Real-Time Systems

Part 2: Time and Clocks

Introduction: Time and Order

1. The constants of physics are defined in relation to the standard of time: the physical second (e.g., speed: m/s)

   *The global time in cyber-physical real-time systems should be also based on the metric of the physical second.*

2. In distributed systems, the nodes must ensure that the events are processed in the same consistent order (preferably in the temporal order in which the events occurred).

   *A global time base helps to establish such a consistent temporal order on the basis of the time-stamps of the events.*
**Temporal Order**

- The continuum of Newtonian real time can be modeled by a directed timeline consisting of an infinite, dense and ordered set \( \{T\} \) of *instants* \( i \) (points in time).
- The section on the time line between two instants is called *duration* \( d \).
- **Events** \( e \) take place at an instant of time (but have no duration).
- Events that occur at the same instant are said to occur *simultaneously*.
- Instants are totally ordered.
- Events are partially ordered (additional criteria are required to totally order events, such as the node at which the event occurred).
Causal Order

• For real-time applications, the \textit{causal dependencies} among events $e$ are of interest.

• The \textit{temporal order} of two events is \textit{necessary}, but \textit{not sufficient}, for their causal order.

• \textit{Causal order} is \textit{more} than \textit{temporal order}. 
Digital Physical Clocks

• In digital physical clocks, a physical oscillation mechanism that periodically increases a counter is used to measure time.

• The periodic event is called a **microtick**.

• The duration **between two consecutive** microticks is called a **granule** of the clock.

• The granularity of a digital clock leads to a **digitization error** in time measurement.
Digital Physical Clocks: Phased-Locked Loop (PLL)

• Typical frequencies of crystal oscillators: kHz ... MHz
• CPUs, mobile phones, etc. require clock signals with frequencies in the GHz range
• Precise multiplication of the frequency of crystal oscillators is required
  → Phase-Locked Loop (PLL)

„A PLL is a circuit which synchronizes the frequency of the output signal generated by an oscillator with the frequency of a reference signal by means of the phase difference of the two signals.“

(J. Encinas)
Digital Physical Clocks: Reference Clock & Absolute Time-Stamp

- A reference clock is a clock $z$ that runs at frequency $f_z$ and which is in perfect sync with the international standard of time.
- $1/f_z$ is the granularity $g_z$ of clock $z$.
- The granularity of a clock $k$ is given by the number of microticks of the reference clock $z$ between two subsequent microticks of the clock $k$.
- An absolute time-stamp of an event is the time of its occurrence measured by the reference clock.
- The duration between two events $e$ is measured counting the microticks of the reference clock.
- The temporal order of events that occur between two consecutive events of the reference clock cannot be reestablished from their absolute time-stamps.
Digital Physical Clocks: Clock Drift

- The drift rate $\rho$ of a physical clock $k$ with respect to a reference clock $z$ is defined as:
  $$\rho = | \frac{f_k}{f_z} - 1 |$$

- A perfect clock has a drift rate $\rho$ of 0

- Drift rates vary due to changes in ambient temperature or ageing of crystal

- The data sheet of a resonator defines a maximum drift rate $\rho_{\text{max}}$.

- Due to the drift rate, clocks deviate from the reference clock over time if not resynchronized.

<table>
<thead>
<tr>
<th>Clock Type</th>
<th>Drift Rate [s/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quartz</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>Pendulum</td>
<td>$10^{-6}$</td>
</tr>
<tr>
<td>Atom</td>
<td>$1.5 \times 10^{-14}$</td>
</tr>
<tr>
<td>Atom (laser-cooled)</td>
<td>$10^{-15}$</td>
</tr>
</tbody>
</table>
Digital Physical Clocks: Failure Modes

- A digital physical clock can exhibit two types of failures:
  - The counter value could become erroneous (e.g. due to a overflow)
  - The drift rate could depart from the specified drift rate
Digital Physical Clocks: Offset

- Offset: The offset of two clocks is the time difference between the respective microticks of the two clocks – measured in the number of microticks of the reference clock.

[Diagram showing three clocks: Clock A, Clock B, and Reference Clock, with an arrow indicating the time scale labeled 'Time t'. There is an arrow labeled 'Offset' pointing to the difference between Clock A and Clock B, with a note 'here 3' indicating the offset value.]
Digital Physical Clocks: Precision & Internal Synchronization

- **Precision**: The precision $\Pi$ denotes the maximum *offset* of respective microticks of an ensemble of clocks in a duration of interest and measured in microticks of the reference clock.

- Because of the drift rate $\rho$, an ensemble of clocks will drift apart if not resynchronized periodically. The process of mutual resynchronization is called *internal synchronization*. 
Digital Physical Clocks: Accuracy

• **Accuracy**: The accuracy denotes the maximum *offset* of a given clock from the external time reference during a duration of interest.

• To keep a clock within a bounded accuracy it must be periodically resynchronized. This process is called *external synchronization*.

• Note: If all clocks of an ensemble are externally synchronized with an accuracy $A$, then the ensemble is also internally synchronized with a precision of $\leq 2A$ → the converse is not true
Digital Physical Clocks: Time Standards

• A time base origin is called the epoch.
• Three time standards are relevant for (distributed) real-time computer systems:

  1. The International Atomic Time (TAI)
     Defines the second as the duration of 9,192,631,770 periods of the radiation of a specified transition of the cesium atom 133. Epoch: January 1, 1958 at 00:00 h (GMT). TAI is a chronoscopic timescale – a timescale without discontinuities)

  2. The Universal Time Coordinated (UTC)
     Replaced GMT (Greenwich Mean Time) in 1972. Not chronoscopic (leap seconds – one-second adjustment to keep the UTC close to the mean solar time).

  3. UNIX (or POSIX) Time
     Seconds since January, 1st 1970 (UTC) not counting leap seconds.
Global Time

• If all clocks of a distributed system are internally synchronized with precision \( \Pi \), each \textbf{n-th} microtick of a clock can be interpreted as a \textit{macrotick} to approximate a \textit{global time}.

![Diagram of microticks and macroticks]

• The global time is called \textit{reasonable} when the internal synchronization error is less than the duration between two consecutive macroticks (i.e. the global time-stamps for a single event can differ by at most 1 tick).
Interval Measurement

• An interval is delimited by two events ($e_{\text{start}}$ and $e_{\text{stop}}$).

• Interval measurement can be affected by:
  – the synchronization error
  – the digitalization error

• If the global time is reasonable, the interval error is always less than $2g$, where $g$ is the granularity of the global time.
Summary: Fundamental Limits of Time Measurement

• In a distributed real-time system with a global time base (of granularity $g$), the following fundamental limits of measurements can be defined:

1. The time-stamp of an event observed by two nodes can differ by one tick. This, however, is not sufficient to recover the temporal order of the events.

2. The true duration $d$ of an observed interval is bounded by +/- $2g$.

3. The temporal order of events can be recovered from their time-stamps if the difference between their time-stamps is equal or greater $2g$.
Internal Clock Synchronization

- Internal synchronization ensures that the global ticks of all nodes occur within a specified precision $\Pi$ (despite the drift rate of each node).

- Resynchronization interval is called $R_{int}$

- The convergence function $\Phi$ denotes the offset after synchronization.

- The drift offset $\Gamma$ indicates the maximum offset before synchronization.

- Synchronization condition: $\Phi + \Gamma \leq \Pi$
Internal Clock Synchronization: Non-fault tolerant algorithms

• Central Master Synchronization
  1. Master sends synchronization message with value of its time counter to all other nodes
  2. Slave records time-stamp when receiving synchronization message
  3. Slave computes deviation of its clock by taking the message transport latency into account and corrects its clock.

• Φ is determined by the fastest and slowest message transmission times (the latency jitter ε)

• The precision of the central master synchronization is:
  \[ \Pi_{\text{central}} = \varepsilon + \Gamma \]

• Not fault tolerant: Failing master ends synchronization
Internal Clock Synchronization: Fault tolerant algorithms

• Standard procedure of fault-tolerant clock synchronization algorithms:

  - **Phase I**
    - Acquisition of state of the global time counters of all other nodes

  - **Phase II**
    - Analysis of collected data for error detection
    - Execution of convergence function to calculate correction value

  - **Phase III**
    - Adjustment of local time counter
Internal Clock Synchronization: Fault tolerant algorithms

- Main term affecting the synchronization precision is the jitter $\varepsilon$.
- Delay jitter depends on system level of creation and interpretation of time synchronization message:

<table>
<thead>
<tr>
<th>System Level</th>
<th>Jitter Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>500 ms – 5 ms</td>
</tr>
<tr>
<td>Kernel</td>
<td>10 ms – 100 ms</td>
</tr>
<tr>
<td>Hardware</td>
<td>&lt; 1 ms</td>
</tr>
</tbody>
</table>

- It is not possible to internally synchronize the clocks of an ensemble of $N$ nodes to a better precision than:
  \[ \Pi = \varepsilon \cdot (1 - 1/N) \]
Internal Clock Synchronization: Cristian’s Algorithm

• S requests the time from M
• On reception of the request from S, M prepares a response containing the time $T$ from its own clock
• S then sets its time to be $T + t_r/2$
Byzantine Errors

- **Byzantine** errors are errors where a component of a system fails in an **arbitrary** way (e.g., producing inconsistent outputs)

- Clock synchronization in the presence of Byzantine errors can only be guaranteed if: 
  
  \[ N \geq (3k + 1) \]

  where \( N \) is the total number and \( k \) the number of Byzantine faulty clocks.
Internal Clock Synchronization:
State Correction vs. Rate Correction

- Based on correction term calculated by the convergence function the local time can be adjusted using:

  **State Correction**
  - Correct local time immediately
  - Problem: Discontinuity in time (e.g. if clock is set backward, the same time value is reached twice)

  **Rate Correction**
  - Correct the rate (speed) of the clock
  - Digital implementation: Change number of microticks per macrotick
  - Analog implementation: Change parameters of the crystal oscillator
External Clock Synchronization

- External clock synchronization links the global time of a distributed system to an external time reference.
- Typically a designated node of the cluster, the time gateway, receives the time from the external time reference, computes the rate correction and forwards it to the nodes.
External Clock Synchronization: Time Formats

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Epoch</th>
<th>Format</th>
<th>Chronoscopic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Time Protocol (NTP)</td>
<td>January, 1st 1900, 00:00 h</td>
<td>4 Bytes for seconds 4 Bytes for fraction of seconds</td>
<td>No (based on UTC and therefore on leap seconds)</td>
</tr>
<tr>
<td>IEEE 1588</td>
<td>January, 1st 1970, 00:00 h</td>
<td>Seconds based on TAI Fraction of a second in nano seconds</td>
<td>Yes</td>
</tr>
</tbody>
</table>

- Time-Triggered Architecture (TTA) uses a mixture of NTP and IEEE 1588 as time format (full seconds based on TAI and parts of seconds as binary fraction) $\rightarrow$ chronoscopic and fully conformant to the dual system
External Clock Synchronization: IEEE 1588

- IEEE 1588-2002 defines the Precision Time Protocol (PTP)
- Accuracy of < 1μs via Ethernet networks

From: Precision Clock Synchronization, White Paper, Hirschmann
Example: GPS

- The Global Positioning System (GPS) was developed by the US Department of Defense.
- Two services are provided:
  - Precise Positioning Service (PPS) – for military purposes.
  - Standard Positioning Service (SPS) – for civilian purposes. Precision was purposely degraded (Selective Availability SA) before May 2, 2000.
- Accuracies in the range of cm possible with Differential Global Positioning System (DGPS).
Example: High-Speed Printing

- Paper runs at speeds of up to 100 km/h
- All printing stations (for different colours) must be synchronised so that the deviation between individual prints is less than 1μm
- Station rollers can be synchronised by coupling them mechanically by shafts
- Better: precise timing via synchronised clocks in each station
Literature

- Official U.S. Government information about GPS
  http://www.gps.gov/
- http://www.ieee1588.com
- http://www.ptb.de/cms/presseaktuelles/uhrzeitapplikation.html
Backup
CERN’s White Rabbit Project
(based on “White Rabbit: a PTP Application for Robust Sub-nanosecond Synchronization”- Maciej Lipinski, et.al., ISPCS 2011 Munich)

- Goal: Develop an alternate timing and control system for the General Machine Timing at CERN
- Synchronization of up to 2000 nodes with sub-nanosecond accuracy, an upper bound on frame delivery and a very low data loss rate
- Based on and compatible with Ethernet (IEEE 802.3), Synchronous Ethernet (ITU-T Std. G.8262, 2007) and IEEE 1588-2008.
- For sub-nanosecond EVERYTHING matters: oscillators; media, PHY, board asymmetry, temperature, …
**\( \pi/\Delta \) Precedence**

- An event set \( \{E\} \) is called \( \pi/\Delta \) precedent if it fulfills the following condition for any two elements \( e_i \) and \( e_j \) of this set:
  \[
  |z(e_i) - z(e_j)| \leq \pi \lor |z(e_i) - z(e_j)| > \Delta
  \]
  where \( z \) is the reference clock, \( \pi \) and \( \Delta \) are durations (\( \pi \ll \Delta \)).

- \( \pi/\Delta \) Precedence: A subset of events that happen about the same time (within \( \pi \)) are separated by at least \( \Delta \) from another subset.