Telecommunication Report

Shaik Awez Mehtab	Gautam	Saksham Jain	Mohana Evuri
23B1080	23B0957	23B1074	23B1017

Fall 2025

1 Design

1.1 Physical Layer Encoding

We are encoding the message in a grid of colored pixels. Suppose that the number of colors is C, then using a grid of k blocks we can encode a message of length C^k. For encoding, we group x bits together so that $2x \leq C$ (i.e., each block of the grid can uniquely map the correct sequence of x bits). For decoding, the image is captured with a frequency $\frac{1}{T}$ using a camera. It then uses an image detection software to get a 2D grid of numbers, where each numbers represents one color, which we map back to the corresponding sequence of bits. Then we refer to the encoding scheme we used and the error detection plus correction mechanisms to get the final message. Parameters C, k, T (defined in implementation) can be tuned.

1.2 Link Layer Framing

1.2.1 Frame Division

To transmit a message of M bits, we divide it into frames of F bits each. Thus, a total of $\lceil \frac{M}{F} \rceil$ frames need to be transmitted, with each frame displayed for a duration of T seconds. The parameter T is chosen to ensure that the receiver's camera can reliably capture each frame.

1.2.2 Frame Structure

For reliable framing, each frame begins with a *length field* that specifies the payload size in bits. This enables the receiver to accurately determine the message boundaries without relying solely on start or end markers. In addition, the frame is bordered with padding blocks (white cells), which clearly delineate the grid outline.

1.2.3 Start and End Detection

- **Start** – A solid transition screen, shifting from red (resting) to green (start), is shown to mark the beginning of a new message. After one second, the message frames are displayed.
- **End** – The receiver stops reading once the number of bits specified in the length field have been received, and then there is resting color displayed after the end of the message.

1.3 Link Layer Reliability

To ensure reliability at the link layer, we employ two-dimensional (2D) parity for error detection. Since the channel is assumed to introduce at most a single-bit error, we apply forward error correction (FEC) using 2D parity, which detects and corrects single-bit errors. This scheme requires only $M + F + 1$ redundant bits.

2 Implementation

2.0.1 Parameters

After fine-tuning, the system parameters are summarized as follows:

- **Message blocks (k):** 28 (arranged as 4 rows $\times$ 7 columns)
- **Number of colors (C):** 2 (black and white)
- **Length field:** 8 bits (prepended to the message)
- **Frame per second (T):** 1 frame per second
- **Final grid size:** 6×9 (including padding)

2.0.2 Camera Calibration

We set the camera to point to a red rectangle (resting color) displayed on the sender's laptop screen. Then we press enter on the receiver's side which prepares the receiver to start capturing the message.

2.0.3 Grid Generation

Each block corresponds to a single bit of the message and is arranged sequentially from the top-left to the bottom-right of the grid. On top of this structure, two-dimensional (2D) parity bits are added along with the necessary padding to complete the grid.

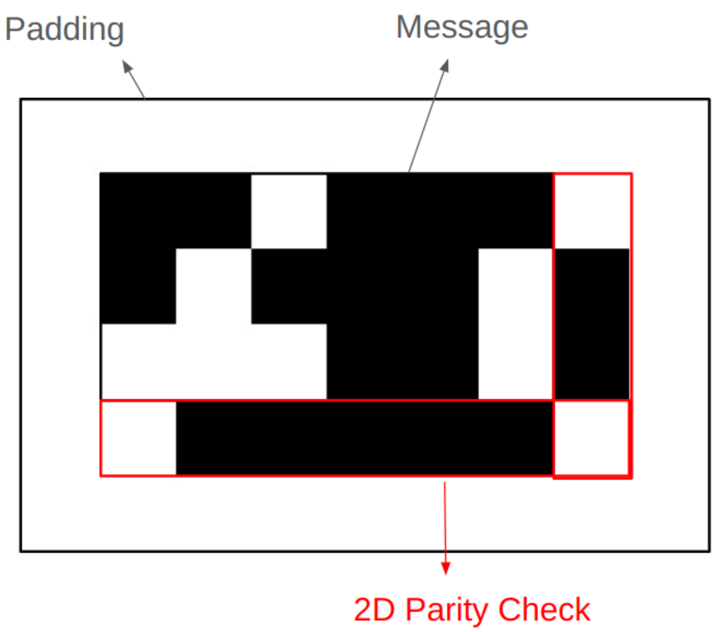

Figure 1: Example of the generated grid with 2D parity and padding.

2.0.4 Grid Detection

Grid detection was implemented using the **OpenCV** library. The camera continuously captures frames and processes them as follows:

1. **Preprocessing:** Each captured frame is converted to grayscale and then binarized using thresholding, separating black and white regions.

2. **Grid localization:** The outermost contour of the grid is identified and a boundary box is drawn around it. This defines the full area of the grid.

3. **Cell extraction:** The bounding box is divided into uniform cells based on the expected number of rows (N_{ROWS}) and columns (N_{COLS}). The value of each cell (0 or 1) is determined by computing the average intensity of the pixels around its center.

4. **Matrix construction:** The resulting binary values are stored in a matrix representing the grid. This matrix is then used for decoding and error correction.

3 Evaluation

3.1 Experimental Setup

Our experimental setup consists of two computers: a sender and a receiver. The receiver is equipped with an external camera that is used to capture and detect the transmitted grid. The sender's screen is positioned directly in front of the camera to ensure clear visibility. The receiver program is initialized first, followed by the sender program, which displays a red screen as resting state. To signal the start of transmission, the sender displays a green screen as a synchronization message, after which it sequentially renders the encoded frames.

3.2 Metrics

We consider the following metrics:

- **Throughput:** kT, where k is the number of bits per frame and T is number of frames per second. This timeout is for synchronization purposes; we have kept it as 1 frame per second for reliability purposes, otherwise the transmission is at the speed of light. Thus, the throughput is ≈ 28 bits per second. This is throughput per message, considering there is no gap between sending of messages.

Together, these metrics provide a comprehensive view of how well the system performs in practice, balancing efficiency, responsiveness, and reliability.