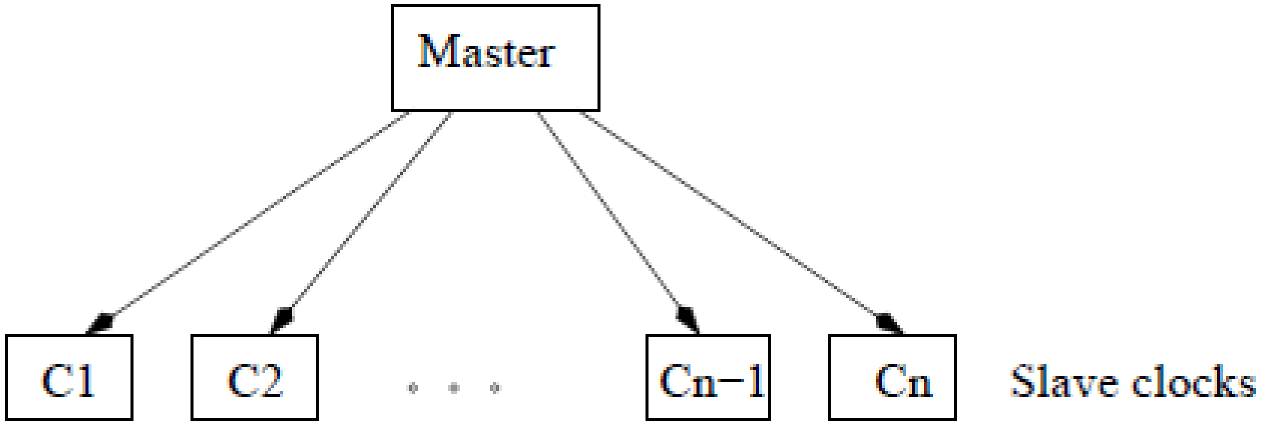
Clocks in Real Time systems

- Clocks in a system are useful for two main purposes
- Determining timeouts
  - To determine failure of a task due to missing of deadline
  - For communication
  - As indicators for possible transmission faults or delays
- Time stamping
  - Message communication among tasks

Clock synchronization

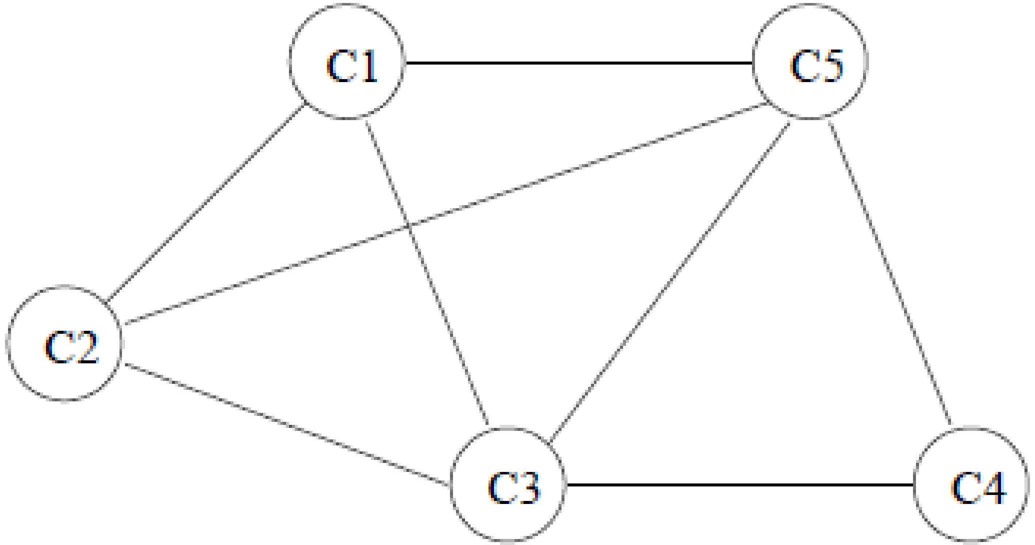
- Goal is to make all clocks in the network to agree on their time values.
- World time standard is called *Universal Coordinated Time (UTC).*
- UTC is based on the *International Atomic Time (TAI)* maintained at Paris by averaging a number of atomic clocks from laboratories around the world
- When the clocks of a system are synchronized with respect one of the clocks of the system, it is called *internal clock synchronization.*
- When synchronization of a set of clocks with external clocks , it is called *external synchronization.*

Centralized clock synchronization



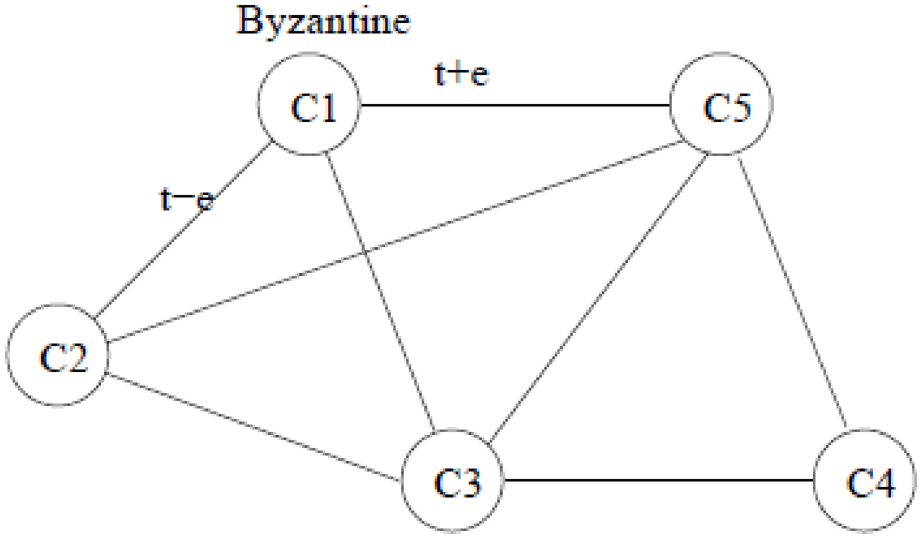
- The server broadcasts its time to all other clocks for synchronization after every ΔT time interval.
- Let the maximum rate of drift between two individual clocks be ρ
- Suppose clocks are synchronized after every ΔT interval, the drift of any clock from the master clock would be bounded by ρΔT
- Maximum drift between any two clocks will be limited to 2ρΔT

Distributed clock synchronization



- Let there be n clocks in a system
- Each clock periodically broadcasts its time value at the end of certain time interval.
- Assume that the clocks in the system are required to be synchronized within Ɛ time units of each other
- If a clock receives a time broadcast that differs from its own time value by Ɛ time units, then it can determine that the sending clock must be a bad one and safely ignore the received time values

- Bad clocks exhibit large drifts – drifts larger than the manufactured specified tolerance.
- A Byzantine clock is a two-faced clock.


Byzantine
C1
t+e
C5
t−e
C2
C3
C4