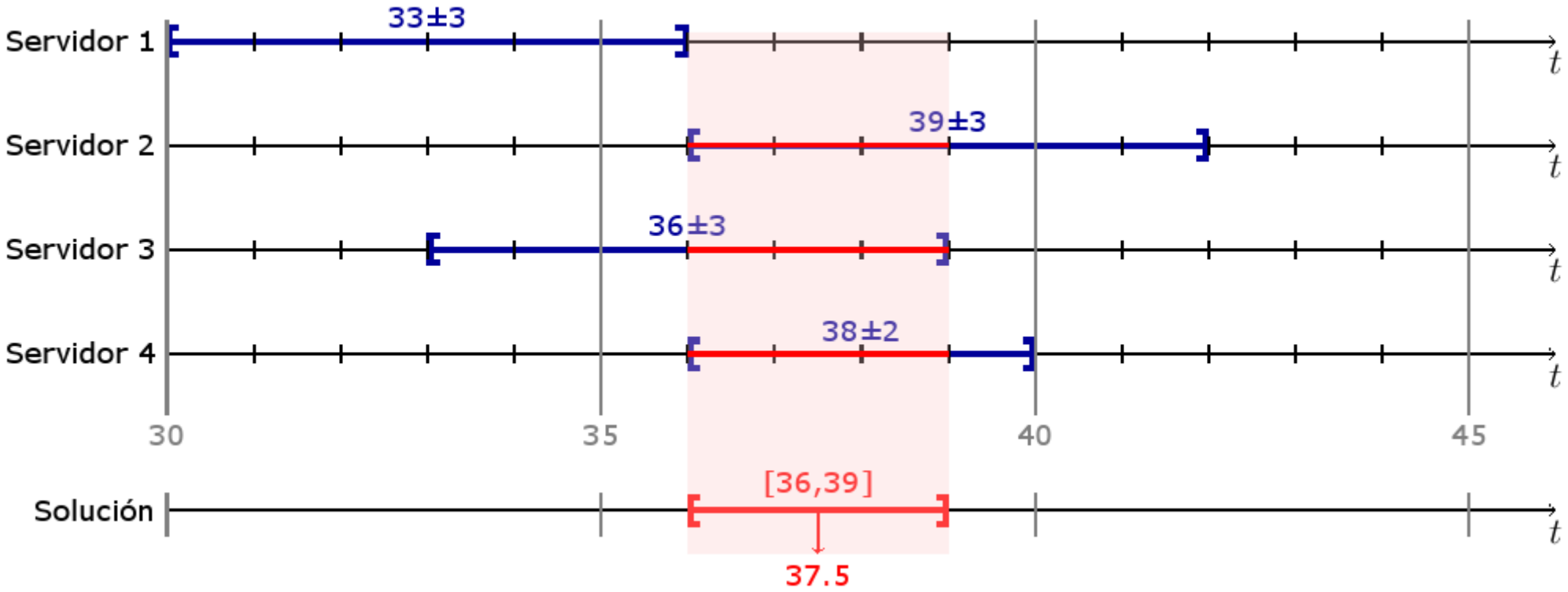
Algoritmo de Marzullo

- Se trata de encontrar un intervalo de tiempo que tenga el **máximo número de solapamientos** con las respuestas de los servidores.



Algoritmo de Marzullo

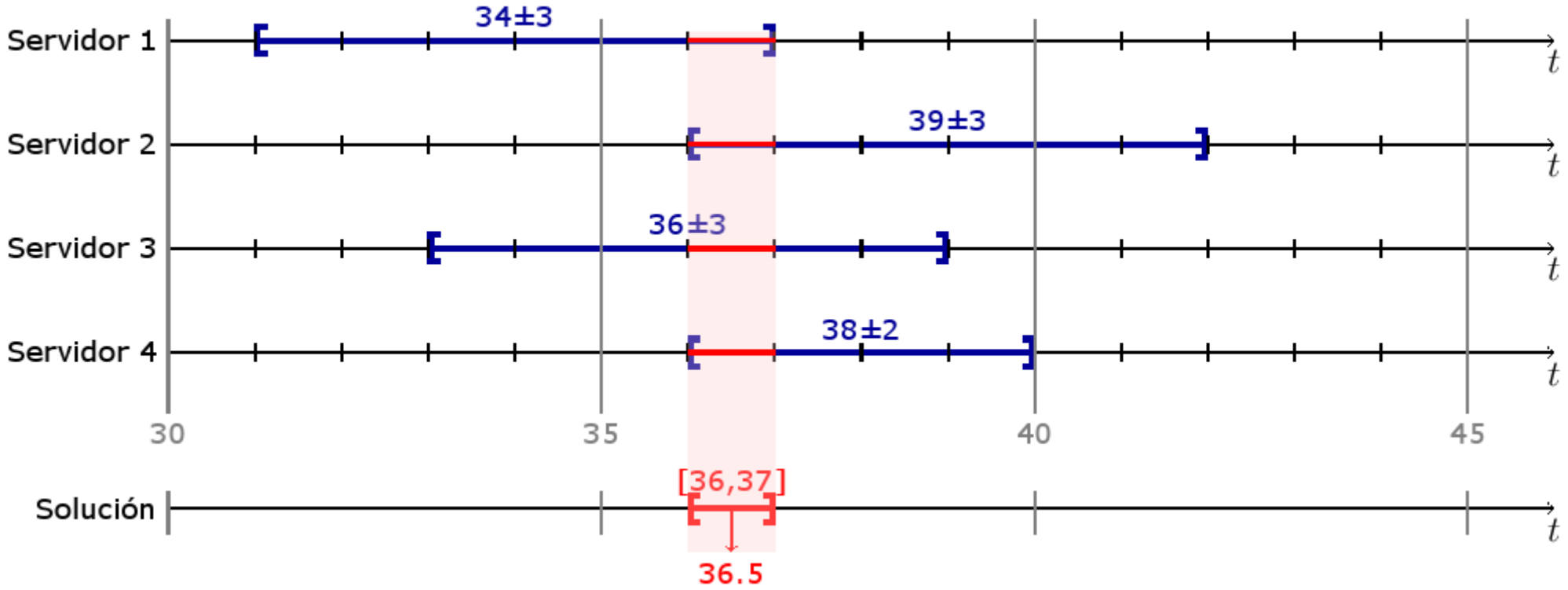
- Otro ejemplo.



- Observar que en este caso el máximo solapamiento posible es de 3 servidores (y uno queda fuera).

Algoritmo de Marzullo

- Una ligera variación en un servidor puede dar lugar a una solución muy diferente.



Algoritmo de Marzullo. Cómo computarlo

- Inicializar: **`Mejor`**=0, **`Contador`**=0
- Recorrer la figura de izquierda a derecha. Cada vez que se abre o cierra un intervalo hacer:
  - Sumar 1 a **`Contador`** si se abre, restar 1 si se cierra.
  - Si **`Contador`>`Mejor`**
    - **`Izquierda`**=punto actual
    - **`Derecha`**=punto siguiente (abra o cierre)
    - **`Mejor`**=**`Contador`**
- Resultado:
  - Intervalo solución: [**`Izquierda`**, **`Derecha`**]
  - Solapamientos conseguidos: **`Mejor`**

Network Time Protocol (NTP)

- NTP es un protocolo estandarizado para mantener la hora en un conjunto de computadores distribuído.
- Utiliza un conjunto de servidores, organizados por estratos:
  - Estrato 0: Son fuentes exactas (no PC). GPS, emisora de WWV…
  - Estrato 1: (servidores de hora). Obtienen la hora directamente del estrato 0.
  - Estrato 2: Obtienen la hora, vía internet, de un servidor de estrato 1.
  - Estrato 3: Obtienen la hora de un servidor de estrato 2.

Network Time Protocol (NTP)

- Para mantener la hora se utiliza una variación del Algoritmo de Marzullo. [Algoritmo de Intersección]
- El algoritmo de intersección mejora al de Marzullo:
  - Busca un intervalo que tenga el máximo solapamiento y que **contenga los puntos medios.**
  - La estimación de la hora no es el punto medio de ese intervalo, sino que tiene en cuenta otros **modelos estadísticos**.
- La implementación es muy compleja, pero está disponible para casi todos los operativos y plataformas.

Puesta en hora del reloj local

- Si la hora real es posterior a la que marca nuestro reloj, basta adelantarlo, pero si es anterior no podemos atrasarlo. ¡El reloj nunca debe retroceder!
- En ese caso se puede hacer que el reloj atrase (vaya más lento), hasta que alcance la hora correcta.
- Se puede aplicar también la técnica de variar la velocidad del reloj para adelantarlo (y así evitar saltos bruscos).