RDCS423 Tutorial Problems and Solutions #2 - Time Handling

1. A *master-slave clock algorithm* was used to synchronize a slave processor clock. At the start of the update cycle the master clock had a time of 10:00:00.000000 and the slave received the master's clock after 10 μsec at its clock time of 10:00:00.000500. In the second phase of the update cycle, the slave responds with a time of 10:00:00.001000 which is transmitted to the master in 30 μsec where the master clock reads 10:00:00.000540. Assuming that nothing is known about the slave clock errors apart from the assumption of a zero-mean Gaussian distribution, what is the clock update that would be sent from the master to the slave?

2. With a *master-slave clock algorithm*, show that a bound on the maximal clock error between slaves would be given by the following expression:

$$\left|2\tau \max_j(\delta_j)\right| + \left|2 \max_j(\in_j)\right|$$

where j = 1 .. number of slaves

δ_j = the drift rate (in sec/sec) for slave j

$\in_j = (\overline{\mu}_i^j - \overline{\mu}_j^i)/2 - (\overline{E}_j^1 - \overline{E}_j^2)/2$

τ = update period (sec)

$\overline{\mu}_i^j$, $\overline{\mu}_j^i$ = mean master-slave and slave-master communication times respectively

$\overline{E}_j^1$, $\overline{E}_j^2$ = mean slave clock error distribution times

3. Given a *fundamental ordering distributed clock algorithm*, develop a bound for the variation of each clock in a distributed network with a communication graph of diameter *d*. Calculate this bound for a case with a clock drift rate of 0.001, message update rate of 10 msec, upper bound on message delays of 10 μsec, and a communication graph diameter of 10 hops.

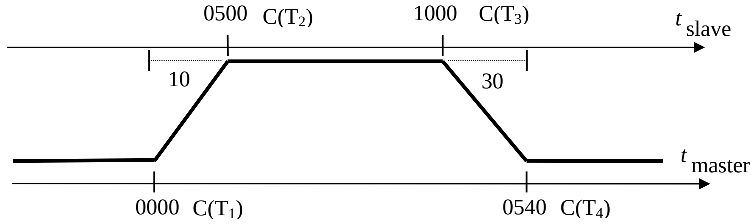
4. With a *distributed clock algorithm* that uses a *minimize maximum error approach*, determine what clock update is performed from node *j* given the following states at node *i* and node *j* at the time of the update cycle:

At node *i*: let the reset time be 00:00:00.000000, the count time is 00:00:00.001000, the drift rate is estimated at 0.01, and the estimated discretization error is 5 μsec.

At node *j*: let the reset time be 00:00:00.000000, the count time is 00:00:00.001020, the drift rate is estimated at 0.01, and the estimated discretization error is 20 μsec. The response delay from node *i* to node *j* is 5 μsec.

Solutions:

1.



- slave computes $d_1 = C(T_2) - C(T_1) = 500$ and sends to master
- master computes $d_2 = C(T_4) - C(T_3) = -460$
- master computes slave clock skew $\xi_1 = (d_1 - d_2)/2 - (\mu_i^j - \mu_j^i)/2 + (E_j^1 - E_j^2)/2$
$= [500 - (-460)]/2 - [10-30]/2$
$= 490$ μsec
- the slave clock skew is sent to the slave for update

2. In the first instance assume no drift and recall that:

$$(d_1 - d_2)/2 = \xi_j + (\mu_i^j - \mu_j^i)/2 - (E_j^1 - E_j^2)/2$$

Now each slave introduces an error via the update algorithm and if this error is

$$\in_j = (\mu_i^j - \mu_j^i)/2 - (E_j^1 - E_j^2)/2 = (d_1 - d_2)/2$$

then $\xi_j = 0$ and the clock skew is found to be zero → this error cannot be removed if (worst case) this is maintained on each successive cycle. For a number of cycles this residual error is:

$$\in_j = (\overline{\mu}_i^j - \overline{\mu}_j^i)/2 - (\overline{E}_j^1 - \overline{E}_j^2)/2$$

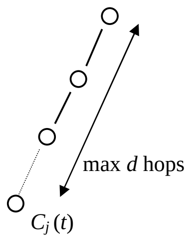
Because each slave may have a persistent error ranging from $|\in_j|$ to $-|\in_j|$ with respect to the master, the maximal clock difference between slaves is $\left|2 \max_j(\in_j)\right|$.

With the drift term δ_j included, in an interval τ, the error introduced is just $\tau\delta_j$. As the drift can range from $|\delta_j|$ to $-|\delta_j|$ with respect to the master → the maximal clock difference between slaves due to drift is $\left|2\tau \max_j(\delta_j)\right|$.

Combine the terms to give the maximal clock error between slaves:

$$\left|2\tau \max_j(\delta_j)\right| + \left|2 \max_j(\in_j)\right|$$

3.



Clock drift rate is δ
Message update rate is τ
Message delay: $\mu < D < \eta$
Communication graph max distance is d
Between any two nodes we have a worst case drift in τ seconds of $2\delta\tau$ seconds.

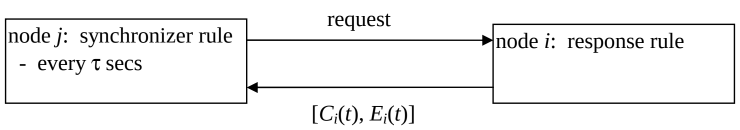
Within a message update interval clocks could have drifted apart by $2\delta\tau$. With the worst communications delay of η seconds the error between directly connected nodes is then $2\delta\tau + \eta$

Also worst case, a sequence of updates must traverse d hops incurring an error of $2\delta\tau + \eta$ on each step $\rightarrow$ the bound on correctness of any two clocks in this network is:

$$\forall i \forall j: \quad |C_i(t) - C_j(t)| < d(2\delta\tau + \eta)$$

with $\delta = 0.001$, $\tau = 0.01$, $\eta = 10^{-5}$, $d = 10$ $\rightarrow$ $|C_i(t) - C_j(t)| < 300$ µsecs

4.



Response from node i: after request from node j

$E_i(t) = \in_i + [C_i(t) - \rho_i]\ \delta_i = 5 + (1000)0.01 = 15$ µs

Send $[C_i(t), E_i(t)]$ = [00:00:00.001000,15] to node j

Synchronizer at node j:

Receive $[C_i(t), E_i(t)]$ from node i

$E_j(t) = \in_j + [C_j(t) - \rho_j]\ \delta_j = 20 + (1020)0.01 = 30.2$ µs

$\rightarrow$ $[C_j(t) - E_j(t), C_j(t) + E_j(t)]$ = [00:00:00.0009988, 00:00:00.0010502]

$[C_i(t) - E_i(t), C_i(t) + E_i(t)]$ = [00:00:00.000985, 00:00:00.001015]

which clearly has a non-empty intersection interval

and $E_i(t) + (1+\delta_j)\mu_i^j = 15 + (1+0.01)5 = 20.05 \leq E_j(t)$

$\rightarrow$ both conditions to use the time from node i are met so the synchronizer at node j will reset its clock, update the error and reset time:

$C_j(t) \leftarrow C_i(t)$ = 00:00:00.001000

$\in_j \leftarrow E_i(t) + (1+\delta_j)\mu_i^j = 20.05$ µs

$\rho_j \leftarrow C_i(t)$ = 00:00:00.001000