— Our assumption: Newtonian universe with an *absolute time*.

— There is a fictitious perfect clock that keeps *absolute* or *real* time.

> "Absolute, true and mathematical time, of itself, and from its own nature, flows equably, without relation to anything external" — I. Newton, *Philosophiae Naturalis Principia Mathematica*, 1687.

— This Newtonian assumption is false but a useful practical basis for our arguments.

— A clock is a mapping,

\[
C_i : \text{real time} \rightarrow \text{clock time}.
\]

— We can also go in the other direction:

\[
c_i : \text{clock time} \rightarrow \text{real time}.
\]

— Our convention: use lower-case for real time and upper case for clock time.
Real clocks drift.

\[ \rho = \max_{t, \Delta} \left| \frac{C_i(t + \Delta) - C_i(t)}{\Delta} - 1 \right|. \]
Clock Synchronization

Two clocks $i, j$ are said to be synchronized at clock time $T$ if, for some specified $\delta > 0$,

$$|c_i(T) - c_j(T)| < \delta$$

An alternative (and almost equivalent) definition is to say that they are synchronized at real time $t$ if, for some specified $\delta > 0$, we have

$$|C_i(t) - C_j(t)| < \delta.$$
Clock Synchronization

Mutual Synchronization:

— Each clock periodically checks its own time with respect to that of the other clocks.
— It adjusts itself (slows down or speeds up) in reaction to this information.
— In the absence of faults, clock synchronization is trivial.
— If clocks suffer Byzantine failure, multiple cliques can form, which then may drift out of sync with one another.
— We will consider algorithms for fault-tolerance clock synchronization implemented in
  * Hardware.
  * Software.
Hardware Synchronization

Basic component is a phase-locked loop.

![Comparator Diagram](image)

V.C.O. = Voltage-Controlled Oscillator

The output voltage of the comparator at any time $t$ is proportional to the difference between the phase of the signal input, $\phi_i(t)$, and that of the VCO, $\phi_r(t)$:

$$v_c(t) = K_c\{\phi_i(t) - \phi_r(t)\},$$

where $K_c$ is called the comparator gain factor.

Assumptions:

— Signal propagation time is negligible.

— The interconnection network is fully connected.
Question: How to pick the reference signal?

* Possibility 1: Pick the fastest signal.
* Possibility 2: Pick the median signal.
* Possibility 3: Pick signal number $k$ for some suitable fixed $k$.

None of these approaches will be guaranteed to work!
— *Condition C1*: If all clocks in $G_1$ ($G_2$) use as reference a signal that is faster\(^1\) (slower) than any clock in $G_2$ ($G_1$), then there must be at least one clock in $G_2$ ($G_1$) which uses as reference either the slowest (fastest) clock in $G_1$ ($G_2$) or a signal faster (slower) than the slowest (fastest) clock in $G_1$ ($G_2$). This condition ensures that multiple non-overlapping cliques do not form.

— *Condition C2*: If a good clock $x$ uses as reference the signal of a faulty clock $y$, there must exist nonfaulty clocks $z_1$ and $z_2$ such that $z_1$ is faster than, or equal to, $y$, and $y$ is faster than, or equal to, $z_2$. Either $z_1$ or $z_2$ may be $x$ itself.

\(^1\)We say that signal $a$ is faster than signal $b$ if $a$ occurs earlier than $b$. 

**Approach:** Pick a reference based on the perceived position of each clock in the tick-sequence. A faster clock will tend to pick a slower clock as a reference.

— **C2:** Condition C2 will be satisfied if we avoid picking either the fastest $m$ or the slowest $m$ clocks in the system.

— **C1:** Satisfying C1 is a bit harder: it can be shown that if we break the good clocks into any two nonempty groups $G_1, G_2$ such that every clock in $G_1$ is faster than any clock in $G_2$, we must have

$$\max_{i \in G_1} f_{p(i)}(N, m) - \min_{i \in G_2} f_{p(i)}(N, m) \geq m.$$
Case 1. $\max_{x \in G_1} f_{p(x)}(N, m) \leq \|G_1\| + m$: If all the faulty clocks appear to the clocks in $G_1$ to be faster than any $G_2$ clock, there will be no reference to any clock outside $G_1$. Hence, at least one clock in $G_2$ must be assured of a reference to a clock in $G_1$. This implies that

$$\min_{y \in G_2} f_{p(y)}(N, m) \leq \|G_1\|$$

Case 2. $\min_{y \in G_2} f_{p(y)}(N, m) \geq \|G_1\| + 1$: By reasoning similar to that in Case 1, this requires that there be at least one clock in $G_1$ whose reference is drawn from $G_2$. But, to be sure of that, we require

$$\max_{x \in G_1} f_{p(x)}(N, m) \geq \|G_1\| + m + 1$$

Case 3. $\max_{x \in G_1} f_{p(x)}(N, m) > \|G_1\| + m$ or $\min_{y \in G_2} f_{p(y)}(N, m) < \|G_1\| + 1$: In such a case, no potential exists for the formation of nonoverlapping cliques.
— If \( \max_{x \in G_1} f_p(x)(N, m) \leq \|G_1\| + m \), then \( \min_{y \in G_2} f_p(y)(N, m) \leq \|G_1\| \).

— Else if \( \min_{y \in G_2} f_p(y)(N, m) \geq \|G_1\| + 1 \), then \( \max_{x \in G_1} f_p(x)(N, m) \geq \|G_1\| + m + 1 \).

It follows from these requirements that
\[
\max_{x \in G_1} f_p(x)(N, m) - \min_{y \in G_2} f_p(y)(N, m) \geq m
\]

But we require that any reference be neither the first \( m \) nor the last \( m \) clocks, i.e., that any reference lie in the interval of positions \( m + 1, \ldots, N - m \). So, we must have:

\[
(N - m) - (m + 1) \geq m \Rightarrow N \geq 3m + 1
\]
To ensure that C1 and C2 are satisfied for *every* partition $G1, G2$, it is sufficient to use the following functions $f_i$:

$$f_i(N, m) = \begin{cases} 
2m & \text{if } i < N - m \\
 m + 1 & \text{otherwise}
\end{cases}$$