ECE428-HW1

Shengjian Chen (3180111657)

QUESTION 1

Ring-based failure detection scheme with processes $p_1, p_2, ..., p_n$. And p_i sends heartbeat to p_{i+1} each T seconds;

(a) Detect Pair Failure

For each Node p_i, instead of sending only to p_{i+1}, it also send a backward one to p_{i-1} each T seconds. Then with the squeezing, we can find any pair of processes when they failed. The bandwidth cost should be doubled compared with the original one since each T will double the heartbeats.

(b) Maximum Detection Time

The maximum detection time should be in the following situation: Just after p_i sends its heartbeat, it fails. Then it should be waiting for the whole cycle long to receive its next heartbeat from p_{i-1}. Each transmit takes $(T + \Delta)$. So the maximum detection time should be $n \times (T + \Delta)$

(c) Identify

For each process p_i, when it doesn't receive the heartbeat after its timeout, send out a unique heartbeat to p_{i+1} and require it to send the ACK back. If p_i did not receive the ACK in $2 \times (T + \Delta)$, then the process p_{i+1} fails.

QUESTION 2

(a) timeout

The receiver's time should be quicker to make timeout larger. The timeout should be: (max_delay - min_delay + T)$\times$drift rate. So it would be $1.01 \times (T + \Delta)$

(b) Maximum Detection Time

Now the sender's time should be quicker. The maximum detection time should plus Δ, so max detection time is $1.01 \times (T + \Delta) + \Delta$.

(c) Cristian Re-run

After the last synchronize, the accuracy bound is $(-20/2ms, 20/2ms)$ which is $(-10ms, 10ms)$. While the clock keeps on running, the time difference due to the time drift is $0.01 \times T$. So we need to make $10 + 0.01 \times T <= 100$. Thus, T = 9000ms.

QUESTION 3

(a). Worst Case Skew

With using the MST to connect the whole servers, we find the path that goes C-A-B-E-D and it connects all the servers in the graph. Then this longest path has the total RTT sum of 5+10+10+5 = 30. Thus, the worst-case skew of this graph is 30/2 = 15

(b). One-way Transmission Time Upper-Bound

Since in this problem, there is no clock drifts, and we know that the real time in both servers A and B have the following condition that $T_A = T_B - o$. With o is the skew time we need to calculate in sub-problem c), let's find o's possible range first.

In the transmission 3, we see that the message was send in $T_B = 27$, while the receive time in A is $T_A = 23$. So we can get that the minimum skew, o is 4,($o >= 4$) since the "worst" case is when transmitting time is 0 from B to A, which we could have the minimum skew.

As for the maximum for o, we need to compare the time difference when B receive the corresponding message from A.

$$t_2 = 25 - 18 = 7$$
$$t_4 = 37 - 29 = 8$$
$$t_6 = 56 - 45 = 11$$

Thus, we have it that the o's range is [4,7]. Further, we can calculate each transmission time with this possible o's range.

$$t_1 <= 12 - (10 - 7) = 9$$
$$t_2 <= (25 - 4) - 18 = 3$$
$$t_3 <= 23 - (27 - 7) = 3$$
$$t_4 <= (37 - 4) - 29 = 4$$
$$t_5 <= 40 - (39 - 7) = 8$$
$$t_6 <= (56 - 4) - 45 = 7$$

(c). Skew

As calculated before, the o's range is [4,7]. And to minimize the error range, B's clock should be changed back for 5.5 unit times.

QUESTION 4

(a). Lamport timestamp

First, let A:1, then we can have M:2, I:2 ... each event is the max(prev,receive) + 1 Thus we can have the following:

A:1, B:4, C:8, D:9
E:4, F:5, G:6, H:7
I:2, J:3, K:8, L:9
M:2, N:3, O:4, P:8

(b). Vector timestamp

With A to be the first vector, (1, 0, 0, 0); Then, for each process, the received vector is the sending vector plus the unit vector in corresponding process. Then each dimension in the vector is the largest one.

A:(1,0,0,0), B:(2,0,0,2), C:(3,4,2,2), D:(4,4,2,2)
E:(1,1,0,2), F:(1,2,0,2), G:(1,3,2,2), H:(1,4,2,2)
I:(1,0,1,0), J:(1,0,2,0), K:(1,4,3,2), L:(1,4,4,2)
M:(1,0,0,1), N:(1,0,0,2), O:(1,0,2,3), P:(1,4,2,4)

(c). concurrent

1) *i. C:* in P1, D happens after C;

in P2, C happens after the last H

in P3, C happens concurrent with K

in P4, C happens after H-E-N; so it happens concurrent with O, P

in total: K, O, P

2) *ii. I:* similarly as before, in total: M, N, B, E, F

3) *iii. O:* in total: G, H, C, D, K, L, B, E, F

Question 5

It is quite complicated, lets first assume the marker(c_{34}) reach after event O. Then for P4 the recording state is O and it send markers to both P1 and P2(P3 ignored since has already recorded). This time, P1 and P2 could be B/H or C/H.

The first marker received must be after G in P2. So, we might have:

1. B,G,J,N
2. B,G,J,O
3. B,H,J,O

In the above, if the marker receive in P2 is G, then it cannot be P in P4, because the marker send in P2 through c_{24} is before H and FIFO in c_{24}.Starting from 5, the first marker receive for P2 is from P1 and the receive time is after H. But it could still be possible for P4 to receive its first marker from P1 before O or even after P. Now if the marker first receive in P1 is after C, then it cannot be the marker send in P2 between G and H. So, C,H,J is fixed. Thus, we could have

4. C,H,J,O
5. C,H,J,P

Now if the first marker in P1 is after D,

6. D,H,J,O
7. D,H,J,P