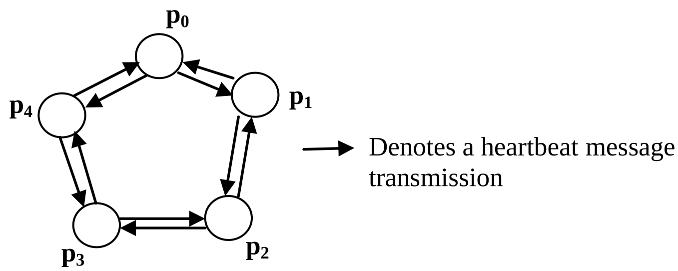
ECE428/CS425/CSE424: Distributed Systems **Spring 2011**
Homework 1 Solutions

1. The 5 node topology shown below uses a modified heartbeating protocol for detecting failures, as follows:



- All processors maintain a sequence number.
- p_i sends $p_{(i+1)\bmod 5}$ and $p_{(i-1)\bmod 5}$ a heartbeating message every T time units (0 <= i <= 4).
- If p_i has not received a new heartbeat from any of the two neighboring processors for the past 3*T time units, since it received the last heartbeat, then p_i detects the corresponding processor as failed and communicates the information to all the other processors.

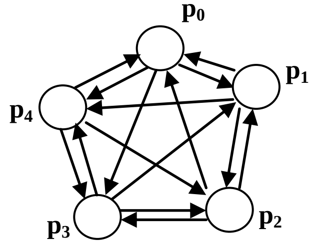
a. Enumerate all possible combinations of simultaneous processor failures (including single processor failures) that can be reliably detected by this protocol. Assume a synchronous system.

Single processor failures: p_0, p_1, p_2. p_3, p_4
2 simultaneous processor failures: p_0p_1, p_0p_2, p_0p_3, p_0p_4, p_1p_2, p_1p_3, p_1p_4, p_2p_3, p_2p_4, p_3p_4
3 simultaneous processor failures: $p_0p_1p_3$, $p_0p_2p_3$, $p_0p_2p_4$, $p_1p_2p_4$, $p_1p_3p_4$

b. Explain why this protocol cannot reliably detect all possible 3 simultaneous processor failures. How will you modify the protocol to detect up to 3 simultaneous processor failures reliably? (Use a figure similar to the one shown above.)

If a processor p_i fails simultaneously along with both processors $p_{(i+1)\bmod 5}$ and $p_{(i-1)\bmod 5}$ to which it sends the heartbeat messages, then there is no way of detecting its failure. Therefore, not all 3 simultaneous processor failures can be detected using this method.

If however, we modify the protocol so that every processor sends heartbeat messages to one more processor, in addition to $p_{(i+1)\bmod 5}$ and $p_{(i-1)\bmod 5}$, then up to 3 simultaneous processor failures can be reliably detected. One possible way of doing this is shown below:

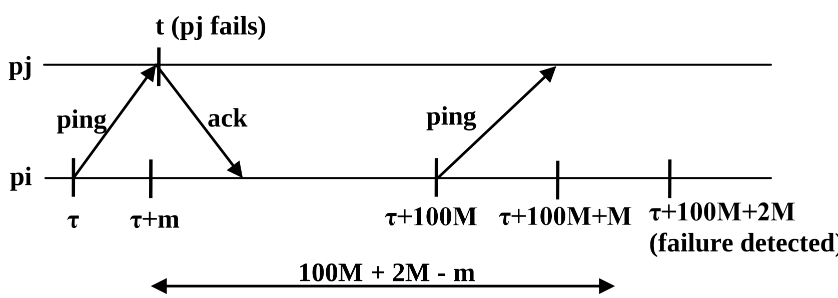


2. Consider the ping-ack mechanism for failure detection between two processors pi and pj (pi sends a ping, pj responds with an ack). Assume a synchronous system where the message delays between the two processors are bounded between [m,M] (both inclusive). Let the pings be sent every 100*M time units.

    a. What is the earliest time by which pi can correctly conclude that pj has failed (in the absence of an ack) after pi sends a ping?

    2*M. To correctly conclude pj's failure, pi should wait for M time units for the ping to reach pj, and another M time units expecting an ack. Thus in the absence of an ack pi can conclude pj's failure after 2*M time units.

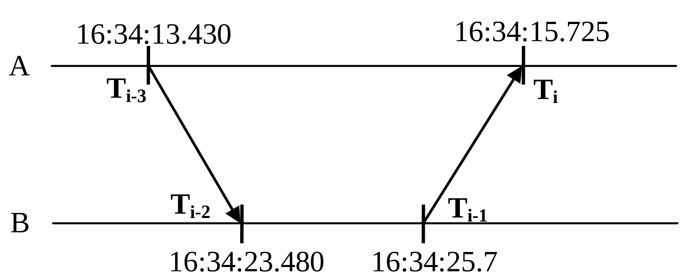
    b. Suppose pj fails at time *t*, what is the maximum time that may elapse before the failure is detected by pi?



    (100*M + 2*M - m). Please refer to the figure above. Let pi send a ping at time τ. Assume that time *t* at which pj fails occurs immediately after sending an ack. The earliest the failure could have happened is 'm' time units after pi sends a ping at time (τ+m). The failure can only be detected during the next ping exchange, which will happen (100*M) time units from the τ. Additionally, the processor pi has to wait M units for this ping to reach pj and another M time units while waiting on an ack. The total time elapsed since the node failure will therefore be (100*M + 2*M - m).

3. Problem 11.7 from textbook

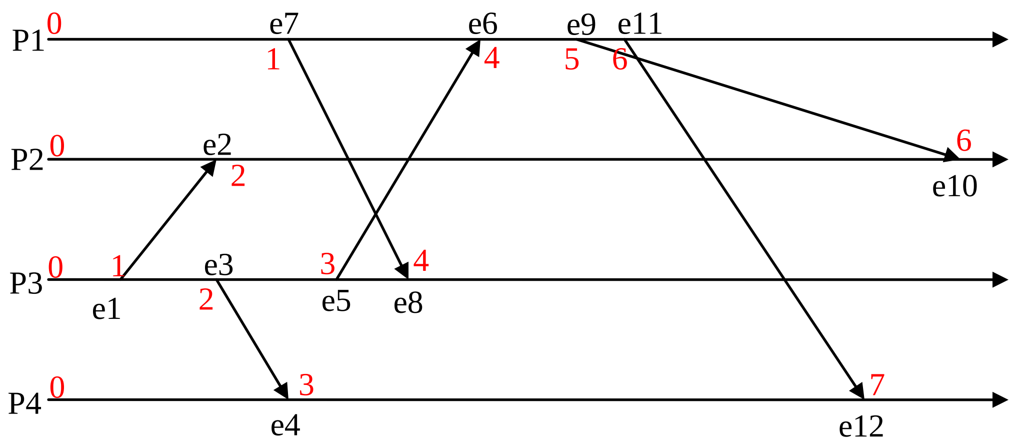
    A pictorial representation of the messages between the servers is shown below:



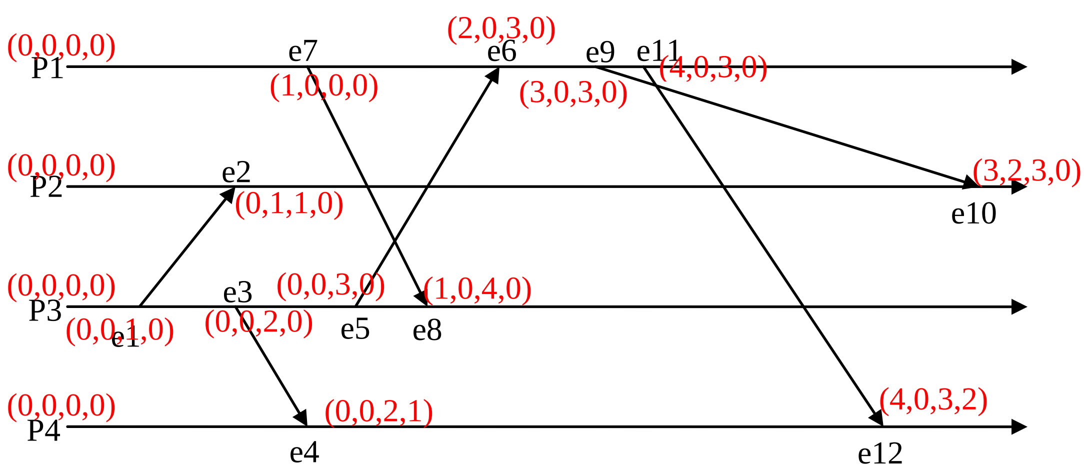
    Using the equations on pg. 444 of the textbook (4th edition),

    Estimate of the offset = $(T_{i-2} - T_{i-3} + T_{i-1} - T_i)/2$ = 10.0125 seconds
    Accuracy of the estimate = $T_{i-2} - T_{i-3} + T_i - T_{i-1}$ = 0.075 seconds

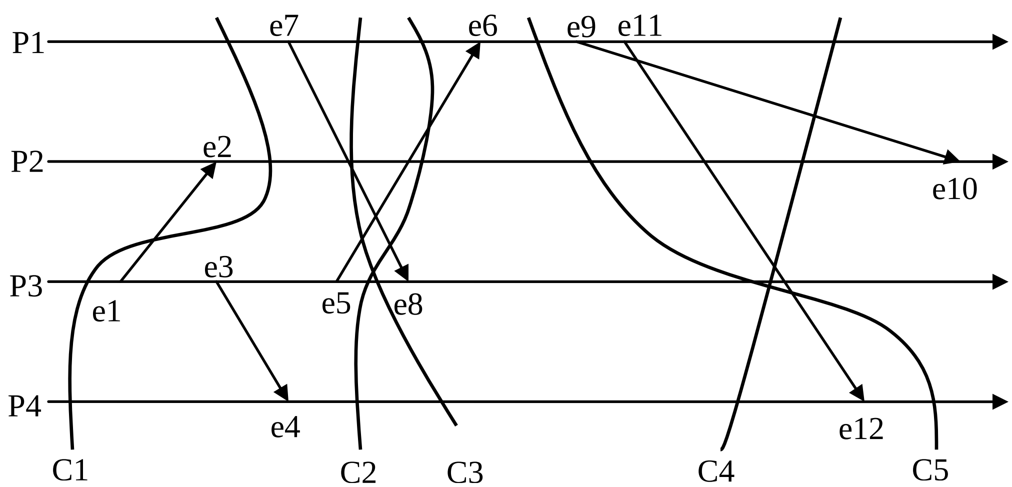
4. Determine the logical timestamp of each event shown below.



5. Determine the vector timestamp of each event shown below.
A vector timestamp (a,b,c,d) represents (a=P1's timestamp, b=P2's timestamp, c=P3's timestamp, d=P4's timestamp).

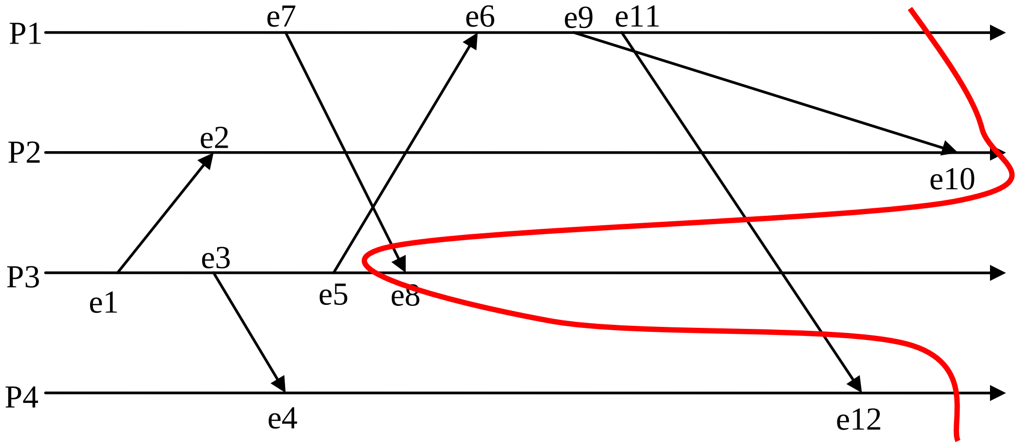


6. Determine which of the cuts C1 through C5 shown below are consistent?

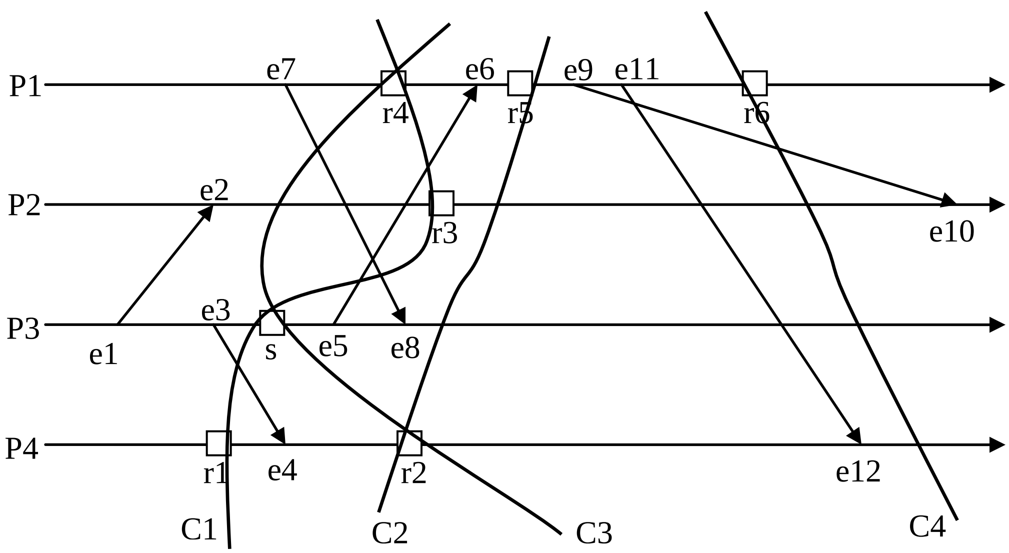


Cuts C2, C3, and C4 are the consistent cuts

7. Construct a consistent cut which covers the largest number of events, but excludes e8.



8. Let us assume that P3 triggers snapshot algorithm at s. Which of the cuts C1, C2, C3, and C4 could be recorded by the algorithm?



Only the cut C3 could be recorded by the snapshot algorithm.