Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

DIGITAL COMMUNICATION SYSTEMS

FINAL REPORT

TEAM MEMBERS:

21BEC1438 - B. KALYAN KUMAR
21BEC1563 - B. ABHIRAM
21BEC1211 - M. VARUN KUMAR

Digital Trunking Radio Communication System

ABSTRACT:

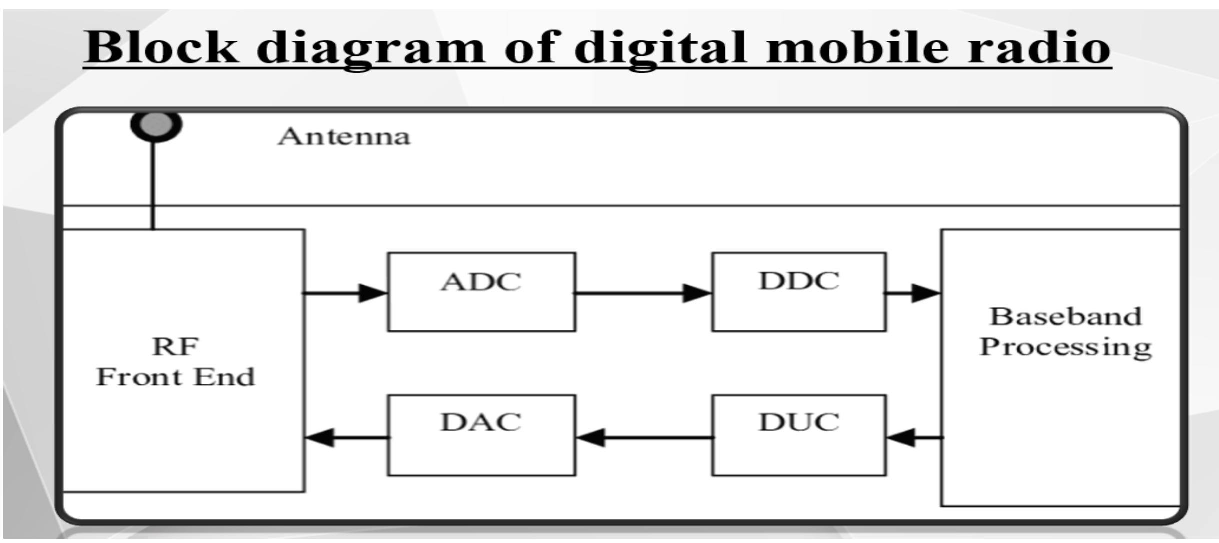
It focuses on developing an application-based simulation model to accurately estimate the system's coverage area. Factors such as terrain, antenna characteristics, transmit power, and receiver sensitivity are considered in the model. The simulation aims to optimize the deployment of digital trunking radio systems in industries like public safety and transportation.
The objective of this is to predict the area coverage of a radio communication system based on a digital trunking infrastructure for independent or private use. The research aims to achieve this objective by predicting the signal coverage area of the repeater and radio devices and observing the communication performance of the Digital Mobile Radio (DMR) Tier-III trunking communication system.

THEORY:

DMR, which stands for Digital Mobile Radio, is a widely used standard for digital-based radio communication systems designed for voice and data transmission in non-public wireless networks. It provides efficient and reliable communication capabilities for professional and commercial applications.

Here are some key features associated with DMR:

1. Digital Voice Communication
2. Time Division Multiple Access
3. Private and Non-Public Networks
4. Data Services

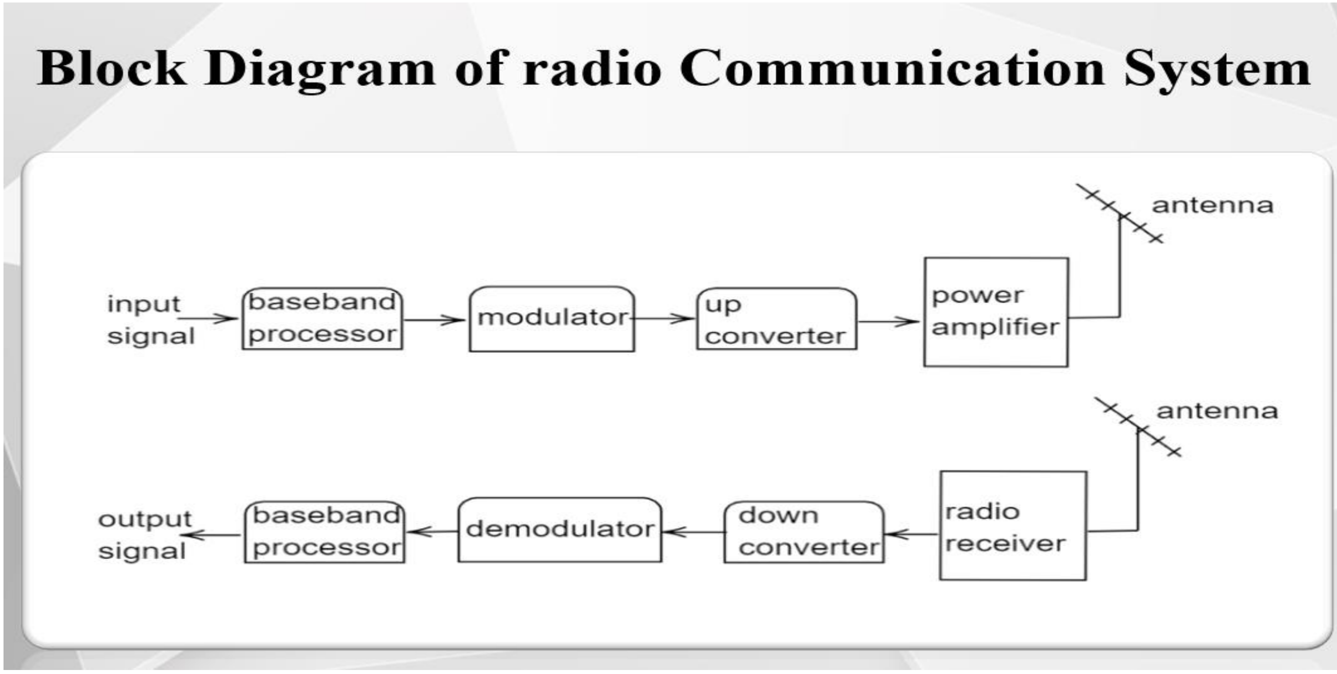


Radio Communication System Infrastructure:

1.Mobile radio devices: These are the common devices designed for use in vehicles or other mobile applications. They typically include a transceiver that allows to transmit and receive radio signals.

2. Portable radio devices: portable radios are handheld communications devices that offer wireless communication on the go. They are commonly used by individuals such as police officers, firefighters, or security personnel.

3. Power supply devices: power supply devices provide the necessary electrical power for the operation of radio communication equipment. They can be in the form of batteries, rechargeable batteries, or direct connections to a power source.



Transmission Techniques:

1. Audio data Transmission

2. Video Data Transmission

3. Text Data Transmission

1. Audio data Transmission:

For audio data transmission, a common approach is to use Pulse Code Modulation (PCM). PCM samples the analog audio signal at a fixed rate and converts each sample into a digital code. The digital codes are then transmitted over the digital trunking system.

In PCM, the analog audio signal is quantized into a discrete set of levels, and each sample is represented by a fixed number of bits. The bit rate is determined by the sampling rate and the number of bits used for each sample. The digital trunking system needs to have sufficient bandwidth to accommodate the required bit rate for audio data transmission.

2. Video Data Transmission:

Video data transmission in a digital trunking system typically involves using digital video compression techniques, such as MPEG (Moving Picture Experts Group) compression standards. These standards employ various algorithms to compress the video data and reduce the required bandwidth for transmission.
MPEG standards, such as MPEG-2, MPEG-4, and H.264, use techniques like spatial and temporal compression, motion estimation, and entropy coding to compress video

frames. The compressed video frames are then transmitted over the digital trunking system.

The specific parameters and configurations for video compression depend on factors such as the desired video quality, available bandwidth, and system constraint

3.Text Data Transmission:

Text data transmission can be achieved using various encoding schemes, such as ASCII or Unicode, where each character is represented by a specific binary code. The text data can be transmitted as a stream of binary codes over the digital trunking system.

Depending on the desired data rate and the number of characters to be transmitted, the system needs to allocate sufficient bandwidth to accommodate the text data transmission.

Types of Data & Modulation Techniques:

1. Analog Audio Data:

Modulation Techniques: Amplitude Modulation (AM), Frequency Modulation (FM).

Explanation: Analog audio data can be modulated using AM or FM techniques. AM varies the amplitude of the carrier signal, while FM varies the frequency. Both techniques preserve the analog nature of the audio signal.

2. Digital Audio Data:

Modulation Techniques: Pulse Code Modulation (PCM), Delta Modulation (DM).

Explanation: Digital audio data can be encoded using PCM, where the audio signal is sampled and quantized into a binary format. Delta Modulation is a simpler technique that encodes the difference between consecutive samples.

3. Analog Video Data:

Modulation Techniques: Amplitude Modulation (AM), Frequency Modulation (FM).

Explanation: Analog video signals can be modulated using AM or FM techniques. AM and FM carry the video signal by varying the amplitude or frequency of the carrier signal, respectively.

4. Digital Video Data:

Modulation Techniques: Quadrature Phase Shift Keying (QPSK), Quadrature Amplitude Modulation (QAM)

Explanation: Digital video data is typically transmitted using modulation schemes such as QPSK or QAM. These techniques modulate both phase and amplitude to represent multiple bits per symbol, allowing for higher data rates

APPENDIX:

```
package dcs_project;
import java.lang.Math;
import java.util.*;

public class Areacoverage {
```

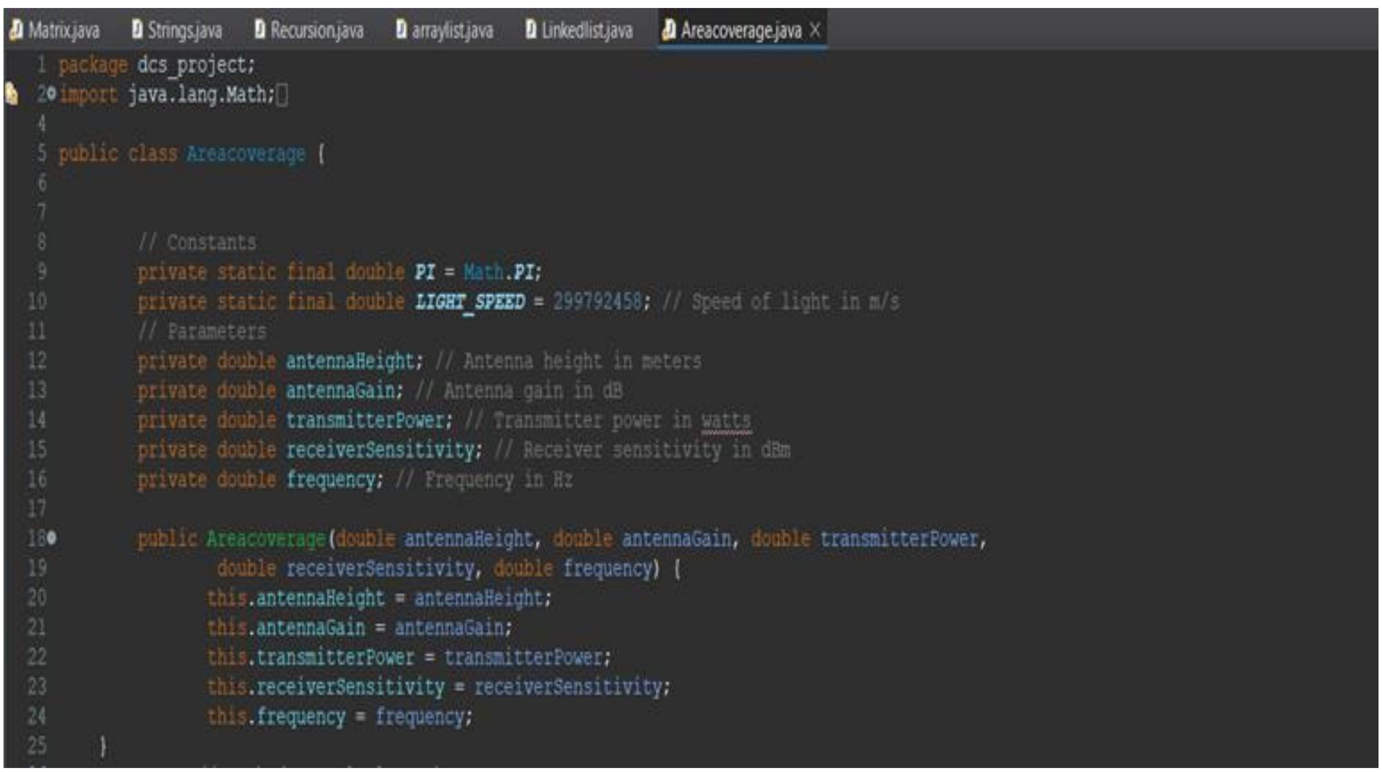
```
// Constants
private static final double PI = Math.PI;
private static final double LIGHT_SPEED = 299792458;
// Speed of light in m/s
// Parameters
private double antennaHeight; // Antenna height in meters
private double antennaGain; // Antenna gain in dB
private double transmitterPower; // Transmitter power in watts
private double receiverSensitivity; // Receiver sensitivity in dBm
 private double frequency; // Frequency in Hz

Public Areacoverage(double antennaHeight, double antennaGain, double
transmitterPower,
            double receiverSensitivity, double frequency) {
           this.antennaHeight = antennaHeight;
           this.antennaGain = antennaGain;
           this.transmitterPower = transmitterPower;
           this.receiverSensitivity = receiverSensitivity;
           this.frequency = frequency;
}

// Method to calculate the area coverage
      public double calculateAreaCoverage() {
// Convert receiver sensitivity from dBm to watts
        double receiverSensitivityWatts = Math.pow(10,
(receiverSensitivity / 10 - 3));
//Calculate the free space path loss
            double wavelength = LIGHT_SPEED / frequency;
                 double freeSpacePathLoss = (4 * PI * antennaHeight *
antennaHeight * frequency * frequency) / (LIGHT_SPEED *
LIGHT_SPEED);
```

```
// Calculate the received power
            double receivedPower = transmitterPower + antennaGain - freeSpacePathLoss;
// Calculate the coverage radius
            double coverageRadius = Math.sqrt((receiverSensitivityWatts * wavelength * wavelength) / (16 * PI * receivedPower));
// Calculate the coverage area
            double coverageArea = PI * coverageRadius * coverageRadius;
            return coverageArea;
}

  //"sample 1"

//        public static void main(String[] args) {
//            double antennaHeight = 50.0; // meters
//            double antennaGain = 10.0; // dB
//            double transmitterPower = 100.0; // watts
//            double receiverSensitivity = -100.0; // dBm
//            double frequency = 900000000.0; // 900 MHz
//            Areacoverage calculator = new Areacoverage(antennaHeight, antennaGain,transmitterPower, receiverSensitivity, frequency);
//            // Calculate the area coverage
//            double coverageArea = calculator.calculateAreaCoverage();
//            // Display the result
//            System.out.println("Coverage Area: " + "" + " square meters");
//        }
//
//          //"sample 2"
//
//          public static void main(String[] args) {
//              double antennaHeight = 50.0; // meters
```

```
//		double antennaGain = 10.0; // dB
//		double transmitterPower = 1007.0; // watts
//		double receiverSensitivity = -1.0; // dBm
//		double frequency = 900000000.0; // 900 MHz
//	Areacoverage calculator = new Areacoverage(antennaHeight, antennaGain,transmitterPower, receiverSensitivity, frequency);

//		// Calculate the area coverage
//		double coverageArea = calculator.calculateAreaCoverage();
//		// Display the result
//		System.out.println("Coverage Area: " + "" + " square meters");
//	}
//
//	//"sample3"
//	public static void main(String[] args) {
//		double antennaHeight = 50.0; // meters
//		double antennaGain = 10.0; // dB
//		double transmitterPower = 1007.0; // watts
//		double receiverSensitivity = -1.0; // dBm
//		double frequency = 900000000.0; // 900 MHz
//		Areacoverage calculator = new Areacoverage(antennaHeight, antennaGain,transmitterPower, receiverSensitivity, frequency);
//		// Calculate the area coverage
//		double coverageArea = calculator.calculateAreaCoverage();
//		// Display the result
//		System.out.println("Coverage Area: " + "" + " square meters");
//	}

////	"sample4"
//	public static void main(String[] args) {
```

```
//		double antennaHeight = 50.0; // meters
//		double antennaGain = 10.0; // dB
//		double transmitterPower = 1007.0; // watts
//		double receiverSensitivity = -1.0; // dBm
//		double frequency = 900.0; // 900 Hz
//		Areacoverage calculator = new Areacoverage(antennaHeight, antennaGain,transmitterPower, receiverSensitivity, frequency);
//		// Calculate the area coverage
//		double coverageArea = calculator.calculateAreaCoverage();
//		// Display the result
//		System.out.println("Coverage Area: " + coverageArea + " square meters");
//	}

  //"sample5"


     public static void main(String[] args) {
    double antennaHeight = 50.0; // meters
    double antennaGain = 10.0; // dB
    double transmitterPower = 1007.0; // watts
    double receiverSensitivity = -1.0; // dBm
    double frequency = 900.0; // 900 Hz
    Areacoverage calculator = new Areacoverage(antennaHeight, antennaGain,transmitterPower, receiverSensitivity, frequency);
    // Calculate the area coverage
    double coverageArea = calculator.calculateAreaCoverage();
    // Display the result
    System.out.println("Coverage Area: " + coverageArea + " square meters");
 }
```

}



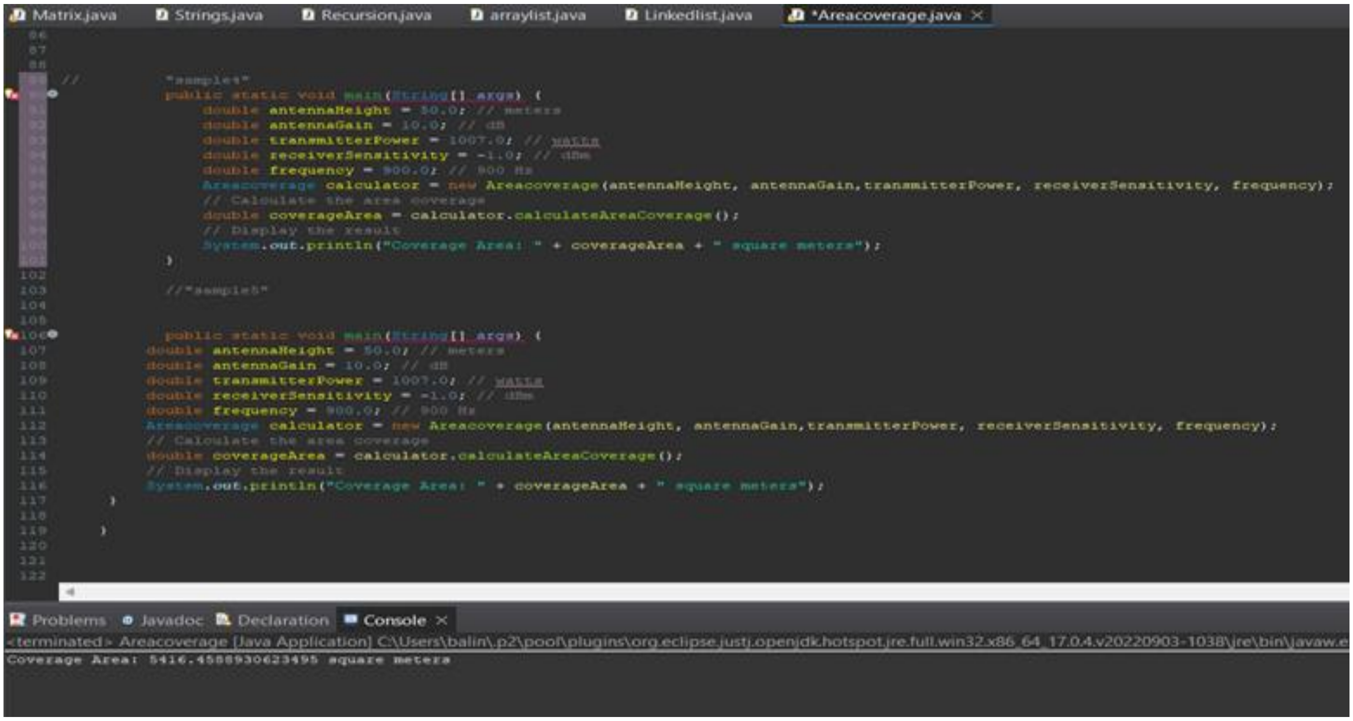

```java
package dcs_project;
import java.lang.Math;

public class Areacoverage {


        // Constants
        private static final double PI = Math.PI;
        private static final double LIGHT_SPEED = 299792458; // Speed of light in m/s
        // Parameters
        private double antennaHeight; // Antenna height in meters
        private double antennaGain; // Antenna gain in dB
        private double transmitterPower; // Transmitter power in watts
        private double receiverSensitivity; // Receiver sensitivity in dBm
        private double frequency; // Frequency in Hz

        public Areacoverage(double antennaHeight, double antennaGain, double transmitterPower,
                double receiverSensitivity, double frequency) {
            this.antennaHeight = antennaHeight;
            this.antennaGain = antennaGain;
            this.transmitterPower = transmitterPower;
            this.receiverSensitivity = receiverSensitivity;
            this.frequency = frequency;
    }
```

Matrix.java | Strings.java | Recursion.java | arraylist.java | Linkedlist.java | *Areacoverage.java

```java
            // Method to calculate the area coverage
        public double calculateAreaCoverage() {
    // Convert receiver sensitivity from dBm to watts
                    double receiverSensitivityWatts = Math.pow(10, (receiverSensitivity / 10 - 3));
    //Calculate the free space path loss
                    double wavelength = LIGHT_SPEED / frequency;
                    double freeSpacePathLoss = (4 * PI * antennaHeight * antennaHeight * frequency * frequency) / (LIGHT_SPEED * LIGHT_SPEED);
    // Calculate the received power
                    double receivedPower = transmitterPower + antennaGain - freeSpacePathLoss;
    // Calculate the coverage radius
                    double coverageRadius = Math.sqrt((receiverSensitivityWatts * wavelength * wavelength) / (16 * PI * receivedPower));
    // Calculate the coverage area
                    double coverageArea = PI * coverageRadius * coverageRadius;
                    return coverageArea;
    }



      //"sample 1"

      public static void main(String[] args) {
          double antennaHeight = 50.0; // meters
          double antennaGain = 10.0; // dB
          double transmitterPower = 100.0; // watts
          double receiverSensitivity = -100.0; // dBm
          double frequency = 900000000.0; // 900 MHz
          Areacoverage calculator = new Areacoverage(antennaHeight, antennaGain,transmitterPower, receiverSensitivity, frequency);
          // Calculate the area coverage
          double coverageArea = calculator.calculateAreaCoverage();
          // Display the result
          System.out.println("Coverage Area: " + "" + " square meters");
      }
```

Matrix.java | Strings.java | Recursion.java | arraylist.java | Linkedlist.java | *Areacoverage.java

```java

        //"sample 2"

        public static void main(String[] args) {
            double antennaHeight = 50.0; // meters
            double antennaGain = 10.0; // dB
            double transmitterPower = 1007.0; // watts
            double receiverSensitivity = -1.0; // dBm
            double frequency = 900000000.0; // 900 MHz
            Areacoverage calculator = new Areacoverage(antennaHeight, antennaGain,transmitterPower, receiverSensitivity, frequency);
            // Calculate the area coverage
            double coverageArea = calculator.calculateAreaCoverage();
            // Display the result
            System.out.println("Coverage Area: " + "" + " square meters");
        }

        //"sample3"
        public static void main(String[] args) {
            double antennaHeight = 50.0; // meters
            double antennaGain = 10.0; // dB
            double transmitterPower = 1007.0; // watts
            double receiverSensitivity = -1.0; // dBm
            double frequency = 900000000.0; // 900 MHz
            Areacoverage calculator = new Areacoverage(antennaHeight, antennaGain,transmitterPower, receiverSensitivity, frequency);
            // Calculate the area coverage
            double coverageArea = calculator.calculateAreaCoverage();
            // Display the result
            System.out.println("Coverage Area: " + "" + " square meters");
        }

```



CONCLUSION:

Accurate area coverage prediction and communication performance are crucial for digital trunking radio systems. They ensure reliable communication within designated areas, especially in critical situations. Predicting coverage accurately allows for optimal resource allocation, minimizing costs and maximizing efficiency. It also aids in network design and optimization, ensuring seamless communication and reduced signal degradation. Prioritizing communication performance enhances operational efficiency and enables swift emergency response.