Alexandros Karakasidis
Assignment 2

Exercise 2.3
Such a service could belong to the levels of the communication subsystem in order to facilitate the advantages of layered architectures. In this way, the middleware can be independent of the multicast system.

On the other hand, the multicast system could be implemented as a module of the middleware system and use the properties of the lower levels of the protocol suite. This scheme also has the advantage that the protocols of the lower levels need not to be altered. This seems to be the best approach.

Exercise 2.6
This can work under the constraint that we use an additional initial byte, and that the code representation of this first byte is symmetric, so that machines featuring both big and little endian encoding can decide which is the representation. For example if this byte had all its bits equal to zero, the trailing bytes can be encoded in the little endian format and vice versa.

Exercise 2.8
The main drawback of letting a server always have the same endpoint is that, if a server changes location, all clients using the objects of the DCE have to be informed of this change. Using a daemon, only this daemon has to be informed for this change. In other words, the utilization of the daemon, offers transparency to the client.

Exercise 2.9
An example implementation of a reference to a transient remote object is that of a client, accessing a URL server handling the respective objects. The objects loaded in the URL server are transient, as long as they exist only when the server is up and running. Sample code for an ATM client attaching to a URL server follows.

```
// Client.java

public class Client {

  public static void main(String[] args) throws Exception {
    if (args.length == 0) {
      System.out.println( "Usage: vbj Client URL_IOR");
      return;
    }
    org.omg.CORBA.ORB orb = org.omg.CORBA.ORB.init(args,null);
      com.visigenic.vbroker.URLNaming.Resolver resolv =
        com.visigenic.vbroker.URLNaming.ResolverHelper.narrow(
          orb.resolve_initial_references("URLNamingResolver"));
      org.omg.CORBA.Object obj = resolv.locate(args[0]);
      Bank.AccountManager manager =
Bank.AccountManagerHelper.narrow(obj);
      String name = args.length > 1 ? args[1] : "Jack B. Quick";
      Bank.Account account = manager.open(name);
```

```
      float balance = account.balance();
      System.out.println
        ("The balance in " + name + "'s account is $" + balance);
  }
}
```

Exercise 2.15

Given that we have only primitives for transient asynchronous communication (e.g. UDP). In order to achieve synchronous communication, the client must be blocked until a response from the server is received. This can be done, by implementing the following scheme:

- The client sends a packet to a server using asynchronous primitives.
- This packet passes through the operating system or the middleware, which blocks the client
- The message is transmitted.
- When the server's operating system, or middleware, receives this packet delivers it to the server, and it sends an answer using the same primitives to the client.
- Considering that the client's system may use buffers, when it receives a message for the client, it unblocks the application.

Alternatively, the application itself might be transmitting UDP packets periodically, and sleeping. This is continued, until it receives a UDP packet from the server. Again an operating system, or middleware buffer is necessary.

Exercise 2.22

Transient synchronous communication face scalability problems, especially in large-scale and widely-dispersed interconnected networks, because some of the parts of the network may not be always accessible. A possible solution is to revert to persistent communication in order to overcome the problems arising by the large scale interconnection (failures etc). One other possible solution is to use devices (e.g. proxies) which would act as representatives and interfaces between different network portions.

Exercise 2.24

Given a collection of computers, which is organized in a ring. In order to achieve maximum delay, we will use a token as following:

- The token is circulated only towards one direction (either clockwise or counter-clockwise).
- A computer can transmit only when it holds the token, and adjacent communication is not allowed.

Following this policy, the token will have maximum delay. The constraint, that only the adjacent communication is prohibited, stems from the fact, that a node in a token ring, does not know the exact position of the other nodes, in order to select the longest path, so this can be reduced, only to immediate neighbor communication.

Exercise 2.26

Let t be the duration of the maximum burst. In order to achieve maximum burst, the bucket has to be full, i.e. contain 1MB. Since during the maximum burst, data may keep coming, the total contents of the bucket equal to

$$1MB+20MB*t.$$

The maximum transmission rate is 50MB/s and if this remains during the entire burst procedure, the bucket output rate is:

$$50MB*t$$

equating the two quantities:

$$1MB+10MB*t = 50MB*t$$

Finally:

$$t = 0{,}025 \text{ sec}$$