Unit V

Mobile Middleware

• Software that supports mediation between other software components across heterogeneous platforms.
• To provide abstractions that reduce development effort, to offer programming paradigms that make developing powerful applications easier.
• A software layer between the operating system and the applications.

Software models with n/w models

Types of middleware for mobile computing

➢ Middleware to support application adaptation
➢ Mobile agent systems
➢ Service discovery frameworks
Adaptation

• It helps applications to deal intelligently with limited or fluctuating resource levels.
  – Adapt behavior and expectations to conserve scarce resources
  – Adjust quality of service (QoS) – guarantee performance
• Mobile audio application - It provides lower quality audio in case of disruption in bandwidth instead of stop delivering any audio.
• Mobile Video application – It provides lower quality video in case of disruption in bandwidth.
• Mobile Video game application – When the battery level is very low during the running of gaming application, it adjusts to decreased battery level by modifying resolution to conserve power.

Types of adaptation

• Application-aware strategies
• OS aware strategies

Mobile Agents

• It consists of software with data, which can move from one computing system to another autonomously.
• It functions for a device or system on the present host.
• Described as an autonomous software which runs on a host with some data
• Dynamically moving software to another host (which has other required data) as and when required
• A powerful tool for distributed applications and retrieval of remote host information
Mobile Computing Unit 5

- Has dynamic software that runs on different hosts at different times
- Makes available the resources of its host to the resource-scarce devices, discovers new resources, and monitors the distribution of resources
- Also manages the network and the distributed computing systems
- Very effective alternative to the use of application-specific server(s) and device middleware for retrieving information and messages
  - Programs migrate directly to servers
  - Gain access to data available in servers
  - And only return to their “home” to deliver results.

**Benefits of mobile agents**

- Disconnected operation is supported
- Access to large amounts of data
- Allow the functionality of servers to be expanded dynamically

**Applications of mobile agents**

- Electronic Commerce
- Network resource management
- Information retrieval

**Mobile agent systems**

- Aglet
- JADE
- Grosshopper
- Voyager
- PMADE
Example of a Mobile agent-based architecture

- Agent moves at instants T1, T2, and T3 to process a request, get email, and get records from a database, respectively.
- When a mobile agent moves at instant T1, T2, or T3, it saves its own state at the host and transmits this saved state to the next host in order to resume execution of the codes starting from the saved state.
Characteristics of a mobile agent

1. Mobility of code and data from one computing system (host) to another
2. Ability to learn in order to adapt code and data to the host computing system
3. Ability to clone, extend, or dispose itself after its role finishes
4. Compatibility to the hosts
5. Ability to continuously and autonomously process requests and send responses and alerts
   • An alert—an unsolicited message, record, or information

Advantages of a mobile agent

1. Asynchronous running of codes on diversified heterogeneous hosts
2. Reduced computational and data requirements on the devices with limited resources
3. Tolerance to connection failures
4. Only the agent source (for example, device middleware, which sends the agent) needs to be modified in order to redefine the functions expected from the agent
5. There is no need of a centralized or an application-specific server
6. An agent can send the requests to a computing system as well as generate responses for requests from the system. An agent thus has certain similarities to peer-to-peer architecture
7. The connection protocol and the connecting network between host and source are immaterial

Summary

• Mobile Agent—an autonomous software which runs on a host with some data for the device or system
• Dynamically moving (migrating) software to another host (which has other required data) as and when required
• No need of a centralized or an application-specific server when using agent
Service Discovery Middleware

- A printer becomes usable and discoverable as soon as it is plugged in.
- Plug and Play technology
- A collection of protocols for developing highly dynamic client server applications that standardizes a number of common mechanisms for interaction between clients and services.

Service advertisement allows services to announce their presence when they enter the network and to announce their departure from the network.

Service Discovery allows clients to discover dynamically services present either in their local network or in internet.

Service Discovery Framework

- Jini – provides a spontaneous discovery and service interaction framework.
- Salutation - a discovery mechanism
- Universal Plug and Play UPnP- a programming language and platform independent discovery mechanism by relying on HTTP and XML
- Service Location Protocol - a vendor independent discovery mechanism

Jini

- A runtime infrastructure that resides on the network and provides mechanisms that enable to add, remove, locate, and access services.
- The runtime infrastructure resides on the network in three places:
  - in lookup services that sit on the network;
  - in the service providers (such as Jini-enabled devices)
  - in clients.

Protocols used

- Discovery - clients and services to locate lookup services.
- Join - a service to register itself in a lookup service
• Lookup - a client to query a lookup service for services that can help the client accomplish its goals.

**Jini** provides a spontaneous discovery and service interaction framework. Jini functions on top of the Java RMI distribution middleware. A Jini-enabled device can function as both a service consumer and provider.

Jini contains three different types of processes: discovery, joining and service lookup. In *discovery* the Jini device broadcasts an announcement upon joining a new network. The announcement contains both information on how to contact services provided by the device as well as groups it intends to join. In *joining* the device provides information about the services to the Jini lookup service through a RMI stub it receives in joining from the lookup service. Service identifier is based on the Java interface and attributes describing the service. The lookup service registers the services for a certain period of time. This is called the lease. In *service lookup* the Jini device queries the lookup service. The query produces a set of stubs that can be used to interact with the remote object. Java Reflection API can be used to inspect what methods are implemented by each of the objects accessible through the stubs. Of course it might be that e.g. web services – wrappers are provided to the interfaces, in which case the client could use this interaction mechanism instead of further relying on RMI.

Jini is not involved in the interaction between the client application and the service producer. The lookup service is basically a directory. Jini Surrogate Architecture provides support for interoperations with Universal Plug and Play.

**Salutation** is a discovery mechanism, which supports brokering (i.e. protocol-gateway functionality) in addition to discovery. Salutation consists of three Network Entities: the *Salutation Manager, client* and *services*. Salutation Managers exist independently on brokers as well as parts of the clients and services. The services are registered to the Salutation Manager.

The Salutation Manager role depends on which of the three available Salutation protocols are utilized. In the *Native Personality* Salutation Manager merely acts as a directory, which the client queries to locate services. Query can consist of a service type and attributes required of the given service type. Interaction between clients and services is
handled without Salutation Manager involvement. In the *Emulated Personality* client first discovers the service. However, after this, interaction messaging between the client and service is brokered by the Salutation Manager, which acts as a protocol gateway between the two participants. This is required if no common interaction protocol exists. In the *Salutation Personality* the Salutation Manager sets up a common protocol (Salutation Manager Protocol) and protocol data structures between the client and the service after service discovery. After the Salutation Manager has made the initial setup, interaction happens directly between the client and the service according to the Salutation Manager Protocol.

**Universal Plug and Play**, UPnP, provides a programming language and platform independent discovery mechanism by relying on HTTP and XML. UPnP relies on listening and responding to HTTP-based requests on a particular multicast channel. UPnP architecture consists of *control points* (i.e. UPnP client devices) and *UPnP devices*. UPnP can be seen to contain five phases: discovery, description, control, event notification and presentation. In *discovery* UPnP devices advertise their services to the control points in the environment. No directory or lookup service is used. The discovery limits to basic device information. UPnP device communication is based on Simple Service Discovery Protocol (SSDP). In *description*, an XML descriptor of the UpnP device and its services is provided to the control point based on the request from the control point. In *control*, control points can send requests to the UPnP devices based on the received XML descriptor. Control interaction is based on SOAP. In *event notification*, a UPnP device notifies control points that registered a service in control phase of changes to the services they registered to, following a publish-subscribe –style messaging paradigm. In *presentation*, a UPnP device can (but is not required to) provide a URL to the control point. The URL provides location of UpnP device control interface. Control point can fetch the URL and use it to enable control of UPnP device to control point end-user.

**Service location protocol**, SLP, is intended to provide a vendor independent discovery mechanism. SLP consists of three entities, which are central elements of the architecture:
user agents perform the discovery, service agents announce the services and directory agents act as the lookup service, collecting the service addresses (i.e. service URLs) and responds to user agent service lookup requests. Directory agents can be found by service and user agents through passive discovery, in which case both parties listen to directory agent periodic multicasts on a protocol-specific multicast address. Alternatively, in active discovery SLP requests are multicast by user and service agents. Alternatively, if a DHCP server is defined to the user and service agents, the DHCP server can provide this information as one type of active discovery. Directory agent is not mandatory to SLP. User and service agents can locate each other through use of multicast. Use of directory agent limits the amount of multicast usage, ensuring better scalability with increase of number of user or service agents or participant changes. SLP is not involved in the interaction between the client application and the service providing application.

Jini defines a runtime infrastructure that resides on the network and provides mechanisms that enable you to add, remove, locate, and access services. The runtime infrastructure resides on the network in three places: in lookup services that sit on the network; in the service providers (such as Jini-enabled devices); and in clients. Lookup services are the central organizing mechanism for Jini-based systems. When new services become available on the network, they register themselves with a lookup service. When clients wish to locate a service to assist with some task, they consult a lookup service.

The runtime infrastructure uses one network-level protocol, called discovery, and two object-level protocols, called join and lookup. Discovery enables clients and services to locate lookup services. Join enables a service to register itself in a lookup service. Lookup enables a client to query a lookup service for services that can help the client accomplish its goals.

**Concepts in service Discovery Framework**

- Services – Discovery of Services
Mobile Computing Unit 5

- Garbage Collection – remove outdated information from the network
- Eventing – timely notification of important event like paper is jammed during printing service is used.
- Security – Authentication on both client and services side
- Interoperability – between different kinds of available service discovery devices like Jini enabled device should be interoperable with Universal plug and play client.

**Ad hoc and sensor networks**

**Overview of ad hoc networking:**

Ad hoc networking refers to a network with no fixed infrastructure. When the nodes are assumed to be capable of moving, either on their own or carried by their users, these networks are referred to as mobile ad hoc networks (MANETs).

**Properties of an Ad hoc Network:**

- No preexisting infrastructure
- Limited access to a base station
- Power limited devices
- No centralized mechanisms

**Wireless Sensor network (WSNs)**

Wireless Sensor network (WSNs) is composed of tiny electronics devices, known as sensor node, distributed over a region to observe some phenomenon. Some typical measures that can be observed by sensor nodes are temperature, pressure, humidity and light etc. With recent developments, there are many applications, where sensor nodes (S-nodes) are deployed to interact with environment. This allows us real time information whenever it is needed immediately. In fact, any kind of automation can be possible by using set of sensor nodes. For example sensors automatically detect the fire, set of sensors monitor the human and animal movements etc.
Mobile Computing Unit 5
A smart sensor node architecture

- It consists of an RF transceiver for communication, a *microcontroller* [CPU, memory, and ADC (analog-to-digital converter)], and an *energy source or a power supply*
- A charge pump traps the charge from the radiations (for example, from WiFi transceiver or an access-point)
- Alternatively, an energy-harvesting module can be used to trap solar radiation and store the energy
- The *RF transceiver enables a node to receive data packets from nearby nodes and route these to next hop of the packet*
- A wireless sensor node disseminates information to the network, central computer, or controller
- tinyOS and CORMOS (communication oriented runtime system for sensor networks)
- tinyDB, SensorWare, and GSN (global sensor networks) application-oriented middleware
- Sensor networks deploy special routing protocols such as CGSR, DSR, or AODV

**Applications**
The applications of WSNs are innumerable, since each sensor node is capable of monitoring a wide variety of ambient conditions such as temperature, humidity, lightning condition, pressure and noise levels. Below, we have given some typical application areas for WSNs.

1. Military applications: Sensor nodes can be spread across a battlefield or enemy area and be programmed to track and monitor enemy troop movements or movement of terrorists and can be used to locate their exact positions. They can also be used instead of mines to immediately detect any movement, thus safeguarding human lives.

Sensor networks can also be used to detect the use of biological or chemical weapons, and relay this information to commanders, so as to allow sufficient time for soldiers to take defensive measures in the field.

Improved battlefield communication is another benefit of WSN for military applications. Here, the soldier with a personal digital assistant (PDA) acquires an extended sense by interacting with the surrounding WSN. Target field imaging, security and tactical surveillance, and intrusion detection are other similar applications.

But military applications impose several special requirements on the WSN. Firstly, auto-deployment and self-organization should be supported. Secondly, detection of a sensor node should be difficult, otherwise an adversary may determine the location of sensor nodes easily, and compromise them. In such a situation, information with the compromised node can be stolen, or invalid information can be injected into the system.

Thus, secure communication is vital but may not be feasible in the form of large keys and robust protocols because of the limited computation power available. This is a major challenge in the deployment of WSN for military applications.
2. Environmental applications: Sensor networks are being increasingly used for environmental concerns. Examples include tracking the nesting habits of seabirds by monitoring a large geographic region with human presence, or attaching the sensors directly to large mammals to monitor their behaviour.

Monitoring of river currents is another application of WSNs, to measure their water inflow and mixture from various sources. Water-quality monitoring may also be useful to determine contamination with bacteria or other harmful pollutants.

A major application in this category is the spread of sensor nodes across a forest to monitor temperatures and give early warnings of fire outbreaks. Weather prediction, climate monitoring, distributed computing, pollution tracking, seismic detection, detecting ambient conditions such as temperature, movement, sound, light, or the presence of certain objects, inventory control and disaster management are other similar applications.

The major requirement of environment sensors is the need for rugged operation in hostile surroundings and extended sleep periods to maximize the lifetime of the network.

3. Medical applications: WSNs can be used in medical applications by using the sensor node as a device that can reside on or within the human body and perform tasks that are currently done by costly machines. These include glucose monitors for continuous reading of insulin levels in diabetic patients; heart monitors for keeping track of the functioning of the heart, especially for patients with irregular heartbeats or coronary diseases; and artificial retinal and cortical implants to electronically transmit information to visually impaired persons.

Another example of the use of WSNs in medical applications is the vital statistics repository, which takes the form of a smart card holding medical information on persons with severe allergies to certain substances or medications.
Other medical applications include use in telemonitoring of human physiological data, tracking and monitoring doctors and patients inside a hospital and insurance cards.

Medical applications using sensors also have special requirements. They must be safe and biocompatible to merit continuous functioning inside the human body and not damage the tissues. The sensors must be designed for long-term operation and have enough power so that frequent surgery is avoided. This means that they must be highly fault-tolerant and provide redundancy and graceful degradation in failure scenarios.

4. Industrial applications: For use in industrial applications, low-cost sensor nodes could be attached to equipment to monitor performance. They could also be attached to parts as they move through an assembly pipeline on the shop floor. Thus, inefficiencies in plant process flow can be recognized quickly, rush orders could be expedited more easily and customer queries could be answered faster and more accurately.

Radio frequency identity (RFID) tags are already being placed on items like gasoline and other merchandise to allow fast and accurate scanning at checkout and for inventory tracking. These tags could be replaced by wireless sensors at fixed locations and used for tracking.

The use of sensor nodes for industrial and commercial applications requires their cost to be made very low so that they can be used in bulk. The protocols in use should, therefore, also be highly scalable.

5. Urban applications: WSNs can be used for various urban applications like transportation and traffic systems, auto-identification by driving license, parking availability, security monitors in shopping malls, parking garages, city streets and home security.
• Medical Applications - patients can be monitored by the sensors attached to them
• Industrial Applications
  Industrial plant wireless sensor networks—Industrial plants use large number of sensors in instruments and controllers
• Environmental Applications - Environmental parameters like temperature, pressure, light, rainfall, and seismic activity management systems
• Traffic monitoring using traffic density wireless sensor networks
  • Traffic density information aggregated at a central server
  • The server relays this information to motorists on wireless Internet
  • A traffic control server sends the traffic reports on Internet
• Military applications - The voice of a person can be sensed by a wireless sensor network and deployed in remote border areas.
  • This monitors the enemy troop and machines movement
  • It is used worldwide for monitoring movement of goods, movement of books in library, and supply chain management systems
• Home automation—including security using a wireless sensor network

**Differences from mobile ad hoc networks**

Mobile ad hoc networks (MANETs), which are also made up of a number of wireless, mobile nodes. However, there are significant differences between MANETs and WSNs. These are as follows:

1. The number of sensor nodes in a sensor network is much more than that in an ad hoc network. Usually sensor networks consist of 1,000 to 10,000 sensor nodes covering the area.
2. Sensor nodes are generally static and cooperate together to transfer the sensed data.
3. In mobile ad hoc networks, the number of nodes is much less, but their mobility is very high.
4. Sensor nodes mainly use the broadcast communication paradigm, whereas most ad hoc networks are based on point-to-point communication.

5. Another difference between the two is that sensor nodes have a much lower power consumption requirement, of the order of 0.75 mW.

**Characteristics**

1. Large number of Nodes (1000 s in number)
2. Sensor nodes can be readily deployed in large number in various types of unstructured environments
3. They rely on wireless channels for transmitting and receiving data from other nodes
4. They are of self configuring capacity
5. Self Healing
6. Dynamic Routing

**Challenges**

- Constrained resources
- No centralized authority
- Limited power
- Wireless Communication
- Limited Computation and storage
- Limited input and output options

**Sensor network Security**

- **Small keys:** Wireless sensor nodes have significantly less storage space than other wireless computers. Storing large keys is not practical. Therefore security protocols must be adopted with these smaller keys which can provide sufficient levels of security.
• **Limited Computation**

• **Changing network membership:** Over the lifetime of a sensor network, the active membership of the sensor network varies.

• **Arbitrary topology:** Sensor networks are deployed in different ways. For eg. Implied in specific locations- neighbors are predetermined. Deployed with less precision- well defined distribution. Deployed in an arbitrary fashion.

**Mobility**

• The overhead is unavoidable because of the effects that mobility has on the wireless communication protocols.

• Mobility may occur because of application needs.

• Loss of connectivity owing to mobility is possible during any sufficiently long data transfer.

• Connectivity may be lost not only between a pair of nodes but also with the rest of the network.

• In addition to the higher data loss rate owing to the wireless medium, movement of nodes leads to data loss. Depending on the importance of the data that are lost, the packet may be retransmitted or dropped.

• Since the physical location of a mobile node changes, the neighbors of that node also change.

• A set of nodes needs to maintain consistency.

**Protocols and Autoconfiguration**

• Neighborhood discovery

• Topology discovery

• Medium Access Control schedule construction

  Hidden terminal problem

  Exposed Terminal problem

  Solution using RTS and CTS
**Mobile Computing Unit 5**

**Mobility Requirements**

The characteristics and applications of wireless sensor networks are changing so rapidly. Existing technologies may not be ideal or even suitable for future networks.

- Movement detection using Global Positioning System
- Patterns movement
- Changing group dynamics
- Resynchronization

**Summary**

- Sensing of the light level, temperature, location shift, time stamps of GPS satellites vibration, pressure, weather data, noise levels, traffic density, and nearby passing vehicles
- Smart sensors have computational, communication, and networking capabilities
- Constrained by their small size, limited energy availability, and limited memory
- Operate at limited computational speed
- Limited bandwidth
- RF transceiver for communication, a *microcontroller* [CPU, memory, and ADC(analog-to-digital converter)], and an *energy source*

Multiple Choice Questions

1) An autonomous software which runs on a host with some data for the device or system (a)Mobile agent (b)Mobile OS (c)application software (d)None of the above

2) a network with no fixed infrastructure (a)wireless network (b)Ad hoc network (c)client server network (d)None of the above

3) When the nodes are assumed to be capable of moving, either on their own or carried by their users, these networks are referred to as (a)wireless network (b)Ad hoc network (c)MANETs (d)None of the above
4) __________ is composed of tiny electronics devices, distributed over a area/region to observe some phenomenon.  (a)wireless network  (b)Ad hoc network  (c)MANETs  (d)Wireless Sensor Network

5) State true/false. Ad hoc networks are not necessarily mobile.  
(a)true  (b)false  (c)  (d)

6) self organization or self configuration is one of the features of __________  
(a)wireless network  (b)Ad hoc network  (c)Wireless sensor network  (d)None of the above

7) Spatial reuse of the bandwidth is used in __________ networks for energy efficient communication.  (a)Wireless sensor network  (b)MANETs  (c)Adhoc network  (d)None of the above

8) The TinyOS system, libraries and applications are written in which one of the following languages  (a)NesC  (b)C++  (c)Java  (d)C

9) Smart dust is a synonym for which of the following?  (a)Wireless sensor networks  (b)Mobile ad hoc networks  (c)Wearable computers  (d)Both (a) and (c)

10) State True/false. Sensor nodes mainly use the broadcast communication paradigm, whereas most ad hoc networks are based on point-to-point communication.  (a)true  (b)false  (c)  (d)

11) A sensor node consists of  (a)RF transceiver  (b)Micro controller  (c)energy source  (d)all of the above

12) State true/False. Global position systems are used to detect the movement of sensor networks.  (a)true  (b)false  (c)  (d)

13) Sensor networks deploy special routing protocol(s)  
(a)CGSR  (b)DSR  (c)AODV  (d)all of the above

Part A Questions
1. What is mobile middleware? Explain.
2. What is difference between sensor network and MANETs?
3. Explain MANETs.
4. What are the benefits of mobile agents?
5. Explain Jini.
6. What do you mean by application adaptation?
7. What are the ways of deploying sensor networks?
8. What is meant by mobile agents?
9. What are the advantages of mobile agents?
10. Draw the structure of a wireless sensor node.
11. List out the properties of an ad hoc network.
12. List out the requirements due to mobility in wireless sensor networks.

Part B
1. Explain the applications of wireless sensor network in detail.
2. Explain the challenges faced by wireless sensor networks.
3. Explain the service discovery framework in detail.
4. Describe the energy efficient strategies used in sensor networks.