Reading Assignment 2
sample answers

1) Briefly describe the similarities and differences between RPC and RMI. (3)

_Similarities:_
- Both RPC and RMI follow the call/return style, where communication is initiated by the client, and the server reactively responds.
- Both RPC and RMI provide communication via local stubs, which support an interface for method calling.
- In RPC and RMI calling and returning is implemented via message passing.
- Both use separate mechanisms for dynamic binding (e.g., object registry, JINI).

_Differences:_
- Unlike RPC, RMI does not hide the distribution in the language. The remote objects are declared as “remote”.
- Unlike RPC stubs, RMI stubs need not be compiled into the client. They can be downloaded at runtime.
- In RMI, Marshalling is simplified as the arguments are passed by value only.
- The JVMs in Java take care of hiding the platform dependencies.
- RPC was designed to support communication among different languages, whereas existing implementations of RMI use the same language at both the ends.

2) Under what circumstances would you prefer to use asynchronous over synchronous communication and vice versa? (2)

_Synchronous communication is preferred:_
- When it is fine/desirable to block the caller until an answer is received.
- When the communication framework is reliable by nature.

_Asyncronous communication is preferred:_
- When the caller can proceed without receiving an answer (e.g. when no result is expected to be returned by the receiver, or if the result can come later).
- When either of the components can initiate communication.
- When failures in the communication or peers are to be expected, rather than exceptional.

3) Briefly describe how communication failures are detected and handled by an application, both in synchronous and asynchronous communication. (2)

Timers need to be set in both cases. In the synchronous case, the timer may be supported by the application, or by the middleware (who then initiates a retransmission automatically, or returns an error to the application).

In the asynchronous case, the timer needs to be set by the application.

4) Briefly describe how communication failures can be masked both in the case of persistent and transient communications.

In case of transient communication, the failure needs to be immediately recognized and recovery initiated.

In case of persistent communication, applications are already developed to handle long
delays between sending a request and receiving an answer so recovery procedure may be
relaxed as middleware will keep messages till they are delivered.

5) Briefly describe similarities and differences between asynchronous message
passing and communication by data sharing. (3)

Similarities:
• They are both asynchronous communication mechanisms.
• Any component can initiate communication.

Differences:
• In message passing, the middleware monitors the communication channel and
call the receiving component when a message comes in. In data sharing, the
receiving component needs to monitor the shared structure itself to see when it is
modified.
• In data sharing it is up to the components to implement synchronization of access
(by using semaphores, monitors, etc.).
• Asynchronous message passing better supports interaction complexity (i.e., more
responsive components), while data sharing better supports data volume.

Note: both data sharing and asynchronous message passing may be used to implement
synchronous communication: when using message passing, the first component does
nothing until a replying message comes in; when using data sharing, the first component
does nothing until the data piece representing the result is updated.