Homework 1 Sample Solution

1. (a) Define each of the following attributes of a distributed system: ***transparency****,* ***openness*** *and* ***scalability****. (b) Discuss two benefits that each of these attributes provide. (c) Describe two different ways in which you would implement each of these attributes.*

(a): **Transparency** of a distributed system is the ability of the system to present itself to users and applications as if it were a single computer system. **Openness** means that the system is made up of components that may be obtained from a number of different sources, which together work as a single distributed system. In other words, it can run on hardware provided by multiple vendors and use a variety of standards-based software components. **Scalability** is a measure of the ease with which a system can be increased in size. The metrics to measure the scalability of a system are based on its size (# of users/applications supported), geographical scalability (distance between nodes) and administrative scalability (number of administrative domains).

(b): **Transparency** simplifies the interface that is presented to a user as it can be used to hide details not important to the user. It also hides the fact that inevitable failures and recoveries may occur. The main advantage of an **open** distributed system is that such a system will provide interoperability with other systems and also allows applications to be easily ported to other implementations of the same system. A well-implemented **scalable** system will be able to grow without significantly affecting the overall performance; it can also be used to provide redundancy, which in turn can be translated into better performance.

(c): **Transparency** can be implemented on a multitier system in which from the user's point to view, he or she only connects to the front tier without knowing that the information may be coming from several other sources the front tier connects to. Similarly, in can also be implemented when a programmer hides the details of a distributed application to make it look as a single computer. **Openness** can be implemented by having a well-defined interface to a distributed system; additionally it can be implemented by ensuring that the communication mechanisms are open. **Scalability** can be implemented by designing the applications in such a way that most of computation is delegated to the clients instead of the servers. Additionally we could replicate data on multiple systems.

2. (a) Discuss the differences and similarities between a ***distributed operating system*** *(****DOS****), a* ***network operating system*** *(****NOS****) and* ***middleware****. (b) Draw a diagram of the network stack and show where each of these pieces of software fits into the network stack.*

(a): **Distributed Operating System** is a tightly coupled OS for shared memory multiprocessors and homogenous multi-computers. A **Network Operating System** is a loosely coupled OS for heterogeneous multi-computers connected using LAN,

WAN etc. **Middleware** is an additional layer on top of NOS that implements general-purpose services.

The main difference between DOS and NOS is in how each views and manages local and global resources. DOS views system resources as globally owned and manages them as such. NOS builds on capabilities provided by local OS and extends it to satisfy new needs. The main function of middleware is to provide transparency.

*3. Most **enterprise computing systems** are implemented as **three-tier distributed system**, with a Web Server in the first tier that provides an interface for the client, an Application Server in the second tier that provides business logic processing, and a Database Server in the third tier that provides access to a database. (a) Why do you think enterprise computing systems are designed in this way? (b) How do the components in the three tiers communicate with one another?*

(a): Designing enterprise computing systems as three-tier systems has the following advantages:

I. Splitting into three layers makes for simple layers with easy integration.
II. Fault detection becomes easy.
III. It is convenient to upgrade/replace any of the layers.
IV. During development, all three layers can be developed in parallel, thus reducing design time.

From the client's point of view, he or she only needs to understand this interface to the application. No other knowledge of working of the application is required. From the processing point of view, this organization separates actual processing from application level. Finally, from the data's point of view, it separates processing and actual data as data may be accessed at different speeds (e.g. data accessed from cache vs. from disk).

If instead of a three-tier implementation, we had a two-tier implementation (user layer-data layer), then it would not have been easy to provide transparency, openness and scalability.

(b): The components typically communicate with each other by first establishing a connection and then by sending and receiving information packets. The actual physical media will be some sort of high-speed network.

*4. (a) Explain the difference between the **client-server model** and the **peer-to-peer model** of distributed computing. (b) For each of these models, give an example application for which it is most appropriate to use that model and explain why.*

(a): The main difference is the notion of a **server**. The server in the **client-server model** is a shared resource for the clients. The clients interact with it by sending request and receiving replies. In the **peer-to-peer model** there is no single server and all the clients can talk to each other to share information.

(b): An example application for client-server model of distributed system architecture is online banking system. It is more appropriate to use this model for the above mentioned application for the following reasons:

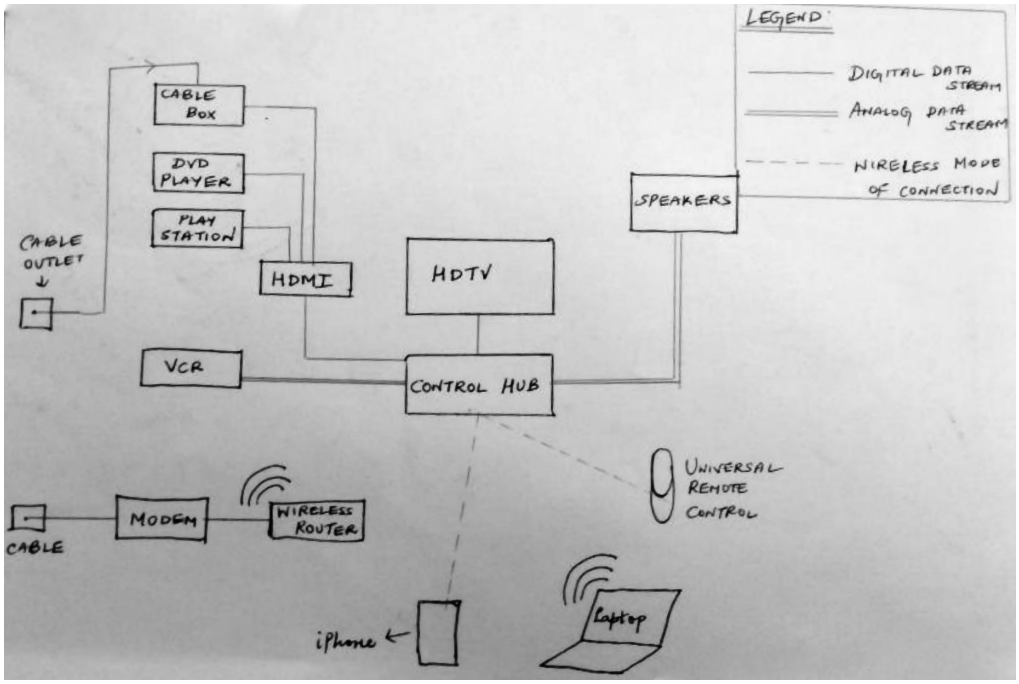
- Data stored in the central servers are confidential and cannot be shared with every machine in the network. Transactions involving critical data are often performed in the banking application and these are to be protected from unauthorized accesses.
- There is always a defined order in processing the requests from the clients by the server (i.e.) consistency is maintained in the system.

Example applications for peer-to-peer model of distributed system architecture are file sharing applications like bittorrent. Its more appropriate to use this model for this application for the following reasons:

- A peer can request for a piece of data from other peers in the system who could be holding that information, i.e., decentralized data storage mechanism. This avoids overloading a single machine with too many requests. This scheme will distribute the network traffic throughout the system instead of concentrating it in a fewer number of channels.
- The system is easily scalable as more number of users can be easily added to the system without much additional infrastructure needed. This feature is very essential for the file sharing applications where there could be a sudden spurt in the number of users wanting to use the application.
- Due to decentralized data storage system, there will be no overloading of request queues in any machine in the system and this helps in faster processing of the requests.

*5. The **Internet of Things** is being promoted on TV by companies such as Cisco. In a domestic Internet of Things, home appliances can interact with each other, and can be monitored and controlled remotely. Such appliances might include kitchen appliances, laundry and cleaning appliances, lighting, heating and cooling devices, fire and burglar alarms, garden watering and maintenance devices, and anything else that you can think of. (a) Design a domestic Internet of Things that involves two or more such appliances and draw a high-level diagram of your design. (b) Do you need a centralized server to manage your distributed system? Explain why or why not. (c) Explain how the appliances can interact with each other without your explicit intervention. (d) Explain how each appliance can be, and needs to be, monitored and controlled.*

(a): The devices considered are the ones that could be interconnected in a home multimedia system: HDTV, VCR (Video Cassette Recorder), DVD player, Game Console, Speakers, PCs, modem with cable connection, wired/wireless router, remote controller, mobile phones, some storage devices like disks, etc.



(b,d): Yes, there is a control hub which takes care of the overall management of the system. The control hub in a way monitors the connections between various devices. For example, it has to check whether the input from the VCR or DVD or set top box is connected to the HDTV depending upon the choice of the user. The control hub is also responsible for providing access to the system for authorized users only. It can maintain a table storing the MAC addresses of the authorized devices. When a new device wants to establish a connection the control hub ensures that the device is in its authorized users list before providing access for the requesting device.

(c): The user can control the system through his/her mobile phone, remote-controllers etc. The user can wish to view his/her program of interest on the TV. The control hub can store a list of recommended material for each user. As soon as the user connects to the system, the control hub sends the list of recommended material that could be of interest to that user to the device through which the user connects with the system.

*6. Today's **embedded systems**, such as those in modern automobiles, contain lots of electronic devices. (a) Give an example of a feedback-control system in an automobile. (b) Describe at a high-level how the devices in that system provide feedback and are controlled. (c) Provide a high-level diagram of the feedback control system to illustrate your description.*

(a-c): A possible example is given in the first discussion session.