



Complete, fold and seal
(see note 2 below)

SURNAME: Humphreys

FORENAME(S): Andrew

12 Pages

1. Please give the information requested in the boxes below.
2. Enter your name in the space provided to the right, fold over and seal down (using the label provided) to hide your name.
3. The answers must be legibly written. The Examiners can either ignore illegible script or have it typed and the expense of typing charged to the candidate.
4. Write your answers in ink.
5. Begin each answer on a fresh page. Do all rough work in the answer book and cross it through.
6. No part of this book is to be torn out; it must be handed in complete.
7. If more books than one are used, fasten them together using the tags available from the invigilator(s).
8. **Every book must be handed in. In no circumstances may answer books, used or unused, be taken out of the examination venue.**

[illegible]

(Gweler yr ochr arall am y fersiwn Cymraeg)

Question 1

(a) Protocol sequence diagram is UDP

UDP does not set up a connection and just sends packets with no care if the packets fail due to network issues also no check is done to see if the other ~~client~~ ~~host~~ device had all or any of the data sent.

Methods to prevent lost data could be simple acknowledgements of the data to say that it has received it correctly also a sequential number could be added to the data to allow tracking of the data and allow for reflagging when data has been missed. Adding in these extra steps does increase on the network load but provides more reliability.

6

(b) A pre-emptive kernel should be used, this allows for ~~tasks~~ tasks to be completed as they come in with equal priority a higher priority task could then be assigned

A re-entrant function is one that uses a ~~global~~ variable that is accessed by many parts of ~~the~~ code. e.g. a global variable.

```
global int i;
```

```
func myFunc()  
{  
    return i*5;  
}
```

This is an example of a non-reentrant function.

Question 2

- (a) Ethernet is a non-deterministic MAC which uses a bus topology, a non-deterministic MAC allows sending data without having to wait for a signal before transmitting. Disadvantages to this is that if two or more devices transmit at the same time this causes a collision and all devices on the ^{collision domain} network have to wait until there is no noise before sending data. The ~~two~~ devices that caused the collision are then given a random wait time before being allowed to send their data as to reduce the chance of more collisions. This means that if collisions are high then the chance of sending data is reduced. Disadvantages with network multiplayer games is that movement appears jerky as players can bounce from one location to another and ~~have~~ ^{can have} large impact on games like FPS where player position and bullets fired are an important requirement.

3

3

2

- (b) IP 199.7.10.0 is a type C address, we borrow bits from the address to allow splitting up of the network to make things like private access or cluster servers.

128	64	32	16	8	4	2	1
•	•	•	•	•	•	•	•
SN 2	4	8	16	32	64	128	256
Hz 256	128	64	32	16	8	4	2

Number of subnets needed = 9

First four bits gives 16. $16 > 9$
 $\times -2$

Subnet mask therefore = 255.255.255.240 ✓

First host	Range	Broadcast
199.7.10.16 ✓	17-30	31 ✓ 30
199.7.10.32 ✓	33-46	47
199.7.10.48 ✓	49-62	63

Range is end

Range of ips are calculated +1 from host as cannot have same ip as the host and -2 from next host as next host -1 is the broadcast ip address

10

Question 3

- (a) Deadlock is where processes are after resources that it cannot ~~ob~~ obtain for one reason or another. Examples of this could be process 1 has resource 1 and needs resource 2, process 2 has resource 2 and needs resource 1. as both processes do not give up their resource to reclaim the next they are stuck waiting for each other. 2

Another example of this is

Starvation can be caused if processes are waiting a long time to claim resources, process A may run for ~~the length long periods between other processes~~ long periods with claimed resources leaving other processes on hold whilst it waits. 2

Another deadlock situation is where a process holds onto a resource required by other processes for the duration of the application 0.5

(b) Mutexes allow for sharing of critical data and allows access on a one by one basis, this stops data ~~modifying~~ modification whilst another process is trying to ~~read~~ read the data

Semaphores allow for a resource pool where many can share the resource, this resource pool can ~~be released~~ be released and allocated when required so could start with no resource and add allocation at a later time.

An application of those could be a server that gets messages from clients, these messages could come in at any time and might not want to be processed straight away. Messages are then put onto a queue for storage. Problems can arise if you are trying to access this queue whilst deleting the last message processed as another message may be in the middle of being added. Also there can be issues where you ~~read~~ have two clients sending data at the same time, you will get a write access error without mutexes or semaphores in place.

Question 4

- (a) Consistency is where all nodes are correct across devices. The more consistent the game data is the more realistic it will be as in terms of asset location and player position. It will be but the more data that is sent over the network to keep this consistent ~~it causes~~ network issues.

4g. 3

Responsiveness ~~is~~ is influenced by consistency, the more consistent the data the less responsive it will become due to latency issues caused by consistency, because of this there is generally a balance that is required so that it's reasonably consistent and reasonably ~~not~~ responsive.

B 1

~~Sette~~ G26

(b) ^{Some} A ~~range~~ of responses to minimise the amount of data are, Send less data or compress the data. Doing this restricts the amount of data required to be sent in the first place

Another method could be to get the clients to do the work ~~rather~~ and send the result back rather than having the ~~the~~ server ~~do~~ it.

(C) Offloading work to a client allows the server to be freed up for doing other tasks, it also allows for less data to be sent back and forth due to a client calculating data ~~and not~~ instead of the server doing the work and having to ~~send~~ send that to the client, client already has that data. The data could also be passed from client to client bypassing the server.

Dead-reckoning could be used to estimate things like player position allowing for less data to be sent over the ~~server's~~ network.

(d) Interpolation is where data is ^{estimated} ~~guessed~~ by using previous values, for example, player position could be estimated from their last position if you know what ~~velocity~~ ^{direction} and speed it was travelling. Extrapolation allows for an amount of guessing to be carried out before it sends the true position to the client. This ~~also~~ decreases load on the network but can give jerky movements if estimates are left too long.

- (e) A client's representation of the world is always wrong, sending data more often can speed up the process but causes a more load on the server, clients could send updates to each other rather than to the server.

0.5

9 1/2