



# Construcción de Sistemas Distribuidos

## “Sincronización”

Rogelio Ferreira Escutia



# Sincronización

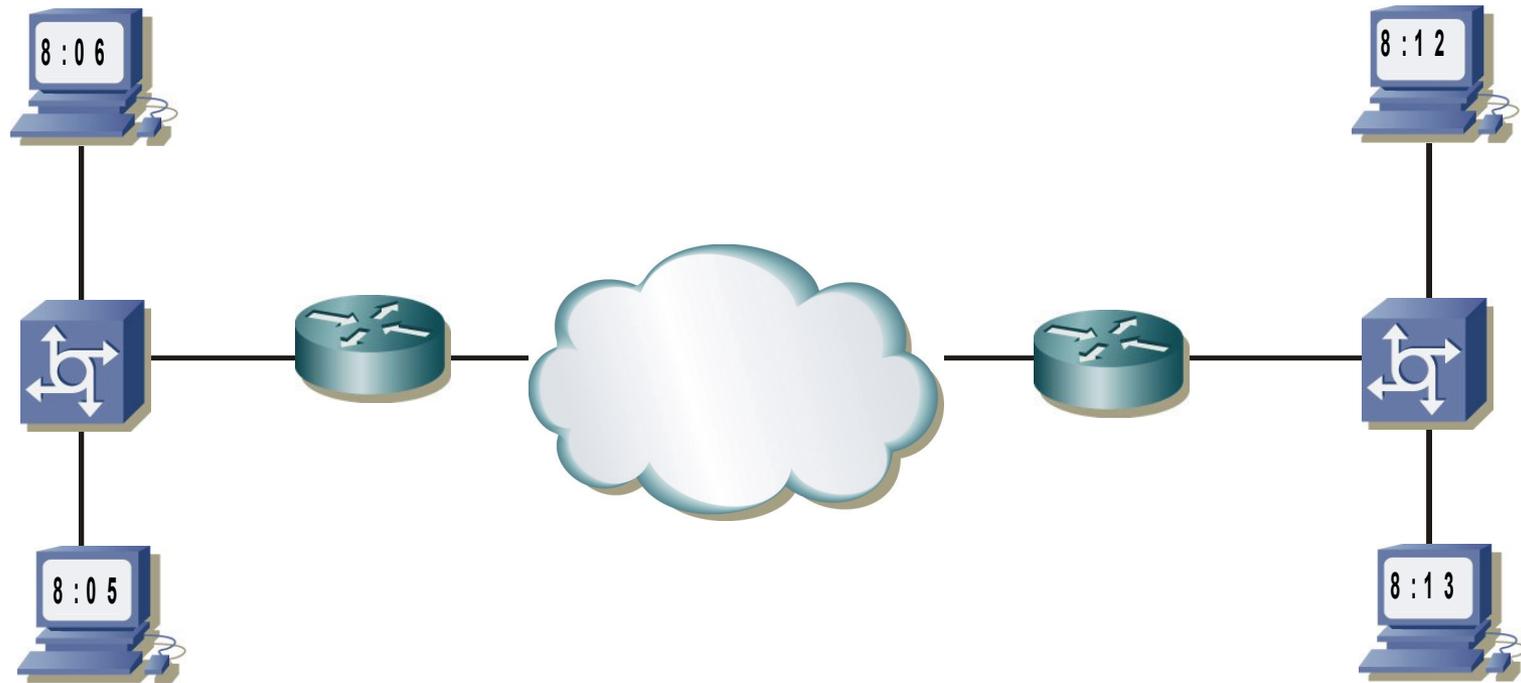
---

## Objetivo:

- Ordenar los eventos en forma cronológica para saber cuándo se efectuó un evento (fecha, hora, proceso a realizar y computadora que lo hizo).

# Sincronización

Diferencias de relojes internos en una red



# *Sincronización*

---

**Hay 2 tipos de sincronización del reloj:**

- **Externa: Cuando se toma el reloj de un dispositivo externo a la computadora.**
- **Interno: Se toman los relojes internos de las computadoras con cierto margen de atraso/adelanto de los mismos.**

# *Sincronización*

---

## Principales Problemas:

- **El más importante es la consistencia natural de los datos distribuidos en una red.**
- **Existen también problemas de autenticación**
- **Envío de peticiones a un servidor.**
- **Manejo de la cola de peticiones**

# *Sincronización*

¿ A que velocidad corre un auto 1 que va a la velocidad de la luz, si es observado por un auto 2 ?

**300,000 Km/seg**

- Verdadero si el auto 2 está inmóvil.
- Falso si el auto 2 va a la misma velocidad. Para el auto 2, el auto 1 esta inmóvil.
- Si el auto 2 va más rápido que el auto 1, observará que el auto 1 se está atrasando.

# *Sincronización*

**Einstein aseguró en su “Teoría de la Relatividad” las posibles consecuencias de asegurar que la velocidad de la luz es constante para todos los casos**

**Problema de usar relojes en una red:**

- **Cuando se envía información en forma simultánea desde varias computadoras no podemos asegurar la cantidad de tiempo que tardará en llegar a su destino ya que el envío será por medio de una red, con la implicaciones de retraso que esto implica.**

# *Sincronización*

---

## **Formas de sincronizar computadoras:**

- **Marcas de tiempo (sólo se pueden conseguir aproximaciones)**

## **2) Paso de mensajes**

**La mejor es la sincronización por medio de eventos (en caso de que el proceso lo admita).**

# Sincronización

## Reloj interno:

- Internamente cada computadora contiene un reloj físico, el cual es un dispositivo que cuenta las oscilaciones de un cristal con una frecuencia bien conocida. Estos conteos se dividen y almacenan en registros que dan por consiguiente una hora establecida.
- Esta hora sirve para hacer marcas de tiempo tomando en cuenta la resolución del reloj (un periodo de reloj debe ser mas corto que el evento mas corto de duración).

# *Sincronización*

---

## **Problemas de los relojes:**

- **El problema de exactitud del reloj reside en que cada pulso de un reloj puede llegar a ser diferente de todos los demás, para relojes contruidos de igual manera y aún dentro del mismo reloj hay variaciones.**
- **Las diferencias pueden ser mínimas pero en forma acumulada pueden ser considerables.**

# Sincronización

---

## Variaciones de relojes:

- Para un reloj basado en un cristal de cuarzo existe un atraso /adelanto de 1 segundo cada  $10^6$  (1,000,000 de segundos), es decir cada 11.6 días.
- Para un reloj atómico es de  $10^{13}$  segundos el cual se basa en pulsaciones de Cesio 133 (Cs).

# *Sincronización*

---

## **Relojes Astronómicos:**

- **También se puede medir el tiempo por medio de la rotación de la tierra pero es inexacto.**

# *Sincronización*

---

## **Tiempo Universal:**

- **Existe un estándar conocido como Tiempo Universal (Universal Time Coordinated – UTC).**
- **Se basa en relojes atómicos coordinados con la hora astronómica.**
- **Esta hora se sincroniza desde bases terrestres y es enviada a satélites.**

# *Sincronización*

**Exactitud:**

**UTC:**

- De 0.1 a 10 milisegundos, esto se debe principalmente a las condiciones atmosféricas de transmisión.

**GOES:**

- (Geostationary Operational Environmental Satellites) Retraso de 0.1 milisegundo.

**GPS:**

- (Global Positioning System) De 1 milisegundo.

**Una computadora puede sincronizarse de estos sistemas satelitales.**

# Sincronización

---

## Relojes de Computadoras:

- Los relojes físicos de las computadoras logran una exactitud de 10 milisegundos el cual, para los procesos de un S.D. pueden ser 100,000 instrucciones ejecutadas en un procesador de 10 MIPS, lo que significa una muy baja exactitud

# *Sincronización*

## Formas de Ajustar el Reloj:

- Se puede atrasar / adelantar la hora del sistema en base al UTC, pero esto puede ocasionar problemas con los procesos que están corriendo.
- Se puede hacer que el reloj pueda trabajar mas rápido o lento hasta alcanzar el reloj original, pero esta característica no es soportada por todos los tipos de hardware.

# *Sincronización*

## **Solución:**

**Existe una solución, haciendo un ajuste a la hora que se le da a una aplicación en base a la hora de la computadora (+ -) un tiempo de distorsión.**

**S (t) = tiempo del software**

**H (t) = tiempo del hardware**

**$\delta$  (t) = factor de compensación.**

$$\mathbf{S(t) = H + \delta (t)}$$

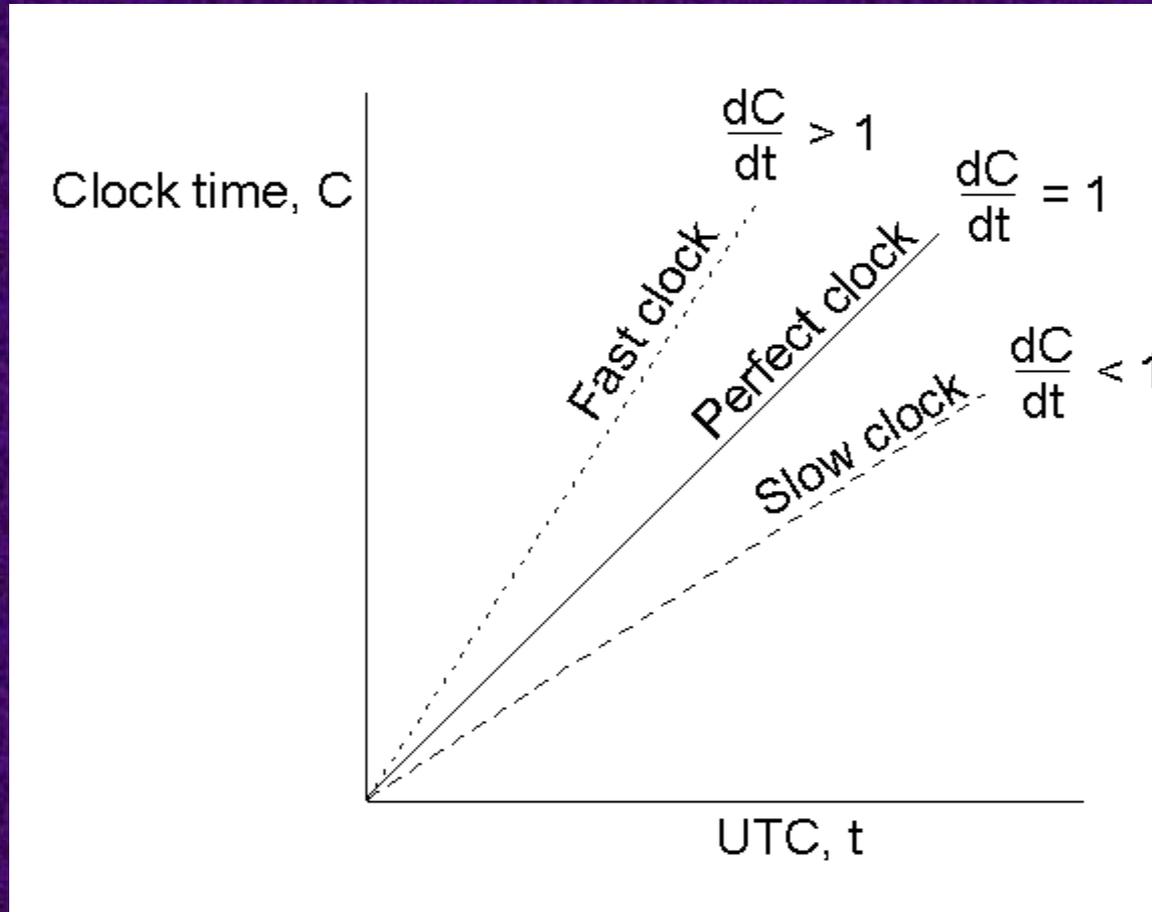
# Sincronización

## Forma de ajustar el reloj



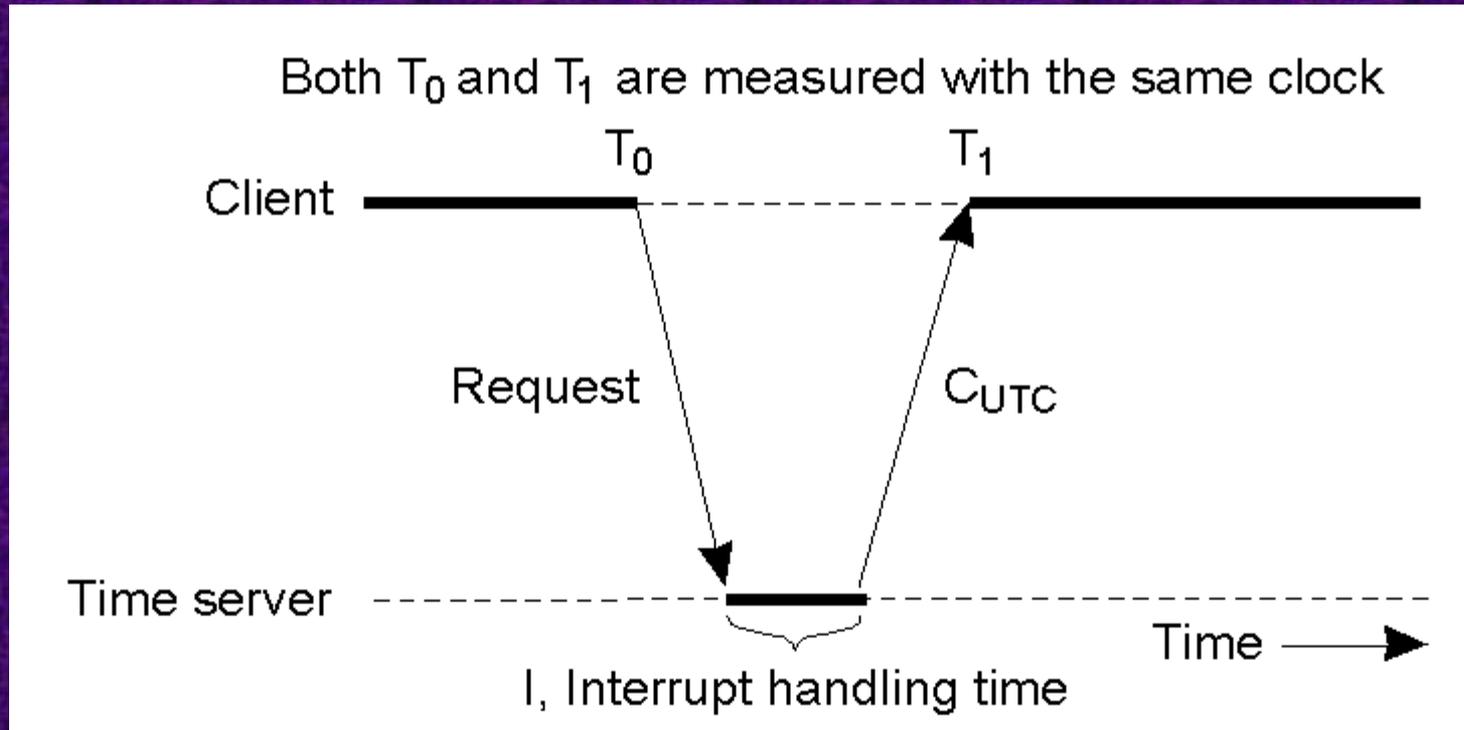
# Sincronización

## Sincronización de relojes



# Sincronización

## Sincronización de relojes: Algoritmo de Cristian



# *Sincronización*

---

## Sincronización de relojes: Algoritmo de Cristian:

**P** = proceso

**S** = servidor

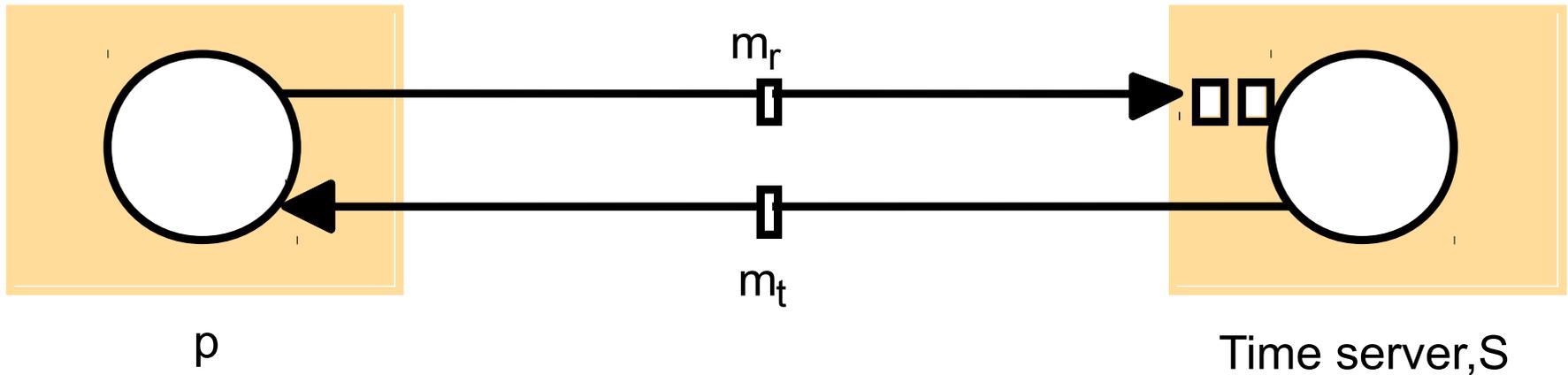
**mr** = message Request = mensaje de petición

**mt** = message Time = mensaje de tiempo

**P** envía una petición a **S**, regresando **mt** con una marca de tiempo **t**.

# Sincronización

## Retraso de envío



# *Sincronización*

---

## **DESVENTAJAS**

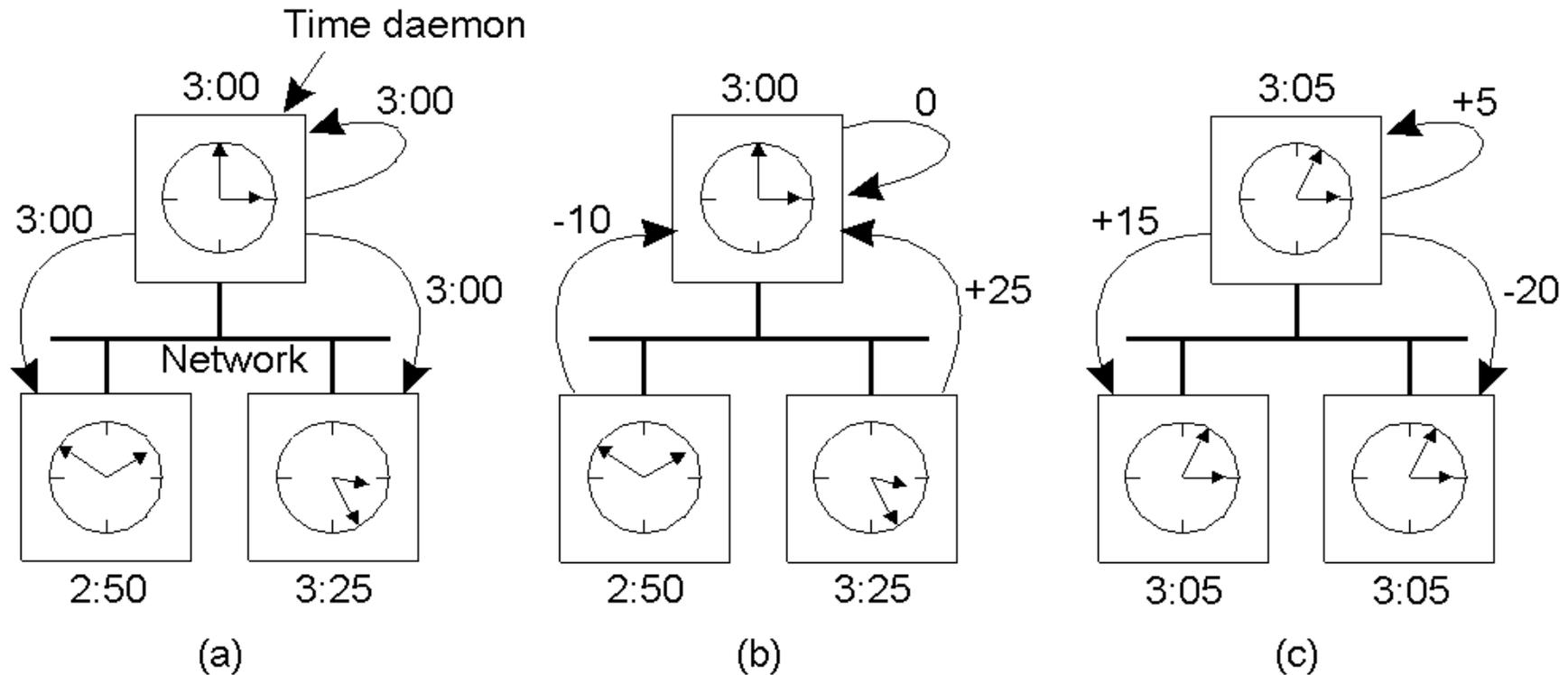
La sincronización es a partir de un servidor, por lo que se falla, falla la sincronización del sistema.

Para solucionar el problema Cristian sugiere que existan varios servidores con señales de sincronización los cuales obtienen su señal de UTC's.

Un cliente para sincronizarse emite una señal multicast hacia los servidores y la primera que recibe es la que toma para sincronizarse.

# Sincronización

## Algoritmo Berkeley



# *Sincronización*

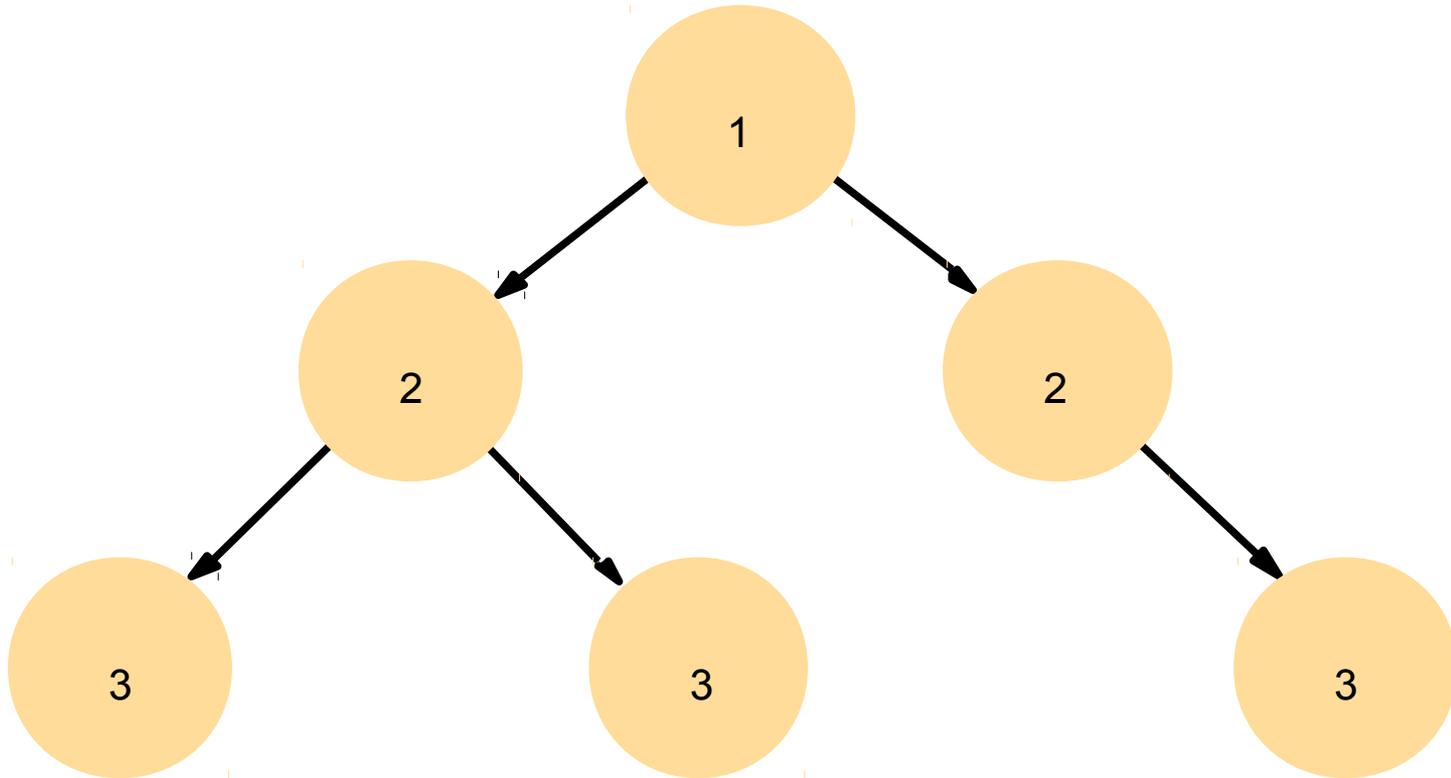
## **NTP (Network Time Protocol)**

**Fue propuesto por Mills en 1991 y ha sido adoptado como es estándar de Internet.**

**El objetivo es sincronizar cliente a un UTC con gran exactitud a través de Internet. El NTP utiliza técnicas estadísticas para determinar la exactitud con que un paquete viaja por Internet, aplicando filtros a los paquetes que llevan un tiempo de transporte grande.**

**NTP utiliza técnicas de autenticación para comprobar que los tiempos de certificación provienen de servidores confiables.**

# Sincronización



Note: Arrows denote synchronization control, numbers denote strata.

# *Sincronización*

---

## **Algoritmo de Lamport:**

**En 1978 Lamport explicó que debido a que no se pueden sincronizar adecuadamente los relojes a través de un sistema distribuido, no se puede utilizar el tiempo físico para encontrar el orden de 2 eventos cualesquiera que sean.**

# *Sincronización*

---

Lamport propone utilizar un esquema parecido al de la causalidad física.

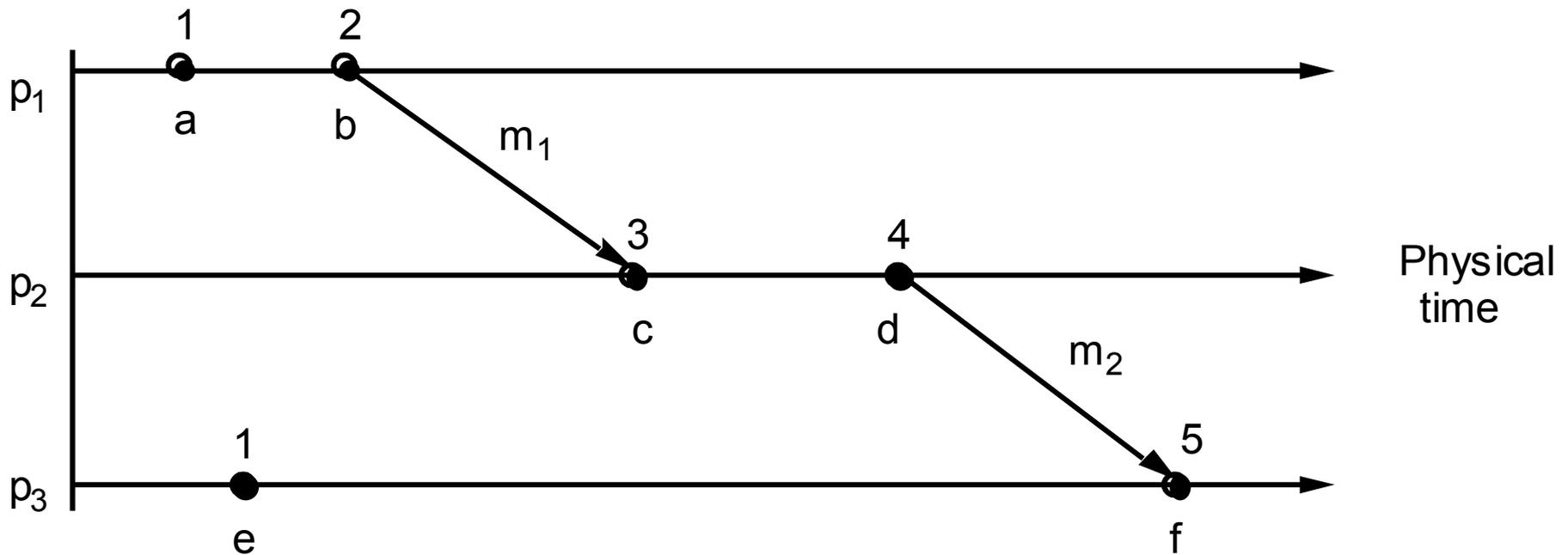
a) Cuando 2 eventos ocurren en un mismo proceso estarán en el orden en que se observan.

b) Cuando se envía un mensaje entre procesos, el evento de envía, el mensaje ocurrirá antes del evento de recibir el mensaje.

Lamport llamó a esto, relación happened-before formalmente

# Sincronización

## Algoritmo de Lamport:



**Fin**

***Sincronización***