ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A QOS Oriented Distributed Routing Protocol for Hybrid Wireless Network

Lavanya Karri
M.Tech,
Department of Computer Science & Engineering,
Avanthi Institute of Engineering and Technology,
Vizianagaram, A.P. India.

P.Srilakshmi
Associate Professor,
Department of Computer Science & Engineering,
Avanthi Institute of Engineering and Technology,
Vizianagaram, A.P. India.

Abstract:

As wireless communication gains popularity, significant research has been devoted to supporting real-time transmission with stringent Quality of Service (Quos) requirements for wireless applications. At the same time, a wireless hybrid network that integrates a mobile wireless ad hoc network (MANET) and a wireless infrastructure network has been proven to be a better alternative for the next generation wireless networks. By directly adopting resource reservation-based Quos routing for MANETs, hybrids networks inherit invalid reservation and race condition problems in MANETs. How to guarantee the Quos in hybrid networks remains an open problem. In this paper, we propose a Quos-Oriented Distributed routing protocol (QOD) to enhance the Quos support capability of hybrid networks. Taking advantage of fewer transmission hops and any cast transmission features of the hybrid networks, QOD transforms the packet routing problem to a resource scheduling problem. QOD incorporates five algorithms: 1) a Quos-guaranteed neighbour selection algorithm to meet the transmission delay requirement, 2) a distributed packet scheduling algorithm to further reduce transmission delay, 3) a mobility-based segment resizing algorithm that adaptively adjusts segment size according to node mobility in order to reduce transmission time, 4) a traffic redundant elimination algorithm to increase the transmission throughput, and 5) a data redundancy elimination-based transmission algorithm to eliminate the redundant data to further improve the transmission Quos. Analytical and simulation results based on the random way-point model and the real human mobility model show that QOD can provide high Quos performing terms of overhead, transmission delay, mobility-resilience, and scalability

Keywords:

Common type systems (CTS), Data Redundancy, IJETS TM, Instruction detection system, Password generation, Wireless networks.

1. INTRODUCTION

1.1 Introduction to Quos Oriented Distributed Routing Protocol:

As wireless communication gains popularity, significant research has been devoted to supporting real-time transmission with stringent Quality of Service (Quos) requirements for wireless applications. At the same time, a wireless hybrid network that integrates a mobile wireless ad hoc network (MANET) and a wireless infrastructure network has been proven to be a better alternative for the next generation wireless networks. By directly adopting resource reservation-based Quos routing for MANETs, hybrids networks inherit invalid reservation and race condition problems in MANETs. How to guarantee the Quos in hybrid networks remains an open problem. In this paper, we propose a Quos-Oriented Distributed routing protocol (QOD) to enhance the Quos support capability of hybrid networks. Taking advantage of fewer transmission hops and any cast transmission features of the hybrid networks, QOD transforms

Purpose:

The main purpose is to develop a solution to bridge the gap between the Educational Institutions and interactive parent communities by providing them with a common platform. The basic idea is to spread the information regarding the student's status to the parent's through Mail and SMS. The parent's can request for further information by posting the comments. Students can also check their status in the site.

Scope:

This site can be used by the students and their respective parents who are registered for that particular institution. The administrator has all the permission like manage academic details, student information and can send messages. The Staff on the other hand can only manage student information and can send messages. Students and parents can check the details in the site and they can post the comments to the administrate.

1.2 Project Overview: Packet routing problem to a resource scheduling problem. QOD incorporates.

FIVE Algorithms:

1) A Quos-guaranteed neighbor selection algorithm to meet the transmission delay requirement, 2) A distributed packet scheduling algorithm to further reduce transmission delay, 3) A mobility-based segment resizing algorithm that adaptively adjusts segment size according to node mobility in order to reduce transmission time. 4) A traffic redundant elimination algorithm to increase the transmission through put, and 5) A data redundancy elimination-based transmission algorithm to eliminate the redundant data further improve the transmission Quos. Analytical and simulation results based on the random way-point model and the real human mobility model show that QOD can provide high Quos performance in terms of overhead, transmission delay, mobility-resilience, and scalability.

A. Wireless Network:

Wireless connects directly to mobile users for video playing and devices together. Wireless technologies are widely used in emergency, services, military, education and entertainment. including laptops and handheld devices, for example the interaction in real time are increased. The evolution and the mobile networking environment. networking that relies on cables to connect networkable networks have been developed with various wireless of high Quality of Service(Quos) to support wireless and purpose of wireless internet users of smart phone in last three services are extensively expanded, so the networks are in need The rapid improvement of Wi-Fi capable mobile devices through wireless mobile devices and video streaming Wireless is a more modern alternative to traditional wired years. The usage of people watching video, playing games

B. Hybrid Wireless Network:

A hybrid wireless network is an extension to an infrastructure network, where a mobile host may connect to an access point (AP) using multi hop wireless routes via other mobile hosted APs are configured to operate on one of multiple available channels. Mobile hosts and wireless routers can select their operating channels dynamically through channel switch in .structure for the next generation wireless networks. It can help to tackle the stringent end-to end Quos requirements of different applications. Hybrid networks synergistically combine infrastructure networks and MANETs to leverage each other. For example it integrates a mobile Wireless Ad Hoc Network (MANET) and wireless infrastructure has proved a better alternative next generation wireless networks.

C. Quality of service (Quos):

It is the overall performance of a computer network, particularly the performance seen by the users of the network. To quantitatively measure quality of service, several related aspects of the network service are often considered, such as error rates, bandwidth, throughput, transmission delay, availability, jitter, etc. Quality of service is particularly important for the transport of traffic with special requirements. In particular, much technology has been developed to allow computer networks to become as useful as telephone networks for audio conversations, as well as supporting new applications with even stricter service demands. QOS provide high performance in terms of overhead, transmission delay Mobile resilience and scalability. Hybrid wireless network has proved a better network structure for next generation of wireless networks and help to tackle the stringent end to end QOS requirement for different applications. The proposed Quos-Oriented Distributed routing protocol (QOD) to enhance the Quos support capability of hybrid networks. Taking advantage of fewer transmission hops and any cast guarantee the Quos requirement in hybrid wireless networks. of the hybrid networks, QOD transforms the packet routing problem to a resource scheduling problem. Analytical and simulation results based International Journal On Engineering Technology and Sciences – IJETS™ ISSN (P): 2349-3968, ISSN (O): 2349-3976 on the random way-point model and the real human mobility model in terms of overhead, transmission delay, mobility-resilience, and scalability.

SYSTEM REQUIREMENTS:

2.1 TECHNOLOGY OVERVIEW:

The project entitled as “a quos oriented distributed routing protocol for hybrid wireless network” developed using Microsoft .net is a set of Microsoft software technologies for rapidly building and integrating XML web services, Microsoft windows-based applications, and solutions. The .NET framework is a language-natural platform for writing programs that can easily and securely interoperate.

Graphical user interface:

Graphical user interface (GUI) is straightforward and easy to navigate. The GUI provide various screens with appropriate incorporate icons, hyperlinks etc…, to facilitate screen navigation and data entry. The user has the ability to return homepage from any location within the application. The fallowing GUI form for user interaction can be conferred in the distributed routing protocol for hybrid wireless network.

Modules:

1. Login
2. Home page
3. Client home page
4. Administrator home page
5. Files upload form at admin side
6. Transaction from admin to client
7. Upload file
8. New register
9. Request from client to admin
10. Request from admin to client
11. Verifying files
12. Contact us

Module Description:

Login:

Admin and user will login with respective username and password on authentication they will access their information.

Homepage:

On authentication success the prior actor will be on his home page to perform his/her requirement

About us:

In this module information about developers of mats yapp and their details.

Administrator Home Page:

On Successful login Administrator will be on his home page to perform his/her actions.

User Home Page:

On Successful login user will be on his/her home page to perform actions.

New Registration:

In client side a new user can also register and access with provided their username and password.

View uploaded files:

Server will show the files which the administrators upload files.

Request from Client to Server:

The client asks some files which they download at that time client will send a request to server.

Technology used in this Project

Features of. Net

Microsoft .NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral platform for writing programs that can easily and securely interoperate. There's no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script. The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate. ".NET" is also the collective name given to various software components built upon the .NET platform. These will be both products (Visual Studio.NET and Windows.NET Server, for instance) and services (like Passport, .NET My Services, and so on).

.NET FRAMEWORK:

The .NET Framework has two main parts:

1. The Common Language Runtime (CLR).
2. A hierarchical set of class libraries.

The CLR is described as the "execution engine" of .NET. It provides the environment within which programs run. The most important features are

- ♦ Conversion from a low-level assembler-style language, called Intermediate Language (IL), into code native to the platform being executed on.
- ♦ Memory management, notably including garbage collection.
- ♦ Checking and enforcing security restrictions on the running code.
- ♦ Loading and executing programs, with version control and other such features.
- ♦ The following features of the .NET framework are also worth description:

Managed Code:

The code that targets .NET, and which contains certain extra Information - "metadata" - to describe itself. Whilst both managed and unmanaged code can run in the runtime, only managed code contains the information that allows the CLR to guarantee, for instance, safe execution and interoperability.

Managed Data:

With Managed Code comes Managed Data. CLR provides memory allocation and Deal location facilities, and garbage collection. Some .NET languages use Managed Data by default, such as C#, Visual Basic.NET and JScript.NET, whereas others, namely C++, do not. Targeting CLR can, depending on the language you're using, impose certain constraints on the features available. As with managed and unmanaged code, one can have both managed and unmanaged data in .NET applications - data that doesn't get garbage collected but instead is looked after by unmanaged code.

Common Type System:

The CLR uses something called the Common Type System (CTS) to strictly enforce type-safety. This ensures that all classes are compatible with each other, by describing types in a common way. CTS define how types work within the runtime, which enables types in one language to interoperate with types in another language, including cross-language exception handling. As well as ensuring that types are only used

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

in appropriate ways, the runtime also ensures that code doesn’t attempt to access memory that hasn’t been allocated to it.

Common Language Specification:

The CLR provides built-in support for language interoperability. To ensure that you can develop managed code that can be fully used by developers using any programming language, a set of language features and rules for using them called the Common Language Specification (CLS) has been defined. Components that follow these rules and expose only CLS features are considered CLS-compliant.

THE CLASS LIBRARY:

.NET provides a single-rooted hierarchy of classes, containing over 7000 types. The root of the namespace is called System; this contains basic types like Byte, Double, Boolean, and String, as well as Object. All objects derive from System. Object. As well as objects, there are value types. Value types can be allocated on the stack, which can provide useful flexibility. There are also efficient means of converting value types to object types if and when necessary. The set of classes is pretty comprehensive, providing collections, file, screen, and network I/O, threading, and so on, as well as XML and database connectivity. The class library is subdivided into a number of sets (or namespaces), each providing distinct areas of functionality, with dependencies between the namespaces kept to a minimum.

LANGUAGES SUPPORTED BY .NET:

The multi-language capability of the .NET Framework and Visual Studio .NET enables developers to use their existing programming skills to build all types of applications and XML Web services. The .NET framework supports new versions of Microsoft’s old favorites Visual Basic and C++ (as VB.NET and Managed C++), but there are also a number of new additions to the family. Visual Basic .NET has been updated to include many new and improved language features that make it a powerful object-oriented programming language. These features include inheritance, interfaces, and overloading, among others. Visual Basic also now supports structured exception handling, custom attributes and also supports multi-threading. Visual Basic .NET is also CLS compliant, which means that any CLS-compliant language can use the classes, objects, and components you create in Visual Basic .NET. Managed Extensions for C++ and attributed programming are just some of the enhancements made to the C++ language. Managed Extensions simplify the task of migrating existing C++ applications to the new .NET Framework. C# is Microsoft’s new language. It’s a C-style language that is essentially “C++ for Rapid Application Development”. Unlike other languages, its specification is just the grammar of the language. It has no standard library of its own, and instead has been designed with the intention of using the .NET libraries as its own. Microsoft Visual J# .NET provides the easiest transition for Java-language developers into the world of XML Web Services and dramatically improves the interoperability of Java-language programs with existing software written in a variety of other programming languages. Active State has created Visual Perl and Visual Python, which enable .NET-aware applications to be built in either Perl or Python. Both products can be integrated into the Visual Studio .NET environment. Visual Perl includes support for Active State’s Perl Dev Kit. Other languages for which .NET compilers are available include

- FORTRAN
- COBOL
- Eiffel

Fig1 .Net Framework

ASP.NET XML WEB SERVICES	Windows Forms
Base Class Libraries	
Common Language Runtime	
Operating System	

C#.NET is also compliant with CLS (Common Language Specification) and supports structured exception handling. CLS is set of rules and constructs that are supported by the CLR (Common Language Runtime). CLR is the runtime environment provided by the .NET Framework; it manages the execution of the code and also makes the development process easier by providing services.C#.NET is a CLS-compliant language. Any objects, classes, or components that created in C#.NET can be used in any other CLS-compliant language. In addition, we can use objects, classes, and components created in other CLS-compliant languages in C#.NET .The use of CLS ensures complete interoperability among applications, regardless of the languages used to create the application.

CONSTRUCTORS AND DESTRUCTORS:

Constructors are used to initialize objects, whereas destructors are used to destroy them. In other words, destructors are used to release the resources allocated to the object. In C#.NET the sub finalize procedure is available. The sub finalize procedure is used to complete the tasks that must be performed when an object is destroyed. The sub finalize procedure is called automatically when an object is destroyed. In addition, the sub finalize procedure can be called only from the class it belongs to or from derived classes.

GARBAGE COLLECTION:

Garbage Collection is another new feature in C#.NET. The .NET Framework monitors allocated resources, such as objects and variables. In addition, the .NET Framework automatically releases memory for reuse by destroying objects that are no longer in use. In C#.NET, the garbage collector checks for the objects that are not currently in use by applications. When the garbage collector comes across an object that is marked for garbage collection, it releases the memory occupied by the object.

OVERLOADING:

Overloading is another feature in C#. Overloading enables us to define multiple procedures with the same name, where each procedure has a different set of arguments. Besides using overloading for procedures, we can use it for constructors and properties in a class.

MULTITHREADING:

C#.NET also supports multithreading. An application that supports multithreading can handle multiple tasks simultaneously, we can use multithreading to decrease the time taken by an application to respond to user interaction.

STRUCTURED EXCEPTION HANDLING:

C#.NET supports structured handling, which enables us to detect and remove errors at runtime. In C#.NET, we need to use Try…Catch…Finally statements to create exception handlers. Using Try…Catch…Finally statements, we can create robust and effective exception handlers to improve the performance of our application.

THE .NET FRAMEWORK:

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet.

OBJECTIVES OF .NET FRAMEWORK:

1. To provide a consistent object-oriented programming environment whether object codes is stored and executed locally on Internet-distributed, or executed remotely.
2. To provide a code-execution environment to minimizes software deployment and guarantees safe execution of code.
3. Eliminates the performance problems.

There are different types of application, such as Windows-based applications and Web based applications.

4.3 Features of SQL-SERVER:

The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services.

SQL-SERVER database consist of six type of objects,
They are,

1. TABLE
2. QUERY
3. FORM
4. REPORT
5. MACRO

TABLE:

A database is a collection of data about a specific topic.

VIEWS OF TABLE:

We can work with a table in two types,
1. Design View
2. Datasheet View

Design View:

To build or modify the structure of a table we work in the table design view. We can specify what kind of data will be hold.

Datasheet View:

To add, edit or analyses the data itself we work in tables datasheet view mode.

Query:

A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dynast (if you edit it) or a snapshot (it cannot be edited).Each time we run query, we get latest information in the dynast. Access either displays the dynast or snapshot for us to view or perform an action on it, such as deleting or updating.

Problem Analysis:

Problem the foregoing survey of related literature on WMN, the following has been identified as open issues:

1. Current Wireless Mesh Network implementations lack a central control, and due to this problem, as network size increases, providing end-to-end quality of service guarantees for different service types in multi-hop WMNs is still a challenge.
2. Efficient operation of practical WMNs depends on close interactions between different layers of the Protocol Stack. How to exploit cross-layer interactions in the design of high-performance WMNs and how to optimize the performances of WMNs are still challenging issues.
3. In the design of Quos multicast routing protocol, important issues including system architecture design, capacity, performance analysis and optimization, quality of service, resource management, scheduling, MAC and Network Routing Protocol design still need to be addressed.
4. Although address resolution protocol (ARP), Dynamic Host Configuration Protocol (DHCP), and Internet Protocol (IP) may work for WMNs, an efficient support for broadband multicast traffic is still needed.
5. Other Open issues in relation to wireless mesh networks are: Quos routing, Multicast routing

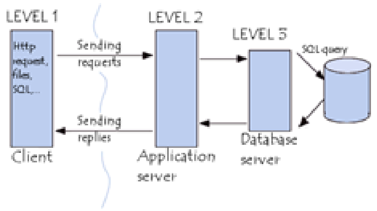
A BRIEF INTRODUCTION TO THREE TIRE ARCHITECTURE:

In 3-tier architecture, there is an intermediary level, meaning the architecture is generally split up between:

1. A Client, i.e. the , which requests the resources, equipped with a user interface for presentation purposes

2. The application server (also called **middleware**), whose task it is to provide the requested resources, but by calr

The three tiers in three-tier architecture are:



1. Presentation Tier:

Occupies the top level and displays information related to services available on a website. This tier communicates with other tiers by sending results to the browser and other tiers in the network.

2. Application Tier:

Also called the middle tier, logic tier, business logic or logic tier, this tier is pulled from the presentation tier. It controls application functionality by performing detailed processing.

3. Data Tier:

Houses database servers where information is stored and retrieved. Data in this tier is kept independent of application servers or business logic.

Usage:

A lot of critics of the 3-Tier Architecture seem to think that the ability to switch from one DBMS to another is not worth the effort as it rarely happens in real life. While it is true that, once developed, an application will rarely be switched to a different DBMS, a lot of organizations are now wishing that they had that ability when they compare the costs of their proprietary databases with the modern and far cheaper open source equivalents. Switching to a different DBMS **after** an application has been built is not the only benefit when you consider that not everyone works on a single legacy application for a single organization. There are software houses who develop and maintain a variety of applications for a variety of customers, so speed of development, cost of development, and providing their customers with more choices is what differentiates them from the competition and is likely to be a selling point instead of a rarely-used option. Consider the following:

- Suppose you develop an application which you want to sell as a package to lots of different customers? Do you want to restrict your potential customers to a DBMS of **your** choice, or one of **their** choice? By having all the DBMS logic in its own component in its own layer you can deliver the same application code to everybody and let them decide on what DBMS to use at installation time.
- Suppose that, instead of developing actual end-user applications, you develop a framework for building end-user applications? Do you want to restrict the potential users of this framework to a DBMS of **your** choice, or one of **their** choice?

By giving your customers the ability to easily switch from one DBMS to another without enormous expense and effort you will be pleasing your customers and displeasing your competitors. The ability to switch from one DBMS to another does not, or should not; restrict you to just one DBMS at a time. If each component in the Business layer is responsible for creating and communicating with a component in the Data Access layer it should be possible to allow more than one component to exist at the same time

HTML (Hyper Text Markup Language):

FEASIBILITY STUDY:

2.2 REQUIREMENTS SPECIFICATION:

A requirement is a feature that the system must have or a constraint that it satisfy to must satisfy to be accepted by the clients. requirements aims at defining the requirements elicitation and analysis. The main purpose is to develop a solution to bridge the gap between the Educational Institutions and interactive parent communities by providing them with a common platform.

The basic idea is to spread the information regarding the student's status to the parent's through Mail and SMS. The parent's can request for further information by posting the comments. Students can also check their status in the site. This site can be used by the students and their respective parents who are registered for that particular institution. The administrator has all the permission like manage academic details, student information and can send messages. The Staff on the other hand can only manage student information and can send messages. Students and parents can check the details in the site and they can post the comments to the administrator.

2.2.1HARDWARE REQUIREMENTS:

- ➢ System : Pentium IV 2.4 GHz.
- ➢ Hard Disk : 40 GB.
- ➢ Floppy Drive: 1.44 Mb.
- ➢ Monitor : 15 VGA Colour.
- ➢ Mouse : Logitech.
- ➢ Ram : 512 Mb.

2.2.2 SOFTWARE REQUIREMENTS:

- ➢ Operating system : Windows XP/7/LINUX.
- ➢ Implementation : NS2
- ➢ NS2 Version : NS2.2.28
- ➢ Front End : OTCL (Object Oriented Tool Command Language)
- ➢ Tool : Cygwin (To simulate in Windows OS)

SYSTEM ANALYSIS:

3. SYSTEMN ANALYSIS:

3.1 Identification Need:

EXISTING SYSTEM:

Hybrid wireless networks (i.e., multi-hop cellular networks) have been proven to be a better network structure for the next generation wireless networks and can help to tackle the stringent end-to-end Quos requirements of different applications. Hybrid networks synergistically combine infrastructure networks and MANETs to leverage each other. Specifically, infrastructure networks improve the scalability of MANETs, while MANETs automatically establish self-organizing networks, extending the coverage of the infrastructure networks. In a vehicle opportunistic access network (an instance of hybrid networks), people in vehicles need to upload or download videos from remote Internet servers through access points (APs) (i.e., base stations) spreading out in a city. Since it is unlikely that the base stations cover the entire city to maintain sufficiently. Strong signal everywhere to support an application requiring high link rates, the vehicles themselves can form a MANET to extend the coverage of the base stations, providing continuous network connections.

DISADVANTAGES OF EXISTING SYSTEM:

- Difficult to guarantee Quos in MANETs due to their unique features including user mobility, channel variance errors, and limited bandwidth.
- Although these protocols can increase the Quos of the MANETs to a certain extent, they suffer from invalid reservation and race condition problems.

PROPOSED SYSTEM:

In order to enhance the Quos support capability of hybrid networks, in this paper, we propose a Quos-Oriented Distributed routing protocol (QOD). Usually, a hybrid network has widespread base stations. The data transmission in hybrid networks has two features. First, an AP can be a source or a destination to any mobile node. Second, the number of transmission hops between a mobile node and an AP is small. The first feature allows a stream to have any cast transmission along multiple transmission paths to its destination through base stations, and the second feature enables a source node to connect to an AP through an intermediate node.

ADVANTAGES OF PROPOSED SYSTEM:

- The source node schedules the packet streams to neighbours based on their queuing condition, channel condition, and mobility, aiming to reduce transmission time and increase network capacity.

- Taking full advantage of the two features, QOD transforms the packet routing problem into a dynamic resource scheduling problem.

3.2 Feasibility Study:

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During that the proposed system is not a burden company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- BEHAVIORAL FEASIBILITY

Technical Feasibility:

A technically feasible system is always accurate, reliable, and secure so that this feasibility is a most required feature of a system. The proposed system is technically sound. It can work on simple personal computer, its hardware and software requirements are easily available. The system is capable of producing output within a given limited time. It has ability to process small volume of transaction at speed.

Behavior Feasibility:

This type of feasibility describes about user-friendly nature of the system so that an end user can easily communicate with the system. This project is very user-friendly with a good graphical mode of operation provided which is very easy to understand and to work with. So, the system is operationally feasible.

Economical Feasibility:

An evaluation of development cost weighed against the ultimate income of benefit from the development of the proposed system is made. Conducting 'Cost Benefit Analysis' determines whether the system is worthwhile to invest in. The proposed system is economically feasible as the cost of development is very less.

SYSTEM DESIGN:

4. SYSTEM DESIGN:

4.1 Data Flow Diagram:

The DFD is also called as dabble chart. it is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.

Flow Diagram of Administrator:

Flow Diagram of user:

4.4 INTRODUCTION TO UML DIAGRAMS:

The unified modeling language (UML) is one of the most exiting tools in the world of system development. UML is the brain child of Grady Booch, James ram Baugh the UML enables system builders to create blue prints the capture their visions in a standard, easy to understand way and communicate them to others. system analysts would try to assess the needs of their clients, generate requirement analysis in some notation that the analyst understood ,give that analysis to a programmer or a team of programmers and hope that the finally product was the system the client wanted. Uml is a notation that resulted from the unification of objects modeling technique and object oriented software technology.uml has been designed for broad range of applications .hence; it provides constructs for broad range of systems and activities.

Components of Uml:

The Uml consists of graphical elements that combine to form diagrams because it's a language; the Uml has rules for combining these elements. The purpose of these diagrams is to present multiple views of a system and this set of multiple views is called a model. Uml model describes what a system is suppose to do it doesn't tell how to implement a system.

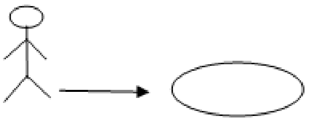
An Overview of Uml In Five Notations:

- ✓ Use case diagrams
- ✓ class diagrams
- ✓ sequence diagrams
- ✓ state chart diagrams

- ✓ activity diagrams
- ✓ collaboration diagrams

USE CASE DIAGRAM:

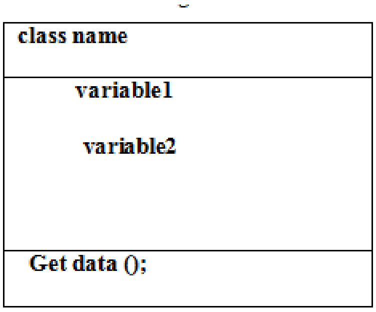
the use case is a description of a systems behavior from a user's standpoint. For system developers this is a valuable tool: it's a tried and true technique for gathering system requirements from user's point of view. Uml the use case diagram represents as follows the littlie stick figure is called an actor. The ellipse represents the use case. The actor is the entity that initiates the use case can be a person or any other system.



Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from the external point of view. The actors are outside the boundary of the system, whereas the use cases are inside the boundary of the system.

CLASS DIAGRAM:

A class is a category or group of things that have similar attributes and common behavior. a rectangle is a icon that represents a class. it is divided into three areas. The upper most area contains the name, the middle area holds the attributes and the lower area shows operations. a class diagram consists of a number of these rectangle connected by the lines that show the classes relate to one another. Class diagrams provide the representation that developers work form. Class diagrams provide the representations that developers work form. Class diagrams help on the analysis side too. They enable analyst talk to client in the client's terminology and thus simulate the clients the important details about the problems they want solved.



Class diagrams to describe the structure of the system. Classes are abstraction that specifies the common structure and behavior of a set of objects. Class diagrams describe the system in terms of objects, classes, attributes, operations and their associations.

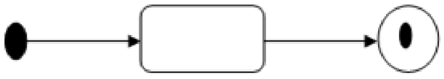
Object Diagram:

An object is an instance of a class-a specific thing that has a specific value of the attributes and behavior. an icon is a rectangle similar to a class icon but the name is underlined. The name of a specific instance is on the left side of a colon, and the name of the class is on the right side of the colon.

Object name: class name

State Chart Diagram:

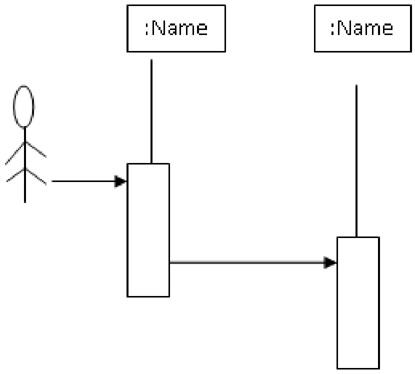
At any given time, on object is in a particular state. The Uml state diagram captures this bit of reality. The symbol at the top of the figure represents the figure represents the start state and symbol at the bottom represents the end state. State chart diagrams describe the behavior of an individual object as a number of states and transitions between these states. a state chart represents a particular set of values for an object. The sequence diagram focuses on the messages exchanged between objects, the state chart diagrams focuses on the transition between states.



Sequence Diagrams:

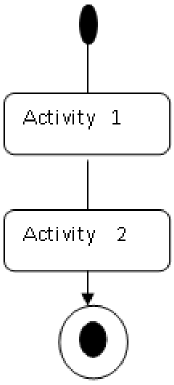
Class diagrams and object diagrams represent static information.

In a functioning system, however, objects interact with one another, and these interactions occur over time. The Uml sequence diagram shows the time based dynamics of the interaction. The sequence diagram consists of objects represented in the usual way as named rectangles (with underline messages), messages represented as solid line arrows and time represented as vertical progression. Mainly these are used for formalize the behavior of the system and to visualize the communication among the objects. they are use full for identifying additional objects that participate in the use cases. a sequence diagram represents the interaction that take place among these objects.



Activity Diagrams:

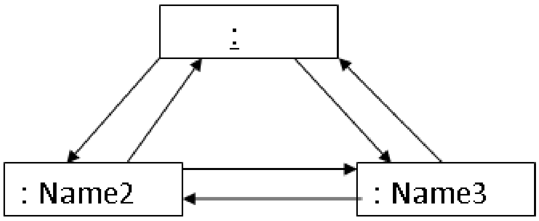
The activities that occur within a use case or within an objects behavior typically occur in a sequence .an activity diagram is designed to be simplified look at what happens during an operations or a process. it is an extension of state diagram. Each activity is represented by a rounded rectangle .the processing within an activity goes to compilation and than an automatic transmission to the next activity occurs. An arrow represents the transition from one activity to the next. The activity diagram has



A starting point represented by a filled in circle, and an endpoint represented by a bull's eye. An activity diagram describes a system in terms of activities. Activities are the state that represents the execution of a set of operations. These are similar to flow chart diagram and dataflow.

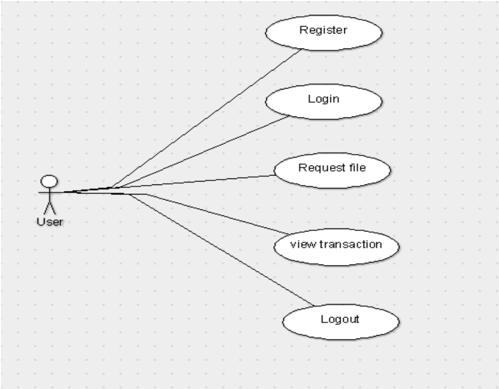
Collaboration Diagram:

The elements of a system work together to accomplish the systems objective and a modeling language must have a way of representing this. The Uml collaboration diagram is designed for this purpose. it is an extension of the object diagram .in addition to the association among objects the collaboration diagram shows the messages the objects send each other. An arrow near the association line between two objects represents the message. The arrow points to the receiving object.

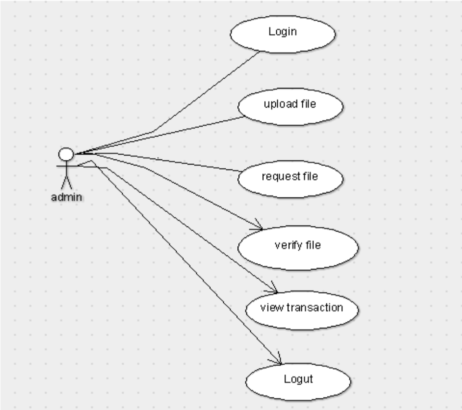


4.4 UML DIAGRAMS:

4.4.1 Use Case Diagram: (User)

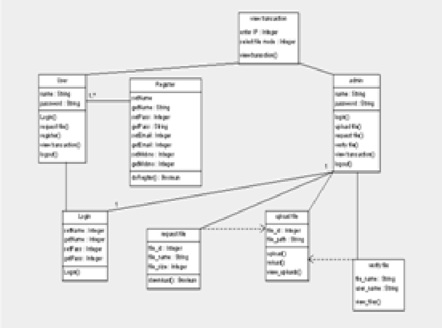


Use Case Diagram (Admin)



4.4.2 Class Diagram:

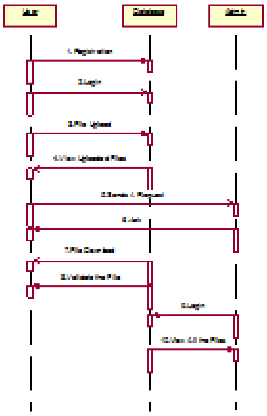
A class is a category of group of things that has similar attributes and common behavior. A Rectangle is the icon that represents the class is divided into three areas. the uppermost area contains the name, the middle area contains the attributes and the lowest areas show the operations. Class diagram provides the representation that develops work from. Class diagram help on the analysis side, too.



4.4.3 Sequence diagrams:

Interaction between objects can be described by means of sequence diagram. An object interacts with an another object by sending messages. the reception of a message by an object triggers the execution of an operation, which in turn may send messages to other objects. Arguments may be passed along with a message and are bound to the parameters of the executing operations in the receiving object.

Sequence Diagram: (Admin and user)

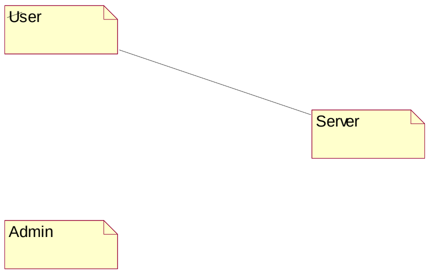


4.4.4 State-Chart Diagram:

State diagrams are used to describe the behavior of a system. State diagram describes all of possible states of an object as events occur. Each diagram usually represents objects of a single class and tracks the different states of its objects through the system. Not all class will require a state diagram and stat diagram are not useful for describing the collaboration of all objects in a use case. State diagrams have very few elements. the basic elements are rounded boxes representing the state of the object and arrows indicating the transition to the next state.

4.4.5 Collaboration:

The elements of a system work together to accomplish the systems objective and a modeling language must have a way of representing this. The Uml collaboration diagram is designed for this purpose. it is an extension of the object diagram .in addition to the association among objects the collaboration diagram shows the messages the objects send each other. An arrow near the association line between two objects represents the message. The arrow points to the receiving object



4.4.6 Activity Diagram:

The activities that occur within a use case or within an objects behavior typically occur in a sequence .an activity diagram is designed to be simplified look at what happens during an operations or a process. it is an extension of state diagram. Each activity is represented by a rounded rectangle .the processing within an activity goes to compilation and than an automatic transmission to the next activity occurs. An arrow represents the transition from one activity to the next. The activity diagram has a starting point represented by a filled in circle, and an endpoint represented by a bull's eye. An activity diagram describes a system in terms of activities. Activities are the state that represents the execution of a set of operations.

SYSTEM TESTING:

5. SYSTEM TESTING:

5.1 Introduction to Testing:

Testing is the process of executing a program with the intent of finding errors. During testing, the program for the test cases is executed with a set if test cases and the output of the program for the test cases is evaluated to determine the program is performing as it is expected. Error is the testing fundamental and is defined as the difference between the actual and ideal testing is usually relied upon to detect these faults in the coding phase. For this, different levels if testing are used which performs different task and aim to test different aspects of the system.

5.2Different types of testing:

Functional Testing:

In functional testing the structure of the program is not considered .Test cases are decided solely on the basis of the requirements or specifications of the program or module and the internal details of the module or the program is not considered for selection of test cases. This is also called "Black Box Testing".

Structural Testing:

Structural testing is concerned with testing the implementation of the program. This testing is also called "White Box Testing".

Unit Testing:

In unit testing different modules are tested against the specifications produced during design for the modules. Unit testing is essentially for verification of the code produced during the coding phase and hence the goal is to test the internal logic of the module.

Integration Testing:

In the integration testing, many test modules are combined into sub systems, which are then tested. The goal here is to see if the modules can be integrated properly, the emphasis being on testing module interaction. After structural testing and functional testing we get error free modules. These modules are to be integrated to get the required results of the system. After checking a module, another module is tested and is integrated with the previous module. After the integration the test cases are generated and the results are rested.

System Testing:

In system testing the entire software is tested. The reference document for this process is the requirement document and the goal is to see whether the software meets its requirements. The system was tested for various test cases with various inputs.

5.3 VERIFICATION AND VALIDATION:

Types of Testing:

1. Black box testing
2. White box testing

Black box Testing:

Black box testing is testing the software without any knowledge of the inner working, structure or language of the module being tested. black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document. it is a in which the software under test is treated, as a black box .you cannot “see” into it. the test provides input and responds to outputs without considering how the software works.

White box testing:

White box testing is a testing in which the software tester has knowledge of inner workings, structure and Language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.