

Software Requirements Specification

for

Dynamic Connection Routing & Management System

Version 0.1

Prepared by

Group – 01

57175	Aditya Choudhury
56132	Rudranil Pal
58648	Sarabesh Kanishkar V
58645	Anirudh M

Wipro Ltd.

Section Number	Section Title
1	Introduction
1.1	Purpose
1.2	Document Conventions
1.3	Intended Audience and Reading Suggestions
1.4	Project Scope
1.5	References
2	Overall Description
2.1	Product Perspective
2.2	Product Functions
2.3	Operating Environment
2.4	Design and Implementation Constraints
2.5	User Documentation
2.6	Assumptions and Dependencies
2.7	System Context Diagram
3	System Features
4	External Interface Requirements
4.1	User Interfaces
4.2	Hardware Interfaces
4.3	Software Interfaces
4.4	Communication Interfaces
5	Other Non-Functional Requirements
5.1	Performance Requirements
5.2	Safety Requirements
5.3	Software Quality Attributes
6	Other Requirements
Appendix A	Glossary

1.Introduction

The Dynamic Connection Routing & Management System (DCRMS) is a high-performance network orchestration framework designed to manage and redirect communication sessions between multiple endpoint nodes. The system utilizes a centralized Routing Engine (Server) and a distributed Node Architecture (Client) to ensure seamless data flow even when specific endpoints are unavailable or occupied.

At its core, the DCRMS acts as a traffic supervisor. It maintains a real-time registry of all active nodes and their current status. When a connection request is initiated, the Routing Engine evaluates the destination node's state against a set of user-defined Redirection Protocols. If a node is marked as "unavailable" or "busy," the system dynamically reroutes the session to an alternative endpoint in real-time. This project simulates complex network behavior, focusing on database persistence, secure node authentication, and multi-level diagnostic logging.

This document provides a detailed description of the software requirements and serves as a foundational reference for system design, development, testing, and maintenance.

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to clearly and comprehensively describe the functional and non-functional requirements of the Dynamic Connection Routing & Management System (DCRMS).

This document aims to:

- Define the expected behavior of the system
- Specify system constraints and operating conditions
- Serve as a communication medium between stakeholders, developers, and testers
- Act as a baseline reference for High-Level Design (HLD) and Low-Level Design (LLD)
- Assist in verification and validation activities during testing

The SRS ensures that all system requirements are well understood before implementation begins and that the final system meets the intended objectives.

1.2 Document Conventions

The following document conventions are used throughout this SRS to maintain consistency and readability:

Font Style: Times New Roman

Main Headings: 18-point font, bold

Sub-headings: 14-point font, bold

Body Text: 12-point font

Alignment: Left-aligned text

Spacing: Standard MS Word spacing

Additional Conventions:

- Requirement identifiers are written in uppercase (e.g., DCRMS_SR_01, DCRMS_CL_02)
- Mandatory requirements are explicitly labeled as **Mandatory**
- Optional requirements are labeled as **Optional**

1.3 Intended Audience and Reading Suggestions

This document is intended for the following audience:

- Project Stakeholders: To understand system capabilities and limitations
- Software Developers: To implement system functionality as specified
- System Architects and Designers: To derive HLD and LLD from requirements
- Test Engineers: To design unit, integration, and system test cases
- Network Administrators: To understand deployment and operational procedures

Reading Suggestions

- Readers new to the project should begin with Section 1 (Introduction) and Section 2 (Overall Description) to gain a conceptual understanding of the system.
- Developers should focus on Section 3 (System Features) for detailed functional requirements.
- System administrators should review Section 4 (External Interface Requirements) and Section 5 (Non-Functional Requirements).
- Testers should review both functional and non-functional requirements to derive test scenarios.
- Reviewers may refer to the glossary for technical terminology.

1.4 Project Scope

The scope of the Dynamic Connection Routing & Management System (DCRMS) includes the following functionalities:

- Maintaining a persistent System Registry (Database) containing node identities, authentication credentials, and active redirection parameters
- Facilitating secure Node Enrollment and De-enrollment with credential verification
- Supporting three distinct Redirection Trigger Modes: Absolute Override, Response Timeout, and Resource Occupied
- Providing a State Controller for nodes to toggle their active redirection rules dynamically
- Performing real-time State Checks on inbound session requests to determine rerouting needs
- Offering an Administrative Console for manual system configuration and node management
- Generating comprehensive Telemetry Logs of all system activities
- Implementing a secure Handshake Protocol for node authentication
- Enabling nodes to modify Redirection Logic at runtime
- Providing a multi-level Diagnostic Module for system monitoring

Out of Scope

- Physical network hardware configuration
- Internet-scale routing protocols (BGP, OSPF, etc.)
- Deep packet inspection or content modification
- Integration with commercial cloud platforms

1.5 References

The following references were used during requirement analysis and system design:

- IEEE Software Engineering Standards for SRS Documentation
- Network Programming Interface Standards
- Database Management System Documentation
- Encryption and Security Protocol Standards
- Logging and Monitoring Framework Documentation

2. Overall Description

This section provides an overview of the system from a high-level perspective. It describes how the system fits into its operational environment, the major functionalities it offers, and the constraints under which it operates.

2.1 Product Perspective

The Dynamic Connection Routing & Management System (DCRMS) is a client-server application that operates in distributed network environments. It is designed as a modular and layered system consisting of:

Central Routing Engine (Server-Side Components)

- System Registry and Database Layer
- Authentication and Security Layer
- Redirection Logic Engine
- Administrative Interface Layer
- Telemetry and Monitoring Layer

Node Architecture (Client-Side Components)

- Handshake and Authentication Module
- State Controller and Rule Management
- Session Handler
- Diagnostic Module

The system architecture follows distributed computing principles with centralized control and decentralized execution, allowing efficient routing decisions while maintaining scalability and flexibility.

2.2 Product Features

- Secure node registration and authentication
- Dynamic routing and redirection
- Session and state management

- Health monitoring via heartbeat mechanism
- Administrative control and telemetry logging
- Fault detection and recovery support

2.3 Operating Environment

- Operating System: Linux (Ubuntu 20.04 or later)
- Kernel: 5.x and above
- Language: C++
- Compiler: gcc / g++
- Execution Environment: Terminal-based

2.4 Design and Implementation Constraints

The system is subject to the following constraints:

- Must maintain backward compatibility with enrolled nodes
- File format must support versioning and backward compatibility
- Binary file storage must ensure data integrity and consistency
- Encryption must be implemented for all sensitive transactions
- Real-time performance must be maintained under specified load
- System must support concurrent connections from multiple nodes
- File I/O operations must be synchronized for multi-threaded access
- Logging must be non-blocking and fault-tolerant
- Administrative interfaces must have role-based access control

2.5 User Documentation

This document is available on the Internet as we will upload the project in the Github repository which is public. Tutorials will be sent along with the software itself for easier understanding of working.

2.6 Assumptions and Dependencies

The assumptions and dependencies relevant to the system are as follows.

- Nodes have reliable network access

- System runs in a controlled enterprise environment
- Administrators have valid system privileges

3. System Features

3.1 Node Registration and Authentication

Description and Priority:

Nodes must register with the system before participating. This is a **high-priority** feature.

Stimulus / Response

- Node sends registration request
- System validates and acknowledges registration

Functional Requirements

DCRMS_01:

The system shall allow distributed nodes to register with the central engine.

The Overall flow begins with a Centralised server engine which can accept/register a new node and similarly remove/unregister the node connected to it.

DCRMS_02:

The system shall authenticate nodes before allowing participation.

When each client tries to connect to the server at that time ,the client has to be authenticated by the server.

3.2 Session Management and Routing

Description and Priority

The system must dynamically manage connection sessions and routing decisions.

Stimulus / Response

- Node requests a session
- System evaluates routing policies
- System responds with routing or redirection details

Functional Requirements

DCRMS_03:

The system shall establish and manage connection sessions between nodes.

DCRMS_04:

The system shall dynamically route or redirect connections based on node availability and policies.

3.3 Health Monitoring

Description and Priority

Continuous health monitoring ensures fault detection.

Functional Requirements**DCRMS_05:**

Nodes shall periodically send heartbeat messages.

DCRMS_06:

The system shall mark nodes inactive if heartbeats are missed.

3.4 State Management

Functional Requirements**DCRMS_07:**

The system shall maintain consistent state information for nodes and sessions.

After each node has been authenticated by the server ,then the system shall keep track of each node and its redirection status throughout its session time.

DCRMS_08:

State transitions shall be atomic and reliable.

3.5 Administrative Control

Functional Requirements**DCRMS_09:**

Administrators shall view node status and session metrics.

DCRMS_10:

Each node/client has a local admin console, using the console : the client status can be changed at runtime.

Client status can be set to available, busy or unavailable.

3.6 Logging and Telemetry

Functional Requirements**DCRMS_11:**

The system shall log connection events, errors, and routing decisions.

DCRMS_12:

The system shall support log levels: FATAL, INFO, WARNING, DEBUG.

Non-Functional Requirements

1. The application server should run (after compilation) on LINUX as well as windows

3.2 Server security

REQ-1: Server side Firewall Protection: The server should listen to port 69 for client connection and make the client side computer access the server port.

REQ-2: Server side Security: Before allowing a client connection the server should match the client IP with the blacklist IP to check if they are blacklisted.

4. External Interface Requirements**4.1 User Interfaces**

- Command-line interface for nodes
- Admin dashboard (CLI / Web-based optional)

4.2 Hardware Interfaces**Minimum Requirements**

- RAM: 1 GB
- Processor: 500 MHz
- Network interface

4.3 Software Interfaces

- Linux OS
- Socket libraries
- Logging frameworks

4.4 Communication Interfaces

- TCP / UDP
- IP-based networking

5. Other Non-Functional Requirements

5.1 Performance Requirements

- Low-latency routing decisions
- Support for multiple concurrent nodes

5.2 Security Requirements

- Authorized node access only
- Optional IP filtering and blacklisting
- Controlled administrative access

5.3 Software Quality Attributes

- Reliability
- Scalability
- Maintainability
- Observability

6. Other Requirements

- Graceful shutdown handling
- Resource cleanup on failures

Appendix A: Glossary

Term	Description
DCRMS	Dynamic Connection Routing & Management System
Central Engine	The server component that manages routing decisions and system state
Node	Client endpoint that participates in the routing network
Redirection Protocol	Rules defining when and how connections should be rerouted
System Registry	Persistent database storing node information and configurations
Handshake Protocol	Initial authentication and setup procedure between node and engine
State Controller	Module allowing nodes to modify their operational state
Telemetry Log	Comprehensive record of system activities and events

Administrative Console	Interface for system administration and monitoring
Redirection Trigger Modes	Conditions that initiate rerouting (Absolute Override, Response Timeout, Resource Occupied)
Diagnostic Module	Component providing debugging and monitoring capabilities
Session	A communication instance between endpoints
Heartbeat	Regular status messages indicating component health