TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

**EXAMINATIONS — 2011**

MID-YEAR

# NWEN 401

# Distributed Systems Design

**Time Allowed:** 2 Hours

**Instructions:** You must attempt ALL questions.

Each question is worth 10 marks.

There are a maximum of 40 marks in total.

Within each question, the marks for subparts are shown.

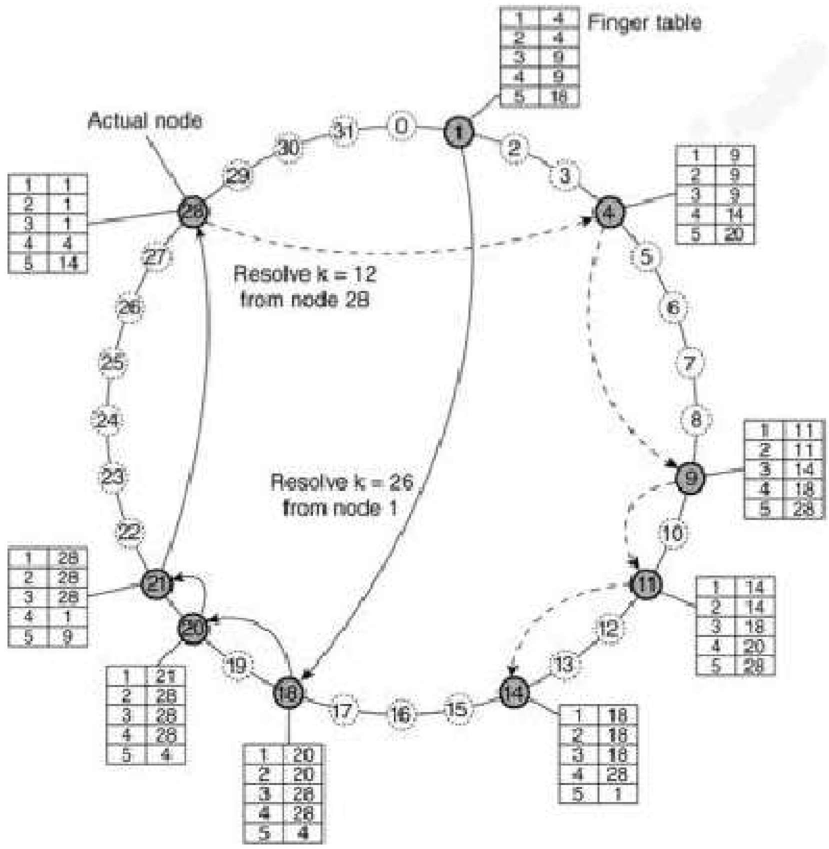
Non-programmable calculators without full alphabetic keys are permitted.

Paper foreign language to English dictionaries are permitted.

| Question | Topic | Marks |
|---|---|---|
| 1 | Peer-to-peer Middleware | 10 marks |
| 2 | Replication and Consistency | 10 marks |
| 3 | Streaming Multimedia Traffic | 10 marks |
| 4 | Synchronization | 10 marks |

## Question 1. Peer-to-peer Middleware. [10 marks]

Consider the Chord system given in the figure below.




**(a)** [4 marks] Node 7 joins the network. What would its finger table be? Show clearly how you compute the second (2nd) entry of the finger table.

*The finger table for node 7 is:*

| | |
|---|---|
| 1 | 9 |
| 2 | 9 |
| 3 | 11 |
| 4 | 18 |
| 5 | 28 |

*Entry #2 is computed as* $succ(p+2^{i-1}) = succ(7 + 2^1) = succ(9) = 9$.

**(b)** [6 marks] Which other nodes' finger tables will be changed, and what are their new finger tables?

*Other finger tables that need to be changed are those in node 4, node 21 and node 1:*

*node 4*

| | |
|---|---|
| *1* | *7* |
| *2* | *7* |
| *3* | *9* |
| *4* | *14* |
| *5* | *28* |

*node 21*

| | |
|---|---|
| *1* | *28* |
| *2* | *28* |
| *3* | *28* |
| *4* | *1* |
| *5* | *7* |

*node 1*

| | |
|---|---|
| *1* | *4* |
| *2* | *4* |
| *3* | *7* |
| *4* | *9* |
| *5* | *18* |

## Question 2. Replication and Consistency [10 marks]

**(a)** [2 marks] Consider the quorum-based consistency protocol. What problem could occur in a system consisting of 12 replicas that has a read quorum of 7 and a write quorum of 6.

> *Two clients could both write to separate groups of 6, that is a write-write conflict. This would be detected during a read because the version numbers would be identical so it would be impossible to distinguish what was the correct value.*

**(b)** [2 marks] Why is the following data store not sequentially consistent? Is it causally consistent? Be sure to explain your answer.

| | | | | | |
|---|---|---|---|---|---|
| P1: | W(x)a | | | | |
| P2: | | W(x)b | | | |
| P3: | | | R(x)b | | R(x)a |
| P4: | | | | R(x)a | R(x)b |

> *It is not sequentially consistent because P3 and P4 are reading the effects of concurrent writes (by respectively P1 and P2) in different orders. It is causally consistent because there are no causal relationships that need to be obeyed.*

**(c)** [3 marks] Consider a system that combines read-your-writes consistency with writes-follow-reads consistency. Is this system also sequentially consistent? Explain your answer.

> *No, it is not sequentially consistent. Although the combination effectively provides location-independent consistent behavior for a single process, it does not guarantee that when there are two concurrent write operations at different locations, that the effect of those writes will be seen everywhere in the same order by all clients.*

**(d)** [3 marks] Is it possible to have write-write conflicts in client-centric consistency models? Explain your answer

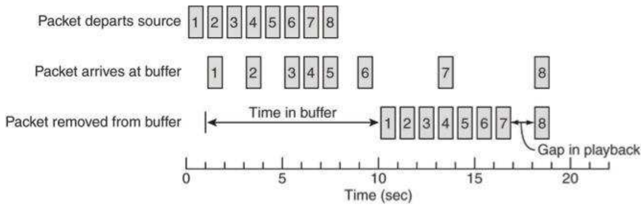
> *Yes. These models say nothing about concurrent access at all. Therefore, it could very well be possible that while Alice is modifying x at location A, that Bob is modifying x as well, but at another location B.*

## Question 3. Streaming Multimedia Traffic [10 marks]

Multimedia traffic like video or voice have stringent delay requirements but allow some degree of packet loss. If these packets are carried over the best-effort internet, packet delays and losses are inevitable.

**(a)** [5 marks] Explain, with the help of a diagram, how a distributed system can alleviate the effects of variations in packet delay.

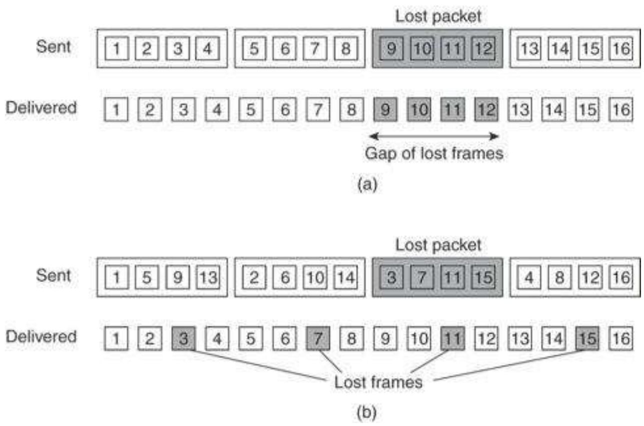
*To reduce the effects of large variances in packet delays, the receiver can store the packets in a buffer (up to a maximum time limit) and pass the packets to the application at a regular rate that is expected by the application.*




*The solution may not be entirely foolproof and some packets may take much longer time to arrive at the receiver such that all the other packets have already been passed to the application leaving the buffer with no more packets send to the receiver at the expected rate. Under such circumstances, a larger buffer may be needed to store more packets before passing the packets to the application.*

**(b)** [5 marks] Explain, with the help of a diagram, how a distributed system can reduce the adverse effects suffered by applications due packet losses.

*When a single packet contains consecutive frames from an audio/video stream, losing that packet may result in the loss of content being noticeable, e.g. large gap in speech or video. To reduce that adverse effect, interleaving frames as shown in the figure can alleviate the loss.*



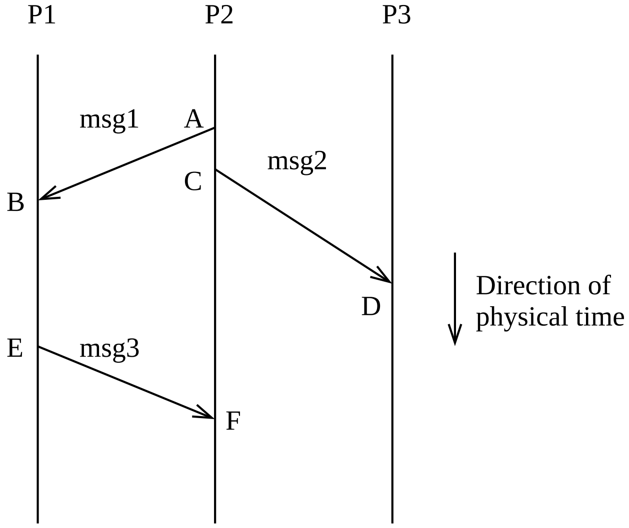

*With frame interleaving, when a packet is lost, the frames that are lost with that packet are not consecutive that may make the loss noticeable. The missing frames are distributed over time and therefore less noticeable when the audio/video is played back the application.*

*The following is also acceptable but will be awarded less marks:*

*Forward error correction (FEC) techniques whereby the sender encodes the packets that it send in such a way that any k out of n packets received is enough to reconstruct the lost packets.*

## Question 4. Synchronisation. [10 marks]

Consider the messages as sent by the three processes shown in the figure below. A send event and receive event is shown for each of the three messages sent between the processes, for example *A* is a send event and *B* is a receive event for a message *msg1* sent from P2 to P1.




**(a)** [2 marks] Define Lamports *happened-before* relation.

> *The happened-before relation says that if (1) a and b are two successive events in the same process, then $a \rightarrow b$, (2) if a is the sending of a message m and b is the receipt of m, then $a \rightarrow b$, or (3) if $a \rightarrow b$, $b \rightarrow c$, then $a \rightarrow c$.*

**(b)** [2 marks] Identify the *concurrent* events shown in the Figure.

> *$A \rightarrow B \rightarrow E \rightarrow F$ and $A \rightarrow C \rightarrow D$. We see that B is concurrent with C and D, E is concurrent with C and D, F is concurrent with D*

**(c)** [4 marks] Use Lamport's logical clocks algorithm to assign timestamps to the events *A-F*. Assume that each process starts off with the same logical time of 0 and that we only consider only internal events related to send and receive events. Show your reasoning.

> *Note that we must always receive at a point later than the send event so we increment the local clock for both sends and receives.*
> *C(process) is the local clock for process*
> *T(event) is the timestamp for an event*
> *T(msg) is the timestampe for a message*
> *A, C(P2)=0, send C(P2)++=1, T(A)=T(msg1)=1*
> *B, C(P1)=0, T(B)=C(P1)=max(C(P1)=0,T(B)=T(msg1)=1)+1=2*
> *C, C(P2)=1, send C(P2)++=2, T(C)=T(msg2)=2*

*D, C(P3)=0, T(D)=C(P2)=max(C(P3)=0,T(msg2)=2)+1=3*
*E, C(P1)=2, send C(P1)++=3, T(E)=3, T(msg3)=3*
*F, C(P2)=2, T(F)=C(P2)=max(C(P2)=2,T(msg3)=3)+1=4*
*Solving this shows an understanding of the algorithm and the happened before relationship.*

*T(A)=1, T(B)=2, T(C)=2, T(D)=3, T(E)=3, T(F)=4*
*This matches the relationships from the earlier part of the question.*

**(d)** [2 marks] Using at least ONE example drawn from part (b) and (c), show that the Lamport clock timestamps reflect the happened-before relation but fail to capture causality.

*P1's E may have been caused by P2's A but cannot have been caused by C because event C has not been seen by P1. The relationship between A and E is reflected by the fact that E's timestamp $>$ A's timestamp. On the other hand, E's timestamp is $>$ than C's timestamp even though there can be no causal relationship between these events.*

********************************