1 Network applications

a) Explain the role of the DNS system in e-mail distribution. (2p)

b) What are the roles of the client and the server in a client/server application model? How does this differ from a peer-to-peer application model? (4p)

c) Some applications use RTP in combination with UDP to add timing information to each packet. Why is this timing information needed? Describe situations where a packet can arrive "too early" or "too late", and how the packet is then treated. (4p)

2 The Transport Layer

a) Briefly describe how TCP and UDP differ from each other with respect to reliability, congestion control, in-order delivery and delay variations. (4p)

b) TCP and UDP provide services that are abstractions of two different fundamental types of interprocess communication. Name these two and describe how the information is communicated in each case. (3p)

c) Suggest one application that would prefer TCP and one that would prefer UDP. Motivate why each application would prefer one or the other. (3p)

d) What are the port numbers in TCP and UDP used for? (2p)

e) TCP includes control mechanisms for flow control and congestion control. What is the purpose of each of these mechanisms? (3p)

f) TCP uses buffers on both the sending and the receiving side. Describe the purpose of the buffer at each side. (3p)
Both TCP and UDP include a checksum to verify the integrity of the data in each segment/datagram. Given that most link layer technologies include a checksum that is much better at detecting errors, why is the checksum needed at all in the transport layer? (2p)

3 The network layer

a) What is the purpose of the ICMP protocol? Also give one example of when it is used. (3p)

b) When delivering data to another computer on the same network, it is necessary to translate the IP address into a MAC address (sometimes called a link layer address or a hardware address). How is this translation done? (3p)

c) A packet that is sent over the Internet may traverse different types of network technologies where the supported frame sizes may differ. What happens if a packet that arrives to a router turns out to be too large for the next link it is to be sent over? (2p)

d) The 32-bit IPv4 address can have 4,294,967,296 different values and there currently are about 450 million computers attached to the Internet. Still, there is a shortage of available Internet addresses. Why? (2p)

4 The Link layer

a) Describe how Ethernet and 802.11 networks differ in the coordination about which node has access to the physical medium. How are the conflict that arises when two different nodes try to transmit at the same time resolved in each case? (2p)

b) Why is there a limit on how long cables you can use in an Ethernet network? (2p)

c) Describe what happens when two nodes want to transmit data simultaneously in an Ethernet and an 802.11 network respectively. Why isn’t the Ethernet approach used in 802.11 networks? Why isn’t the 802.11 approach used in Ethernet networks? (4p)

d) Nodes in an Ethernet do not have a synchronized clock. In addition to this, the clock can drift differently in different nodes. This means that reception of a frame can occur at any time, and then possibly at a pace slightly different from the internal clock at the receiver. How are these problems resolved in Ethernet? (2p)
5  OS and Architecture

a) When designing an application in which several tasks are carried out ”simultaneously”, it is possible to divide the application into different processes or into different threads. What are the advantages and disadvantages of each approach? (4p)

b) What is the role of the middleware? (2p)

c) What thread operations are the most costly? (2p)

d) What is meant by the end-to-end design principle? (2p)

e) Shared memory regions can be used for different purposes. Name at least two. (2p)

f) Why is it so much more costly to create a new process in comparison to creation of a new thread? (3p)

g) What are the main differences between a monolithic kernel and a microkernel? (2p)

h) What are the core components of an operating system? (3p)

6  Security

a) Outline how key distribution can be solved in a system with symmetric and asymmetric encryption respectively. (4p)

b) Firewalls can be implemented as packet filters or application gateways. What are the main differences between these two design alternatives? (3p)

c) Assume that you want to encrypt a media stream that uses UDP for transport. Would you choose a block or stream cipher? Motivate your choice. (3p)

d) Describe two desirable properties of a cryptographic hash function (sometimes referred to as message digest). (2p)

e) The Diffie-Hellman key exchange protocol is vulnerable to a ”man-in-middle” attack. Explain how an adversary sitting between two participants can trick them into finding they have established a shared secret between themselves, when in fact they have each established a secret with the adversary. Outline how Diffie-Hellman could be extended to protect against this possibility. (5p)
In this problem, the notation $E_{key_1}(M)$ represents encryption of the message M using $key_1$, while $D_{key_2}(M)$ denotes decryption of the message M using $key_2$.

Alice wants to send an authenticated and encrypted message to Bob using public-key cryptography. She therefore takes her message M and performs the combined encryption/authentication operation $E_{public_{Bob}}(E_{private_{Alice}}(M))$. What operation will Bob have to perform in order to both decrypt the message and ensure it’s really from Alice?

### 7 Programming

a) A "null" RMI that takes no parameters, calls an empty object and returns no values delays the caller for about 2 milliseconds. Explain what contributes to this delay.

b) Compare programming using sockets API with RMI. What are the advantages and disadvantages of each approach?

c) The `accept` method used in a TCP server returns a new socket upon a new incoming connection. Why is a new socket created in addition to the already existing server socket?

d) Assume you have a Java client/server application that uses TCP. How should the server be implemented to avoid clients from blocking each other?

e) What does marshalling mean?

f) What is an IDL (Interface Definition Language) used for in Corba? Why is no IDL needed when using Java RMI?

g) What is the purpose of the middleware? Name one example of middleware used in the course.

h) What is required from an object to be able to use it as an argument or result in Java RMI?

i) What are the advantages of using callbacks instead of polling to update clients with information?

### 8 Agreement and Coordination

a) Outline how a ring-based algorithm for mutual exclusion works.
b) Outline how a ring-based algorithm for elections works. (3p)

c) Show by way of example that the usage of a ring-based algorithm for mutual exclusion not necessarily guarantees access to a critical section in happened-before order. (4p)

9 Synchronization

a) Describe how clock synchronization can be done in a system with an external time reference (master clock). (3p)

b) Describe how clock synchronization can be done in a system without an external time reference (master clock). Abrupt changes in time should be avoided. (3p)

c) It is desirable to avoid abrupt time changes either backward or forward in time. Why? Specify two applications that would experience problems and why there would be a problem with abrupt clock changes. (4p)

10 Transactions

a) A desired property for transactions is serializability. Why is this a desired property, and what does the term mean? (2p)

b) Consider two transactions \( T \) and \( U \) defined as follows:

<table>
<thead>
<tr>
<th>( T )</th>
<th>( U )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x = \text{read}(i); )</td>
<td>( \text{write}(i,55); )</td>
</tr>
<tr>
<td>( \text{write}(j,44); )</td>
<td>( \text{write}(7,66); )</td>
</tr>
</tbody>
</table>

Describe an interleaving of \( T \) and \( U \) that is not serially equivalent if locks are released early. (3p)

c) With timestamp ordering in distributed transactions, suppose a \( \text{write} \) operation \( \text{write}(T_1, x) \) can be passed to the data manager, because the only, possibly conflicting operation \( \text{write}(T_2, x) \) had a lower timestamp. Why would it make sense to let the scheduler postpone passing \( \text{write}(T_1, x) \) until transaction \( T_2 \) finishes? (2p)

d) We have repeatedly said that when a transaction is aborted, everything is restored to its previous state, as though the transaction had never happened. We lied. Give an example where it is impossible to reset everything. (3p)
e) Explain why serial equivalence requires that once a transaction has released a lock on an object, it is not allowed to obtain any more locks. (2p)

f) One important property of a transaction is atomicity. What does this mean? (2p)

g) Consider the following two transactions:

<table>
<thead>
<tr>
<th>T</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>x = read(i);</td>
<td>write(i,10);</td>
</tr>
<tr>
<td>write(j,20);</td>
<td>write(j,30);</td>
</tr>
</tbody>
</table>

Is this interleaving serially equivalent? Could it occur in the presence of two-phase locking? In order to earn points on this subproblem, it is not enough to just answer just yes or no; you must also motivate your answer. (3p)

h) Assuming that the recovery manager appends the data entry corresponding to each write operation to the log file immediately instead of waiting until the end of the transaction, describe the information written to the log on behalf of the transactions T and U from the previous subproblem. (3p)

11 Replication and Consistency

a) To increase the availability of a replicated database, one may want to introduce Quorum-based reads and writes. What is meant by Quorum-based reads and writes? (2p)

b) How is consistency guaranteed, e.g. that all reads return the same value from the database, in a Quorum-based system? (2p)

c) One approach for serializability of transactions in distributed databases is optimistic concurrency control. Briefly describe what this is and what advantages and disadvantages there are with this approach. (4p)

d) Even with optimistic concurrency control, locking may be needed in order to get consistency guarantees. What is it that must be protected? (2p)

e) How do the replica managers in a passive replication scheme reach consensus on the effect of a request? (3p)

f) How can linearizability be achieved in the absence of clock synchronization with sufficient precision? (3p)
g) What are the differences between causal and total ordering? Also give examples of application scenarios where each type may be required. (4p)

12 Small questions

a) What is meant by Remote Procedure Calls (RPC)? (1p)

b) What is meant by Exactly-once-semantics in the context of client/server communication? (1p)

c) What does the Best-effort service model mean? (1p)

d) What are the normal two options to configure the IP address of a node? (1p)

e) Why is reliable ordered multicast impossible to realize in the Internet? (1p)

f) Briefly describe why three nodes will not be able to reach consensus in the presence of byzantine failures. (1p)

g) The physical distance between nodes is one of the hardest obstacles in distributed systems. Why? (1p)

h) What is a name server in a distributed system, and what is it used for? (1p)

i) What is meant by Two-phase commit in a distributed system? (1p)

j) What role does a Kerberos server have in a distributed system? (1p)

k) What is meant by byzantine failures? (2p)

l) Why is collision detection impossible in a wireless link layer technology? (2p)

m) What is meant by Two-phase commit in a distributed system? (2p)

n) Give two advantages of a distributed system compared to a centralized system. (2p)

o) Give two advantages of a centralized system compared to a distributed system. (2p)