

**Military College of Signals - NUST**  
**Solution - Final Exams: BESE-14 (A&B)**  
**Distributed Computing**

**Part 1 (14 Marks)**

**Solution 1**

[1]

UDP: Reason is that, in voice transmission, real time traffic is important to be delivered as fast as possible. TCP, having more overheads, is not suitable for real time traffic.

**Solution 2**

[3]

When the client accesses a server, it makes an invocation of an operation in a server running in a remote computer. The following factors can affect responsiveness:

1. Server overloaded;
2. Latency in exchanging request and reply messages (due to layers of OS and middleware software in client and server)
3. Load on network.

The use of **Caching** helps with all of the above problems. In particular client caching reduces all of them.

**Proxy server** caches help with *Server overloading problem*.

**Replication of the service** also helps with *Server overloading problem*.

**Solution 3**

[1+3]

Operations that can be performed repeatedly on a given data but results of the operations remain same.

Pressing a lift (elevator) request button

Idempotent.

Writing data to a file

Idempotent

Appending data to a file

Not Idempotent

**Solution 4**

[3]

Values of its instance variables define state of an object.

**Encapsulation:**

State of an object can be accessed only by the valid methods of the object (authorized methods). Unauthorized methods are not allowed to access the instance variables.

In most of the programming languages like Java and C++, encapsulation is enforced by defining the instance variables as *private* and legitimate methods to access these variables as *public*.

In distributed systems, only those remote objects can be invoked remotely that are included in the Remote Interface of the object.

**How does it enforce Concurrency**

Possibility of concurrent RMIs from object in different computers imply that an object may be accessed concurrently. Therefore the possibility of conflict may arise which must be handled. Access to variables through only legitimate methods can be utilized in ensuring concurrency.

**Solution 5**

[2]

**Mobile Code and RMI**

**Solution 6**

[1]

TCP attempts to match the speeds of the processes. If writer is much faster than the reader, the writer process is blocked until the reader has consumed sufficient data.

## Part 2 (14 Marks)

### Solution 1

[2+3]

#### a) Message format

<b>messageType</b>	<i>int (0=Request, 1= Reply)</i>
<b>requestId</b>	<i>int</i>
<b>objectReference</b>	<i>RemoteObjectRef</i>
<b>methodId</b>	<i>int or Method</i>
<b>arguments</b>	<i>array of bytes</i>

- b) Detailed description of timeouts, re-sending of messages, dealing with duplicate messages and their discarding by using history.

### Solution 2

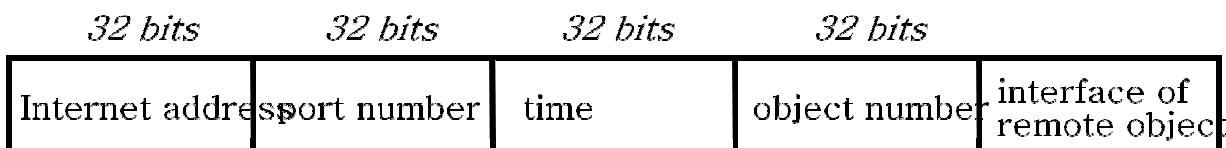
[4]

A remote object reference is an identifier for a remote object that is valid throughout a DS. It is passed in the invocation message to specify which object to be invoked. It can be passed as arguments and returned as results of remote method invocations.

#### Uniqueness is achieved as shown by the field in the representation

Internet address and port number makes it unique for a specific computer and a process.

Time and object number makes it unique within a process.



#### Example

### Solution 3

[2]

Only 1 record, because

Send is blocking , therefore there can only be one record to be kept in history (Details omitted from Solution)

### Solution 4

[3]

- Get: for retrieving the contents of a web object referenced by the specified URL
- Head: for retrieving a header from the server only, not the object itself
- Post: to send data to a process on the server which executes and processes it.
- Put: to request a server to store the contents enclosed with the request to the server machine in the file location specified in the URL
- Delete, Options, Trace

## Part 3 (11 Marks)

### Solution 1

[4]

Each process has a remote object module to support RRM. This module is called by the components of RMI software, e.g. when a request message arrives the Remote Object Table (explained below) is used to find out which local object to be invoked. RRM records the

correspondence b/w Local Object Reference in that process & Remote Object Reference (which are system wide).

#### Operations of RRM

RRM performs two operations

- i. Translating b/w local and remote object reference.
- ii. Creating Remote Object Reference

The actions of RRM can be further explained as follows

1. When a Remote Object is to be passed as argument for the first time, the RR-Module is asked to create an ROR, which it adds to its table. (Server Side)
2. When an ROR arrives in a Request-Reply message, the RR-Module is asked for the corresponding LOR, which may refer either to a proxy (client) or to an RO (server).
3. In the case the ROR is not in the table (this can happen on client side when it sees the ROR first time), the RMI software creates a new proxy & ask the RR-Module to add it to the table

#### Entries in Remote Object Tables

RO Table includes:

- An entry of all the RO held by the process (in Server)
- An entry of each local proxy (at Client)

#### **Solution 2**

**[1]**

Proxy (Client): Skeleton (Server)

#### **Solution 3**

**[1]**

Proxy (Client), Dispatcher (Server), Skeleton (Server)

#### **Solution 4**

**[2]**

When client binds to a distributed system object an (virtual) implementation of the objects interface called a proxy is loaded into the clients address space.

There is one proxy for every Remote Object for which a process holds the ROR.

Proxy implements methods in ROR but it implements them quite differently. Each method of proxy marshals a reference to the target object, its own method id and its arguments into a request message and sends it to the target. Then it waits for the reply message. Un-marshals it and returns the results to the invoker.

Whereas, servants actually implement the methods in ROR.

#### **Solution 5**

**[2]**

ROR can be passed as arguments and returned as results of remote method invocations.

ROR can initially be registered in Binder (Java Registry in case of Java RMI) and Client can access it through a Lookup Service

#### **Solution 6**

**[1]**

Server Side

### **Part 4 (11 Marks)**

#### **Solution 1**

**[2]**

Web crawler retrieves Web pages which are then indexed and used by Search engines.

#### **Solution 2**

**[2]**

On Server Side

**Solution 3**  
**Content Delivery Network**

[2]

**Solution 4**  
**Software as a Service**

[2]

**Solution 5**

[1]

Omission Failures  
Server Crash  
Network Failure

6. Draw an RDF Model of scenario: *MCS is a constituent institute of NUST. NUST is headed by a Rector and MCS is headed by a Commandant. Rector's email ID is rector@nust.edu.pk .* [2]

**Solution 6**

[2]

