1. What is socket abstraction? Name the main protocols used in interprocess communication.

Socket abstraction is the endpoint for interprocess communication, and is used by UDP and TCP. Socket abstraction contains the data structure to hold the information for communication and the system calls that manipulate the socket structure.
Main protocols: TCP, UDP

2. Why can't binary data be represented directly in XML, for example, by representing it as Unicode byte values? XML elements can carry strings represented as base64. Discuss the advantages or disadvantages of using this method to represent binary data.

Binary data cannot be represented directly in XML because binary data may contain special characters like '<', ' ', '>' and null characters, which messes up with XML structure.
Advantages: being readable on any computer. Can be read by humans.
Disadvantages: larger than binary files, needs more space to store. Requires longer processing and transmission time.

3. 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) Generally 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 order.

Senders should attach a sequence number to packets since there may be multiple senders and the ordering of packets is not guaranteed. And recipients should check the sequence numbers.
Since only a small proportion of messages are dropped in general, retransmitting the whole messages would waste a lot of resources. The recipients can send messages to the senders to announce which parts of messages are missing, and the senders send the missing parts. The senders should keep all the messages until the recipients receive all the messages.
Since sometimes it can be a long time before the senders receive the acknowledgements, and it is impossible for the senders to keep the messages all the time, we can set a time limit of keeping the messages. The senders dropped the messages after the limit.

4. What are the main characteristics of Skype?

The architecture of Skype is based on a peer-to-peer infrastructure containing users' machines and super nodes. The super nodes are selected by a series of requirements from the ordinary users' machine. Super nodes are linked with each other, each super node connects to multiple ordinary hosts.
Ordinary hosts are authenticated by a well-known login server, they connect to a selected super node and they keep a cache of super node identities.
Super nodes are connected with each other, and each super node contains the information of part of ordinary nodes. Super nodes also cache some search results to improve performance.
When the required user is found, Skype uses TCP to establish a connection for signaling call requests, and uses TCP or UDP for the streaming audio.

5. An Election interface provides two remote methods: vote: This method has two parameters through which the client supplies the name of a candidate (a string) and the 'voter's number' (an integer used to ensure each user votes once only). The voter's numbers are allocated sparsely from the range of integers to make them hard to guess.
result: This method has two parameters through which the server supplies the client with the name of a candidate and the number of votes for that candidate.

Which of the parameters of these two procedures are input and which are output parameters?

Input: name of a candidate, voter's number
Output: name of a candidate, votes for the candidate

6. Define the interface to the Election service in CORBA IDL and Java RMI. Note that CORBA IDL provides the type long for 32-bit integers. Compare the methods in the two languages for specifying input and output arguments.

CORBA IDL:

```
interface Election {
        void vote(in string name, in long number);
        void result(out string name, out long votes);
}
```

Java RMI:

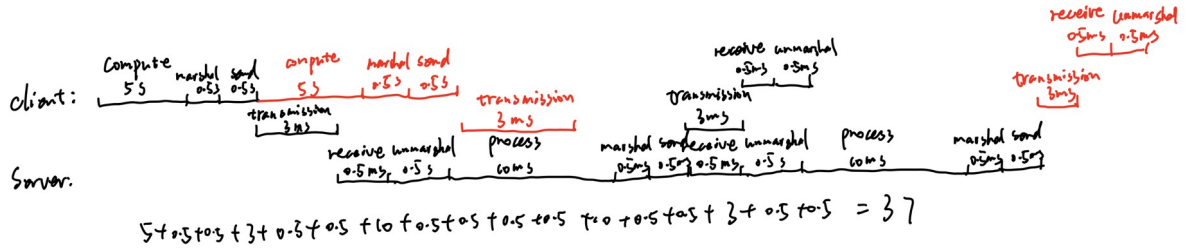
```
class Result {
        String name;
        int votes;
}
public interface Election extends Remote {
        void vote(String name, int number) throws RemoteException;
        Result result() throws RemoteException;
}
```

7. Outline an implementation for the Election service that ensures that its records remain consistent when it is accessed concurrently by multiple clients.

Create a table to record the vote process, there are at least two attributes: phone_number and name(nameID). The attribute of the number should be unique, so even if there are multiple threads adding a new record to the database, only one can succeed. The result of votes can be calculated from the table, just counting the number of votes for each candidate, and returning the result.

8. A client makes remote method invocations to a server. The client takes 5 milliseconds to compute the arguments for each request, and the server takes 10 milliseconds to process each request. The local operating system processing time for each send or receive operation is 0.5 milliseconds, and the network time to transmit each request or reply message is 3 milliseconds. marshaling or unmarshaling takes 0.5 milliseconds per message.
Calculate the time taken by the client to generate and return from two requests:
(i) if it is single-threaded;
(ii) if it has two threads that can make requests concurrently on a single processor. You can ignore context-switching times. Is there a need for asynchronous invocation if the client and server processes are threaded?

(i)
one request:
client compute time + marshal time + OS sending time + unmarshal time + server processing time + marshal results + OS sending time + network transmitting time + OS receiving time + unmarshal time = 5 + 10 + 4 * 0.5 + 4 * 0.5 + 2 * 3 = 25 ms
two request: 2 * 25 = 50ms
(ii)
The first request is in black line, the second request is in red line. The total process time is 37ms.



$5+0.5+0.5+3+0.5+0.5+10+0.5+0.5+0.5+0.5+0+0.5+0.5+3+0.5+0.5 = 37$

References:

https://www.tutorialspoint.com/what-is-sockets

https://www.techrepublic.com/article/send-binary-data-in-xml/