Solutions for Homework 2, ECE 428 Distributed Systems, Fall 2005

(a) The desired solution is obtained with one message taking minimum delay (min), and the other taking the maximum delay ($max = d$). Here is one way this may occur.

Let t_i denote real time.

Let $HC_j(t_i)$ denote the hardware clock at process j at real time i, where $j = 1, 2$. Assume that $HC_j(t_i) - HC_j(t_k) = t_i - t_k$.

Define the following:

t_1 = time at which process 2 sends message to process 1 with timestamp $HC_2(t_1)$.

t_2 = time at which process 1 sends message to process 2 with timestamp $HC_1(t_2)$.

t_3 = time at which process 2 receives the message from process 1

t_4 = time at which process 1 receives the message from process 2

Assume that $t_1 < t_2 < t_3 < t_4$ and $t_4 - t_1 = max = d$ and $t_3 - t_2 = min$, where $max - min = u$ (the uncertainty).

Now, let adj_1 and adj_2 denote the adjustments computed by processes 1 and 2 using the specified algorithm. Then we have that

$$adj_1 = \tfrac{1}{2}\left(HC_2(t_1) + d - \tfrac{u}{2} - HC_1(t_4)\right)$$

$$adj_2 = \tfrac{1}{2}\left(HC_1(t_2) + d - \tfrac{u}{2} - HC_2(t_3)\right)$$

Let us consider the difference in the software clock at the two processes when real time is t_4 (right after the message is received at process 1). The difference is given by:

software clock at process 1 at t_4 − software clock at process 2 at t_4

$$\begin{aligned}
&= HC_1(t_4) + adj_1 - HC_2(t_4) - adj_2 \\
&= HC_1(t_4) + \frac{1}{2}\left(HC_2(t_1) + d - \frac{u}{2} - HC_1(t_4)\right) - HC_2(t_4) - \frac{1}{2}\left(HC_1(t_2) + d - \frac{u}{2} - HC_2(t_3)\right) \\
&= HC_1(t_4) - HC_1(t_4)/2 - HC_1(t_2)/2 + HC_2(t_1)/2 - HC_2(t_4) + HC_2(t_3)/2 \\
&= \frac{1}{2}\left(HC_1(t_4) - HC_1(t_2)\right) + HC_2(t_1)/2 - \left(HC_2(t_1) + t_4 - t_1\right) + \left(HC_2(t_1) + t_3 - t_1\right)/2 \\
&= (t_4 - t_2)/2 - t_4 + t_1 + t_3/2 - t_1/2 \qquad \text{Recall that } HC_j(t_i) - HC_j(t_k) = t_i - t_k. \\
&= t_1/2 - t_2/2 + t_3/2 - t_4/2
\end{aligned}$$

$$
\begin{aligned}
&= -(t_4 - t_1)/2 + (t_3 - t_2)/2 \\
&= -max/2 + min/2 \\
&= -(max - min)/2 \\
&= -u/2
\end{aligned}
$$

Hence the skew is $u/2$.

(b) Note that the marker in the Chandy-Lamport algorithm may be delayed arbitrarily, but must be delivered to P1 and P2 before message F and G, respectively, due to FIFO channels. To find the latest states, suppose that the marker is indeed delayed that much. This leads to the answer below.

Process P2 records state right before it receives message G from P4.

Process P1 records state right before it receives message H from process P2 (recall that process P2 records state before sending H).

Process P3 may record the state at anytime after all shown events at process P3 (The homework states that there are no messages other than those shown in the figure. However, if there were to be other messages in the future, then P3 may delay recording its state only until it receives a message sent by other processes after recording their state – the state must be recorded by P3 before receiving such a message.)

(c)(i) The majority vote defaults to 0 if there is a tie. Suppose that there are 5 processes, P1, P2, P3, P4 and P5. Suppose that none of the processes are faulty, and their values are 0, 1, 1, 0, 1, respectively.

During time T, process P1 receives values 0, 1 and 1 from processes P1, P2, and P3, and decides on 1. Process P2 receives values 0, 1, 1, and 0, from processes P1 through P4, respectively, and decides on 0 (due to the tie). So consensus fails.

(c)(ii) Each process chooses the majority value on receiving first N-1 values from N-1 processes (including itself). With N=5 processes, N-1 = 4. Suppose process P1 receives values 0, 1, 1, 0 from P1, P2, P3, P4 first, and decides on 0 (due to the tie). Process P2 receives values 1, 1, 0, 1 from P2, P3, P4, P5 first, and decides on 1. So consensus fails.