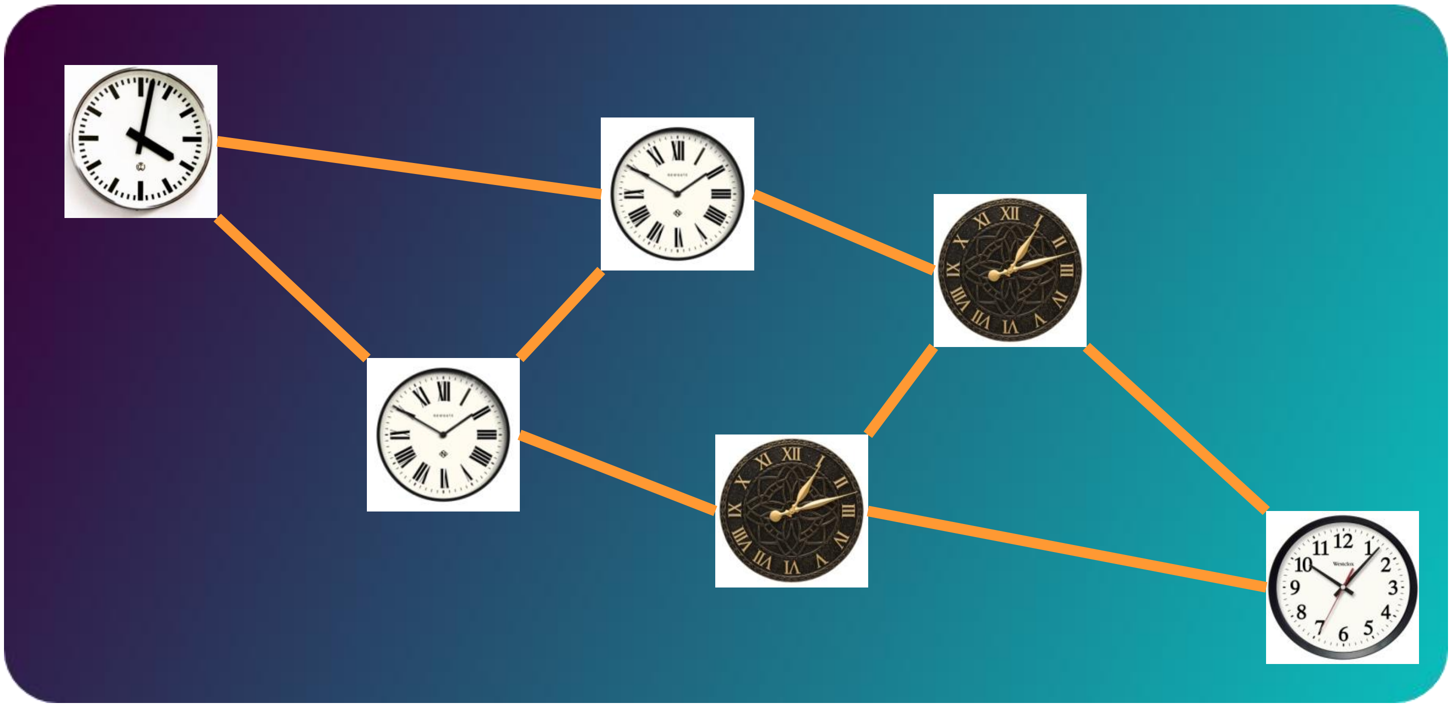
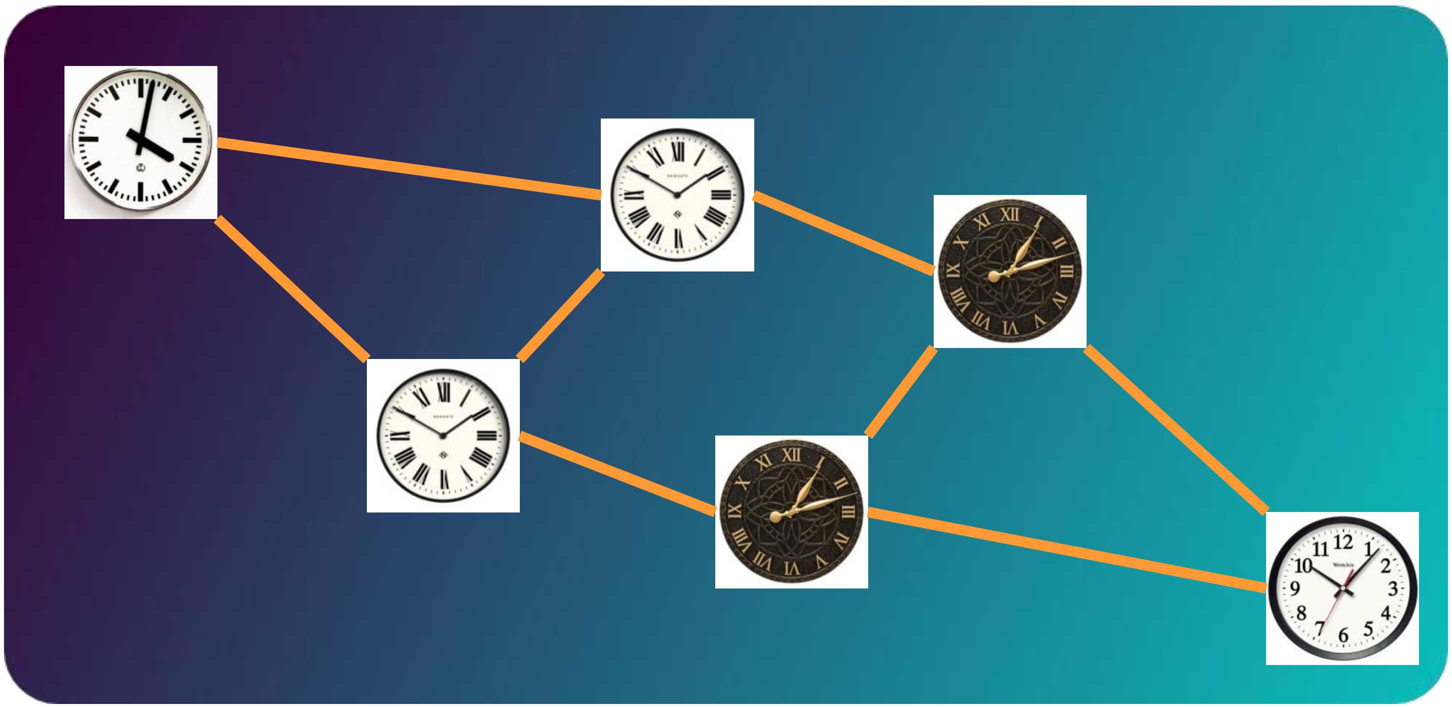
Model & Synchronizing w/o Faults



Model



network G = (V,E)

node = state machine with hardware clock

edge = communication link (message passing)

Model: What Nodes Can Do

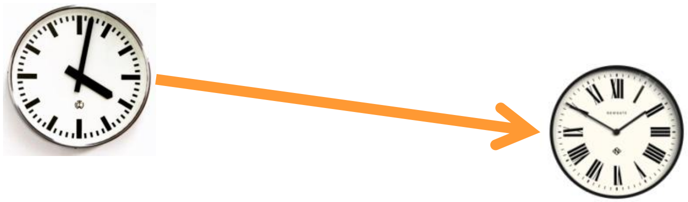
- arbitrary deterministic computations
- computation times satisfy (known) bounds
- hardware clock runs at rates between 1 and ϑ:

$$t - t' \leq H_v(t) - H_v(t') \leq \vartheta(t - t')$$

goal: compute logical clocks such that

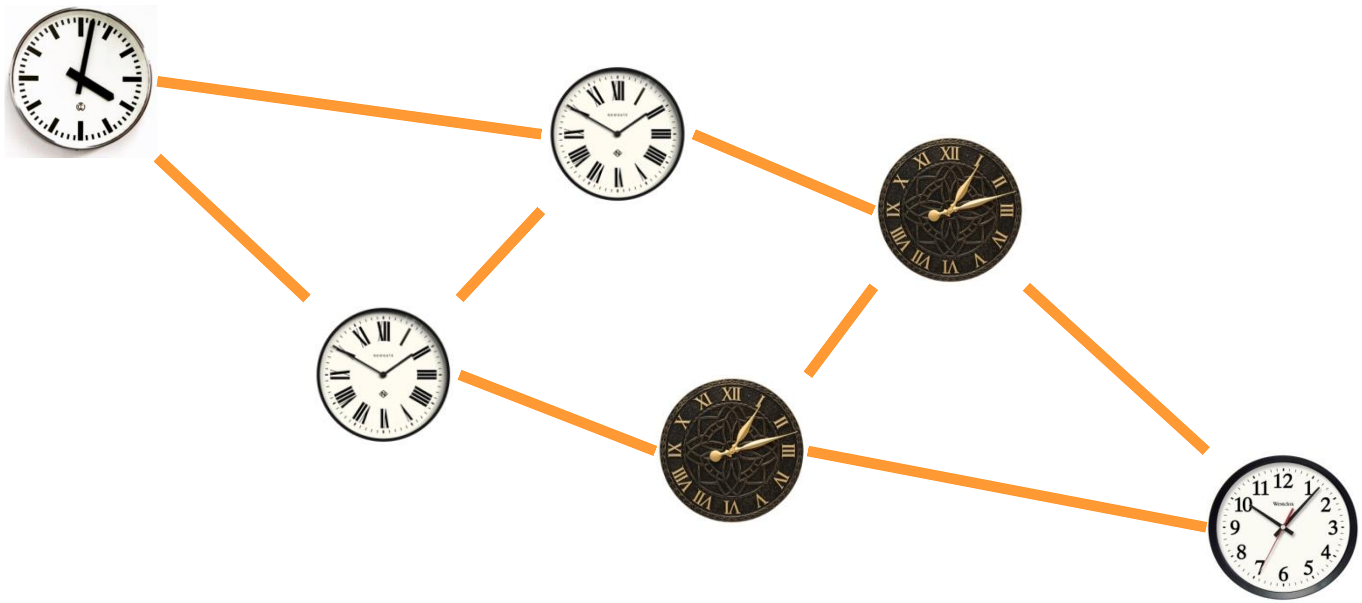
$$H_v(t) - H_v(t') \leq L_v(t) - L_v(t') \leq (1 + \mu)(H_v(t) - H_v(t'))$$

Model: How Communication Works



- communication by message passing
- messages sent as result of computations
- transmission times satisfy (known) bounds
- (end-to-end) delay, i.e., message transmission + computation time, is between d-u and d
- delay d, uncertainty u, and drift ϑ are known and can be used in computations

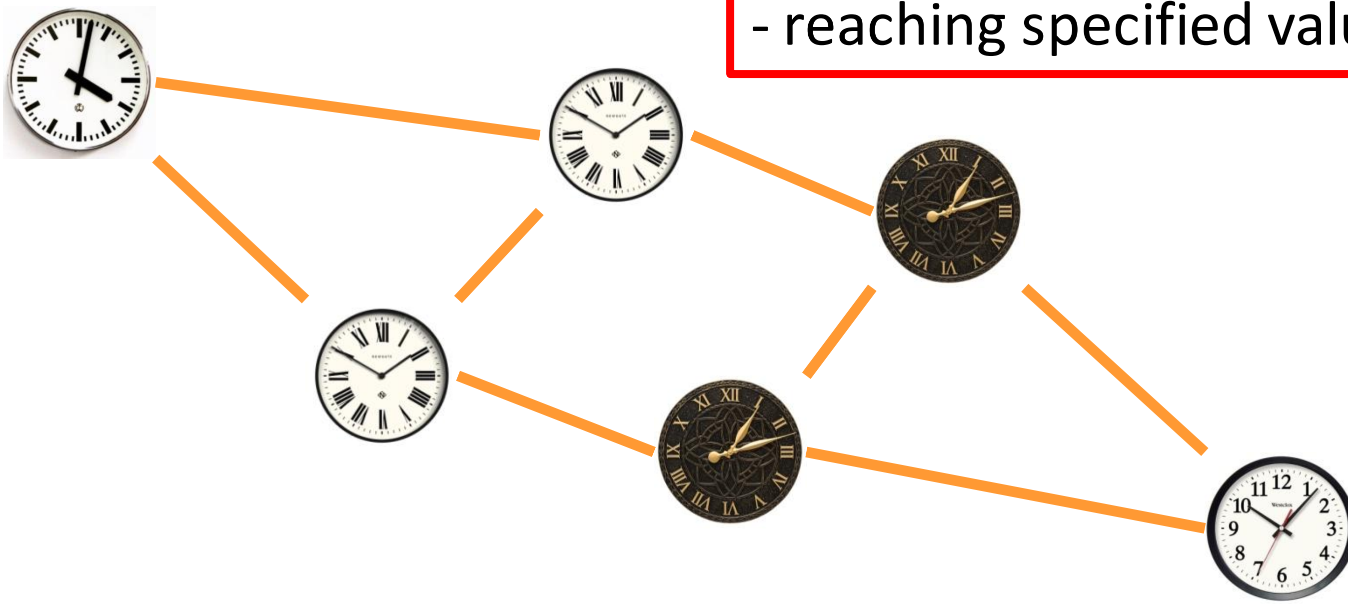
Model: Executions



- fix network G = (V,E) and algorithm
- fix H_v (and a wake-up time) for each node
- (inductively) fix delay of each sent message
- this specifies an execution

Model: Executions

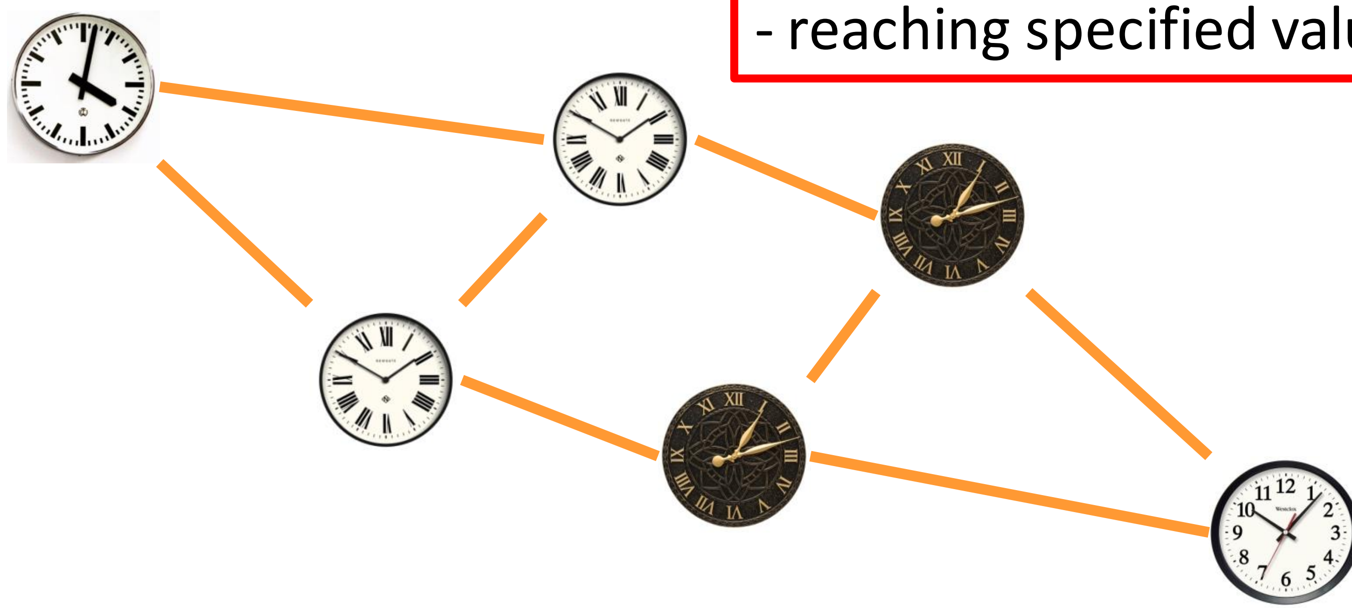
*event-driven; events are:
- waking up (initialization)
- receiving a message
- reaching specified value of H_v



- fix network G = (V,E) and algorithm
- fix H_v (and a wake-up time) for each node
- (inductively) fix delay of each sent message
- this specifies an execution*

Model: Executions

*event-driven; events are:
- waking up (initialization)
- receiving a message
- reaching specified value of H_v



IMPORTANT NOTICE:

Delays include computations, so the time a message is “received” equals the time when any immediately triggered messages are sent!

Example: Max Algorithm

Algorithm 4 Basic Max Algorithm. Parameter $T \in \mathbb{R}^+$ controls how frequently messages are sent. The code lists the actions of node v at time t and provides getL().

if $t = 0$ (i.e., v just woke up) **then**
    $h \leftarrow$ getH()
    $\ell \leftarrow h$ ▹ initialize $L_v(0)$ to $H_v(0)$
end if
if received $\langle \ell' \rangle$ at time t and $\ell' >$ getL() **then**
    $h \leftarrow$ getH()
    $\ell \leftarrow \ell'$ ▹ increase logical clock to received value
end if
if getL() $= kT$ for some $k \in \mathbb{N}$ **then**
    send $\langle kT \rangle$ to all neighbors
end if
procedure getL() ▹ returns $L_v(t)$
    return $\ell +$ getH() $- h$ ▹ logical clock increases at rate $\frac{dH_v}{dt}$
end procedure

- getH() returns $H_v(t)$
- all nodes are assumed to wake up at time 0

Example: Max Algorithm

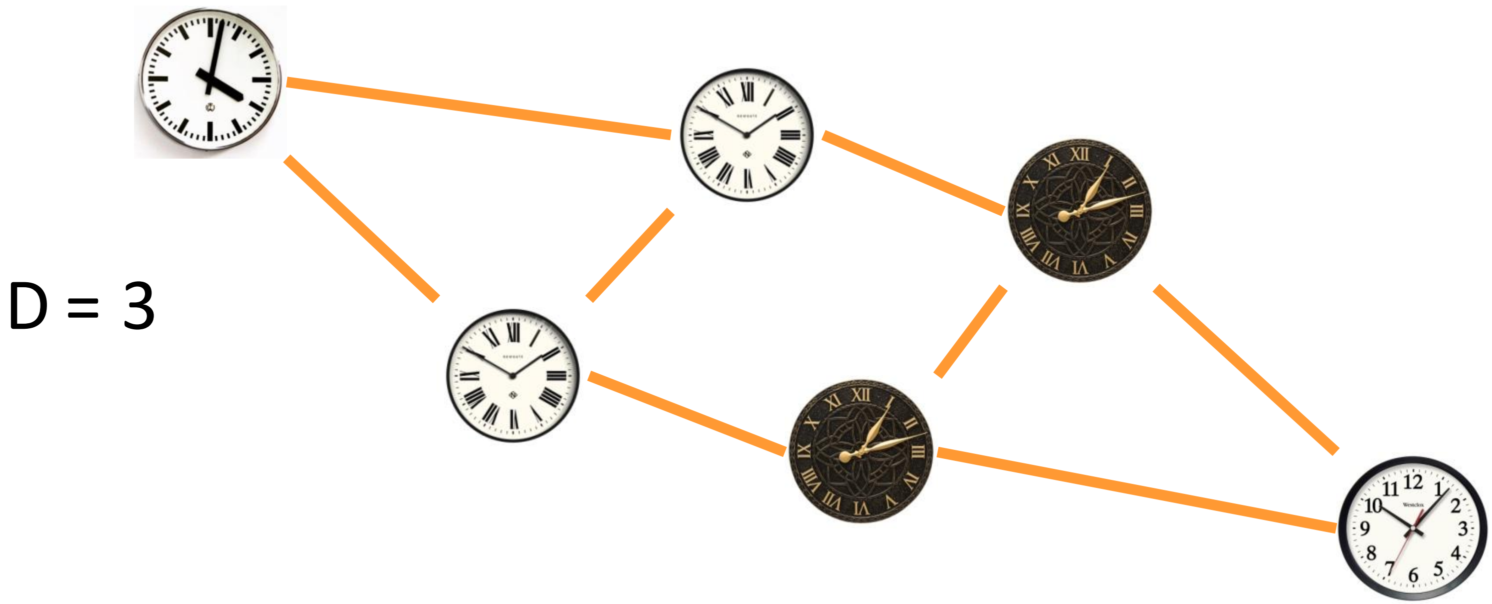
Theorem

The Max Algorithm guarantees

$$\max_{v,w \in V}\{L_v(t)-L_w(t)\} \leq \vartheta dD + (\vartheta-1)T$$

at time $t \geq dD + T$,

where D is the network diameter of G.



Example: Max Algorithm

Theorem

The Max Algorithm guarantees

$$\max_{v,w \in V}\{L_v(t)-L_w(t)\} \leq \vartheta dD + (\vartheta-1)T$$

at time $t \geq dD + T$,

where D is the network diameter of G.

Proof sketch:

- every T (logical) time v broadcast L_v
- receiving nodes adjust their clock (if needed) and broadcast, too (if they still need to)
- in the dD time for a value to spread, v's clock advances by at most ϑdD
- $(\vartheta-1)T$ is added to account for the broadcast interval T