High Level Design & Low Level Design

The purpose of this document is to provide with a template for documenting both HLD & LLD.

**SIMULATION OF SWITCHING FUNCTIONALITY**

**Document Control :**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Revision History** | | | | | | | | |
|  |  |  | |  |  |  |  |  |
| **Date** | **Version** | **Author** | **Brief Description of Changes** | | | | **Approver Signature** | |
| 28/11/2022 | 0.1 | Whole team | SRS briefing | | | |  | |
| 29/11/2022 | 0.2 | Whole Team | Designing | | | |  | |
|  |  |  |  | | | |  | |

**TABLE OF CONTENTS**

[1. Introduction 3](#_Toc368912248)

[1.1. Intended Audience 3](#_Toc368912249)

[1.2. Project Purpose 3](#_Toc368912251)

[1.3. Project Scope 3](#_Toc368912253)

[1.4. Functional Overview 3](#_Toc368912256)

[1.5 Assumptions & Dependencies 3](#_Toc368912257)

[1.6 Risks 3](#_Toc368912258)

[2. Design Overview 3](#_Toc368912259)

[2.1. Design Objectives 3](#_Toc368912260)

[2.2 Recommended Architecture 3](#_Toc368912261)

[2.2.1 UML Diagrams 3](#_Toc368912262)

[2.3 Data Flow Diagram 3](#_Toc368912263)

[3. System Architecture 3](#_Toc368912274)

[3.1. Functional Requirements 3](#_Toc368912275)

[3.2. Hardware interfaces.........................................................................................................................3](#_Toc368912276)

[3.3. Software interfaces..........................................................................................................................3](#_Toc368912277)

[3.4. Technical requirements 3](#_Toc368912278)

[3.5 Non functional Requirements 3](#_Toc368912279)

[4. Detailed System Design .................................................................................................................3](#_Toc368912281)

[4.1. Key Entities 3](#_Toc368912282)

[4.2. Detailed-Level Database Design.........................................................................................................3](#_Toc368912283)

[4.2.1. Data Mapping Information 3](#_Toc368912284)

[4.3.Business Process workflow 3](#_Toc368912286)

[4.4. Variables.....................................................................................................................................................3](#_Toc368912287)

[4.5. Business Logic 3](#_Toc368912287)

[4.6.Data Migration...................................................................................................................................3](#_Toc368912282)

[4.6.1 Architectural Reprsentation 3](#_Toc368912283)

[4.6.2 Architectural Goals And Constraints 3](#_Toc368912284)

[5. Environment Description 3](#_Toc368912300)

[5.1. Language Support 3](#_Toc368912301)

[5.2. Deployment Considerations 3](#_Toc368912302)

[5.3. Integration Requirements 3](#_Toc368912303)

[5.4. Network 3](#_Toc368912304)

[5.5. Operating System 3](#_Toc368912305)

1. **Introduction**

A switch operates on the data link layer / layer 2 of the OSI model. In a local area network (LAN) using Ethernet, a switch determines where to send each incoming message frame by looking at the physical device address (or MAC address). Switches maintain tables that match each MAC address, to the port where that MAC address station is connected. If a frame arrives at a port and the destination MAC address is unknown to the switch infrastructure, then it is flooded to all ports in the switching domain.

Our simulator will implement the switch functionality. The switch will have 4 ports, where each port can be enabled or disabled based on the requirement through the command line interface. Once the port is allocated to a process, the respective port can send or receive the frames. Switch, upon receiving a frame from a port, based on the destination MAC address in the frame header, it will look up the MAC address table, and send the frame to the destined port. If the destination MAC address is not available in the MAC it will flood all the stations with the frames.

## Intended Audience

This application is intended to be read by the Client.

## Project Purpose

Our Project main purpose is to increase the efficiency in sending messages through the stations. It will make sending and receiving of messages easier and the user can get the message in the required station.

## Project Scope

The simulation of switching functionality mainly takes place in layer 2 switch. The switch is enabled to handle as many ports to transfer the messages or frames between the stations.

The significant use of this functionality can be seen in networking areas. In LAN, WAN, almost in every networks this functionality can be used where there will be n number of stations connected.

## Functional Overview

The system supports stations to send as many messages to the required station. Users or user interface can send messages between the stations they want. The switch will look at the MAC table and if port address is found it forwards the messages from one station to the other.

## Assumptions & Dependencies

Assuming we are sending the frames from one station using port through switch to another station .The switch depends on the port on which it receives the frames.

## Risks

All assumptions, functional overview and design parameters are documented without evaluation which are to be implemented without missing. To eliminate its struggles with developing an optimized simulation and to properly manage resources, the use of deterministic simulation was explored but deterministic simulation was unable to create accurate switching.

# 2. Design Overview

It is a simulator which implements the switch functionality. The switch will have 4 ports, where each port can be enabled or disabled based on the requirement through the command line interface. Once the port is allocated to a process, the respective port can send or receive the frames. Switch, upon receiving a frame from a port, based on the destination MAC address in the frame header, it will look up the MAC address table, and send the frame to the destined port. If the destination MAC address is not available in the MAC it will flood all the stations with the frames.

## 2.1 Design Objectives

To effectively implement switching functionality between the layers using l2 switch.

### 2.2 Recommended Architecture

UML Architecture:

UML stands for Unified Modelling Language. Any real-world system is used by different users. The users can be developers, testers, business people, analysts, and many more. There are 14 types of UML are there. Some of them are,

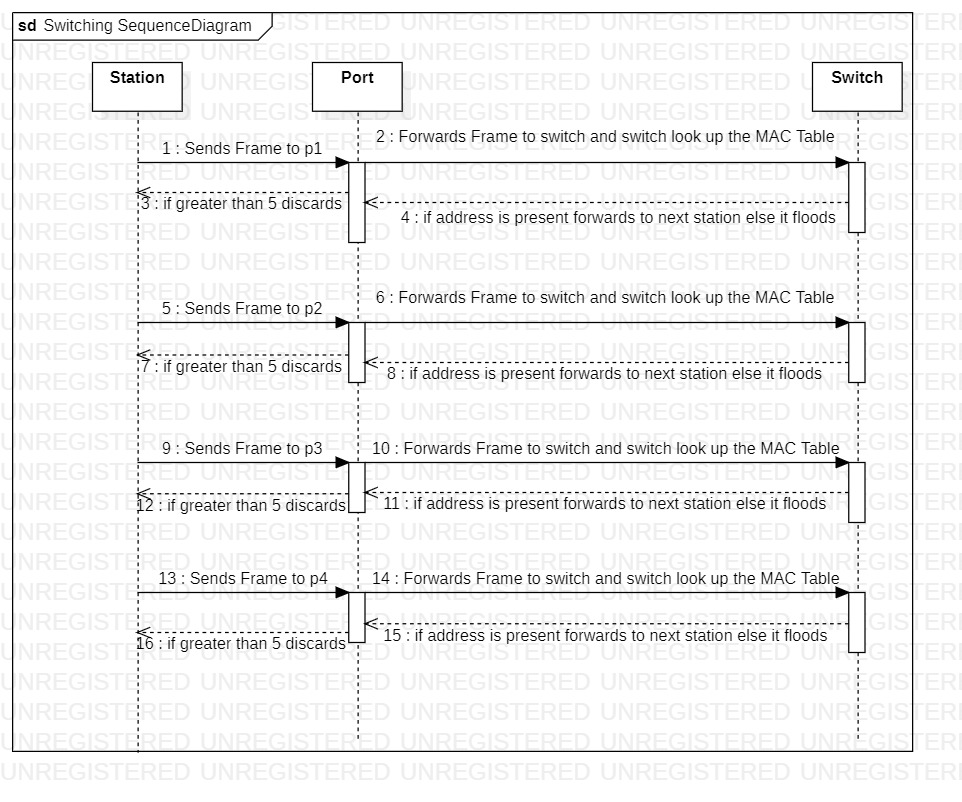
* Use Case Diagram
* Sequence Diagram
* Class Diagram

## 2.2.1 UML Diagrams

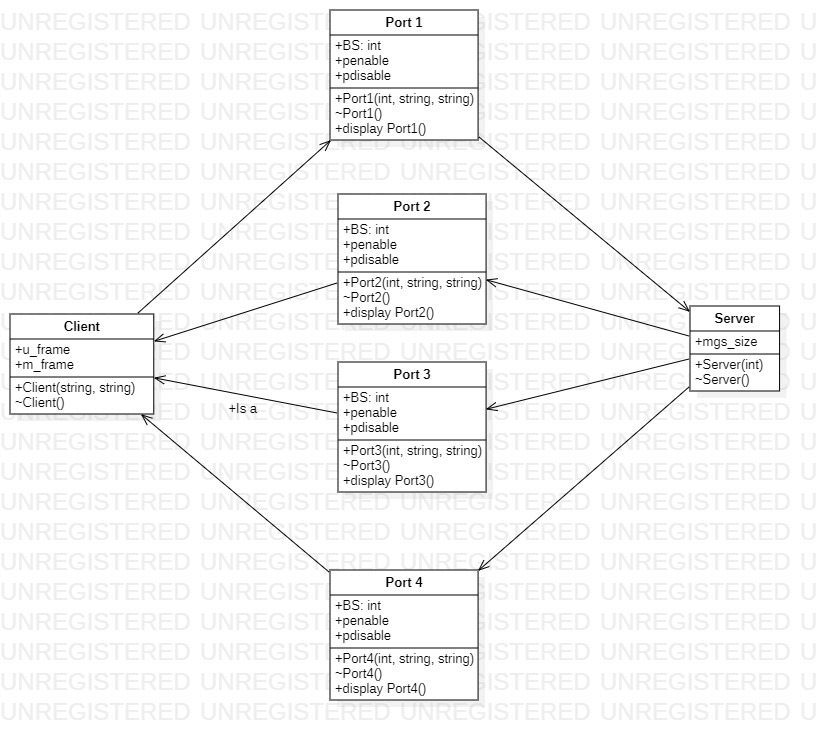
## System Use-Case Diagram

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. Here the switch have four ports to which the station can connect to. The stations are connected directly the user interface and they are connected to switch through corresponding ports. The switch can be connected to only four ports and should have the buffer size of five frames. The station sends the frames to the switch using ports.

### Sequence Diagram

 A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction. Here the switch have four ports to which the station can connect to. The stations are connected directly to the user interface and they are connected to switch through corresponding ports. The switch can be connected to only four ports and should have the buffer size of five frames. The station sends the frames to the switch using ports. The switch will look up at the MAC table for port address. If the port address is present, it forwards the frames to next station or else it floods all the stations with frames.

**Class Diagram**



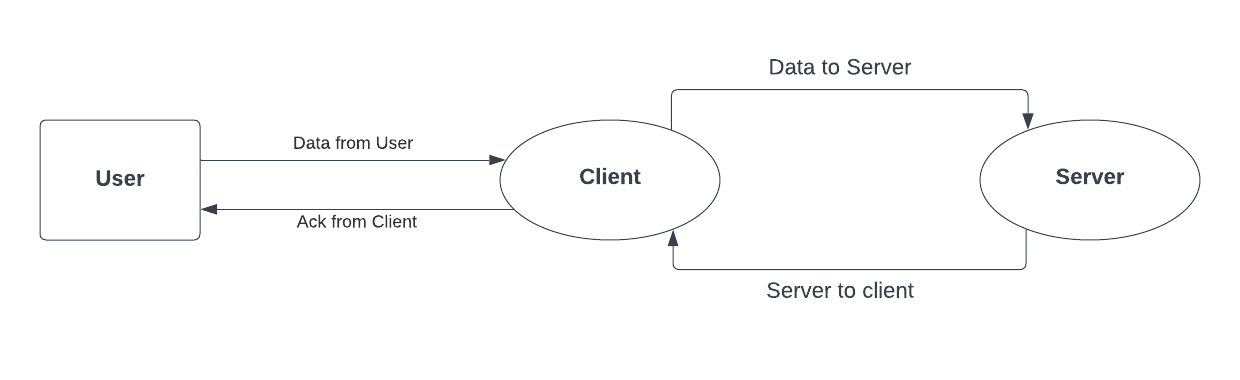
This is the most used UML diagram in the field of software engineering design. It is called as a main building block of any object oriented solution. Usually it illustrates the classes in a system, attributes and operations of each class and also the relationship between each class.

Here the station & the switch acts as the main classes & the 4-ports acts as child classes .In all classes, we declared variables & functions.

**2.3 DATA FLOW DIAGRAMS**

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both. It shows how data enters and leaves the system, what changes the information, and where data is stored.

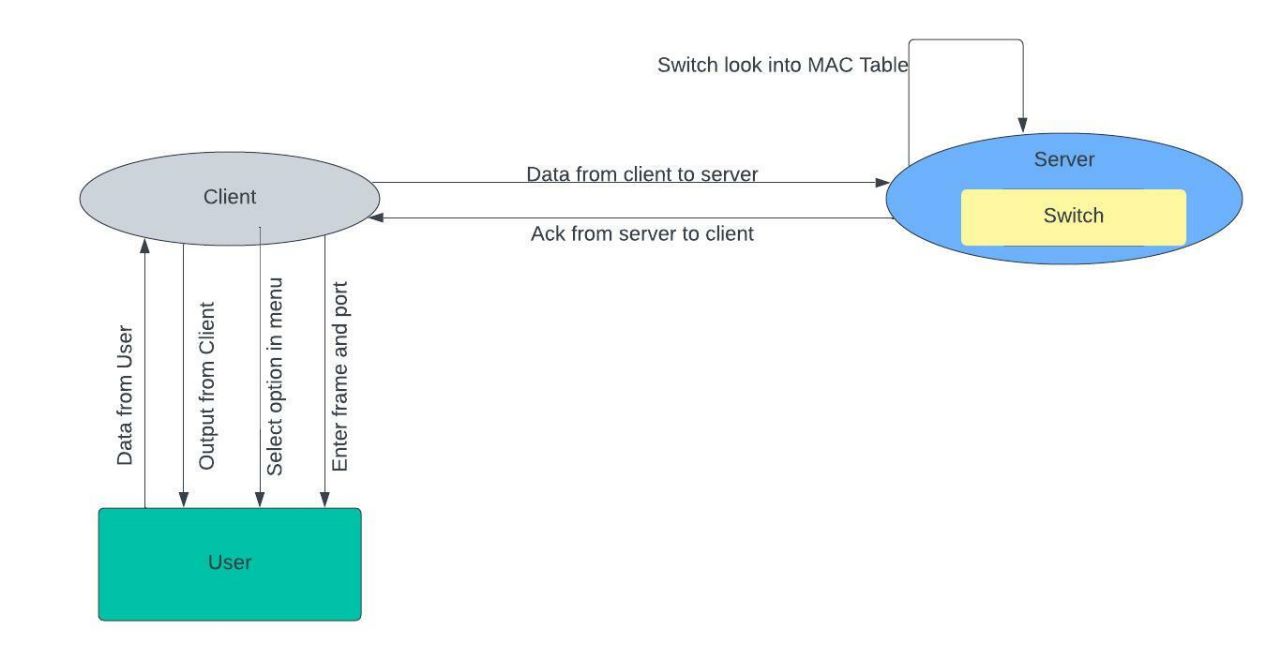
**LEVEL 0 DFD**

****

It is also known as a context diagram. It’s designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

The DFD Level 0 diagram for simulation of switching functionality is also known as the context diagram of the system. The level 0 or context diagram presents the main idea as the basis for the subsequent levels. The basic idea is represented by a single process consisting of the main process, users, and data.

**Level 1 DFD**

****

In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into sub processes.

The stations are connected directly to the user interface and they are connected to switch through corresponding ports. The switch can be connected to only four ports and should have the buffer size of five frames. The station sends the frames to the switch using ports. The switch will look up at the MAC table for port address. If the port address is present, it forwards the frames to next station or else it floods all the stations with frames

**3.System Features**

**3.1 Functional Requirements**

**3.1.1 Ports**

Switch will have 4 ports to which stations can connect to.

### 3.1.2 Port Buffer Size

Each port has a buffer size of 5 frames, upon which the port discards the incoming frames.

### 3.1.3 Enable/Disable Port

A port can be enabled or disabled through command line interface.

### 3.1.4 MAC Address Table

Switch should internally store a MAC address table.

### 3.1.5 Traffic Filtering

Switch should not forward a frame out of port on which it is received.

### 3.1.6 Unicast Frame

Switch should be able to handle a unicast frame.

### 3.1.7 Broadcast Frame

Switch should be able to handle the broadcast frame.

### 3.1.8 Entry Removing in MAC Table

When a station is disconnected, it should remove its entry from the MAC Table.

### 3.1.9 Station Connection

Station should be able to connect to one of the ports of the switch and can send frames.

### 3.1.10 Accept/Discard Frames

Station should be able to accept or discard the incoming frame.

**3.1.11 Station Termination**

When the switch is terminated, all stations should terminate.

**3.2 Hardware Interfaces**

• GHz processor, 2 GB RAM or more (system memory)

• 20 GB of hard-drive space or more

• VGA capable of 1024×768 screen resolution

• Necessary computer peripherals such as keyboards etc.

• Internet Connectivity (Wired/ Wireless)

**3.3 Software Interfaces**

• Windows/ Linux Based OS/ Mac OS/ Any OS capable of running c ++

• Database

• Server

**3.4 Technical Requirements**

• P thread

• CPP File handling

• System Programming

• CPP Language

**3.5 Non-Functional Requirements**

• CPPUnit to automate unit testing

• Valgrind to detect memory leak

• Make file

• Multi file multi directory solution with two step compilation process.

# 4 Detailed System Design

## 4 .1 Key Entities

Stations, Ports, Switch, MAC table

## 

## 4.2 Detailed-Level Database Design

Here the switch have four ports to which the station can connect to. The stations are connected directly the user interface and they are connected to switch through corresponding ports. The switch can be connected to only four ports and should have the buffer size of five frames. The station sends the frames to the switch using ports.

**4.2.1 Data Mapping Information**

In this application mapping refers to port address and data. It compares the address and port number to enable the particular port.

## 4.3 Business Process workflow

1. Connecting stations to switch through ports

2. Switch have internal MAC table

3. Sending frames from stations through ports

4. Switch look up the MAC table for address

5. If present forward the frames else floods

**4.4 Variables**

1.class IPCMq main mq;

{

int selection=0;

int ret = -1;

unsigned int prio;

string rxMsgFromMain;

}

2. class IPCMq

{

private:

string -mqName;

int -mqFlags;

}

3.class TCPServer server

{

int clNum = 0;

int port = 65123;

server.bindAddress(port);

}

## 4.5 Business Logic

**4.5.1 Client Model**:

4.5.1.1 string Messaging :: create\_message(msg\_types\_e option, string frame)

4.5.1.2 string Messaging :: create\_message(msg\_types\_e option, int p\_enable)

**4.5.2** **Server Model:**

4.5.2.1 int Switching::start\_switch(string swstart)

4.5.2.**2** int Switching::stop\_switch(string swstop)

4.5.2.3 sw.port\_enable(rcvString)

4.5.2.4 sw.port\_disable(rcvString)

## 4.6 Data Migration

Data Migration is the process of transferring data from one storage system or computing environment to another.

### 4.6.1 Architectural Representation

Architecture includes everything i.e Ports, Stations, Switch, MAC Table.

### 4.6.2 Architectural Goals and Constraints

### The goal is to implement switching functionality between the switch & stations using ports.

**5. ENVIRONMENT DESCRIPTION**

### 5.1 Language Support

System programming, IPC, Socket Programming.

### 5.2 Deployment Considerations

Hardware, Software, File storage, Session storage.

### 5.3 Integration Requirements

An Integration is connecting systems, applications and devices together so that you have a better flow of data and processes.

### 5.4 Network

The networks used are LAN, WAN.

**5.5 Operating System**

Linux OS

**Change Log**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **QMS Template Version Control (Maintained by QA)** | | | | | |
|  |  |  |  |  |  |
| **Date** | **Version** | **Author** | | **Description** | |
|  |  |  | |  | |
|  |  |  | |  | |
|  |  |  | |  | |
|  |  |  | |  | |
|  |  |  | |  | |