

## Agenda

- **Last time**
  - Finish P2P
  - Assignment #3 due
- **This time (D222 Thurs 5-6:15pm, D223 Fri 3-4:15pm)**
  - Chapter 11 (time and global states)
- **Next time (Tuesday Apr 3)**
  - More Chapter 11 Time and global states

CS451: Distributed Systems (Spring 2007)

## Before we start: Assignment #4 and #5

- **Old**
  - Assignment #4 (no book): due Tues Apr 10
  - Assignment #5 (no book): out Tues April 3, proposal due Thurs April 12, due April Tues 24
- **New**
  - Assignment #4 (no book):
    - P2P out now, due Tues April 17 OR
    - "comparable" project proposal due Thurs Apr 5, due Tues Apr 17
  - Assignment #5 (only book): out ~~April 5~~, April 10, due Tues April 24

CS451: Distributed Systems (Spring 2007)

## Before we start: Schedule

Sun	Tues	Thurs	Fri		
	DNS P2P (10) 27	P2P, #3 due, #4 out Time/global(11) 29	Time/global (11) 30		
	Time/global 3	Coordination/agreement (12) PA#4 "Comparable project" due 5			
	Coordination/agreement Assignment#5 Out 10	Transactions (13) 12			
	Transactions (13) Dis transactions (14) PA#4 due 17	Dis transactions (14) Replication (15) 19			
	Replication (15) PA#5 due 24	Class WrapUp 26			

CS451: Distributed Systems (Spring 2007)

## Before we start: "comparable project"

- **I think the P2P file system is very beneficial and fun, so I think you should do that**
- **However, you ARE allowed to pursue a different project**
  - Which project? Let's start a dialogue. Contact me.

CS451: Distributed Systems (Spring 2007)

## Chapter 11: Time and Global States: Roamap

- **Problem definition:** what would happen if there's no attempt to synchronize clocks in a distributed system?
- **Ground rules:** Introduce a general model of computing (process, state, "happens before", HW clock, etc.)
- "approximate synchronization" of clocks
  - (internal) Synchronization in a synchronous system
  - Christian's method (external)
  - Berkeley alg. (internal)
  - NTP (external)
- **Logical time and logical clocks**
  - Happened-before
  - Logical clocks
- **Global states**

CS451: Distributed Systems (Spring 2007)

## Introduction<sub>(1)</sub>

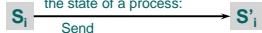
- **Time:** important issue in DSs, must be measured accurately
  - To know when a particular event occurred
- **No global clock**
- Synchronization algorithms developed for:
  - Maintaining the consistency of distributed data
  - Eliminating the processing of duplicate updates
  - Checking the authenticity of a request sent to a server
  - ...

CS451: Distributed Systems (Spring 2007)

### Introduction (3)

- Distributed system = collection of  $N$  processes  $P_i$ ,  $i=1,2,\dots,N$ : Independent, No shared memory
- $S_i$ : state of process  $P_i$  (values of variables or objects in its local operating system environment, e.g., files)
- Process communication: done only by message passing

- **Process  $P_i$ :** Set of actions that can modify the state of a process:
  - Send
  - Receive
  - Operations that transform the state  $S_i$



CS451: Distributed Systems (Spring 2007)

### Introduction (4)

- Event = Occurrence of an action
- Sequence of events in a single process  $P_i$ : unique total ordering denoted by the relation  $\rightarrow_i$

$$e \rightarrow_i e' \text{ IIF event } e \text{ occurs before } e' \text{ at } P_i$$

- History of process  $P_i$  = series of events that take place within it, ordered by the relation  $\rightarrow_i$

$$\text{History}(P_i) = h_i = \langle e_i^0, e_i^1, e_i^2, \dots \rangle$$

- Software clock :  $C_i(t)$

$$C_i(t) = \alpha H_i(t) + \beta$$

Physical clock: hardware clock value of process  $P_i$

CS451: Distributed Systems (Spring 2007)

### Introduction (5)

- $C_i(t)$ : used to timestamp any event
  - **Skew**: instantaneous difference between two clocks; **Drift rate**: how quickly the two clocks diverge
- Synchronization with external sources of highly accurate time: International Atomic Time
- Coordinated Universal Time (UTC):
  - International standard for timekeeping, based on atomic time, and adjusted with astronomical time (leap second)
  - UTC Signals: synchronized and broadcast regularly from land-based radio stations and satellites
    - Radio station WWV in USA
    - GOES and GPS Satellites

CS451: Distributed Systems (Spring 2007)

### Synchronization of a Distributed System

- Synchronization of Physical Clocks
- Synchronization of a Synchronous System
- Network Time Protocol

CS451: Distributed Systems (Spring 2007)

### Synchronization of Physical Clocks (1)

- **External synchronization**: clock  $C_i$  synchronized with an external source of time  $S$  ("accurate to within bound  $D$ ")

$$|S(t) - C_i(t)| < D, i=1,2,\dots,N$$

- **Internal synchronization**: clocks  $C_i$  synchronized with each others ("agree" within the bound  $D$ )

$$|C_i(t) - C_j(t)| < D, i=1,2,\dots,N$$



CS451: Distributed Systems (Spring 2007)

### Synchronization of Physical Clocks (2)

- **Clock's properties:**

- **Correctness**: hardware clock  $H$  is correct if its drift rate falls within a known bound  $\rho > 0$  (e.g.,  $10^{-6}$  seconds/second)

$$(1-\rho)(t-t_0) \leq H(t) - H(t_0) \leq (1+\rho)(t-t_0), t \geq t_0$$

- **Monotonicity**:  $t > t' \Rightarrow C(t) > C(t')$

- Faulty clocks (crash failure, arbitrary failure): don't satisfy these properties

Fault tolerance techniques can mask such failures

CS451: Distributed Systems (Spring 2007)

### Synchronization of a Synchronous System (1)

- Simple case: (internal) synchronization between two processes in a synchronous system
  - (definition of sync system: bounds are known for drift rate, max message transmission, time to execute each step)

If  $T_{Trans}$  is constant:

Set\_Clock( $t + T_{Trans}$ )

Internal synchronization reached

CS451: Distributed Systems (Spring 2007)

### Synchronization of a Synchronous System (2)

- Problem:**  $T_{Trans}$  subject to variation and is unknown
- Solution:** let MAX be the maximum possible value for one message delay, and MIN be the smallest

Scenario	Receiver assign to its clock	Maximal skew of the clock
1	$t + \text{MIN}$	$\text{MAX} - \text{MIN}$
2	$t + \text{MAX}$	$\text{MAX} - \text{MIN}$
3	$t + (\text{MAX} + \text{MIN}) / 2$	$(\text{MAX} - \text{MIN}) / 2$

CS451: Distributed Systems (Spring 2007)

### Synchronization of a Synchronous System (3)

- General case:** optimum bound that can be achieved when synchronizing N clocks =  $U(1 - 1/N)$
- Practical distributed systems:** asynchronous, there is no upper bound MAX

$T_{Trans} = \text{MIN} + x, x > 0$  and value of  $x$  unknown

CS451: Distributed Systems (Spring 2007)

### Synchronization of a Synchronous System (4)

- Cristian's method (1989): external synchronization**
  - Use of a time server: connected to a device that receives signals from a source of UTC

Process P assign to its clock  $t + T_{round} / 2$

**Limit:** server may fail

**Solution:** time should be provided by a group of synchronized servers

CS451: Distributed Systems (Spring 2007)

### Synchronization of a Synchronous System (5)

- Example of Cristian's method**

- Initially, client  $t = 10$ , server = 15**
  - Assume  $m_r$  takes 2 seconds and takes 4 seconds
  - What does client set its time to?  $mt$ 
    - Server timestamp + (client recpt time - client send time) / 2
    - $T = (15+2) + (2+4)/2 = 17 + 3 = 20$  (server thinks the time is 21, right?)
  - Is time monotonic? Not necessarily. What do you do?

CS451: Distributed Systems (Spring 2007)

### Synchronization of a Synchronous System (5)

- Berkeley algorithm (1989): internal synchronization**
  - Developed for collections of computers running Berkeley UNIX
  - Coordinator computer (master):** polls the other computers whose clocks are to be synchronized (slaves)
    - The slaves send back their clock values
    - The master estimates their local clock times by observing the round-trip times: averages the values obtained including its own clock's reading
    - The master sends the amount by which each individual slave's clock requires adjustment
  - Master fails:** another is elected to take over and function exactly as its predecessor; also: a "fault-tolerant average": don't consider outliers

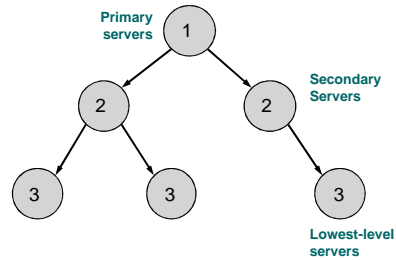
CS451: Distributed Systems (Spring 2007)

## Network Time Protocol (1)

- Cristian's method & Berkeley algorithm: designed for use within intranets
- Network Time Protocol :
  - Architecture for time service and a protocol to distribute time information over the Internet (potentially large delays)
  - Service NTP : provided by a network of servers located across the Internet
    - **Primary servers**: connected directly to an UTC time source (radio, satellite)
    - **Secondary servers**: synchronized with primary servers

CS451: Distributed Systems (Spring 2007)

## Network Time Protocol (2)



CS451: Distributed Systems (Spring 2007)

## Network Time Protocol (3)

- NTP servers synchronize with each other (all via UDP):
  - **Multicast**:
    - Used on high-speed LAN
    - One or more servers periodically multicasts the time to the servers running in other computers connected to the LAN
    - Receiving computers set their clocks assuming a small delay

This mode can achieve only relatively low accuracies

CS451: Distributed Systems (Spring 2007)

## Network Time Protocol (4)

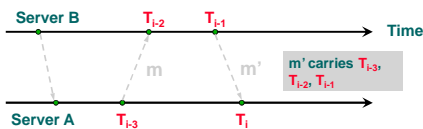
- Synchronization modes (cont'd):
  - **Procedure-call**: (similar to Cristian's algorithm)
    - One server accepts requests from other computers
    - Replies by sending its timestamp (current clock reading)

This mode is suitable where higher accuracies are required

CS451: Distributed Systems (Spring 2007)

## Network Time Protocol (5)

- Synchronization modes (cont'd):
  - **Symmetric mode**:
    - A pair of processes operate in symmetric mode and exchange messages bearing timing information



CS451: Distributed Systems (Spring 2007)