# 98-Comp-B10, Distributed Systems

3 hours duration

**NOTES:**

1. If doubt exists as to the interpretation of any question, the candidate is urged to submit with the answer paper a clear statement of any assumptions made.

2. This is a CLOSED BOOK examination.
   Only non-programmable calculator is permitted

3. Answer any **five of the six** questions.
   Only the first five questions as they appear in the answer book will be marked.

4. All questions carry equal weight.

5. Most questions require an answer in essay format. Clarity and organization of the answer are important.

## Question # 1.

a. Name two advantages and two disadvantages of distributed systems over centralized systems.
b. Explain and illustrate (in graphical form) the client-server architecture of one major Internet applications (for example the Web, email or ftp).
c. Name two types of software resource and two types of hardware resource that can be shared efficiently in distributed systems. Provide examples of their sharing.
d. Explain what is meant by administrative scalability and why it is often such a difficult problem to solve.

## Question # 2.

a. A client sends a 500 byte request message to a service, which produces a response containing 8000 bytes. Estimate the total time to complete the request in each of the following cases, with the performance assumptions listed below:
   i) Using connectionless (datagram) communication (for example, UDP);
   ii) Using connection-oriented communication (for example, TCP);
   iii) The server process is in the same machine as the client.

   *[Latency per packet (local or remote, incurred on both send and receive):5 milliseconds*
   *Connection setup time (TCP only):5 milliseconds*
   *Data transfer rate: 10 megabits per second*
   *MTU: 2000 bytes*
   *Server request processing time: 2 milliseconds*
   *Assume that the network is lightly loaded.]*

b. Comment on the use of Connectionless (UDP) and connection-oriented (TCP) communication for of each of the following application:
   1. Mail access protocols (for example, POP3, IMAP);
   2. File transfer;
   3. Information browsing (for example, HTTP);
   4. Remote procedure call.

## Question # 3.

a. Explain why the total amount of data sent in a protocol might be less significant to performance than the total number of messages sent? Design a variant of the RRA protocol in which the acknowledgement is piggy-backed on, that is, transmitted in the same message as, the next request where appropriate, and otherwise sent as a separate message. (Hint: use an extra timer in the client.)

b. Outline the design of a scheme that uses message retransmissions with IP multicast to overcome the problem of dropped messages. Your scheme should take the following points into account:
   i) There may be multiple senders;

ii) Only a small proportion of messages are dropped;
iii) Recipients may not necessarily send a message within any particular time limit.

Assume that messages that are not dropped arrive in sender ordering

**Question # 4.**

a. A file server uses caching. The average hit rate achieved is 70%. If the requested block is in the cache, file operations in the server cost 7 ms of CPU time; otherwise, it takes an additional 10 ms of disk I/O time otherwise. Estimate the server's throughput capacity (average requests/sec) if it is: (Explain any assumptions you make)
   1. single-threaded;
   2. two-threaded, running on a single processor;
   3. two-threaded, running on a two-processor computer

b. Compare the thread-per-request architecture with the worker pool multi-threading architecture.
c. What the kernel must provide for a user-level implementation of threads (such as Java on UNIX)? In user-level threads implementations, do page faults present a problem?

**Question # 5.**

a. Name and discuss three key design issues for distributed file systems.
b. Sun NFS aims to support heterogeneous distributed systems by the provision of an operating system-independent file service. What are the key decisions that the implementer of an NFS server for an operating system other than UNIX would have to take? What constraints should an underlying filing system obey to be suitable for the implementation of NFS servers?
c. Compare AFS and NFS from stability point view? Are there any limits on AFS scalability, assuming that servers can be added as required? Are there any recent technological developments that would help to offer greater scalability?

**Question # 6.**

a. Adapt the central server algorithm for mutual exclusion to handle the crash failure of any client (in any state), assuming that the server is correct and given a reliable failure detector. Comment on whether the resultant system is fault tolerant. What would happen if a client that possesses the token is wrongly suspected to have failed?
b. What is the gossip architecture? Why does a replica manager need to keep both a 'replica' timestamp and a 'value' timestamp?
c. Write pseudocode for a scheme for integrating two replicas of a file system directory that underwent separate updates during disconnected operation.