

# Sincronização de Relógios

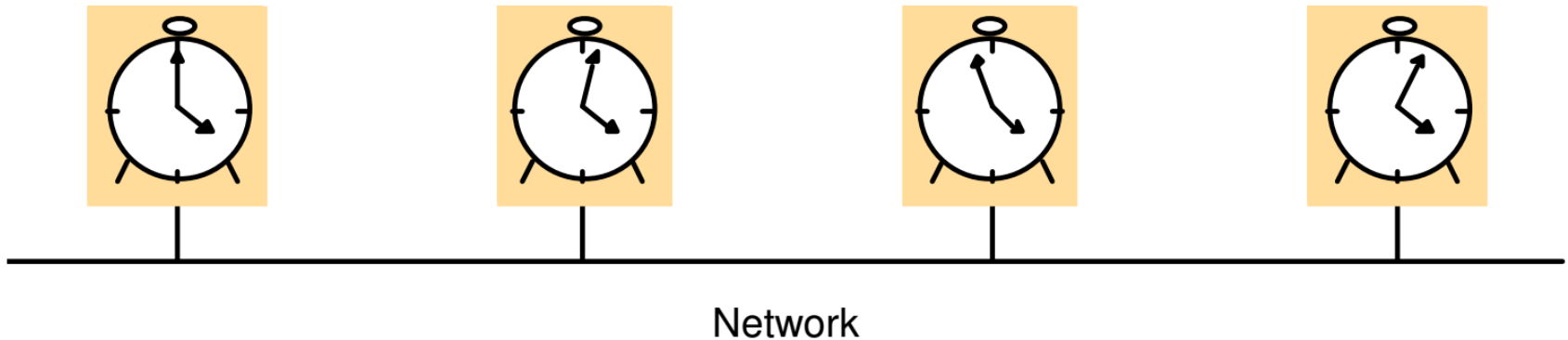
Prof. Vilmar Abreu Junior  
vilmar.abreu@pucpr.br

# Relógio

- Em um sistema centralizado, o tempo é algo trivial;
- Se um processo A solicita o tempo (horário) atual, e **alguns instantes depois** o processo B solicita o tempo, certamente o tempo que B recebeu será **maior** que o tempo que A recebeu;
- Já em um sistema distribuído, isso nem sempre é verdade!

# Distorção (skew) entre relógios

- Quando cada máquina tem seu próprio relógio, um evento A que ocorreu após um evento B pode ser marcado com um horário anterior ao de B;



# Aplicações

- Make
- Financeiras
- Auditoria
- Sistemas colaborativos
- Tolerância a falhas
- Etc

# Tipos de sincronização

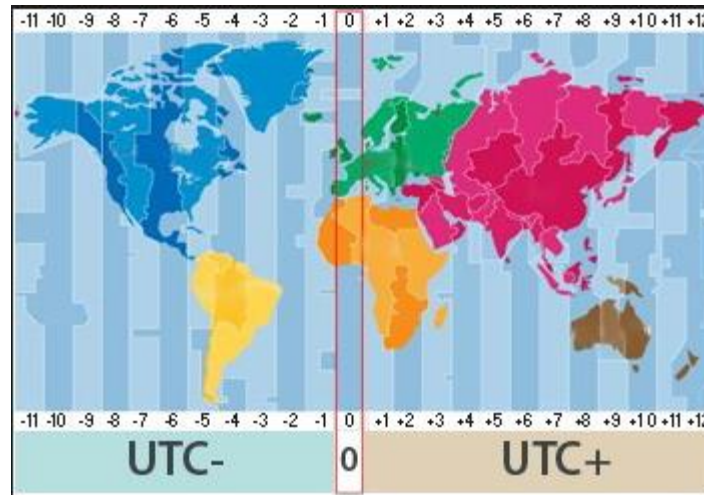
- **Externa:** Os relógios são sincronizados com uma **autoridade** que possui um relógio **“oficial”**;
  - **Exemplo:** Um Sistema bancário precisa saber o horário de uma determinada transação.
- **Interna:** Os relógios são sincronizados **entre si** com um grau de **precisão conhecido**. Dessa forma, é possível **determinar** o intervalo entre dois eventos que ocorreram em dois computadores distintos através do **relógio local**, mesmo não estando sincronizados com uma fonte externa.

# Relógios físicos

- Todos os computadores possuem um **circuito** (cristal) responsável por **determinar** o horário através de **tickets** (oscilações);
- O que realmente importa são os tempos **relativos** dos eventos
  - Exemplo: Relógio atrasado dois segundos utilizando make;
- É **impossível** garantir que os tickets de **diferentes relógios** estão exatamente na mesma frequência;

# UTC (Autoridade)

- O Tempo Universal Coordenado (UTC) é o fuso horário de referência a partir do qual se calculam todas as outras zonas horárias do mundo;



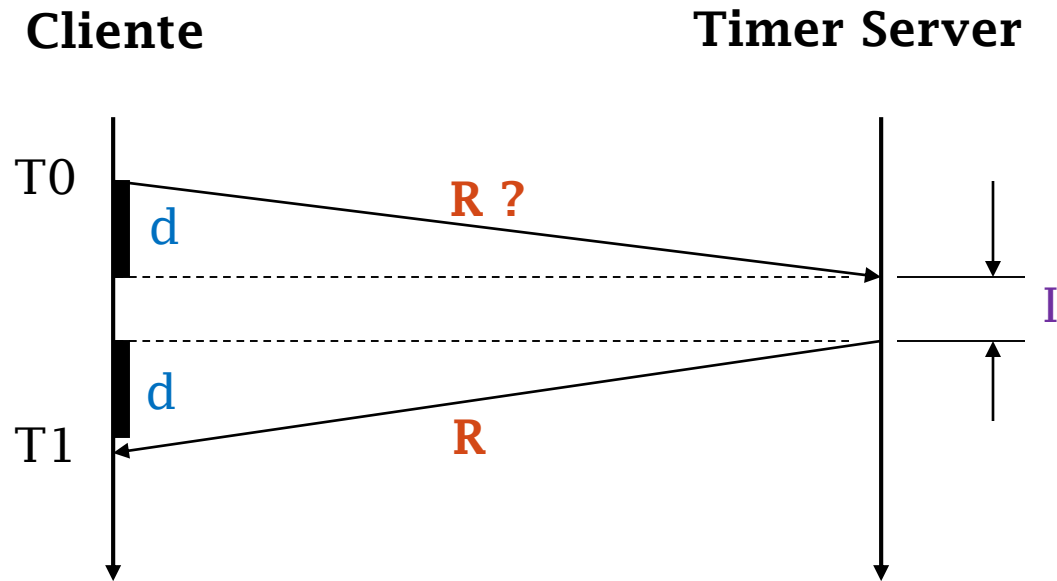


# Algoritmo de Christian

- A rede dispõe de um **time server**;
- Uma **máquina cliente** envia uma mensagem pedindo a hora certa ao **time server**;
- Ao receber a mensagem resposta do **time server**, o cliente adiciona o tempo médio de envio de mensagens à hora recebida;
- Esse tempo médio é calculado pelo próprio cliente considerando as horas de envio e recebimento das mensagens e ainda o tempo gasto pelo time server para processar o pedido.



# Algoritmo de Christian (cont.)



$$d = (T1 - T0 - I) / 2$$

$$T = R + d$$

# Atividade 01

- Uma máquina M envia uma mensagem r ao time server solicitando a hora certa, o qual responde com uma mensagem s, contendo a informação solicitada. Quando r é enviada, o relógio de M marca **3004**, e quando s é recebida o relógio de M marca **3050**.
- Supondo que o time server consome um tempo de **8** entre o recebimento da mensagem vinda de M e o envio da mensagem resposta para M, para qual valor M deve ajustar o seu relógio se o time server respondeu que a hora certa era **3035**?

# Problemas?

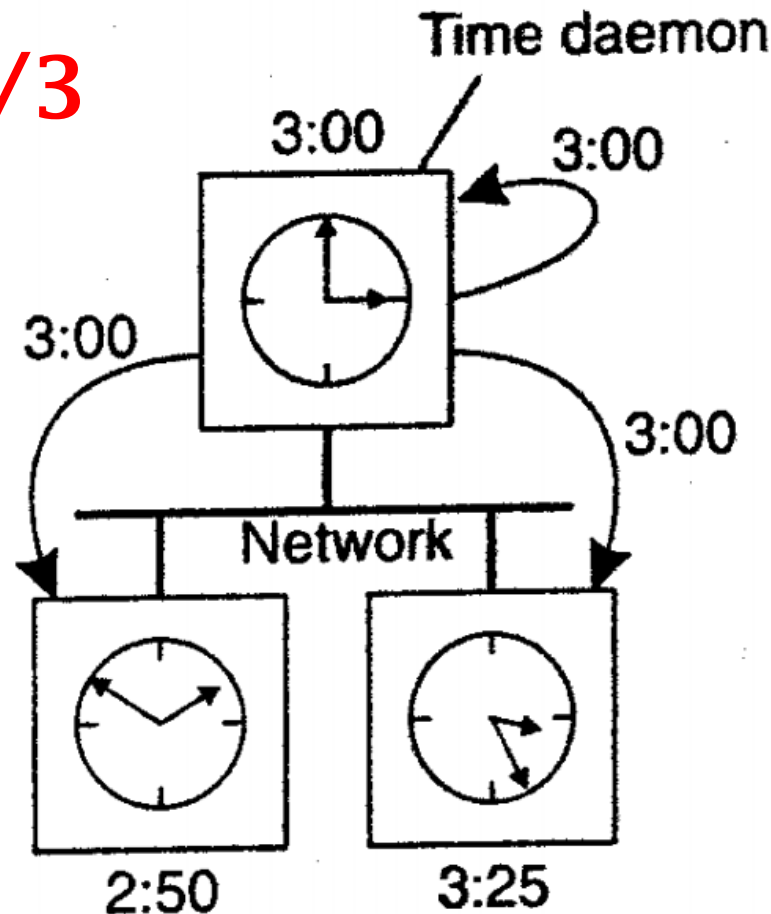
- Hora do cliente não pode “decrecer”
  - Solução: fazer o horário andar mais devagar por um determinado tempo, a fim de sincronizar com o servidor;
- O tempo de comunicação de  $A \rightarrow B$  pode ser diferente de  $B \rightarrow A$

# Algoritmo de Berkeley

- No algoritmo de Christian, o time server é **passivo** - aguarda as requisições;
- No algoritmo de Berkeley, o time server periodicamente **requisita** aos clientes o horário atual;
- Baseado na **média** dos horários recebidos, o time server **notifica** aos clientes o horário correto;
- Recomendado quando não existe uma máquina com acesso ao **UTC**.

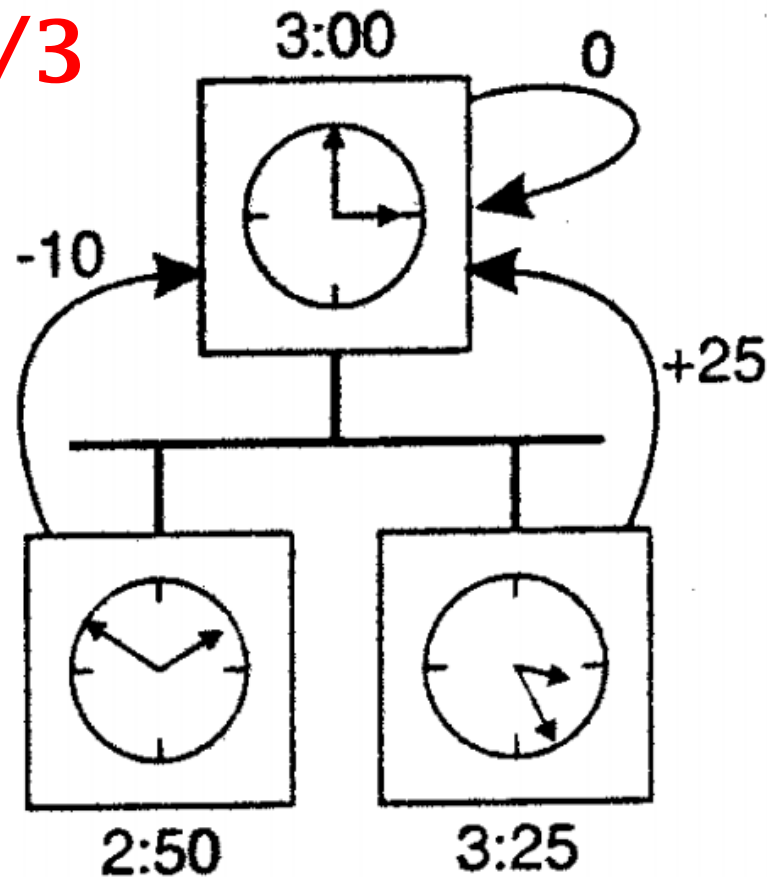
# Algoritmo de Berkeley (cont.)

**Passo: 1/3**



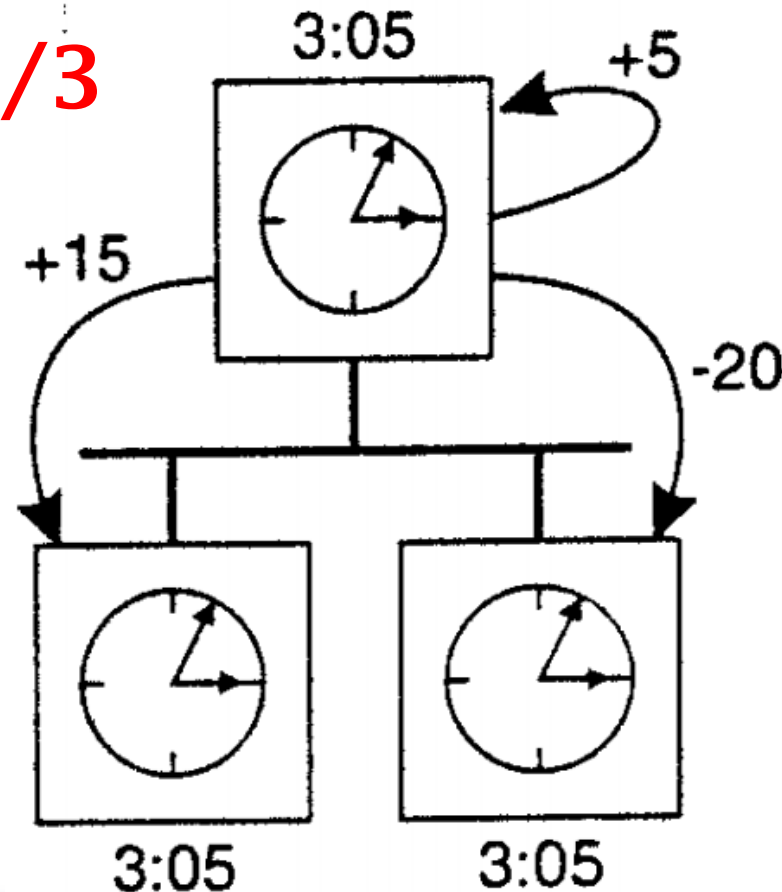
# Algoritmo de Berkeley (cont.)

**Passo: 2/3**



# Algoritmo de Berkeley (cont.)

**Passo: 3/3**





# Atividade 02

- Utilize o algoritmo de Berkley para definir um relógio, onde:
  - Time Server = 20:00
  - Bob = 19:40
  - Alice = 20:35



**PUCPR**  
GRUPO MARISTA

ESCOLA  
**POLITÉCNICA**