Figure 10.1  Skew between computer clocks in a distributed system

Network
Figure 10.2  Clock synchronization using a time server

Figure showing a time server synchronizing with a process. The diagram includes nodes labeled as follows:
- $m_r$ (representing the process)
- $m_t$ (representing the time server)
- $p$ (representing another process)
- Time server, $S$
Figure 10.3  An example synchronization subnet in an NTP implementation

Note: Arrows denote synchronization control, numbers denote strata.
Figure 10.4  Messages exchanged between a pair of NTP peers

<table>
<thead>
<tr>
<th>Server B</th>
<th>$T_{i-2}$</th>
<th>$T_{i-1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server A</td>
<td>$T_{i-3}$</td>
<td>$T_i$</td>
</tr>
</tbody>
</table>

$m$ and $m'$ represent message exchanges.
Figure 10.5  Events occurring at three processes

Figure 10.6 Lamport timestamps for the events shown in Figure 10.5.
Figure 10.7  Vector timestamps for the events shown in Figure 10.5
Figure 10.8  Detecting global properties

a. Garbage collection

b. Deadlock

c. Termination
Figure 10.9 Cuts

Physical time

Inconsistent cut
Consistent cut
Figure 10.10  Chandy and Lamport’s ‘snapshot’ algorithm

*Marker receiving rule for process $p_i$*

On $p_i$’s receipt of a *marker* message over channel $c$:

- *if* ($p_i$ has not yet recorded its state) it
  - records its process state now;
  - records the state of $c$ as the empty set;
  - turns on recording of messages arriving over other incoming channels;

- *else*
  - $p_i$ records the state of $c$ as the set of messages it has received over $c$ since it saved its state.

*end if*

*Marker sending rule for process $p_i$*

After $p_i$ has recorded its state, for each outgoing channel $c$:

- $p_i$ sends one marker message over $c$

  (before it sends any other message over $c$).
Figure 10.11  Two processes and their initial states

$p_1$ \hspace{1cm} $c_2$ \hspace{1cm} $p_2$

$\leftarrow c_1$

<table>
<thead>
<tr>
<th></th>
<th>$p_1$</th>
<th>$c_2$</th>
<th>$p_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>account</td>
<td>$$1000$</td>
<td>(none)</td>
<td>$$50$</td>
</tr>
<tr>
<td>widgets</td>
<td>(none)</td>
<td>(none)</td>
<td>$2000$</td>
</tr>
</tbody>
</table>

Figure 10.12 The execution of the processes in Figure 10.11

1. Global state $S_0$

   $<$1000, 0$>$ $p_1$ $c_2$ (empty) $p_2$ $<$50, 2000$>$
   $c_1$ (empty)

2. Global state $S_1$

   $<$900, 0$>$ $p_1$ $c_2$ (Order 10, $100), M$ $p_2$ $<$50, 2000$>$
   $c_1$ (empty)

3. Global state $S_2$

   $<$900, 0$>$ $p_1$ $c_2$ (Order 10, $100), M$ $p_2$ $<$50, 1995$>$
   $c_1$ (five widgets)

4. Global state $S_3$

   $<$900, 5$>$ $p_1$ $c_2$ (Order 10, $100)$ $p_2$ $<$50, 1995$>$
   $c_1$ (empty)

($M = \text{marker message}$)
Figure 10.13  Reachability between states in the snapshot algorithm

actual execution $e_0, e_1, ...$

$S_{init}$ recording begins

pre-snap: $e'_0, e'_1, ..., e'_{R-1}$

$S_{snap}$

post-snap: $e'_R, e'_{R+1}, ...$

recording recording begins

recording ends

$S_{final}$
Figure 10.14  Vector timestamps and variable values for the execution of Figure 10.9

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure1014.png}
\caption{Vector timestamps and variable values for the execution of Figure 10.9}
\end{figure}
Figure 10.15 The lattice of global states for the execution of Figure 10.14

$S_{ij}$ = global state after $i$ events at process 1 and $j$ events at process 2
Figure 10.16 Algorithms to evaluate possibly $\phi$ and definitely $\phi$

1. Evaluating possibly $\phi$ for global history $H$ of $N$ processes

$L := 0$

$States := \{ (s_1^0, s_2^0, \ldots, s_N^0) \}$

$while (\forall S \in States (\phi(S) = False))$

$L := L + 1$

$Reachable := \{ S': S' \text{ reachable in } H \text{ from some } S \in States \land level(S') = L \}$

$States := Reachable$

$end while$

$output "possibly \phi";$

2. Evaluating definitely $\phi$ for global history $H$ of $N$ processes

$L := 0$

$if (\phi(s_1^0, s_2^0, \ldots, s_N^0)) then States := \{ \} else States := \{ (s_1^0, s_2^0, \ldots, s_N^0) \}$

$while (States \neq \{\})$

$L := L + 1$

$Reachable := \{ S': S' \text{ reachable in } H \text{ from some } S \in States \land level(S') = L \}$

$States := \{ S \in Reachable: \phi(S) = False \}$

$end while$

$output "definitely \phi";$
Figure 10.17 Evaluating definitely $\phi$ 

Level 0

1

2

3

4

5

$F = (\phi(S) = \text{False})$; $T = (\phi(S) = \text{True})$